

MQSeries[®] for Compaq NonStop[™] Kernel



Guida alla gestione del sistema

Versione 5 Rilascio 1

MQSeries® for Compaq NonStop™ Kernel



Guida alla gestione del sistema

Versione 5 Rilascio 1

Nota

Prima di utilizzare questo prodotto e le relative informazioni, consultare la sezione “Appendice Q. Informazioni particolari” a pagina 409.

Prima edizione (giugno 2001)

Questa edizione si riferisce al programma MQSeries for Compaq NSK, Versione 5.1 e tutti i successivi rilasci, versioni e modifiche, se non diversamente indicato nelle nuove edizioni.

Come ultima pagina del manuale è stato predisposto un foglio riservato ai commenti del lettore. Se il foglio è stato rimosso, eventuali commenti possono essere inviati alla:

Selfin S.p.A.
Translation Assurance
via F. Giordani, 7
80122 - NAPOLI

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente dall'IBM e dalla Selfin e diventeranno proprietà esclusiva delle stesse.

© Copyright International Business Machines Corporation 1993, 2001. Tutti i diritti riservati.

Indice

Figure **ix**

Tabelle **xi**

Informazioni su questo manuale **xiii**

A chi si rivolge questo manuale xiii

Cosa bisogna conoscere per comprendere questo manuale xiii

Come utilizzare questo manuale xiii

Informazioni su MQSeries su Internet xiii

Novità presenti in MQSeries for Compaq NSK V5.1. **xv**

Miglioramenti delle prestazioni. xv

Funzioni di MQSeries aggiornate xv

Intercomunicazioni xvii

Facilità di utilizzo di Compaq NSK xvii

Parte 1. Guida **1**

Capitolo 1. Introduzione **7**

MQSeries e accodamento di messaggi 7

 Applicazioni indipendenti dal tempo 7

 Elaborazione dei messaggi guidati 8

Messaggi e code 8

 Che cos'è un messaggio? 8

 Cos'è una coda? 9

Oggetti 10

 Nomi degli oggetti 10

 Gestione degli oggetti 11

 Queue Manager di MQSeries 11

 Code di MQSeries 12

 Definizioni di processo 15

 Canali 15

 Cluster 16

 Namelist 16

Oggetti definiti dal sistema 16

Gestione 17

 Gestione locale e remota 17

Client e server 17

 Applicazioni MQI in un ambiente client-server 18

Estensione delle funzioni di Queue Manager 18

 User exit 18

 Servizi installabili 18

Sicurezza 19

 Funzione OAM (Object Authority Manager) 19

Supporti transazionali 19

Ottimizzazione delle prestazioni, affidabilità, scalabilità e dimensioni 20

Capitolo 2. Architettura di MQSeries for Compaq NSK V5.1. **21**

Panoramica dei processi queue manager. 21

Impacchettamento del prodotto. 23

Eseguibili 24

Messa a confronto dell'architettura della Versione 5.1 e della Versione 2.2.0.1 24

Queue manager — panoramica funzionale 25

Modello di processo di Queue manager 26

File e volumi secondari di MQSeries 27

Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries **29**

Esecuzione delle attività di gestione utilizzando i comandi di controllo 29

 Utilizzo dei comandi di controllo 29

Esecuzione delle attività di gestione utilizzando comandi MQSC 30

 Esecuzione dei comandi MQSC. 30

Esecuzione delle attività di gestione utilizzando comandi PCF. 31

 Attributi MQSC e PCF. 31

 Escape PCF 31

Gestione di TS/MP (PATHWAY) 32

 Specificazione e controllo dei listener TCP/IP 32

 Controllo del server di comando 32

 Specificazione e controllo dei channel initiator. 32

 Specificazione e controllo dei trigger monitor 33

 Specificazione della distribuzione di processi tra CPU. 33

 Aggiunta di nuovi processi di status server. 35

 Aggiunta di nuovi processi del server di coda. 35

 Specificazione della frequenza di aggiornamento dei pannelli monitor MQM 35

 Configurazione di PATHWAY per un Queue Manager 35

 Modifica dei parametri delle classi di server PATHWAY. 42

 Aggiunta di classi di server definite dall'utente a un PATHWAY di MQSeries 43

Capitolo 4. Gestione dei Queue Manager **45**

Introduzione 45

Come creare un queue manager 45

 Eeguire una copia di riserva dei file di configurazione dopo aver creato un Queue Manager 47

Modifica delle proprietà del queue manager 48

 Volume iniziale del queue manager 48

 Numero di processi EC per queue manager 48

 Terminale iniziale del queue manager 49

 Il nome di processo PATHMON per il queue manager 49

 Il CCSID del queue manager 51

 Controllo della conversione dei dati EBCDIC 51

 EMS Collector per il queue manager 52

Il pool di agenti tenuto pronto da ciascun EC nel queue manager	52
Limite massimo di riutilizzo processi e agenti inattivi	52
Priorità dei processi queue manager	53
Numero massimo di canali per il queue manager	53
Numero massimo di canali attivi per il queue manager	53
ID di segmento Guardian utilizzati da MQSeries	54
Porta TCP/IP predefinita	54
Porte TCP/IP su cui è in ascolto il queue manager	54
Processo TCP/IP utilizzato dal queue manager	55
Riconfigurazione della classe server MQS-TCPLISnn per una porta ed un processo TCP/IP non predefiniti	55
Assegnazione di spazio di swap	56
Nome status server predefinito	56
Aggiunta e rimozione di status server non predefiniti	57
Nome queue server predefinito	59
Aggiunta e rimozione di queue server non predefiniti	60
Struttura dei volumi	61
Volume secondario FFST del queue manager	61
Volume secondario file di dati queue manager	61
Volume secondario log errori queue manager	62
Volume secondario coda messaggi del queue manager	63
Volume secondario sincronizzazione canali del queue manager	63
Trasformazione dei nomi di oggetto	63
Lavorare con i gestori code	64
Creazione di un Queue Manager predefinito	64
Creazione dei principal di MQSeries	65
Esecuzione di cleanrdf per un queue manager abilitato a RDF	66
Avvio di un queue manager	66
Ripristino degli oggetti predefiniti e di sistema	66
Analisi dei file di oggetto	66
Interruzione di un queue manager	67
In caso di problemi	67
Chiusure immediate e prioritarie di un queue manager	67
Riavvio di un Queue Manager	68
Rendere predefinito un Queue Manager esistente	68
Eliminazione di un queue manager	68
Utilizzo della funzione MQM (Message Queue Management)	69
Utilizzo del Queue Manager Menu	70
Utilizzo del menu Queues	74
Utilizzo del menu Channels	82
Capitolo 5. Gestione di oggetti MQSeries locali	97
Supporto dei programmi applicativi che utilizzano MQI	97
Esecuzione delle attività di gestione locale utilizzando i comandi MQSC	98
Prima di iniziare	98
Utilizzo interattivo della funzione MQSC	99

Feedback dei comandi MQSC	100
Termine dell'input interattivo MQSC	100
Visualizzazione degli attributi Queue Manager	100
Utilizzo di un Queue Manager diverso dal predefinito	101
Modifica degli attributi di Queue Manager	101
Esecuzione di comandi MQSC da file di testo	102
Utilizzo dei file di comandi MQSC	103
Utilizzo dei prospetti MQSC	103
Esecuzione dei file di comandi MQSC forniti	104
Utilizzo di runmqsc per la verifica dei comandi	104
Risoluzione dei problemi con MQSC	105
Utilizzo delle code locali	106
Definizione di una coda locale	106
Modifica della dimensione di file fisico per le code	107
Definizione di una coda messaggi non recapitati	107
Visualizzazione degli attributi predefiniti di oggetti	108
Copia della definizione di una coda locale	109
Modifica degli attributi di coda locale	109
Modifica del volume di una coda locale	110
Modifica delle opzioni di una coda locale	110
Riassegnazione degli oggetti agli status server e ai queue server	111
Annullamento di una coda locale	111
Eliminazione di una coda locale	112
Esame delle code	112
Utilizzo degli alias di coda	115
Definizione di un alias di coda	115
Utilizzo di altri comandi con gli alias di coda	116
Utilizzo delle code di esempio	117
Definizione di una coda di esempio	117
Utilizzo degli altri comandi con le code di esempio	117
Gestione degli oggetti per triggering	118
Definizione di una coda di applicazione per triggering	118
Definizione di una coda di iniziazione	119
Creazione di una definizione di processo	119
Visualizzazione della definizione di processo	120

Capitolo 6. Automazione delle attività di gestione	121
Comandi PCF	121
Attributi MQSC e PCF	122
Escape PCF	122
Utilizzo di MQAI per semplificare l'impiego di PCF	122
Gestione del server di comando per una gestione remota	123
Avvio del server di comando	123
Visualizzazione dello stato del server di comando	123
Arresto del server di comando	124

Capitolo 7. Gestione di oggetti remoti di MQSeries	125
Canali, cluster e accodamento remoto	125
Gestione remota mediante cluster	126

Gestione di un Queue Manager remoto	127
Preparazione dei Queue Manager alla gestione remota	127
Preparazione di canali e code di trasmissione per la gestione remota	128
Definizione di canali e code di trasmissione	129
Avvio dei canali	130
Emissione remota di comandi MQSC	131
Consigli per l'accodamento remoto	132
Eventuali problemi nell'utilizzo remoto di MQSC	132
Creazione di una definizione locale di una coda remota	133
Analisi del funzionamento delle definizioni locali di code remote	133
Un modo alternativo per inserire messaggi in una coda remota	135
Utilizzo di altri comandi con code remote	135
Creazione di una coda di trasmissione	135
Code di trasmissione predefinite	135
Utilizzo di definizioni di coda remota come alias	136
Alias di Queue Manager	136
Alias di coda di risposta RTQ	136
Conversione dei dati	137
Quando un Queue Manager non è in grado di convertire messaggi in formati incorporati	137
File CCSID	137
Conversione di messaggi in formati definiti dall'utente	138
Modifica del CCSID del Queue Manager	138

Capitolo 8. Protezione di oggetti di MQSeries 139

Perché bisogna proteggere le risorse di MQSeries	139
ID utenti nel gruppo utenti MQM	139
Come ottenere ulteriori informazioni	140
Object Authority Manager (OAM)	140
Come funziona OAM.	141
Gestione dell'accesso mediante i gruppi utente	141
Protezione delle risorse con l'OAM	142
Utilizzo dei gruppi per le autorizzazioni	142
Disabilitazione dell'OAM (Object Authority Manager).	143
Utilizzo dei comandi OAM (Object Authority Manager).	144
Cosa specificare quando si utilizzano i comandi OAM	144
Utilizzo del comando almqusr	144
Utilizzo del comando dspmqusr	145
Utilizzo del comando setmqaut	145
Autorizzazioni di accesso	146
Comando di visualizzazione autorizzazioni	147
Istruzioni per OAM (Object Authority Manager)	147
ID utente	147
Volumi del queue manager	147
Code	148
Autorizzazione per utente alternativo	148
Autorizzazione contestuale	148
Considerazioni sulla sicurezza remota	149
Sicurezza del comando di canale	150
Comprendere le tabelle di specifica delle autorizzazioni	150

Autorizzazioni MQI	151
Autorizzazioni di gestione	154
Autorizzazioni per comandi MQSC nei PCF escape.	154
Comprendere i file di autorizzazione	156
Il database dei principal.	156
Il database OAM	156
Record di autorizzazione classe	158
Record di autorizzazione per tutte le classi	158

Capitolo 9. Handler di code messaggi non recapitati di MQSeries 159

Richiamo del DLQ handler	159
Tabella regole DLQ handler	160
Dati di controllo	160
Regole (schemi e azioni).	162
Parole chiave dello schema di confronto	162
Parole chiave di azione	163
Convenzioni relative alla tabella regole.	165
Elaborazione della tabella regole	166
Metodi per l'elaborazione di tutti i messaggi presenti nella coda DLQ.	167
Esempio di tabella regole per DLQ handler	168

Capitolo 10. Eventi EMS e di strumentazione 171

Eventi di strumentazione MQSeries	171
Tipi di eventi	171
Notificazione di evento attraverso code di evento.	172
Utilizzo delle code di evento con la funzione trigger.	172
Abilitazione degli eventi di strumentazione	172
Messaggi di evento	173
Eventi EMS (Event Management Service)	173
I file modello EMS forniti con MQSeries for Compaq NSK	173
Integrazione dei modelli di evento EMS MQSeries.	174
Definizione di PARAM MQEMSEVENTS	176
Utilizzo di un collector alternativo	177
Scrittura dei programmi per l'elaborazione degli eventi EMS MQSeries	177

Capitolo 11. Descrizione del supporto transazionale e della messaggistica. . 179

Utilizzo di NonStop TM/MP (Transaction Manager).	179
Limiti di Syncpoint	180
Operazioni diverse da syncpoint in messaggi permanenti	180
Operazioni Syncpoint o messaggi non permanenti	181
Requisiti della configurazione per TM/MP e MQSeries for Compaq NSK	181
Monitoring	181
Dimensione traccia di controllo	181
Configurazione gestore risorse.	182
Risoluzione dei problemi	182

Capitolo 12. Ripristino e riavvio 183

Ripristino e tolleranza di guasti	183
Back up e ripristino di MQSeries	184
Esecuzione del backup di MQSeries	184
Ripristino di MQSeries	184
Ripristino e riavvio degli status server e dei queue server	185
Ripristino mediante RDF	185

Capitolo 13. File di configurazione 187

Cosa sono i file di configurazione?	187
File di configurazione MQSeries (MQSINI)	187
Contenuto del file di configurazione MQSeries	187
File di configurazione Queue manager (QMINI)	189
Contenuto del file di configurazione Queue manager	189
File di configurazione Queue manager di esempio	191
Modifica dei file di configurazione	197
Implementazione delle modifiche ai file di configurazione	197
Suggerimenti relativi ai file di configurazione	197

Capitolo 14. Determinazione dei problemi 199

Controlli preliminari	199
Si è mai avuta in precedenza un'esecuzione con esito positivo di MQSeries?	199
Vengono riportati messaggi di errore?	199
Vengono riportati codici di errori che spiegano il problema?	200
Il problema è riproducibile?	200
Sono state effettuate modifiche dall'ultima esecuzione valida?	200
Si è mai avuta in precedenza un'esecuzione con esito positivo dell'applicazione?	201
Se l'applicazione in precedenza non è stata eseguita con esito positivo	201
Il problema è relativo a parti specifiche della rete?	202
Il problema si verifica ad una determinata ora del giorno?	202
Il problema si manifesta in modo discontinuo?	202
Sono stati eseguiti aggiornamenti del servizio?	203
Errori di programmazione frequenti	203
Problemi relativi ai comandi	203
Come procedere	204
Viene riportato un output non corretto?	204
Non si è ottenuta alcuna risposta utilizzando un comando PCF?	204
Si verificano malfunzionamenti in alcune code?	205
Il problema è relativo alle sole code remote?	206
L'esecuzione dell'applicazione o di MQSeries for Compaq NSK procede lentamente?	206
Considerazioni sulla progettazione delle applicazioni	207
Lunghezza del messaggio	207
Ricerca di un particolare messaggio	207
Code contenenti messaggi di diversa lunghezza	208
Frequenza di syncpoint	208

Utilizzo della chiamata MQPUT1	208
Output non corretto	208
Messaggi assenti nella coda	208
Messaggi che contengono informazioni inattese o alterate	210
Problemi relativi ad output non corretti durante l'accodamento distribuito	210
Log di errore	212
File di log	212
Errori precedenti	213
Messaggi per l'operatore	213
Esempio di log degli errori	213
Eventi EMS	214
Code messaggi non recapitati	215
File di configurazione e determinazione dei problemi	215
Utilizzo della traccia MQSeries	215
Nomi dei file di traccia	215
Esempio di dati traccia	216
FFST (First Failure Support Technology™)	216
Esame dei FFST	216

Capitolo 15. Scalabilità e prestazioni 219

Introduzione	219
Progettazione di nuove applicazioni per prestazioni e scalabilità	219
Progettazione per ridurre o eliminare l'utilizzo di risorse condivise	219
Ottimizzazione delle prestazioni iterativa	220
Messaggi permanenti	220
Messaggi non permanenti	220
Canali e messaggi non permanenti	221
Server di code e file di code	222
Memorizzazione di messaggi permanenti	222
Memorizzazione dei messaggi non permanenti	222
Distribuzione della CPU del server di code	222
Riassegnazione delle code ai server di code utilizzando almqfls	223
Coda di trasmissione cluster: SYSTEM.CLUSTER.TRANSMIT.QUEUE	223
Modifica della posizione del file di code utilizzando almqfls	223
Suddivisione dei file di coda	224
File di overflow messaggi	224
Buffer quando si sfogliano i messaggi	225
Altre opzioni del server di code	225
Assegnazione di CPU	226
Programmi applicativi per binding FASTPATH	228
Background	228
Riduzione del carico di MQI	228
Abilitazione del binding FASTPATH	228
Limitazioni durante l'utilizzo del binding FASTPATH	229

Capitolo 16. Integrità e disponibilità dei dati 231

Integrità dei dati	231
Disponibilità	232
Dati permanenti e non permanenti	233
Messaggi permanenti	234

Messaggi non permanenti	235
Congruenza dei database	236
Congruenza dei database interni	236
Congruenza dei database esterni	236
OpenTMF	237
NonStop Tuxedo	238
Transazioni applicative interleaf	238
File dei database più importanti di MQSeries	238
Processi principali	239
Cluster	245
Considerazioni sulla configurazione per la disponibilità	246
Considerazioni sulla configurazione per l'integrità dei dati	246

Parte 2. Riferimenti 249

Capitolo 17. I comandi di controllo di MQSeries 251

Riepilogo dei comandi di controllo	251
Utilizzo di nomi	251
Come leggere i diagrammi di sintassi	252
altmqfls (Modifica attributi del file di coda)	254
altmqusr (Modifica informazioni utente MQSeries)	259
cleanrdf (Eseguire la pulizia RDF)	261
cnvclchl (Converti le definizioni di canale client)	263
crtmqcvx (Conversione dei dati)	265
crtmqm (Crea Queue Manager)	267
dltmqm (Delete queue manager)	271
dspmqaout (Display Authority)	273
dspmqcsv (Display Command Server)	277
dspmqlfs (Visualizza gli attributi file MQSeries)	278
dspmqrtrc (Visualizza output di traccia formattato di MQSeries)	282
dspmqustr (visualizza informazioni utente MQSeries)	283
endmqcsvg (End command server)	285
endmqm (End queue manager)	288
endmqtrc (End MQSeries trace)	291
instmqm (Installa MQSeries for Compaq NSK)	292
runmqchi (Run channel initiator)	293
runmqchl (Run channel)	294
runmqdlq (Run dead-letter queue handler)	295
runmqqlsr (Run listener)	296
runmqsc (Esegue comandi di MQSeries)	298
runmqtrm (Avvia un trigger monitor)	301
setmqaut (Imposta/reimposta l'autorizzazione)	302
strmqcsvg (Avvia server di comandi)	309
strmqm (Avvio Queue Manager)	310
strmqtrc (Avvio traccia di MQSeries)	311
upgmqm (Aggiorna il gestore code V2.2.0.1)	314

Parte 3. Appendici 317

Appendice A. MQSeries for Compaq NSK al primo sguardo 319

Numero programma e parte	319
Requisiti hardware	319
Requisiti software	319

Sicurezza	320
Funzioni di manutenzione	320
Compatibilità	320
Compilatori supportati	320
Gestione delle licenze	321
Selezione della lingua	321
Internazionalizzazione	321

Appendice B. Impostazioni predefinite del sistema 323

Appendice C. Impostazione delle variabili di ambiente TAQL per MQSeries for Compaq NSK 325

Parametri di regolazione dei server di coda	327
---	-----

Appendice D. Confronto tra gruppi di comandi 329

Appendice E. Arresto e rimozione manuale dei queue manager 333

Arresto manuale di queue manager	333
Rimozione manuale dei queue manager	333

Appendice F. MQSeries e Compaq NonStop Server for Java 335

Considerazioni transazionali	335
--	-----

Appendice G. MQSC supportato da MQSeries for Compaq NSK 337

Attributi di MQSC	339
Informazioni sullo stato del canale (DISPLAY CHSTATUS)	339
MAXUMSGS e MAXHANDS	339
HARDENBO e NOHARDENBO	340
CONNAME	340
USERDATA per i programmi con la funzione trigger	340
Utilizzo dei nomi di uscita come attributi di oggetti 340	

Appendice H. Application Programming Reference. 341

Tipi di dati di struttura	341
MQCNO – Connect Options	342
MQGMO – Get Message Options	342
MQMD – Message Descriptor	343
MQPMO – Put Message Options	343
Chiamate MQI	344
MQCLOSE – Chiudi oggetto	345
MQDISC – Disconnetti il queue manager	345
MQINQ – Interroga attributi oggetto	346
MQOPEN – Apri oggetto	346
MQSET – Imposta attributi oggetto	346
Attributi degli oggetti MQSeries	346
Attributi per tutte le code	346
Attributi delle code locali e modello	347
Attributi dei queue manager	347
Conversione dei dati	347

Appendice I. Creazione ed esecuzione delle applicazioni. 349

Scrittura delle applicazioni	349
Utilizzo dell'intervallo di attesa MQGET e del canale DISCONT e HBINT	349
Gestione dell'unità di lavoro (transazione).	350
Considerazioni sulla progettazione generale	351
Interfaccia XA	351
MQGMO_BROWSE_* con MQGMO_LOCK	351
Applicazioni con la funzione trigger.	351
Linguaggi ed ambienti supportati	352
Considerazioni per la creazione di applicazioni con thread	353
Compilazione e binding delle applicazioni.	354
Messa a confronto dei binding FASTPATH e STANDARD.	354
Esecuzione delle applicazioni	355

Appendice J. MQSeries Administration Interface (MQAI) . . . 357

Appendice K. Programmi di esempio di MQSeries for Compaq NSK 359

Creazione di programmi di esempio in C	360
Non nativo (con l'utilizzo della libreria statica non nativa MQMLIB).	360
Nativo (con l'utilizzo della libreria statica nativa MQMLIBN)	360
Nativo (con l'utilizzo di SRL MQSRLLIB)	361
Creazione di programmi di esempio in C++	362
Nativo (con l'utilizzo della libreria statica nativa MQMLIBN)	362
Nativo (con l'utilizzo di SRL MQSRLLIB)	362
Creazione di programmi di esempio in COBOL	363
Non nativo (con l'utilizzo della libreria statica non nativa MQMLIB).	363
Nativo (con l'utilizzo della libreria statica nativa MQMLIBN)	363
Nativo (con l'utilizzo di SRL MQSRLLIB)	364
Creazione di programmi di esempio in TAL	365
Non nativo (con l'utilizzo della libreria statica non nativa MQMLIB).	365
Creazione dei programmi di esempio in OSS (solo modo nativo)	365

Appendice L. User exit 367

User exit supportati	367
Formato nome uscita	368
MQ_LOAD_ENTRY_POINT_EXIT - Caricamento User Exit	368
MQLXP - Struttura parametri MQ_LOAD_ENTRY_POINT_EXIT	368
Installazione user exit	372
Installazione di un'uscita nell'SRL privato di MQSeries.	372
Installazione di un'uscita nella libreria statica nativa di MQSeries	373
Installazione di un'uscita nella libreria statica non-nativa MQSeries	373

Appendice M. Impostazione delle comunicazioni 377

Canali SNA	377
Processi responder LU 6.2	378
Canali TCP/IP	383
Esempi di comunicazione	384
Esempi di comunicazione SNAX	384
Esempi di comunicazione ICE	389
Esempi di comunicazione TCP/IP	393

Appendice N. Client MQSeries 395

Supporto client.	395
Considerazioni sulla sicurezza.	395

Appendice O. Programmable System Management. 397

Eventi di strumentazione	397
Tipi di evento supportati da MQSeries for Compaq NSK	397
Formato dei messaggi di evento	398
Programmable Command Format (PCF)	399
Descrittore messaggio PCF	400
Instestazione PCF (MQCFH).	400
Parametro di stringa PCF (MQCFST)	401
Parametro integer list PCF (MQCFIL)	401
Integer PCF (MQCFIN)	401
String list PCF (MQCFSL)	401
Comandi PCF supportati da MQSeries for Compaq NSK	402
Risposte dei comandi PCF	403
Servizi installabili	403
Interfaccia di Authorization Service	403
Interfaccia di Name Service.	404

Appendice P. Modello di evento EMS utilizzato da MQSeries for Compaq NSK 405

Appendice Q. Informazioni particolari 409

Marchi	410
------------------	-----

Bibliografia 413

MQSeries pubblicazioni multiplatforma	413
Pubblicazioni MQSeries per piattaforme specifiche	413
Manuali in formato elettronico	414
Formato HTML	414
Portable Document Format (PDF)	414
Formato BookManager	415
Formato PostScript	415
Formato Windows Help	415
Informazioni su MQSeries disponibili su Internet	415
Pubblicazioni correlate	415

Glossario dei termini e delle abbreviazioni 417

Indice analitico. 427

Figure

1. Processi MQSeries for Compaq NSK	22	27. Pannello Monitor Channels	90
2. Componenti di MQSeries for Compaq NSK V5.1	26	28. Pannello Channel Status	91
3. Esempio di configurazione PATHWAY	36	29. Pannello Start/Stop Channel	92
4. Main Menu MQM di MQSeries for Compaq NSK	70	30. Pannello Reset Channel	93
5. Pannello Queue Manager Menu	71	31. Pannello Resolve Channel	94
6. Pannello 2 di Queue Manager Menu	72	32. Pannello Copy Channel	95
7. QUEUE MANAGER TRACE MENU	73	33. Code, messaggi e applicazioni	97
8. Pannello Search Criteria (queue)	74	34. Esempio output per QMGR ALL	101
9. Queue Menu	75	35. Estratto del file di comandi MQSC, mymqscin	103
10. Pannello Create Queue	76	36. Estratto del file di notifica MQSC, mymqscou	104
11. Pannello Create Local Queue	77	37. Gestione remota	128
12. Create Local Queue - pannello 2	78	38. Impostazione di canali e code per la gestione remota	129
13. Pannello Create Remote Queue	79	39. Utilizzo dei gruppi con SAFEGUARD per la concessione di autorizzazioni	143
14. Pannello Copy Queue	80	40. Esempio di dati di controllo	161
15. Pannello Display/Modify Local Queue	81	41. Esempio di regola	162
16. Pannello Monitor Local Queues	82	42. File di configurazione MQSeries di esempio (MQSINI)	188
17. Pannello Search Criteria (channel)	83	43. File di configurazione Queue manager di esempio (QMINI)	192
18. Channel Menu	83	44. Esempio di traccia	216
19. Pannello Display/Modify Sender Channel (1)	84	45. Prospetto FFST di esempio	217
20. Pannello Display/Modify Sender Channel (2)	85	46. Esempio MQLOADEXIT	371
21. Pannello Create Channel	85	47. File di impostazione MQSeries SNAX di esempio	380
22. Pannello Create Sender Channel	86	48. File di impostazione MQSeries SNA di esempio per ICE	382
23. Pannello Create Receiver Channel	87		
24. Pannello Create Server Connection Channel	87		
25. Pannello Create Cluster Sender Channel	88		
26. Pannello Create Cluster Receiver Channel	89		

Tabelle

1. eseguibili MQSeries	24	16. Comandi per la gestione di Queue Manager	329
2. Elenco di possibili ISO CCSIDs	51	17. Comandi per la gestione di server di comandi	329
3. Autorizzazione di sicurezza necessaria per chiamate MQI	151	18. Comandi per la gestione di code	330
4. Comandi MQSC e autorizzazioni di sicurezza necessari	154	19. Comandi per la gestione di processi	330
5. Comandi PCF e autorizzazioni di sicurezza necessari	155	20. Comandi per la gestione di canali.	331
6. Messa a confronto dell'overflow di coda con l'overflow di messaggio	234	21. Altri comandi di controllo	331
7. File di database controllati principali	238	22. Interfaccia con linguaggio Java.	335
8. I metodi di protezione utilizzati per i processi più importanti	240	23. MQSC supportato da MQSeries for Compaq NSK	337
9. Autorizzazioni di sicurezza del comando dspmqaut.	274	24. Riepilogo dei linguaggi e degli ambienti supportati.	352
10. Specificazione delle autorizzazioni per i diversi tipi di oggetto.	305	25. Utilizzo della corretta versione di MQI library	354
11. Esempi di file che possono essere cancellati dopo un aggiornamento	314	26. User exit supportati per MQSeries for Compaq NSK	367
12. Oggetti del sistema e oggetti predefiniti per le code	323	27. I tipi di evento supportati da MQSeries for Compaq NSK	397
13. Oggetti del sistema e oggetti predefiniti per i canali	323	28. Struttura MQMD di un messaggio di evento	398
14. Oggetti del sistema e oggetti predefiniti per elenchi di nomi	324	29. Struttura intestazione evento (MQCFH)	399
15. Oggetti del sistema e oggetti predefiniti per processi.	324	30. Descrittore messaggio PCF	400
		31. Intestazione PCF	400
		32. Parametro di stringa PCF	401
		33. Integer list PCF	401
		34. Integer PCF	401
		35. String list PCF	401
		36. Comandi PCF supportati da MQSeries for Compaq NSK	402

Informazioni su questo manuale

MQSeries for Compaq NonStop Kernel, Versione 5 Rilascio 1—in questo manuale indicato come MQSeries for Compaq NSK o MQSeries, come consentito dal contesto—è parte della famiglia di prodotti MQSeries. Questi prodotti forniscono servizi di programmazione di applicazioni che consentono ai programmi applicativi di comunicare fra essi mediante *code di messaggi*. Questa forma di comunicazione è denominata *messaging commerciale*. Le applicazioni implicate possono esistere su nodi diversi in una vasta gamma di tipi di computer e di sistemi operativi. Queste utilizzano un'interfaccia comune di programmazione di applicazioni, denominata Message Queuing Interface o MQI, in modo che i programmi sviluppati su una determinata piattaforma possano essere prontamente trasferiti a un'altra piattaforma.

Questo manuale descrive gli aspetti di gestione del sistema MQSeries for Compaq NSK, Versione 5 Rilascio 1, ed i servizi da esso forniti a supporto della messaging commerciale in un ambiente Compaq NSK. Ciò include la gestione delle code che le applicazioni utilizzano per ricevere messaggi e la garanzia che le applicazioni dispongano dell'accesso alle code necessarie.

A chi si rivolge questo manuale

In primo luogo, questo manuale è rivolto agli amministratori e ai programmatori di sistema, che gestiscono le attività di configurazione e gestione di MQSeries. Inoltre, il volume è utile ai programmatori di applicazioni, i quali necessitano di alcune cognizioni sulle attività amministrative di MQSeries.

Cosa bisogna conoscere per comprendere questo manuale

Per adoperare questo manuale occorre disporre di una buona competenza del sistema operativo Compaq NSK e delle utilità associate. Non è necessario aver già lavorato con prodotti di accodamento messaggi, ma occorre avere alcune cognizioni dei concetti fondamentali dell'accodamento messaggi.

Come utilizzare questo manuale

La parte centrale di questo volume:

- Presenta MQSeries
- Descrive la gestione quotidiana di un sistema MQSeries for Compaq NonStop Kernel, affrontando argomenti quali la gestione di oggetti locali e remoti di MQSeries, la sicurezza, il supporto per le transazioni e la determinazione di problemi

Informazioni su MQSeries su Internet

URL di MQSeries

L'URL della homepage della famiglia di prodotti MQSeries è:

<http://www.ibm.com/software/mqseries/>

MQSeries su Internet

Novità presenti in MQSeries for Compaq NSK V5.1

La seguente nuova funzione è descritta in questa edizione di *MQSeries for Compaq NSK V5.1 Guida alla gestione del sistema*.

Miglioramenti delle prestazioni

Nuovo processo del server di code

È stato introdotto un nuovo processo del server di code (nel queue manager) che consente di memorizzare i messaggi per una o più code locali e gestisce tutte le operazioni di GET e PUT su tali code. Esso fornisce un'implementazione efficace per i messaggi non permanenti e supporta le nuove funzioni di messaggistica per la Versione 5.1. Consultare la sezione "Capitolo 2. Architettura di MQSeries for Compaq NSK V5.1" a pagina 21 per ulteriori informazioni.

modifiche nell'operazione dello status server

Lo status server sostituisce l'approccio basato sul file per lo stato del canale. Supporta le informazioni sullo stato degli oggetti che non sono code locali e fornisce un accesso efficace alle informazioni sullo stato del canale. Consultare la sezione "Capitolo 2. Architettura di MQSeries for Compaq NSK V5.1" a pagina 21 per ulteriori informazioni.

messaggi non permanenti

L'utente può ora trarre vantaggio dai miglioramenti apportati alle prestazioni offerti dai messaggi non persistenti.

supporto per il binding FASTPATH per le applicazioni attendibili

Se l'applicazione lo consente, è possibile connettersi ad un queue manager utilizzando il binding FASTPATH per trarre vantaggio dai miglioramenti apportati alle prestazioni. Le applicazioni FASTPATH prevedono delle limitazioni ed è opportuno utilizzarle correttamente perché questa forma di binding fornisce minore protezione per i dati interni importanti del queue manager. Consultare la sezione "Appendice I. Creazione ed esecuzione delle applicazioni" a pagina 349 per ulteriori informazioni.

miglioramenti apportati alla memorizzazione sul disco per i messaggi permanenti

La memorizzazione dei messaggi permanenti sul disco è stata modificata per migliorare le prestazioni per i messaggi di tutte le dimensioni. Per i file di coda non sono richiesti file chiave alternati; inoltre, è stato introdotto un nuovo tipo di memorizzazione sul disco per i messaggi molto lunghi. In tal modo, viene ottimizzata l'efficacia della memorizzazione per i messaggi fino a 100 MB.

Funzioni di MQSeries aggiornate

Cluster del Queue Manager di MQSeries

È possibile connettere i Queue Manager di MQSeries per formare un cluster di Queue Manager. All'interno di un cluster, i Queue Manager possono rendere le code ospitate disponibili a tutti gli altri Queue Manager. Qualsiasi Queue Manager può inviare un messaggio a tutti gli altri Queue Manager nello stesso cluster senza la necessità di definizioni di

canale esplicite, definizioni di coda remota o code di trasmissione per ciascuna destinazione. I principali vantaggi dei cluster di MQSeries sono:

- Minor numero di attività di gestione del sistema
- Disponibilità accresciuta
- Bilanciamento del carico di lavoro

Per informazioni su questa funzione, consultare il manuale *MQSeries Queue Manager Clusters*.

MQSeries Administration Interface (MQAI)

MQSeries for Compaq NSK V5.1 ora supporta MQSeries Administration Interface (MQAI), un'interfaccia di programmazione che semplifica l'utilizzo dei messaggi PCF per configurare MQSeries. Per ulteriori informazioni su MQAI e per la descrizione dei comandi, consultare *MQSeries Administration Interface Programming Guide and Reference*.

Supporto MQI avanzato

MQSeries for Compaq NSK V5.1 ora supporta la funzione di messaggistica avanzata fornita nelle Versioni 5.0 e 5.1 di MQSeries su altre piattaforme. Ciò include l'elaborazione di elenchi di distribuzione, i messaggi di riferimenti, i messaggi segmentati e molte altre nuove opzioni. Per ulteriori informazioni, consultare *MQSeries Application Programming Guide e MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

aumento della dimensione dei messaggi e delle code di messaggi

I messaggi possono contenere fino a 100 MB. Le code di messaggi possono contenere fino a 4 GB.

creazione automatica di oggetti predefiniti

Quando si utilizza il comando **crmqm** per creare un queue manager, gli oggetti predefiniti del sistema vengono creati automaticamente.

arresto sincrono, controllato di un queue manager

È stata aggiunta una nuova opzione al comando **endmqm** per consentire l'arresto sincrono controllato di un Queue Manager.

Supporto Java

MQSeries for Compaq NSK V5.1 ora può essere utilizzato con i compilatori Java, che consentono la codifica delle applicazioni in Java. Consultare la sezione "Appendice F. MQSeries e Compaq NonStop Server for Java" a pagina 335 per ulteriori informazioni.

supporto per applicazioni OSS

MQSeries for Compaq NSK V5.1 ora può essere utilizzato con le applicazioni NSK OSS che utilizzano C, C++, Cobol e Java.

Gestione Web

Con MQSeries for Compaq NSK V5.1 è possibile eseguire le seguenti attività utilizzando un sistema Microsoft® Windows NT® ed un browser HTML, ad esempio Netscape Navigator o Microsoft Internet Explorer:

- Connettersi come Amministratore di MQSeries
- Selezionare un Queue Manager ed inviare ad esso comandi MQSC
- Creare, modificare ed eliminare script MQSC.

migliorato meccanismo di uscite utente

Il meccanismo per il binding e la configurazione del codice delle uscite utente da utilizzare con MQSeries è stato notevolmente migliorato per fornire un'interfaccia più vicina agli standard ed un meccanismo comune per tutte le uscite utente. Consultare la sezione "Appendice L. User exit" a pagina 367 per ulteriori informazioni.

Intercomunicazioni

TCP/IP

MQSeries for Compaq NSK V5.1 consente ora ad un queue manager di utilizzare processi server TCP/IP Guardian multipli. Ciò si traduce in migliori configurazioni per il bilanciamento del carico sull'hardware della rete e produce ridondanza delle connessioni di rete per il queue manager e per le applicazioni. Consultare la sezione "Appendice M. Impostazione delle comunicazioni" a pagina 377 per ulteriori informazioni.

SNA

MQSeries for Compaq NSK V5.1 dispone di un meccanismo migliorato per la gestione e il controllo dell'inizializzazione remota dei canali per il protocollo SNA. Questo nuovo meccanismo utilizza un processo listener eseguito nel PATHWAY ed è supportato per i prodotti SNAX e InSession ICE. L'attributo di canale non standard AUTOSTART non è più supportato. Consultare la sezione "Appendice M. Impostazione delle comunicazioni" a pagina 377 per ulteriori informazioni.

Canali I canali ora supportano gli *heartbeat* e la possibilità di trasmettere i messaggi non permanenti al di fuori di un'unità di lavoro per garantire migliori prestazioni.

MQSeries for Compaq NSK V5.1 ora supporta la definizione facoltativa automatica dei canali per i canali inizializzati in modo remoto da altri queue manager o client.

Facilità di utilizzo di Compaq NSK

Compaq NSK Fix Command incluso con `runmqsc`

`runmqsc` ora include la funzione Compaq NSK Fix Command che consente di richiamare e modificare i comandi MQSC. Per ulteriori informazioni, consultare il "Utilizzo del comando NSK Fix di Compaq" a pagina 30.

Programma di utilità `altnmqfls` avanzato

Il programma di utilità `altnmqfls` è stato modificato notevolmente per fornire la gestione dettagliata delle opzioni di memorizzazione per i messaggi. Per ulteriori informazioni, consultare "Capitolo 15. Scalabilità e prestazioni" a pagina 219 e "`dspmqfls` (Visualizza gli attributi file MQSeries)" a pagina 278.

Pannelli MQMC

I pannelli di gestione MQMC forniti come parte dell'ambiente PATHWAY di queue manager PATHWAY sono stati aggiornati per supportare le funzioni avanzate di MQSeries in questo rilascio.

Parte 1. Guida

Capitolo 1. Introduzione	7
MQSeries e accodamento di messaggi	7
Applicazioni indipendenti dal tempo	7
Elaborazione dei messaggi guidati	8
Messaggi e code	8
Che cos'è un messaggio?	8
Dimensione dei messaggi	8
Cos'è una coda?	9
In che modo le applicazioni inviano e ricevono i messaggi?	9
Code predefinite e dinamiche.	9
Richiamo dei messaggi dalle code	10
Oggetti	10
Nomi degli oggetti	10
Gestione degli oggetti	11
Attributi degli oggetti	11
Queue Manager di MQSeries	11
Chiamate MQI	12
Code di MQSeries	12
Utilizzo di oggetti coda	12
Code locali utilizzate da MQSeries.	13
Definizioni di processo	15
Canali	15
Cluster	16
Namelist	16
Oggetti definiti dal sistema	16
Gestione	17
Gestione locale e remota	17
Client e server	17
Applicazioni MQI in un ambiente client-server	18
Estensione delle funzioni di Queue Manager	18
User exit	18
Servizi installabili	18
Sicurezza	19
Funzione OAM (Object Authority Manager)	19
Supporti transazionali	19
Ottimizzazione delle prestazioni, affidabilità, scalabilità e dimensioni	20
Capitolo 2. Architettura di MQSeries for Compaq NSK V5.1	21
Panoramica dei processi queue manager.	21
Impacchettamento del prodotto.	23
Eseguibili	24
Messa a confronto dell'architettura della Versione 5.1 e della Versione 2.2.0.1	24
Queue manager — panoramica funzionale	25
Modello di processo di Queue manager	26
File e volumi secondari di MQSeries	27
Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries	29
Esecuzione delle attività di gestione utilizzando i comandi di controllo	29
Utilizzo dei comandi di controllo	29

Esecuzione delle attività di gestione utilizzando comandi MQSC	30
Esecuzione dei comandi MQSC.	30
Utilizzo del comando NSK Fix di Compaq	30
Esecuzione delle attività di gestione utilizzando comandi PCF	31
Attributi MQSC e PCF.	31
Escape PCF	31
Gestione di TS/MP (PATHWAY)	32
Specificazione e controllo dei listener TCP/IP	32
Controllo del server di comando	32
Specificazione e controllo dei channel initiator.	32
Modifica della coda di iniziazione predefinita per channel initiator	33
Specificazione e controllo dei trigger monitor	33
Specificazione della distribuzione di processi tra CPU.	33
Aggiunta di nuovi processi di status server.	35
Aggiunta di nuovi processi del server di coda.	35
Specificazione della frequenza di aggiornamento dei pannelli monitor MQM	35
Configurazione di PATHWAY per un Queue Manager	35
Modifica dei parametri delle classi di server PATHWAY.	42
Aggiunta di classi di server definite dall'utente a un PATHWAY di MQSeries	43
Capitolo 4. Gestione dei Queue Manager	45
Introduzione	45
Come creare un queue manager	45
Eseguire una copia di riserva dei file di configurazione dopo aver creato un Queue Manager	47
Modifica delle proprietà del queue manager	48
Volume iniziale del queue manager	48
Numero di processi EC per queue manager	48
Ripartizione del carico di lavoro nell'ambito del sistema	49
Terminale iniziale del queue manager	49
Il nome di processo PATHMON per il queue manager	49
Il CCSID del queue manager	51
Controllo della conversione dei dati EBCDIC	51
EMS Collector per il queue manager	52
Il pool di agenti tenuto pronto da ciascun EC nel queue manager	52
Limite massimo di riutilizzo processi e agenti inattivi	52
Priorità dei processi queue manager	53
Numero massimo di canali per il queue manager	53
Numero massimo di canali attivi per il queue manager	53
ID di segmento Guardian utilizzati da MQSeries	54
Porta TCP/IP predefinita	54

Porte TCP/IP su cui è in ascolto il queue manager	54	Supporto dei programmi applicativi che utilizzano MQI.	97
Processo TCP/IP utilizzato dal queue manager	55	Esecuzione delle attività di gestione locale utilizzando i comandi MQSC	98
Riconfigurazione della classe server MQS-TCPLISnn per una porta ed un processo TCP/IP non predefiniti	55	Prima di iniziare.	98
Assegnazione di spazio di swap	56	Nomi degli oggetti MQSeries	98
Nome status server predefinito	56	Sensibilità al maiuscolo/minuscolo nei comandi MQSC	99
Aggiunta e rimozione di status server non predefiniti	57	Utilizzo interattivo della funzione MQSC	99
Nome queue server predefinito.	59	Feedback dei comandi MQSC	100
Aggiunta e rimozione di queue server non predefiniti	60	Termine dell'input interattivo MQSC	100
Struttura dei volumi	61	Visualizzazione degli attributi Queue Manager	100
Volume secondario FFST del queue manager	61	Utilizzo di un Queue Manager diverso dal predefinito	101
Volume secondario file di dati queue manager.	61	Modifica degli attributi di Queue Manager	101
Volume secondario log errori queue manager	62	Esecuzione di comandi MQSC da file di testo	102
Volume secondario coda messaggi del queue manager	63	Utilizzo dei file di comandi MQSC	103
Volume secondario sincronizzazione canali del queue manager	63	Utilizzo dei prospetti MQSC	103
Trasformazione dei nomi di oggetto	63	Esecuzione dei file di comandi MQSC forniti	104
Lavorare con i gestori code	64	Utilizzo di runmqsc per la verifica dei comandi	104
Creazione di un Queue Manager predefinito	64	Risoluzione dei problemi con MQSC	105
Creazione dei principal di MQSeries	65	Utilizzo delle code locali	106
Esecuzione di cleanrdf per un queue manager abilitato a RDF	66	Definizione di una coda locale.	106
Avvio di un queue manager.	66	Modifica della dimensione di file fisico per le code	107
Ripristino degli oggetti predefiniti e di sistema	66	Definizione di una coda messaggi non recapitati	107
Analisi dei file di oggetto.	66	Visualizzazione degli attributi predefiniti di oggetti.	108
Interruzione di un queue manager.	67	Copia della definizione di una coda locale.	109
In caso di problemi.	67	Modifica degli attributi di coda locale	109
Chiusure immediate e prioritarie di un queue manager	67	Modifica del volume di una coda locale	110
Riavvio di un Queue Manager	68	Modifica delle opzioni di una coda locale	110
Rendere predefinito un Queue Manager esistente.	68	Riassegnazione degli oggetti agli status server e ai queue server	111
Eliminazione di un queue manager	68	Annullamento di una coda locale.	111
Utilizzo della funzione MQM (Message Queue Management).	69	Eliminazione di una coda locale	112
Utilizzo del Queue Manager Menu	70	Esame delle code	112
Modifica degli attributi di Queue Manager	72	Utilizzo degli alias di coda	115
Traccia degli oggetti di MQSeries	72	Definizione di un alias di coda	115
Utilizzo del menu Queues	74	Utilizzo di altri comandi con gli alias di coda	116
Creazione di una coda.	75	Utilizzo delle code di esempio.	117
Copia di una coda	79	Definizione di una coda di esempio	117
Modifica di una coda	80	Utilizzo degli altri comandi con le code di esempio	117
Cancellazione di una coda	81	Gestione degli oggetti per triggering.	118
Controllo di una coda	81	Definizione di una coda di applicazione per triggering.	118
Utilizzo del menu Channels	82	Definizione di una coda di iniziazione	119
Modifica di un canale	84	Creazione di una definizione di processo	119
Creazione di una definizione di canale	85	Visualizzazione della definizione di processo	120
Controllo di un canale.	89		
Cancellazione di un canale	90	Capitolo 6. Automazione delle attività di gestione.	121
Visualizzazione dello stato di un canale	90	Comandi PCF	121
Avvio e arresto di un canale.	91	Attributi MQSC e PCF	122
Reimpostazione di un MSN (Message Sequence Number)	92	Escape PCF	122
Risoluzione di un canale	93	Utilizzo di MQAI per semplificare l'impiego di PCF	122
Copia di un canale	94	Gestione del server di comando per una gestione remota	123
		Avvio del server di comando	123
Capitolo 5. Gestione di oggetti MQSeries locali	97		

Visualizzazione dello stato del server di comando	123	Cosa specificare quando si utilizzano i comandi OAM	144
Arresto del server di comando	124	Elenchi di autorizzazioni	144
Capitolo 7. Gestione di oggetti remoti di MQSeries	125	Utilizzo del comando almqusr	144
Canali, cluster e accodamento remoto	125	Utilizzo del comando dspmqusr	145
Gestione remota mediante cluster	126	Utilizzo del comando setmqaut	145
Gestione di un Queue Manager remoto.	127	Comandi di autorizzazione e servizi installabili	146
Preparazione dei Queue Manager alla gestione remota	127	Autorizzazioni di accesso	146
Preparazione di canali e code di trasmissione per la gestione remota	128	Comando di visualizzazione autorizzazioni	147
Definizione di canali e code di trasmissione	129	Istruzioni per OAM (Object Authority Manager)	147
Avvio dei canali	130	ID utente	147
Definizione automatica dei canali.	131	Volumi del queue manager	147
Emissione remota di comandi MQSC	131	Code	148
Funzionamento dei Queue Manager su MVS/ESA	132	Autorizzazione per utente alternativo	148
Consigli per l'accodamento remoto	132	Autorizzazione contestuale	148
Eventuali problemi nell'utilizzo remoto di MQSC	132	Considerazioni sulla sicurezza remota	149
Creazione di una definizione locale di una coda remota	133	Sicurezza del comando di canale	150
Analisi del funzionamento delle definizioni locali di code remote	133	Comandi PCF	150
Esempio	133	Comandi canale MQSC	150
Come funziona	133	Comprendere le tabelle di specifica delle autorizzazioni	150
Un modo alternativo per inserire messaggi in una coda remota	135	Autorizzazioni MQI	151
Utilizzo di altri comandi con code remote	135	Autorizzazioni di gestione	154
Creazione di una coda di trasmissione	135	Autorizzazioni per comandi MQSC nei PCF escape	154
Code di trasmissione predefinite	135	Autorizzazioni per comandi PCF	155
Utilizzo di definizioni di coda remota come alias	136	Comprendere i file di autorizzazione	156
Alias di Queue Manager	136	Il database dei principal	156
Alias di coda di risposta RTQ	136	Il database OAM	156
Conversione dei dati	137	Record di autorizzazione classe	158
Quando un Queue Manager non è in grado di convertire messaggi in formati incorporati	137	Record di autorizzazione per tutte le classi	158
File CCSID	137	Capitolo 9. Handler di code messaggi non recapitati di MQSeries	159
Conversione di dati predefinita	137	Richiamo del DLQ handler	159
Conversione di messaggi in formati definiti dall'utente	138	Tabella regole DLQ handler	160
Modifica del CCSID del Queue Manager	138	Dati di controllo	160
Capitolo 8. Protezione di oggetti di MQSeries	139	Regole (schemi e azioni).	162
Perché bisogna proteggere le risorse di MQSeries	139	Parole chiave dello schema di confronto	162
ID utenti nel gruppo utenti MQM	139	Parole chiave di azione	163
Come ottenere ulteriori informazioni	140	Convenzioni relative alla tabella regole	165
Object Authority Manager (OAM)	140	Elaborazione della tabella regole	166
Come funziona OAM.	141	Metodi per l'elaborazione di tutti i messaggi presenti nella coda DLQ.	167
Gestione dell'accesso mediante i gruppi utente	141	Esempio di tabella regole per DLQ handler	168
Quando un utente appartiene a più di un gruppo utenti	141	Capitolo 10. Eventi EMS e di strumentazione	171
Set di gruppi ed il gruppo principale	141	Eventi di strumentazione MQSeries	171
Protezione delle risorse con l'OAM	142	Tipi di eventi	171
Utilizzo dei gruppi per le autorizzazioni	142	Notificazione di evento attraverso code di evento.	172
Disabilitazione dell'OAM (Object Authority Manager).	143	Utilizzo delle code di evento con la funzione trigger.	172
Utilizzo dei comandi OAM (Object Authority Manager).	144	Abilitazione degli eventi di strumentazione	172
		Messaggi di evento	173
		Eventi EMS (Event Management Service)	173
		I file modello EMS forniti con MQSeries for Compaq NSK	173
		Integrazione dei modelli di evento EMS MQSeries.	174

Definizione di PARAM MQEMSEVENTS	176	Il problema si verifica ad una determinata ora del giorno?	202
Utilizzo di un collector alternativo	177	Il problema si manifesta in modo discontinuo?	202
Scrittura dei programmi per l'elaborazione degli eventi EMS MQSeries	177	Sono stati eseguiti aggiornamenti del servizio?	203
Capitolo 11. Descrizione del supporto transazionale e della messaggistica	179	Errori di programmazione frequenti	203
Utilizzo di NonStop TM/MP (Transaction Manager).	179	Problemi relativi ai comandi	203
Limiti di Syncpoint	180	Come procedere	204
Operazioni diverse da syncpoint in messaggi permanenti	180	Viene riportato un output non corretto?	204
Operazioni Syncpoint o messaggi non permanenti	181	Non si è ottenuta alcuna risposta utilizzando un comando PCF?	204
Requisiti della configurazione per TM/MP e MQSeries for Compaq NSK	181	Si verificano malfunzionamenti in alcune code?	205
Monitoring	181	Il problema è relativo alle sole code remote?	206
Dimensione traccia di controllo	181	L'esecuzione dell'applicazione o di MQSeries for Compaq NSK procede lentamente?	206
Configurazione gestore risorse.	182	Considerazioni sulla progettazione delle applicazioni	207
Risoluzione dei problemi	182	Lunghezza del messaggio	207
Capitolo 12. Ripristino e riavvio	183	Ricerca di un particolare messaggio	207
Ripristino e tolleranza di guasti	183	Code contenenti messaggi di diversa lunghezza	208
Back up e ripristino di MQSeries	184	Frequenza di syncpoint	208
Esecuzione del backup di MQSeries	184	Utilizzo della chiamata MQPUT1.	208
Ripristino di MQSeries	184	Output non corretto	208
Ripristino e riavvio degli status server e dei queue server	185	Messaggi assenti nella coda	208
Ripristino mediante RDF	185	Messaggi che contengono informazioni inattese o alterate	210
Capitolo 13. File di configurazione	187	Problemi relativi ad output non corretti durante l'accodamento distribuito	210
Cosa sono i file di configurazione?	187	Log di errore	212
File di configurazione MQSeries (MQSINI)	187	File di log	212
Contenuto del file di configurazione MQSeries	187	Errori precedenti	213
File di configurazione Queue manager (QMINI)	189	Messaggi per l'operatore	213
Contenuto del file di configurazione Queue manager	189	Decodifica dei numeri EC nel file MQERRLG	213
File di configurazione Queue manager di esempio	191	Esempio di log degli errori	213
Modifica dei file di configurazione	197	Eventi EMS	214
Implementazione delle modifiche ai file di configurazione	197	Code messaggi non recapitati	215
Suggerimenti relativi ai file di configurazione	197	File di configurazione e determinazione dei problemi	215
Capitolo 14. Determinazione dei problemi	199	Utilizzo della traccia MQSeries	215
Controlli preliminari	199	Nomi dei file di traccia	215
Si è mai avuta in precedenza un'esecuzione con esito positivo di MQSeries?	199	Esempio di dati traccia	216
Vengono riportati messaggi di errore?	199	FFST (First Failure Support Technology™)	216
Vengono riportati codici di errori che spiegano il problema?	200	Esame dei FFST	216
Il problema è riproducibile?	200	Capitolo 15. Scalabilità e prestazioni.	219
Sono state effettuate modifiche dall'ultima esecuzione valida?	200	Introduzione	219
Si è mai avuta in precedenza un'esecuzione con esito positivo dell'applicazione?	201	Progettazione di nuove applicazioni per prestazioni e scalabilità	219
Se l'applicazione in precedenza non è stata eseguita con esito positivo	201	Progettazione per ridurre o eliminare l'utilizzo di risorse condivise	219
Il problema è relativo a parti specifiche della rete?	202	Ottimizzazione delle prestazioni iterativa	220
		Messaggi permanenti	220
		Messaggi non permanenti	220
		Canali e messaggi non permanenti	221
		Server di code e file di code	222
		Memorizzazione di messaggi permanenti	222
		Memorizzazione dei messaggi non permanenti	222
		Distribuzione della CPU del server di code	222
		Riassegnazione delle code ai server di code utilizzando almqfls	223
		Coda di trasmissione cluster: SYSTEM.CLUSTER.TRANSMIT.QUEUE	223

Modifica della posizione del file di code utilizzando altnmqfls	223
Suddivisione dei file di coda	224
File di overflow messaggi	224
Buffer quando si sfogliano i messaggi	225
Altre opzioni del server di code	225
Caricamento all'avvio	225
Conservazione nella cache	226
Controllo NPM.	226
Measure Counter (calcoladimensioni)	226
Assegnazione di CPU	226
Programmi applicativi per binding FASTPATH	228
Background	228
Riduzione del carico di MQI	228
Abilitazione del binding FASTPATH.	228
Limitazioni durante l'utilizzo del binding FASTPATH	229
Capitolo 16. Integrità e disponibilità dei dati	231
Integrità dei dati	231
Disponibilità	232
Dati permanenti e non permanenti	233
Messaggi permanenti.	234
Messaggi non permanenti	235
Congruenza dei database	236
Congruenza dei database interni	236
Congruenza dei database esterni	236
OpenTMF	237
NonStop Tuxedo	238
Transazioni applicative interleaf	238
File dei database più importanti di MQSeries	238
Processi principali	239
Cluster	245
Considerazioni sulla configurazione per la disponibilità.	246
Considerazioni sulla configurazione per l'integrità dei dati	246

Capitolo 1. Introduzione

Questo capitolo illustra MQSeries for Compaq NonStop Kernel Versione 5.1 (MQSeries for Compaq NSK V5.1) dalla prospettiva di gestione descrivendo i concetti fondamentali relativi a MQSeries e alla messaggistica. Esso contiene le seguenti sezioni:

- "MQSeries e accodamento di messaggi"
- "Messaggi e code" a pagina 8
- "Oggetti" a pagina 10
- "Code di MQSeries" a pagina 12
- "Definizioni di processo" a pagina 15
- "Canali" a pagina 15
- "Oggetti definiti dal sistema" a pagina 16
- "Gestione" a pagina 17
- "Client e server" a pagina 17
- "Estensione delle funzioni di Queue Manager" a pagina 18
- "Sicurezza" a pagina 19
- "Supporti transazionali" a pagina 19
- "Ottimizzazione delle prestazioni, affidabilità, scalabilità e dimensioni" a pagina 20

MQSeries e accodamento di messaggi

MQSeries consente ai programmi applicativi di utilizzare l'accodamento di messaggi per partecipare all'elaborazione di messaggi guidati. I programmi applicativi possono comunicare attraverso diverse piattaforme utilizzando appropriati prodotti software per l'accodamento dei messaggi. Ad esempio, le applicazioni Compaq NSK e MVS/ESA™ possono comunicare tra loro attraverso MQSeries for Compaq NSK e MQSeries per OS/390®. Le applicazioni sono protette dai meccanismi delle comunicazioni di base.

I prodotti MQSeries implementano un'interfaccia comune di programmazione applicativa denominata MQI (Message Queue Interface), indipendentemente dalla piattaforma su cui sono eseguite le applicazioni. Ciò facilita il trasferimento delle applicazioni da una piattaforma all'altra.

MQI è descritta in dettaglio nel manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

Applicazioni indipendenti dal tempo

Grazie all'accodamento, lo scambio dei messaggi tra i programmi di invio e quelli di ricezione può avvenire in modo indipendente dal tempo. Ciò significa che le applicazioni di invio e di ricezione sono disaccoppiate, di conseguenza l'applicazione di invio può continuare l'elaborazione senza attendere la conferma di ricezione del messaggio da parte dell'applicazione ricevente. In effetti, non è necessario che l'applicazione di destinazione sia in esecuzione quando viene inviato il messaggio. Tale applicazione potrà ricevere il messaggio quando verrà avviata.

MQSeries e accodamento di messaggi

Elaborazione dei messaggi guidati

Una volta giunti a una coda, i messaggi possono avviare l'applicazione automaticamente utilizzando un meccanismo denominato *triggering*. Se necessario, le applicazioni possono essere arrestate una volta terminata l'elaborazione dei messaggi.

Messaggi e code

I messaggi e le code sono gli elementi di base di un sistema di accodamento.

Che cos'è un messaggio?

Un *messaggio* è un stringa di byte che assume significato per l'applicazione in cui viene utilizzato. I messaggi vengono utilizzati per il trasferimento di informazioni tra le applicazioni o tra diverse parti della stessa applicazione. Le applicazioni possono essere eseguite sulla stessa piattaforma o su piattaforme diverse.

I messaggi MQSeries si compongono di due parti:

- *dati applicativi*

Il contenuto e la struttura dei dati è definito dal programma applicativo che li utilizza.

- *descrittore messaggio*

Il descrittore messaggio identifica il messaggio e contiene altre informazioni di controllo, quali il tipo di messaggio e la priorità assegnata al messaggio dall'applicazione che lo invia.

Il formato del descrittore messaggio è definito da MQSeries. Per informazioni più dettagliate sul descrittore messaggio, consultare la guida *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

Il formato del descrittore messaggio è definito da MQSeries. Per informazioni più dettagliate del descrittore messaggio, consultare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

Esistono due tipi di messaggio: messaggi permanenti e messaggi non permanenti. Essi si differenziano nei seguenti modi:

- I messaggi permanenti sopravvivono al riavvio di un Queue Manager. I messaggi non permanenti non sopravvivono al riavvio di un Queue Manager.
- I messaggi non permanenti di norma non vengono scritti sul disco e quindi l'aggiunta e la rimozione di tali messaggi da una coda sono più rapide e utilizzano un numero minore di risorse rispetto ai messaggi permanenti.
- In determinate condizioni di malfunzionamento, i messaggi non permanenti non sono affidabili quanto i messaggi permanenti.
- Non è possibile inserire i messaggi permanenti in una coda dinamica temporanea.

Dimensione dei messaggi

In MQSeries, la dimensione massima di un messaggio è 100 MB (dove 1 MB equivale a 1 048 576 byte). La dimensione del messaggio può essere limitata da:

- la dimensione massima di messaggio definita per la coda ricevente.
- la dimensione massima di messaggio definita per il Queue Manager.
- la dimensione massima di messaggio definita da entrambe le applicazioni di ricezione e di invio.
- lo spazio di memoria disponibile per il messaggio.

Per trasferire tutte le informazioni richieste da un'applicazione potrebbe essere necessario l'invio di più messaggi.

L'aumento delle dimensioni massime consentite per un messaggio potrebbe provocare conseguenze negative. Inoltre, le dimensioni del messaggio potrebbero risultare eccessive per la coda o il Queue Manager. In tale evenienza, è possibile dividere un messaggio in segmenti e quindi raggrupparle questi ultimi in un messaggio logico. Il raggruppamento logico dei messaggi consente alle applicazioni di raggruppare messaggi simili e di garantire la sequenza dei messaggi. Per ulteriori informazioni sulla segmentazione e il raggruppamento dei messaggi, consultare il manuale *MQSeries Application Programming Guide*.

Cos'è una coda?

Una *coda* è una struttura di dati utilizzata per memorizzare messaggi. I messaggi possono essere inseriti in una coda da programmi applicativi o da Queue Manager come una normale operazione.

Ogni coda è gestita da un *Queue Manager*. Il Queue Manager è responsabile della gestione delle code di sua pertinenza e della memorizzazione nelle code appropriate di tutti i messaggi ricevuti.

La dimensione massima di una coda è 4 GB. Per informazioni relative alla pianificazione del quantitativo di memoria necessario alle code, consultare *MQSeries Planning Guide* oppure visitare il seguente sito web per ottenere le relazioni sulle prestazioni specifiche delle piattaforme.

<http://www.ibm.com/software/mqseries/txppacs/txpm1.html>

In che modo le applicazioni inviano e ricevono i messaggi?

Le applicazioni inviano e ricevono i messaggi utilizzando le *chiamate MQI*. Per inviare un messaggio a una coda, ad esempio, l'applicazione:

1. Apre la coda di destinazione inoltrando una chiamata MQI MQOPEN.
2. Inoltra una chiamata MQI MQPUT per inserire il messaggio nella coda.
3. Un'altra applicazione può richiamare il messaggio dalla stessa coda inoltrando una chiamata MQI MQGET.

Per ulteriori informazioni sulle chiamate MQI, consultare il manuale *MQSeries for Tandem Non Stop Kernal, V5.1 Guida operativa*.

Code predefinite e dinamiche

Le code possono essere caratterizzate in base al modo in cui vengono create:

- Le *Code predefinite* sono create dall'amministratore del sistema utilizzando l'appropriato gruppo di comandi. Ad esempio, il comando DEFINE QLOCAL crea una coda locale predefinita. Le code predefinite sono permanenti; ciò significa che esistono indipendentemente dalle applicazioni che le utilizzano e sopravvivono al riavvio di MQSeries.
- Le *Code dinamiche* sono create da un'applicazione che inoltra una richiesta OPEN specificando il nome di una *coda modello*. La coda così creata è basata su una definizione di coda modello, che è la coda modello. È possibile creare una coda modello utilizzando il comando MQSC DEFINE QMODEL. Gli attributi di una coda modello, ad esempio il numero massimo di messaggi memorizzabili nella coda, vengono trasferiti a tutte le code dinamiche create in base a quel modello.

Messaggi e code

Le code modello hanno un attributo che specifica se la coda dinamica deve essere permanente o temporanea. Le code permanenti sopravvivono all'applicazione e al riavvio; le code temporanee vanno perse quando si riavvia MQSeries.

Richiamo dei messaggi dalle code

In MQSeries, le applicazioni opportunamente autorizzate possono richiamare messaggi da una coda sulla base di questi algoritmi di richiamo:

- FIFO (First-In-First-Out).
- Priorità di messaggio, come definito nel descrittore di messaggio. I messaggi con pari priorità sono richiamati in base al FIFO.
- La richiesta di un programma relativa a un messaggio specifico.

La richiesta MQGET dell'applicazione determina il metodo utilizzato.

Oggetti

Molte operazioni descritte in questo manuale prevedono la gestione di *oggetti* MQSeries. In MQSeries Versione 5.1, i tipi di oggetti comprendono Queue Manager, code, definizioni di processi, canali, cluster e namelist.

La manipolazione o la *gestione* di oggetti prevede:

- Avvio e arresto dei Queue Manager
- Creazione di oggetti, in particolare code, per le applicazioni.
- Utilizzo dei canali per creare percorsi di comunicazione verso i Queue Manager di altri sistemi remoti. Tutto ciò è illustrato in dettaglio nel manuale *MQSeries Intercommunication*.
- Creazione di *cluster* di Queue Manager per semplificare i processi di gestione generali o per ottenere un bilanciamento del carico di lavoro.

Questo manuale contiene informazioni dettagliate relative alla gestione nei seguenti capitoli:

- “Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries” a pagina 29
- “Capitolo 4. Gestione dei Queue Manager” a pagina 45
- “Capitolo 5. Gestione di oggetti MQSeries locali” a pagina 97
- “Capitolo 7. Gestione di oggetti remoti di MQSeries” a pagina 125

Nomi degli oggetti

Ogni istanza di un Queue Manager è caratterizzata da un nome di oggetto. Tale nome di oggetto deve essere univoco all'interno della rete di Queue Manager per l'adeguata identificazione del Queue Manager di destinazione a cui viene inviato un messaggio.

Il nome oggetto deve essere univoco all'interno di un Queue Manager e un tipo di oggetto. Ad esempio, è possibile che esistano una coda e un processo con lo stesso nome, mentre è impossibile che esistano due code con lo stesso nome.

Un nome oggetto può essere composto da un massimo di 48 caratteri, ad eccezione dei *canali*. Gli oggetti canale possono essere composti da un massimo di 20 caratteri. Per ulteriori informazioni sui nomi, consultare la sezione “Utilizzo di nomi” a pagina 251.

Gestione degli oggetti

MQSeries fornisce funzioni per la creazione, la modifica, la visualizzazione e l'eliminazione di oggetti. Esse comprendono:

- Comandi MQSC (MQSC), che possono essere immessi digitandoli dalla tastiera oppure leggendoli da un file
- MQM (interfaccia basata sullo schermo)
- Comandi PCF (Programmable Command Format), che possono essere utilizzati da un programma.
- Comandi di controllo, che è possibile immettere in modo interattivo dalla riga comandi del sistema operativo.

Per ulteriori informazioni, consultare la sezione "Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries" a pagina 29.

Attributi degli oggetti

Le proprietà di un oggetto sono definite dai relativi attributi di oggetto. È possibile specificare o modificare alcuni attributi di oggetto, ma solo visualizzarne altri. Ad esempio, la dimensione massima di messaggio che una coda può supportare è definita dal relativo attributo *MaxMsgLength*. È possibile specificare questo attributo di oggetto durante la creazione di una coda. L'attributo *DefinitionType* specifica il modo in cui è stata creata la coda. È possibile visualizzare solo l'attributo *DefinitionType*.

In MQSeries, esistono due modi di far riferimento a un attributo di oggetto:

- Utilizzando il relativo nome PCF, ad esempio, *MaxMsgLength*. Il nome PCF è il nome formale di un attributo.
- Utilizzando il relativo nome MQSC, ad esempio, *MAXMSGL*.

Il nome formale di un attributo è il relativo nome PCF. Dal momento che l'utilizzo della funzione MQSC costituisce una parte fondamentale di questo manuale, il nome MQSC di un dato attributo sarà più ricorrente di quello PCF negli esempi forniti.

Queue Manager di MQSeries

Un Queue Manager fornisce servizi di accodamento messaggi alle applicazioni. Esso garantisce che:

- Gli attributi degli oggetti vengano modificati in base ai comandi ricevuti.
- Eventi particolari, quali gli eventi trigger o gli eventi di strumentazione, vengano generati solo quando esistono le condizioni ideali.
- I messaggi vengano inseriti nella coda appropriata, così come richiesto dall'applicazione che ha generato la chiamata MQPUT. L'applicazione viene informata nel caso ciò risulti impossibile e viene visualizzato il codice di errore appropriato.

Ogni coda appartiene a un determinato Queue Manager e rispetto a questo viene definita *coda locale*. Il Queue Manager, a cui si connette un'applicazione, è definito Queue Manager locale rispetto a quella applicazione. Rispetto all'applicazione, le code che appartengono al Queue Manager locale, si definiscono code locali.

Una *coda remota* è semplicemente una coda che appartiene a un altro Queue Manager.

Oggetti

Un *Queue manager remoto* è un qualunque Queue Manager diverso da quello locale. Un Queue Manager remoto può risiedere su una macchina remota attraverso la rete oppure può risiedere su una determinata macchina come Queue Manager locale.

MQSeries supporta diversi Queue Manager sulla stessa macchina.

Chiamate MQI

Un oggetto Queue Manager può essere utilizzato per varie chiamate MQI. Ad esempio, è possibile richiedere informazioni relative agli attributi di oggetto mediante la chiamata MQINQ MQI.

Nota: I messaggi vengono sempre inseriti negli oggetti coda piuttosto che agli oggetti Queue Manager. Non è possibile inserire un messaggio in un oggetto Queue Manager.

Code di MQSeries

Le code vengono definite in MQSeries for Compaq NSK utilizzando:

- Comandi MQSC DEFINE
- Funzione MQM (Message Queue Management) di MQSeries for Compaq NSK
- Comando PCF Create Queue
- Comandi MQAI

Questi comandi specificano il tipo di cosa e i relativi attributi di oggetto. Ad esempio, una coda locale dispone di attributi oggetto che specificano quando le applicazioni fanno riferimento a tale coda nelle chiamate MQI. Esempi di attributi di oggetto sono:

- Se le applicazioni possono ricevere messaggi dalla coda (GET enabled)
- Se le applicazioni possono inserire messaggi nella coda (PUT enabled)
- Se l'accesso alla coda è esclusivo per una sola applicazione o è condiviso tra più applicazioni
- Il numero massimo di messaggi che possono essere memorizzati in una coda nello stesso tempo (capacità massima della coda)
- La dimensione massima dei messaggi che è possibile inserire nella coda

Per ulteriori informazioni:

- Su MQSC, consultare il manuale *MQSeries Guida di riferimento per i comandi*
- Su MQM, consultare il manuale "Utilizzo della funzione MQM (Message Queue Management)" a pagina 69
- Sui comandi PCF, consultare il manuale *MQSeries Programmable System Management*

Utilizzo di oggetti coda

In MQSeries, esistono diversi tipi di oggetti coda. Ogni tipo di oggetto può essere manipolato dai comandi del prodotto ed è associato alle code reali in diversi modi:

• Oggetto coda locale

Un oggetto coda locale identifica una coda locale appartenente al Queue Manager al quale l'applicazione è connessa. Tutte le code sono code locali, nel senso che ogni coda appartiene a un Queue Manager e, per quel Queue Manager, la coda è una coda locale.

• Oggetto coda remota

Un oggetto coda remota identifica una coda appartenente a un altro Queue Manager. Tale coda deve essere definita come coda locale di quel Queue

Manager. Le informazioni specificate durante la definizione di un oggetto coda remota consentono al Queue Manager locale di individuare il Queue Manager remoto, in modo da destinare a quest'ultimo i messaggi da inviare alla coda remota.

Prima che le applicazioni possano inserire messaggi in una coda appartenente a un altro Queue Manager, occorre aver definito una coda di trasmissione e i canali tra i Queue Manager, a meno che non siano stati raggruppati uno o più Queue Manager in un cluster. Per ulteriori informazioni sui cluster, consultare la sezione "Gestione remota mediante cluster" a pagina 126.

- **Oggetto coda di alias**

Un oggetto coda di alias consente alle applicazioni di accedere a una coda riferendosi ad essa indirettamente nelle chiamate MQI. Quando si utilizza un nome coda di alias in una chiamata MQI, il nome viene interpretato come una coda locale o remota al momento dell'avvio. Questo consente di modificare le code utilizzate dalle applicazioni, senza modificare in alcun modo l'applicazione: basterà modificare solo la definizione della coda di alias in modo da riflettere il nome della nuova coda a cui si riferisce l'alias.

Una coda di alias non è una coda, ma un oggetto che consente di accedere a un'altra coda.

- **Oggetto coda modello**

Un oggetto coda modello definisce un gruppo di attributi di coda che vengono utilizzati come modello per la creazione di una coda dinamica. Le code dinamiche vengono create dal Queue Manager quando un'applicazione inoltra una richiesta MQOPEN specificando un nome di coda che è appunto il nome di una coda modello. La coda dinamica così creata è una coda locale, i cui attributi sono ricavati dalla definizione della coda modello. Il nome della coda dinamica può essere specificato dall'applicazione oppure creato dal Queue Manager, che lo restituisce all'applicazione.

Le code dinamiche così definite possono essere code temporanee, che non sopravvivono al riavvio del prodotto, oppure code permanenti che invece sopravvivono.

Code locali utilizzate da MQSeries

MQSeries utilizza varie code locali per particolari scopi relativi alle proprie operazioni. È *necessario* definirle prima che MQSeries le utilizzi.

Code di applicazione: Una coda che viene utilizzata da un'applicazione (attraverso MQI) si definisce coda di applicazione. Questa può essere una coda locale sul Queue Manager a cui è connessa l'applicazione, oppure una coda remota di pertinenza di un altro Queue Manager.

Le applicazioni possono inserire messaggi nelle code locali o remote. Tuttavia, possono ricevere messaggi solo da una coda locale.

Code di iniziazione: Le code di iniziazione sono di norma utilizzate nel triggering. Un Queue Manager inserisce un messaggio trigger in una coda di iniziazione qualora si verifichi un evento trigger. Un evento trigger è una combinazione logica di condizioni che viene rilevata da un Queue Manager. Ad esempio, può essere generato un evento trigger quando il numero di messaggi in una coda raggiunge la capacità predefinita per quella coda. Questo evento fa sì che il Queue Manager inserisca un messaggio trigger nella coda di iniziazione specificata. Questo messaggio trigger viene ricevuto da un *trigger monitor*, un'applicazione particolare che controlla una coda di iniziazione. Quindi il trigger monitor avvia il programma applicativo specificato nel messaggio trigger.

Code di MQSeries

Prima che un Queue Manager possa utilizzare la funzione triggering, occorre definire almeno una coda di iniziazione relativa a tale Queue Manager.

Consultare la sezione “Gestione degli oggetti per triggering” a pagina 118. Per ulteriori informazioni relative alla funzione triggering, consultare il manuale *MQSeries Application Programming Guide*.

Code di trasmissione: Una coda di trasmissione memorizza temporaneamente i messaggi destinati a un Queue Manager remoto. È necessario definire almeno una coda di trasmissione per ogni Queue Manager remoto, a cui il Queue Manager locale deve inviare messaggi direttamente. Tali code sono utilizzate anche nella gestione remota. Consultare la sezione “Gestione di un Queue Manager remoto” a pagina 127. Per informazioni relative all’utilizzo delle code di trasmissione nell’accodamento distribuito, consultare il manuale *MQSeries Intercommunication*.

Code di trasmissione del cluster: Ogni Queue Manager all’interno di un cluster possiede una coda di trasmissione del cluster denominata `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Per impostazione predefinita, su ogni Queue Manager viene creata una definizione di tale coda.

Un Queue Manager che compone un cluster può inviare messaggi tramite la coda di trasmissione del cluster a qualsiasi Queue Manager all’interno dello stesso cluster.

I Queue Manager del cluster possono comunicare con i Queue Manager che non fanno parte del proprio cluster. A tale scopo, il Queue Manager devono definire i canali e una coda di trasmissione verso l’altro Queue Manager, così come avviene in un normale ambiente di accodamento distribuito.

Durante la risoluzione del nome, la coda di trasmissione del cluster assume prevalenza sulla coda di trasmissione predefinita. Quando un Queue Manager esterno al cluster inserisce un messaggio in una coda remota, per impostazione predefinita viene utilizzata la coda di trasmissione predefinita, a meno che vi sia una coda di trasmissione con lo stesso nome del Queue Manager di destinazione.

Se il Queue Manager fa parte di un cluster, per impostazione predefinita viene utilizzato `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, a meno che il Queue Manager di destinazione sia estraneo al cluster.

Code messaggi non recapitati: Una coda messaggi non recapitati memorizza i messaggi che non è possibile inoltrare alla corretta destinazione. Ciò accade, ad esempio, quando la coda di destinazione è piena. La coda messaggi non recapitati predefinita è denominata `SYSTEM.DEAD.LETTER.QUEUE`. Su altre piattaforme queste code vengono denominate diversamente (undelivered-message).

Per l’accodamento distribuito è necessario definire una coda messaggi non recapitati su ciascun Queue Manager attivo.

Code comandi: La coda comandi, denominata `SYSTEM.ADMIN.COMMAND.QUEUE`, è una coda locale a cui le applicazioni opportunamente autorizzate possono inviare i comandi MQSeries for Compaq NSK da elaborare. Questi comandi vengono poi richiamati da un componente MQSeries denominato server di comandi. Il server di comandi analizza i comandi e trasmette quelli validi al Queue Manager affinché siano elaborati e restituisce le eventuali risposte alle appropriate code di risposta.

Code di risposta RTQ (Reply-to queue): Quando un'applicazione riceve un messaggio di richiesta può restituire un messaggio di risposta all'applicazione che lo ha inviato. Tale messaggio viene inserito in una coda denominata coda di risposta, che di norma è una coda locale rispetto all'applicazione di invio. Il nome della coda di risposta è specificato dall'applicazione di invio come parte del descrittore di messaggio.

Code di evento: MQSeries for Compaq NSK supporta eventi di strumentazione, che possono essere utilizzati per controllare i Queue Manager indipendentemente dalle applicazioni MQI. Gli eventi di strumentazione possono essere generati in diversi modi, ad esempio:

- Un'applicazione che tenta di inserire un messaggio in una coda non disponibile o inesistente
- Una coda che sta diventando piena
- Viene avviato un canale

Quando si verifica un evento di strumentazione, il Queue Manager inserisce un messaggio di evento in una coda di evento. Questo messaggio può così essere rilevato da un'applicazione di controllo, che può informare un amministratore oppure iniziare attività di ripristino nel caso in cui l'evento indichi un problema.

Nota: Gli eventi di trigger sono diversi dagli eventi di strumentazione in quanto sono prodotti da condizioni diverse e non generano messaggi di evento.

Per ulteriori informazioni relative agli eventi di strumentazione, consultare il manuale *MQSeries Programmable System Management*.

Definizioni di processo

Un oggetto *definizione di processo* definisce un'applicazione che viene avviata come conseguenza di un evento trigger in un Queue Manager di MQSeries for Compaq NSK. Consultare la sezione "Code di iniziazione" a pagina 13 per ulteriori informazioni.

Gli attributi di questo oggetto sono l'ID applicazione, il tipo di applicazione e i dati specifici per l'applicazione.

Per creare una definizione di processo utilizzare il comando DEFINE PROCESS di tipo MQSC oppure il comando Create Process di tipo PCF.

Canali

I *canali* sono oggetti che forniscono un percorso di comunicazione tra i Queue Manager. I canali vengono utilizzati nell'accodamento distribuito di messaggi per trasportare i messaggi tra i diversi Queue Manager. Essi separano le applicazioni dai protocolli di comunicazione di base. I Queue Manager possono risiedere sulla stessa piattaforma o su piattaforme diverse. Per la comunicazione tra Queue Manager, è necessario definire un oggetto canale relativo al Queue Manager che invia i messaggi ed un altro oggetto canale relativo al Queue Manager che riceve i messaggi.

MQSeries for Compaq NSK supporta i protocolli di trasferimento TCP/IP e SNA LU6.2 (SNAX o ICE).

Canali

Per informazioni sui canali e sul loro utilizzo, consultare il manuale *MQSeries Intercommunication* e la sezione "Preparazione di canali e code di trasmissione per la gestione remota" a pagina 128.

Cluster

In una normale rete MQSeries che utilizza l'accodamento distribuito, tutti i Queue Manager sono indipendenti. Affinché un Queue Manager possa inviare un messaggio a un altro Queue Manager, occorre aver definito una coda di trasmissione, un canale diretto al Queue Manager remoto e una definizione di coda remota per ogni coda in cui inserire messaggi.

Un cluster è un gruppo di Queue Manager impostato in modo che i Queue Manager possano comunicare direttamente fra loro attraverso un'unica rete, senza la necessità di complesse code di trasmissione e definizioni di canali e coda.

Per informazioni relative ai cluster, consultare la sezione "Capitolo 7. Gestione di oggetti remoti di MQSeries" a pagina 125 e il manuale *MQSeries Queue Manager Clusters*.

Namelist

Una namelist è un oggetto MQSeries che contiene un elenco di altri oggetti MQSeries. Solitamente, le namelist vengono utilizzate da applicazioni quali trigger monitor, per identificare un gruppo di code. Il vantaggio dato dall'utilizzo di una namelist consiste nel fatto che essa è indipendente dall'applicazione; pertanto, può essere aggiornata senza dover arrestare nessuna delle applicazioni che la utilizzano. Inoltre, se un'applicazione subisce un malfunzionamento, la namelist non viene coinvolta e può essere ancora utilizzata dalle altre applicazioni.

Le namelist vengono anche utilizzate con i cluster di Queue Manager in modo da poter gestire una lista di cluster a cui si riferisce più di un oggetto MQSeries.

Oggetti definiti dal sistema

Gli *oggetti predefiniti di sistema* sono un insieme di definizioni di oggetto che vengono creati automaticamente ogni volta che viene creato un queue manager.

I nomi degli oggetti predefiniti hanno l'estensione SYSTEM.DEFAULT; ad esempio, SYSTEM.DEFAULT.LOCAL.QUEUE è la coda locale predefinita; SYSTEM.DEFAULT.RECEIVER è il canale ricevente predefinito. Non è possibile rinominare questi oggetti; l'estensione predefinita di questi nomi è necessaria.

Quando si definisce un oggetto, tutti gli attributi non specificati vengono copiati dal corrispondente oggetto predefinito. Ad esempio, gli attributi, che non vengono specificati durante la creazione di una coda locale, vengono copiati dalla coda predefinita SYSTEM.DEFAULT.LOCAL.QUEUE.

Dopo che è stato creato un Queue Manager, è possibile utilizzare il comando **runmqsc** per sostituire qualsiasi valore predefinito con altre definizioni.

Consultare la sezione "Appendice B. Impostazioni predefinite del sistema" a pagina 323 per ulteriori informazioni relative alle impostazioni predefinite.

Gestione

In MQSeries, si eseguono attività di gestione inviando dei *comandi*. Sono forniti quattro gruppi di comandi. La scelta del gruppo dipende dalle attività che si desidera eseguire e dalle relative modalità di utilizzo. I gruppi di comandi sono descritti nella sezione “Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries” a pagina 29. Le attività di gestione comprendono:

- Avvio e arresto di Queue Manager.
- Creazione di oggetti, in particolare code, per le applicazioni.
- Utilizzo dei canali per creare percorsi di comunicazione verso Queue Manager di altri sistemi remoti. Tale processo è illustrato in dettaglio nel manuale *MQSeries Intercommunication*.

Gestione locale e remota

La gestione locale consente di eseguire attività di gestione su qualsiasi Queue Manager precedentemente definito nel sistema locale. È possibile accedere ad altri sistemi, ad esempio, attraverso il programma **telnet** di emulazione TCP/IP ed eseguirvi attività di gestione. In MQSeries questa può considerarsi una gestione locale poiché non si utilizzano canali, visto che le comunicazioni sono gestite dal sistema operativo.

MQSeries supporta la gestione da una singola postazione attraverso la *gestione remota*. Ciò consente di inviare dal proprio sistema locale comandi che vengono elaborati da un altro sistema. Non è necessario collegarsi a quel sistema, sebbene occorra aver definito i canali in maniera corretta. Il Queue Manager e il server di comando del sistema di destinazione devono essere in esecuzione. Ad esempio, è possibile emettere un comando remoto per modificare una definizione di coda in un Queue Manager remoto.

Alcuni comandi non possono essere inviati in questo modo; in particolare, la creazione o l'avvio dei Queue Manager e l'avvio dei server di comando. Per eseguire queste attività, è necessario accedere al sistema remoto ed inviare da esso i comandi oppure creare un processo in grado di inviare tali comandi.

Client e server

MQSeries for Compaq NSK supporta configurazioni client-server per applicazioni MQI. Non esistono client di MQSeries for Compaq NSK, ma solo un server di MQSeries for Compaq NSK; tuttavia, i client su altre piattaforme possono collegarsi al server di MQSeries for Compaq NSK.

Un *client MQI* è un componente del prodotto MQSeries che è installato su una macchina per ricevere le chiamate MQI e trasferirle a una macchina server MQI. Qui vengono elaborate da un Queue Manager. Di norma, il client e il server risiedono su macchine diverse, ma possono anche risiedere sulla stessa macchina.

Un *server MQI* è un Queue Manager che fornisce servizi di accodamento a uno o più client. Tutti gli oggetti di MQSeries, ad esempio le code, esistono solo sul sistema del Queue Manager, cioè sulla macchina server MQI. Un server può supportare anche normali applicazioni locali MQI.

Per ulteriori informazioni, consultare il manuale *MQSeries Intercommunication* e la sezione *MQSeries Client*.

Applicazioni MQI in un ambiente client-server

Se collegate a un server, le applicazioni client MQI possono inviare chiamate MQI così come fanno le applicazioni locali. L'applicazione client inoltra una chiamata MQCONN per collegarsi a uno specifico Queue Manager. Successivamente vengono elaborate tutte le chiamate MQI aggiuntive che specificano l'handle di connessione restituito dalla richiesta di connessione.

È necessario collegare le applicazioni alle librerie client appropriate. Consultare il manuale *MQSeries Application Programming Guide* per ulteriori informazioni. Attualmente, non viene fornito alcun client MQI per Compaq NSK; tuttavia, poiché Compaq NSK è un server MQI, può accettare collegamenti da parte di qualsiasi client MQI di MQSeries in esecuzione su altre piattaforme.

Estensione delle funzioni di Queue Manager

Le funzioni di un Queue Manager possono essere estese da:

- User exit
- Servizi installabili

User exit

User exit consente di inserire il proprio codice di programmazione in una funzione di Queue Manager. Sono supportati due tipi di user exit:

- *Channel exit*

Queste uscite modificano il funzionamento dei canali. Le uscite del canale Channel exit sono descritte nel manuale *MQSeries Intercommunication*.

- *Data conversion exit*

Queste uscite creano frammenti di codice sorgente che possono essere inviati a un programma applicativo per convertire dati da un formato a un altro. Le uscite di conversione dati sono descritte nel manuale *MQSeries Application Programming Guide*.

- *Cluster workload exit*

Quest'uscita può essere utilizzata per modificare la modalità di selezione tra più istanze su una coda remota da parte di un cluster. Le informazioni sulla definizione della chiamata sono contenute nel manuale *MQSeries Queue Manager Clusters*.

Nota: Il meccanismo che consente di abilitare le user exit in MQSeries è stato modificato nella Versione 5.1. Rivedere attentamente la descrizione del meccanismo di uscita per determinare le modifiche che sarà necessario apportare allo scopo di trasferire il codice user exit dalle precedenti versioni di MQSeries for Compaq NSK.

Per ulteriori informazioni su tali uscite, consultare la sezione "Appendice L. User exit" a pagina 367.

Servizi installabili

I servizi installabili realizzano una maggiore estensione delle uscite in quanto sono dotati di interfacce API (Application Programming Interface) standard a ingressi multipli.

L'implementazione di un servizio installabile si definisce *componente servizio*. È possibile utilizzare i componenti forniti con il prodotto oppure è possibile compilarne uno per eseguire le funzioni desiderate. Attualmente, vengono forniti i seguenti servizi installabili:

- **Servizio di autorizzazione**

L'Authorization Service consente di creare una propria funzione di sicurezza.

Il componente di servizio predefinito che implementa il servizio è OAM (Object Authority Manager), fornito con il prodotto. Per impostazione predefinita, OAM è abilitato. È possibile utilizzare l'interfaccia Authorization Service per creare altri componenti che sostituiscano o incrementino OAM. Per ulteriori informazioni relative a questo componente, consultare la sezione "Capitolo 8. Protezione di oggetti di MQSeries" a pagina 139.

- **Name service**

Name service abilita i Queue Manager alla condivisione delle code consentendo alle applicazioni di identificare le code remote come se fossero code locali.

È possibile scrivere il proprio componente Name service. Consultare il manuale *MQSeries Programmable System Management* per ulteriori informazioni.

Sicurezza

MQSeries for Compaq NSK fornisce funzioni di sicurezza tramite OAM (Object Authority Manager).

Funzione OAM (Object Authority Manager)

L'autorizzazione per l'utilizzo di chiamate MQI e comandi, nonché per accedere agli oggetti, è fornita da OAM (Object Authority Manager), che è abilitato per impostazione predefinita. L'accesso alle entità è controllata tramite i principal e i gruppi utente di MQSeries for Compaq NSK, oltre che attraverso OAM. I nomi dei principal e di gruppo supportati da OAM sono risolti ai nomi di gruppo e utente di Compaq NSK. Nella Versione 5.1, tutti gli utenti di MQSeries devono avere un nome di principal associato a un nome utente di Compaq NSK. Ciò è necessario indipendentemente dal fatto che OAM sia abilitato o no. È disponibile un'interfaccia di riga comandi che consente di aggiungere ed eliminare i principal, oltre a concedere e revocare le autorizzazioni, se necessario.

Inoltre, è possibile utilizzare le funzioni di sicurezza di Compaq NSK per controllare l'accesso ai comandi MQSeries e ai file di database. Se SAFEGUARD è installato e MQSeries è compatibile con esso, questo può trarre vantaggi da alcune delle funzioni estese fornite. Per ulteriori informazioni, consultare la sezione "Capitolo 8. Protezione di oggetti di MQSeries" a pagina 139.

Supporti transazionali

Un programma applicativo può raggruppare un insieme di aggiornamenti in un'*unità di lavoro*. Tali aggiornamenti sono solitamente in relazione logica e devono essere privi di errori per preservare l'integrità dei dati. Se anche un solo aggiornamento si arresta, l'integrità dei dati verrà compromessa.

Un'unità di lavoro esegue il **commit** quando ha esito positivo. Da questo momento tutti gli aggiornamenti realizzati in quella unità di lavoro diventano permanenti o irreversibili. Se invece l'unità di lavoro dovesse avere esito negativo, sarà invece eseguito il *back out* di tutti gli aggiornamenti. Syncpoint coordination è il processo attraverso il quale si esegue il commit o il back out delle intere unità di lavoro.

Sicurezza

Un'unità di lavoro locale si caratterizza per il fatto che le uniche risorse aggiornate sono quelle appartenenti al Queue Manager di MQSeries. In queste unità, syncpoint coordination è prodotto autonomamente dal Queue Manager utilizzando un processo commit a fase singola.

Un'unità di lavoro globale si caratterizza per il fatto che vengono aggiornate anche le risorse relative ad altri gestori di risorse, quali un database XA compatibile. In queste unità deve essere utilizzato un processo commit a doppia fase e l'unità di lavoro deve essere coordinata autonomamente dallo stesso Queue Manager oppure esternamente da un altro gestore di transazioni XA compatibile, quali IBM® CICS®, Transarc Encina o BEA Tuxedo.

In MQSeries for Compaq NSK, tutte le unità di lavoro sono coordinate tramite TM/MP o un prodotto compatibile a più livelli che utilizzi a propria volta TM/MP (ad esempio, NonStop Tuxedo in ambiente OSS). Le applicazioni possono utilizzare TM/MP per coordinare le unità di lavoro che includono aggiornamenti a qualsiasi risorsa che sia compatibile con TM/MP e ai messaggi MQSeries. Non esistono database XA compatibili in ambiente Compaq NSK, ma è implementata un'interfaccia simile. Ad esempio, gli aggiornamenti ai database NonStop SQL o ai file ENSCRIBE possono essere coordinati con operazioni di messaggistica MQSeries per preservare l'integrità del database.

Per ulteriori informazioni, consultare la sezione "Capitolo 11. Descrizione del supporto transazionale e della messaggistica" a pagina 179.

Ottimizzazione delle prestazioni, affidabilità, scalabilità e dimensioni

Sono disponibili funzioni che consentono a MQSeries for Compaq NSK di beneficiare delle funzioni dell'hardware e del software Compaq NSK, che offrono un'infrastruttura applicativa scalabile, affidabile e dalle ottime prestazioni. Tali funzioni includono:

- Flessibilità di configurazione dei Queue Manager e dei relativi oggetti attraverso CPU e dischi al fine di incrementare il parallelismo
- Funzioni del software che garantiscano la tolleranza di qualsiasi malfunzionamento con, al massimo, una momentanea interruzione del servizio e il ripristino automatico del completo funzionamento
- Integrità del database protetta dal software e del relativo file system transazionale
- Compatibilità con prodotti che forniscano protezione per il ripristino d'emergenza.

Per ulteriori informazioni, consultare la sezione "Capitolo 15. Scalabilità e prestazioni" a pagina 219.

Capitolo 2. Architettura di MQSeries for Compaq NSK V5.1

Questo capitolo descrive l'architettura generale di MQSeries for Compaq NSK V5.1. Esso contiene le seguenti sezioni:

- "Panoramica dei processi queue manager"
- "Impacchettamento del prodotto" a pagina 23
- "Eseguibili" a pagina 24
- "Messa a confronto dell'architettura della Versione 5.1 e della Versione 2.2.0.1" a pagina 24
- "Queue manager — panoramica funzionale" a pagina 25
- "Modello di processo di Queue manager" a pagina 26
- "File e volumi secondari di MQSeries" a pagina 27

Queste informazioni possono essere di ausilio nell'impostazione di una configurazione ottimale del prodotto per il sistema operativo utilizzato.

"Capitolo 15. Scalabilità e prestazioni" a pagina 219 e "Capitolo 16. Integrità e disponibilità dei dati" a pagina 231 forniscono inoltre delle specifiche indicazioni di configurazione nel contesto di queste informazioni sull'architettura.

Panoramica dei processi queue manager

La seguente figura illustra i processi che compongono un queue manager MQSeries (PATHMON è mostrato per ragioni di esaustività, poiché il queue manager viene eseguito in PATHWAY). Questa figura fornisce inoltre una panoramica delle interazioni IPC dei processi.

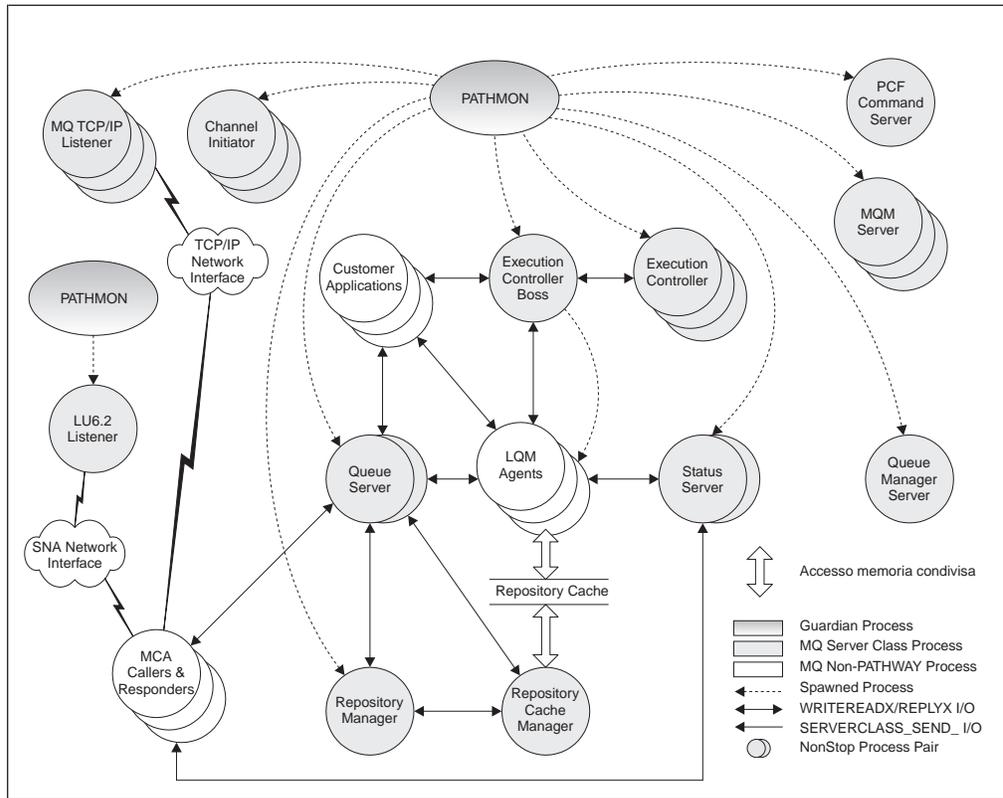


Figura 1. Processi MQSeries for Compaq NSK

Quando viene avviato un queue manager, PATHWAY crea i processi queue manager chiave. Nella configurazione predefinita, PATHMON avvia Execution Controller Boss (ECBoss), gli Execution Controller (EC), i server di stato e di coda ed il Repository Manager.

L'ECBoss gestisce tutte le richieste MQCONN in entrata e le distribuisce tra gli Execution Controller disponibili, fornendo una funzione di ripartizione del carico di lavoro selezionando l'EC impegnato da un numero minore di connessioni.

Gli EC gestiscono e controllano gli altri processi queue manager e le applicazioni MQI.

I Local Queue Manager Agent (LQMA) eseguono le operazioni richieste per elaborare le chiamate MQI per conto delle applicazioni. Gli agenti eseguono il grosso del codice che supporta la MQI. Lo scopo principale degli agenti consiste nel separare i programmi applicativi dalle risorse primarie del queue manager per proteggerle da applicazioni dal funzionamento imprevedibile o dannoso.

Il numero di processi agente dipende dal numero di applicazioni connesse e dal valore MinIdleLQMAgents specificato nel file QMINI. Quando viene avviato il Queue Manager, ciascun EC avvierà il numero specificato di agenti inattivi. Quando gli agenti diventano attivi, l'Execution Controller avvia dei nuovi agenti per conservare il numero minimo di agenti inattivi.

Un componente centrale dell'architettura di MQSeries per la Versione 5.1 è il *queue server*. Il queue server è un NonStop Process Pair che supporta tutte le operazioni di messaggistica per le code locali. Quando viene creato per la prima volta, un queue manager dispone di un singolo queue server predefinito. Sulla base della

configurazione del sistema e dei requisiti di prestazioni, è possibile configurare altri queue server ed assegnare loro delle code locali.

Lo status server gestisce le informazioni sugli stati per tutti gli oggetti diversi dalle code locali. Questo server svolge le attività del server MQSS utilizzato nella versione 2.2.0.1. Inoltre, il server di stato predefinito gestisce le informazioni sugli stati dei canali per il queue manager. Quando viene creato per la prima volta, un queue manager dispone di un singolo status server predefinito. Sulla base della configurazione del sistema e dei requisiti di prestazioni, è possibile configurare altri status server ed assegnare loro degli oggetti MQSeries.

Il queue manager avvia la Channel Initiator Server Class automaticamente quando viene avviato. In questo modo le operazioni di clustering funzionano senza che sia necessario avviare manualmente il Channel Initiator.

Nuovi per la Versione 5.1 sono i processi *repository manager* e *repository cache server* per la gestione delle code cluster. Esiste un processo repository manager in ciascuna CPU dove è in esecuzione un EC (Execution Controller). Il primo dei processi avviati svolge funzioni di repository manager, che coordina le attività di archiviazione di memoria nel queue manager. I processi successivi gestiscono un segmento di memoria condiviso che contiene la cache di archiviazione di memoria per la CPU in cui sono in esecuzione. Può esistere un solo Repository Manager per CPU. La cache di archiviazione di memoria contiene le informazioni sugli oggetti MQSeries raccolti in cluster, incluso le code e gli altri queue manager presenti nel cluster.

Anche nuovo per la Versione 5.1 è il *queue manager server*. Questo server gestisce i prospetti ed i messaggi scaduti.

Gli MCA (Message Channel Agent) trasferiscono i messaggi tra queue manager. Per una maggiore efficienza, in MQSeries for Compaq NSK V5.1, gli MCA sono connessi tramite FASTPATH. Per ulteriori informazioni su FASTPATH, consultare: "Programmi applicativi per binding FASTPATH" a pagina 228.

L'LU6.2 Listener (eseguito in un ambiente PATHWAY separato), è nuovo per la Versione 5.1. "Appendice M. Impostazione delle comunicazioni" a pagina 377 contiene informazioni dettagliate sulle operazioni e la configurazione del listener e del relativo ambiente.

Impacchettamento del prodotto

MQSeries for Compaq NSK V5.1 fornisce tre tipi di binding: nativo dinamico, nativo statico e non nativo.

Per il binding dinamico nativo, MQSeries fornisce una Shared Resource Library (MQSRLLIB) in modalità nativa. In MQSeries for Compaq NSK V5.1 si è tentato di integrare la massima porzione possibile di prodotto nella SRL (Shared Resource Library), riducendo al minimo la quantità di spazio richiesta dal prodotto. Utilizzare il binding dinamico nativo è l'approccio preferito per l'utilizzo di MQSeries poiché le risorse di sistema vengono utilizzate in modo ottimale. Dall'ambiente OSS, questo è il solo binding supportato.

Per il binding statico nativo, MQSeries for Compaq NSK fornisce una Re-Linkable Library (MQSRLLNK). Questa libreria è supportata per i clienti che stanno già utilizzando una SRL e non possono pertanto utilizzare la SRL MQSeries.

MQSeries fornisce una libreria statica non nativa per motivi di compatibilità con le applicazioni già esistenti che non possono utilizzare i binding in modalità nativa.

Eseguibili

La seguente tabella presenta gli eseguibili MQSeries

Tabella 1. eseguibili MQSeries

eseguibile MQSeries	Nome
Execution Controller	MQEC
Execution Controller Boss	MQECBOSS
Local Queue Manager Agent	MLQMAG
Status Server	MQSTSVR
Queue Server	MQQSSVR
Repository Manager	MQREPSVR
Queue Manager Server	MQMGRSVR
Command Server	MQCMDSVR
MQSeries Management Server	MQMSVR
Channel Initiator	RUNMQCHI
Caller MCA	MQMCACAL
TCP/IP Responder MCA	MQTCPRES
LU6.2 Responder MCA	MLLU6RES
Trigger Monitor	RUNMQTRM

Messa a confronto dell'architettura della Versione 5.1 e della Versione 2.2.0.1

La maggiore modifica strutturale tra MQSeries Versione 2.2.0.1 e Versione 5.1 è l'introduzione dei queue server. Ciascun queue server conserva i dati ed i file associati ad uno o più oggetti coda locale. Quando viene creato per la prima volta, un queue manager dispone di un singolo queue server predefinito, responsabile per tutte le code locali associate al queue manager. E' possibile aggiungere altri queue server (utilizzando PATHWAY) ed assegnare loro delle code utilizzando **altnqfls** .

I queue server sono NonStop Process Pair che supportano tutte le operazioni di messaggistica per le code locali. Il queue server supporta anche messaggi non permanenti basati solo sulla memoria, con un notevole miglioramento delle prestazioni rispetto alla Versione 2.2.0.1. MQSeries utilizza un'interfaccia interna per TM/MP per integrare questi messaggi basati sulla memoria nelle transazioni, consentendo così l'integrazione di messaggi sia permanenti che non permanenti nella stessa transazione, senza che sia richiesto l'ulteriore carico di lavoro necessario per scrivere i messaggi non permanenti su disco. I queue server svolgono le funzioni dei server MQSS della Versione 2.2.0.1 solo per le code locali.

Gli MQSS Server sono stati sostituiti dagli Status Server. Gli status server sono NonStop Process Pair che conservano le informazioni sugli stati per gli oggetti diversi dalle code locali.

Una seconda modifica rilevante nell'architettura del prodotto è rappresentata dall'introduzione di segmenti di memoria condivisi. Il file di inizializzazione del queue manager viene distribuito nel queue manager utilizzando memoria condivisa di sola lettura. Questo consente delle prestazioni migliori quando si stabilisce una connessione ad un queue manager. La MQSeries Repository Cache (utilizzata per implementare le funzioni di clustering) utilizza un segmento di memoria condiviso di lettura/scrittura in ciascuna CPU dove sono in esecuzione processi MQSeries.

Versione 5.1 incorpora un supporto più trasparente per i canali LU6.2. Questa versione introduce un LU6.2 listener, responsabile per l'avvio degli LU6.2 responder. Di conseguenza, l'attributo "AUTOSTART" per i canali LU6.2 è stato rimosso.

Queue manager — panoramica funzionale

Il queue manager presenta i seguenti componenti principali:

Application Interface

Fornisce l'ambiente ed il meccanismo per l'esecuzione delle chiamate MQI.

Queue Manager Kernel

Fornisce la maggior parte delle funzioni della MQI. Ad esempio, il triggering è implementato qui.

Object Authority Manager (OAM)

Fornisce un controllo dell'accesso per il queue manager e le sue risorse. Consente di specificare quali utenti e gruppi sono autorizzati ad eseguire determinate operazioni sulle risorse.

Data Abstraction and Persistence (DAP)

Fornisce la memorizzazione ed il ripristino dei dati contenuti nel queue manager. DAP conserva i messaggi.

Message Channel Agent

Sono applicazioni speciali che utilizzano la MQI per la maggior parte delle loro operazioni. La loro funzione principale consiste nel garantire delle trasmissioni affidabili di messaggi tra queue manager. Gli MCA sono collegati tramite FASTPATH.

Command Server

Il Command Server è una speciale applicazione MQI che si occupa di elaborare i messaggi che consentono comandi per la gestione del queue manager.

Common Services

Isola il resto del queue manager dal sistema operativo. Fornisce dei servizi simili a quelli del sistema operativo quali ad esempio la gestione della memoria e la gestione dei processi e delle serializzazioni

Figura 2 a pagina 26 illustra il rapporto tra i vari componenti.

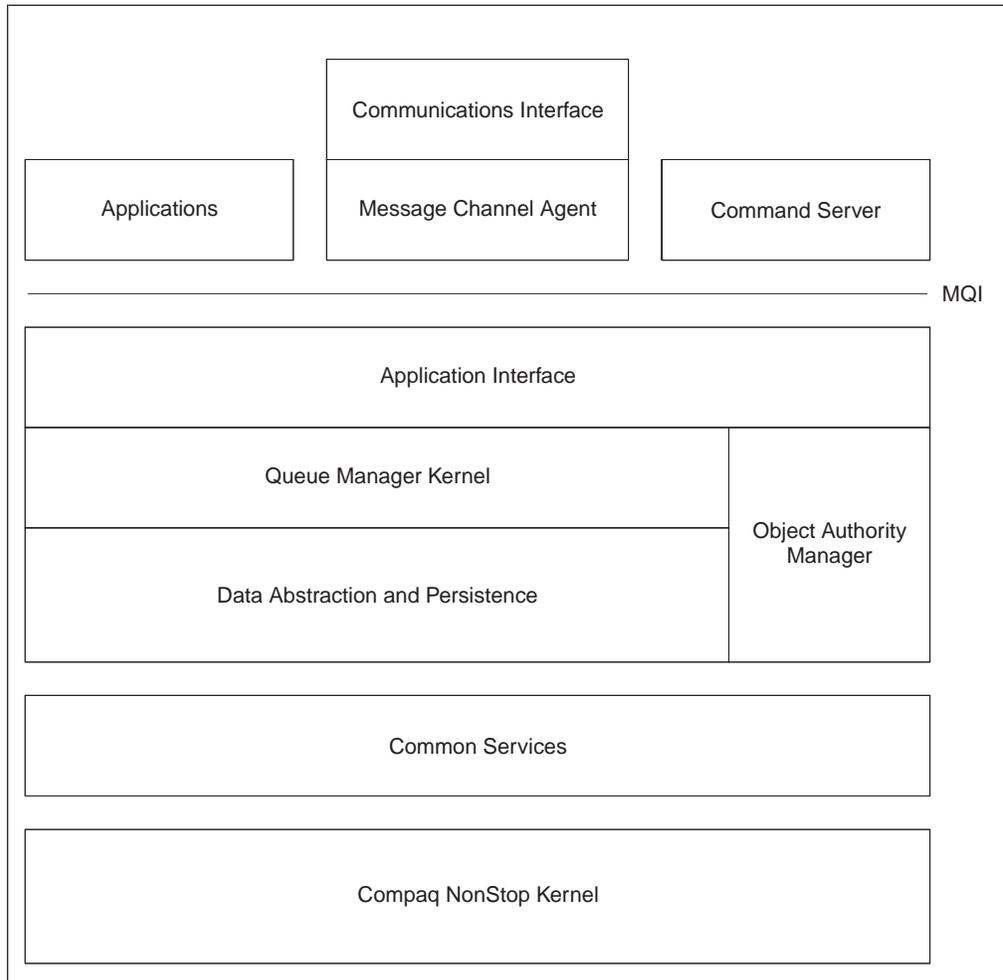


Figura 2. Componenti di MQSeries for Compaq NSK V5.1

Modello di processo di Queue manager

L'applicazione comunica con l'Execution Controller Boss (ECBoss) quando deve stabilire una connessione con un agente. L'ECBoss seleziona l'EC (Execution Controller) meno impegnato (cui sono connesse cioè meno applicazioni) ed inoltra ad esso la richiesta di connessione. L'EC seleziona un LQMA (Local Queue Manager Agent) inattivo. L'EC restituisce una risposta all'applicazione tramite l'ECBoss, che stabilisce quindi la connessione all'LQMA selezionato.

L'Application Interface è divisa in due parti:

- Lo stub dell'applicazione MQI impacchetta le richieste MQ e le passa al processo agente utilizzando l'IPCC (Inter-Process Communication Component).
- L'IPCC fornisce un'interfaccia per la trasmissione di messaggi tra le applicazioni MQI, gli agenti, gli EC e l'ECBoss. La base del componente IPCC di MQSeries è costituita da IPC GUARDIAN standard.

L'applicazione comunica con il suo processo agente tramite l'IPCC. Il processo agente trasforma le chiamate MQI per conto delle applicazioni. Gli scambi IPCC tra l'applicazione e l'agente sono messaggi di richiesta-risposta sincroni. Oltre a

comunicare con l'agente, quando esegue le operazioni di MQPUT e MQGET, l'applicazione trasferisce i dati dei messaggi direttamente al queue server responsabile per la coda.

Per le applicazioni collegate tramite FASTPATH, il codice MQSeries è collegato direttamente all'applicazione utente. Questo migliora le prestazioni ma comporta il rischio che un'applicazione dannosa possa disturbare, anche il modo grave, le operazioni del queue manager e determinare perdite di dati o altri problemi.

File e volumi secondari di MQSeries

MQSeries utilizza un certo numero di file Compaq NonStop ENSCRIBE. La posizione ed i nomi dei file sono di seguito riepilogati.

I file associati ad un queue manager di MQSeries sono distribuiti su vari volumi secondari.

<qmgr> D

Volume secondario Data. Utilizzato per i file che contengono informazioni che interessano tutto il queue manager.

<qmgr> M

Volume secondario Message data. Utilizzato per i file associati ai dati dei messaggi.

<qmgr> L

Log di errore

<qmgr> S

File di sincronizzazione dei canali

<qmgr>

file FFST™

Catalogo oggetti

Il catalogo oggetti (OBJCAT) contiene le informazioni correlate agli oggetti di MQSeries. Il catalogo oggetti si trova nel volume secondario relativo ai dati del queue manager.

File di coda

Ciascuna coda locale dispone di un file di coda, di un file di overflow di coda e di un touch file ad esso associati. I file hanno rispettivamente un prefisso Q, O e T—il resto del nome è parte del nome MQSeries dell'oggetto oppure un nome generato dal sistema. Per determinare i file associati ad una determinata coda, utilizzare il programma di utilità **dspmqls**. Quando viene creata una coda, il queue manager crea i file associati nel volume secondario message del queue manager (<qmgr>M). **altmqfls** consente di spostare questi file in un altro volume, se necessario per migliorare le prestazioni o per ragioni di spazio.

Il queue manager crea dei file di overflow di messaggio per ciascun messaggio di dimensioni superiori alla soglia di overflow di messaggio configurata per la coda. I file di overflow di messaggio si trovano nel volume secondario queue server per impostazione predefinita. Se è necessario modificare il volume secondario dove il server di coda crea i file di overflow di messaggio, utilizzare il programma di utilità **altmqfls**. Un file di overflow di messaggio è un file non strutturato e non sottoposto a procedure di controllo dedicato ad uno specifico messaggio (Consultare: "Server di code e file di code" a pagina 222 per ulteriori informazioni sulla configurazione dei file di coda). Per assicurare un efficace trasferimento dei

dati, i dati vengono scritti nel file di overflow di messaggio utilizzando la modalità di trasferimento di grosse quantità di dati (SETMODE 141) che abilita il trasferimento direttamente dalla memoria del processo applicativo in segmenti da 56 KB.

Per i messaggi permanenti di dimensioni inferiori a circa 3 KB, i dati del messaggio vengono memorizzati nel file di coda. Per i messaggi permanenti di dimensioni comprese tra 3 KB e la soglia di overflow di messaggio configurata per la coda, il server di coda scrive dei record aggiuntivi nel file di overflow di coda. Per i messaggi permanenti di dimensioni superiori alla soglia di overflow di messaggio, il server di coda crea un file di overflow di messaggio per memorizzare il messaggio.

Se si ha bisogno di una coda di partizione o di file di overflow di coda per ottimizzare le prestazioni o per ragioni di dimensioni, consultare: "Suddivisione dei file di coda" a pagina 224 per le informazioni necessarie per eseguire queste operazioni. Se si esegue la partizione dei file, MQSeries distribuisce i dati di messaggio in modo bilanciato tra le partizioni per fornire delle prestazioni ottimali.

"Capitolo 15. Scalabilità e prestazioni" a pagina 219 fornisce ulteriori informazioni sulle opzioni per l'ottimizzazione delle prestazioni correlate alle code ed ai file di coda.

Alias e Remote

Alle code alias e remote è associato un touch file. Il touch file inizia con un prefisso di T e si trova nel volume secondario data del queue manager.

File di elenchi di nomi

A ciascun elenco di nomi è associato un file non strutturato. I file iniziano con un prefisso L e si trovano nel volume secondario data del queue manager. Il resto del nome del file è parte del nome oggetto MQSeries oppure un valore generato dal sistema. Per determinare il nome del file associato ad un determinato elenco nomi, utilizzare il programma di utilità **dspmqls**.

Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries

Questo capitolo illustra i comandi che è possibile utilizzare per eseguire le attività di gestione del sistema sugli oggetti MQSeries. Le attività di gestione includono la creazione, l'avvio, la modifica, la visualizzazione, l'arresto e l'eliminazione di Queue Manager, code, processi e canali. Per eseguire tali attività, occorre selezionare il comando appropriato.

MQSeries for Compaq NSK V5.1 fornisce i seguenti gruppi di comandi di gestione per l'esecuzione di attività di gestione:

- MQSC (comandi MQSeries)
- Comandi PCF (Programmable Command Format)
- Comandi di controllo
- MQAI (MQSeries Administrator Interface)

Inoltre:

- Alcuni comandi TS/MP (PATHWAY) sono utilizzati a scopi di gestione.
- La funzione MQM (Message Queue Management) supporta alcune attività di gestione. La funzione MQM è descritta nella sezione "Utilizzo della funzione MQM (Message Queue Management)" a pagina 69.

Questo capitolo descrive i gruppi di comandi MQSC, PCF e di controllo, oltre a fornire un riepilogo delle funzioni supportate da ciascun gruppo di comandi impostati in "Appendice D. Confronto tra gruppi di comandi" a pagina 329. Le modalità di utilizzo dei comandi TS/MP sono descritte nella sezione "Gestione di TS/MP (PATHWAY)" a pagina 32.

Esecuzione delle attività di gestione utilizzando i comandi di controllo

Sono disponibili i seguenti tipi di comandi di controllo:

- Comandi per la creazione, l'avvio, l'arresto e l'eliminazione dei Queue Manager
- Comandi per l'avvio, l'arresto e la visualizzazione dei server di comando
- Comandi di utilità associati, ad esempio, all'esecuzione di comandi MQSC, alla gestione di accesso agli oggetti MQSeries, all'avvio e all'arresto di una traccia di MQSeries e all'esecuzione dei trigger monitor

Utilizzo dei comandi di controllo

È possibile eseguire comandi di controllo dal prompt TACL di Compaq. I nomi di comando non sono sensibili al maiuscolo/minuscolo (si noti, tuttavia, che i nomi dei Queue Manager *sono* sensibili al maiuscolo/minuscolo). Ad esempio:

```
runmqsc
```

"Capitolo 17. I comandi di controllo di MQSeries" a pagina 251 descrive la sintassi e lo scopo di ciascun comando.

Esecuzione delle attività di gestione utilizzando comandi MQSC

È possibile utilizzare i comandi MQSC per gestire oggetti Queue Manager, comprese le definizioni di Queue Manager, canali, code e processo. Ad esempio, è possibile definire, modificare, visualizzare ed eliminare una coda specificata mediante comandi MQSC.

Il comando DISPLAY QUEUE, visualizzando una coda, consente di visualizzare gli *attributi* della coda. Ad esempio, l'attributo MAXMSGL specifica la lunghezza massima di un messaggio che è possibile inserire nella coda. Il comando non mostra i messaggi presenti nella coda. Tali comandi sono riassunti nella sezione "Appendice D. Confronto tra gruppi di comandi" a pagina 329. Per informazioni dettagliate su ciascun comando MQSC, consultare il manuale *MQSeries Guida di riferimento per i comandi*.

Esecuzione dei comandi MQSC

È possibile eseguire MQSC in modo interattivo, richiamando il comando di controllo **runmqsc** dal prompt TACL di Compaq oppure eseguendo uno script quando il Queue Manager locale è in esecuzione. È possibile eseguire lo stesso comando **runmqsc** in tre modi, a seconda dei flag impostati nel comando:

- *Verification mode (Modo Verifica)*, in cui i comandi MQSC vengono verificati su un Queue Manager locale, ma non effettivamente eseguiti.
- *Direct mode (Modo Diretto)*, in cui i comandi MQSC vengono eseguiti su un Queue Manager locale.
- *Indirect mode (Modo Indiretto)*, in cui i comandi vengono eseguiti su un Queue Manager remoto.

Per ulteriori informazioni relative all'utilizzo della funzione MQSC e ai file di testo, consultare la sezione "Utilizzo interattivo della funzione MQSC" a pagina 99. Per ulteriori informazioni relative al comando **runmqsc**, consultare la sezione "runmqsc (Esegue comandi di MQSeries)" a pagina 298. I comandi MQSC sono riassunti nell'"Appendice D. Confronto tra gruppi di comandi" a pagina 329.

Utilizzo del comando NSK Fix di Compaq

Se si esegue il comando **runmqsc** in modo interattivo (dal prompt NSK TACL), è quindi possibile utilizzare anche la funzione di comando NSK Fix di Compaq che consente di richiamare e modificare comandi MQSC. Ad esempio:

- Digitando **history** o **h** si otterrà un elenco dei dieci comandi più recenti
- Digitando **!n**, dove **n** è il numero di comando, tale comando verrà eseguito nuovamente
- Digitando **h n** o **history n**, dove **n** è un numero, si otterrà un elenco degli **n** comandi più recenti
- Digitando **fc** sarà visualizzato l'ultimo comando immesso per la modifica. Digitando **fc n**, dove **n** è il numero del comando, sarà possibile modificare tale comando. Digitando **fc string**, dove **string** è la parte iniziale di un comando o l'intero comando precedentemente immesso, si otterrà l'ultima ripetizione di tale comando per la modifica. La sintassi segue lo standard NSK. Ad esempio, digitare **d** per eliminare un carattere, **i** per inserire un carattere e **r** per sostituire un carattere.

Esecuzione delle attività di gestione utilizzando comandi PCF

I comandi PCF consentono di programmare attività di gestione nelle applicazioni o in un programma di gestione. Le funzioni MQSC sono realizzate in modo analogo dai comandi PCF. È possibile impostare un programma in modo che un singolo nodo possa emettere comandi PCF verso qualsiasi Queue Manager della rete. È inoltre possibile centralizzare e automatizzare le attività di gestione.

Ogni comando PCF è una struttura di dati che può essere integrata nella parte di un messaggio MQSeries relativa ai dati applicativi. Ciascun comando viene inviato al Queue Manager di destinazione utilizzando la funzione MQPUT di MQI. Il server di comando del Queue Manager che riceve il messaggio lo riconosce come comando e lo esegue. Per ricevere le risposte, l'applicazione realizza una chiamata MQGET, così che i dati relativi alla risposta sono restituiti come struttura di dati nella parte relativa ai dati applicativi del messaggio MQSeries. A questo punto, l'applicazione può analizzare la risposta e eseguirne il contenuto.

Nota: Diversamente dai comandi MQSC, i comandi PCF e le relative risposte non sono in un formato testo leggibile.

È necessario specificare le seguenti voci per creare un messaggio di comando PCF:

Descrittore messaggio

Questo è un descrittore messaggio standard di MQSeries, nel quale:
 il tipo di messaggio (*MsgType*) è MQMT_REQUEST.
 il formato di messaggio (*Format*) è MQFMT_ADMIN.

Dati applicativi

Contiene il messaggio PCF, inclusa l'intestazione PCF, nel quale:
 Il tipo di messaggio PCF (*Type*) specifica MQCFT_COMMAND.
 L'identificativo del comando specifica il comando, ad esempio, *ChangeQueue* (MQCMD_CHANGE_Q).

Per una descrizione completa delle strutture di dati PCF e della loro implementazione, consultare il manuale *MQSeries Programmable System Management*.

Attributi MQSC e PCF

Gli attributi degli oggetti specificati in MQSC sono riportati in lettere maiuscole (ad esempio, RQMNAME), sebbene essi non siano sensibili al maiuscolo/minuscolo. I nomi di questi attributi hanno una lunghezza massima di otto caratteri (ad esempio, QDPHIEV). Gli attributi degli oggetti in PCF sono visualizzati in corsivo e la loro lunghezza non è limitata a otto caratteri. L'equivalente PCF di RQMNAME è *RemoteQMgrName* e quello di QDPHIEV è *QDepthHighEvent*.

Escape PCF

Gli escape PCF sono comandi PCF che contengono i comandi MQSC all'interno del messaggio di testo. È possibile utilizzare i comandi PCF per inviare comandi a un Queue Manager remoto. Per ulteriori informazioni relative all'utilizzo di escape PCF, consultare il manuale *MQSeries Programmable System Management*.

MQAI è un'interfaccia di amministrazione per MQSeries, al momento disponibile su piattaforma Compaq NSK. Essa consente una facile gestione di MQSeries: non è necessario scrivere i messaggi PCF, evitando di conseguenza i problemi associati a

comandi PCF

strutture di dati complesse. Per ulteriori informazioni sull'utilizzo di MQAI, consultare la sezione "Utilizzo di MQAI per semplificare l'impiego di PCF" a pagina 122.

Gestione di TS/MP (PATHWAY)

La maggior parte delle operazioni sul Queue Manager viene compiuta eseguendo i comandi di controllo di MQSeries da TACL. Tuttavia, alcune operazioni richiedono l'utilizzo di PATHCOM per operare direttamente sulle classi di server TS/MP. Inoltre, a causa delle modifiche apportate alla configurazione del sistema, potrebbe essere necessario eseguire alcune operazioni di gestione sulla medesima configurazione TS/MP.

La presente sezione riassume tali attività.

Specificazione e controllo dei listener TCP/IP

Per avviare i listener TCP/IP, è possibile utilizzare il comando di controllo di MQSeries **runmqtsr** (descritto nella sezione "runmqtsr (Run listener)" a pagina 296) oppure utilizzare i comandi PATHCOM THAW SERVER e START SERVER. Per arrestare i listener TCP/IP, utilizzare i comandi PATHCOM FREEZE SERVER e STOP SERVER. Utilizzare il comando PATHCOM STATUS SERVER per visualizzare il numero di listener TCP/IP in esecuzione e i relativi nomi di processo.

Per impostazione predefinita, ciascun Queue Manager dispone di un listener che si trova nella classe di server MQS-TCPLIS00. Utilizzare il comando PATHCOM ADD SERVER per creare classi di server del listener TCP/IP aggiuntive per stabilire un collegamento con più porte TCP/IP. Ciascun listener TCP/IP dovrebbe essere configurato nella relativa classe di server TS/MP per la massima flessibilità. Se si aggiungono listener TCP/IP, è necessario aggiungere anche definizioni di porta TCP/IP al file di inizializzazione del Queue Manager (QMINI), come descritto nella sezione "Porte TCP/IP su cui è in ascolto il queue manager" a pagina 54. Il primo listener da avviare utilizza la prima porta del listener definita in QMINI, il secondo listener utilizza la seconda porta del listener, e così via. Per un esempio delle voci QMINI necessarie a supportare più listener, consultare la sezione "Configurazione di QMINI per il supporto di più listener TCP/IP" a pagina 394.

Controllo del server di comando

Il server di comando viene creato come la classe di server TS/MP MQS-CMDSERV00. In alternativa ai comandi di controllo **strmqcsv**, **endmqcsv** e **dspmqcsv**, è possibile utilizzare i comandi PATHCOM THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER e STATUS SERVER.

Specificazione e controllo dei channel initiator

Il channel initiator predefinito viene creato come la classe di server TS/MP MQS-CHANINIT00. In alternativa all'utilizzo del comando di controllo **runmqchi** (descritto nella sezione "runmqchi (Run channel initiator)" a pagina 293), è possibile utilizzare i comandi PATHCOM THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER e STATUS SERVER per controllare e visualizzare lo stato del channel initiator. Il channel initiator predefinito elabora la coda di iniziazione predefinita, SYSTEM.CHANNEL.INITQ.

Modifica della coda di iniziazione predefinita per channel initiator

Nella Versione 5.1, il Queue Manager avvia automaticamente il channel initiator predefinito. Il channel initiator predefinito dovrà essere in esecuzione per supportare le operazioni cluster.

Se si desidera utilizzare una coda di iniziazione diversa da quella predefinita(SYSTEM.CHANNEL.INITQ), occorre modificare la configurazione di PATHWAY.

Nota: Si consiglia di non modificare la coda di iniziazione se il Queue Manager fa parte di un cluster. La modifica della coda di iniziazione predefinita relativa al channel initiator predefinito disabilita il supporto per i cluster.

È possibile modificare la coda di iniziazione predefinita mentre il Queue Manager è in esecuzione, ma occorre arrestare la classe di server di channel initiator. In PATHCOM, inviare il seguente comando verso la configurazione PATHWAY del Queue Manager:

```
ALTER SERVER MQS-CHANINIT00, STARTUP "-q<init-queue>"
```

dove <init-queue> è il nome della coda di iniziazione alternativa. È possibile avviare il channel initiator e uscire da PATHCOM.

Specificazione e controllo dei trigger monitor

Un singolo trigger monitor predefinito viene creato come classe di server TS/MP MQS-TRIGMON00. È possibile utilizzare i comandi PATHCOM THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER e STATUS SERVER per gestire tale classe di server. Se sono necessari trigger monitor supplementari, è possibile configurarli come classi di server aggiuntive, utilizzando MQS-TRIGMON00 come modello. Si consiglia di utilizzare oggetti classe di server separati per la massima flessibilità. Non è necessario utilizzare TS/MP per controllare i trigger monitor. Ad esempio, è possibile eseguire il trigger monitor da TACL utilizzando il comando di controllo `runmqtrm`.

Il trigger monitor predefinito elabora la coda di iniziazione predefinita,SYSTEM.DEFAULT.INITIATION.QUEUE. È possibile modificare quest'ultima aggiungendo o modificando il messaggio di STARTUP relativo alla classe di server che contiene il trigger monitor. Ciò è necessario se vengono configurati più trigger monitor per il Queue Manager. Utilizzare il comando PATHCOM ALTER SERVER per aggiungere o modificare l'attributo STARTUP.

Specificazione della distribuzione di processi tra CPU

Un aspetto importante della distribuzione del lavoro tra CPU è rappresentato dalla CPU assegnata a ciascun EC nel Queue Manager. Ogni EC crea e gestisce un gruppo di processi unicamente nella relativa CPU. Di conseguenza, se i processi EC vengono distribuiti tra le CPU del sistema, i processi agente saranno distribuiti in maniera simile.

Per impostazione predefinita, se vengono specificati più EC, i processi EC (ciascuno una classe di server distinta) saranno distribuiti, per quanto possibile, tra le CPU disponibili sul sistema. Non esiste un limite prestabilito per il numero di

Gestione di TS/MP

processi EC in un Queue Manager: il numero necessario dipende interamente dal carico che il Queue Manager dovrà gestire. Per impostazione predefinita, esiste un unico processo EC nel Queue Manager.

La classe di server EC predefinita è denominata MQS-EC00. Specificare il flag `-e` sul comando `crtmqm` per creare un Queue Manager con più EC. È possibile modificare il numero di processi EC dopo aver creato il Queue Manager, aggiungendo oppure eliminando le classi di server del processo EC e apportando la corrispondente modifica alla voce `ExpectedNumECs` nella `ECBoss` stanza nel file `QMINI`.

Ciascun processo EC *deve* trovarsi nella relativa classe di server. Qualora fosse necessario creare manualmente processi EC aggiuntivi, utilizzare la classe di server MQS-EC00.

Ciascuna CPU che ospiti un EC dovrà ospitare anche un repository manager. Alla creazione di un Queue Manager, MQSeries crea una classe di server del repository manager (con nomi del formato MQS-REPSVR00) per ogni EC. Se si aggiungono manualmente classi di server EC alle CPU che non ospitavano precedentemente gli EC, sarà opportuno utilizzare la classe di server MQS-REPSVR00 come modello per creare una nuova classe di server del repository manager nella nuova CPU.

MQSeries richiede una classe di server del repository manager in ciascuna CPU laddove una qualsiasi delle condizioni seguenti sia presente:

- la CPU ospita un EC, oppure
- la CPU ospita la classe di server MQS-QMGRSVR00, oppure
- la CPU esegue applicazioni collegate tramite FASTPATH, oppure
- gli utenti della CPU eseguono una qualsiasi delle applicazioni MQSeries: `runmqsc`, `runmqchi`, `runmqchl` o `mqrepdmp`.

L'assegnazione predefinita di CPU ai processi EC o a qualsiasi altra classe di server può essere modificata mediante il comando `PATHCOM ALTER SERVER` con l'attributo `CPU`.

Il status server predefinito viene automaticamente creato da `crtmqm` nella classe di server MQS-STATUS00. Per impostazione predefinita, l'unica assegnazione di CPU eseguita è quella primaria nella CPU 0 e la copia di riserva nella CPU 1 o `CPUS(0:1)`. L'assegnazione di CPU per il server MQS-STATUS00 può essere modificata mediante il comando `PATHCOM ALTER SERVER`. È possibile specificare una determinata copia di riserva di CPU per status server fornendo due numeri di CPU separati da due punti, ad esempio `CPUS(2:12)`. In tal caso, `PATHMON` crea l'assegnazione primaria nella CPU 2 e la copia di riserva nella CPU 12. Se non viene indicata una copia di riserva specifica Compaq NSK il sistema operativo determina in quale posizione sarà creata la copia di riserva.

Anche il server di coda predefinito sarà creato automaticamente da `crtmqm` nella classe di server MQS-QUEUE00. Per impostazione predefinita, l'unica assegnazione di CPU eseguita è quella primaria nella CPU 0 e la copia di riserva nella CPU 1 o `CPUS(0:1)`. L'assegnazione di CPU per il server MQS-QUEUE00 può essere modificata mediante il comando `PATHCOM ALTER SERVER`. È possibile specificare una determinata copia di riserva di CPU per il server di coda fornendo due numeri di CPU separati da due punti, ad esempio `CPUS(2:12)`. In tal caso, `PATHMON` crea l'assegnazione primaria nella CPU 2 e la copia di riserva nella CPU 12. Se non viene indicata una copia di riserva specifica Compaq NSK il sistema operativo determina in quale posizione sarà creata la copia di riserva.

Aggiunta di nuovi processi di status server

Per aggiungere ulteriori status server a un Queue Manager, creare una classe di server utilizzando la classe di status server predefinita MQS-STATUS00 come modello.

Il nome di qualsiasi nuova classe di status server dovrebbe iniziare con la stringa di caratteri MQS-STATUS. Se i nomi della classe di server non si attengono a tale convenzione di denominazione, questi non saranno automaticamente avviati da **strmqm** allo startup del Queue Manager e l'accesso a qualsiasi oggetto configurato per tali classi di status server verrà disabilitato.

Se vengono configurati status server aggiuntivi, sarà necessario attribuire loro nomi di processo univoci. Si consiglia inoltre di configurarli affinché siano eseguiti in diverse CPU, in modo da trarre vantaggio dalla scalabilità fornita dall'architettura di status server.

Aggiunta di nuovi processi del server di coda

Per aggiungere ulteriori server di coda a un Queue Manager, creare una classe di server utilizzando la classe del server di coda predefinita MQS-QUEUE00 come modello.

Il nome di qualsiasi nuova classe del server di coda dovrebbe iniziare con la stringa di caratteri MQS-QUEUE. Se i nomi della classe di server non si attengono a tale convenzione di denominazione, questi non saranno automaticamente avviati da **strmqm** allo startup del Queue Manager e l'accesso a qualsiasi oggetto configurato per tali classi di server di coda verrà disabilitato.

Se vengono configurati server di coda aggiuntivi, sarà necessario attribuire loro nomi di processo univoci. Si consiglia inoltre di configurarli affinché siano eseguiti in diverse CPU, in modo da trarre vantaggio dalla scalabilità fornita dall'architettura del server di coda.

Specificazione della frequenza di aggiornamento dei pannelli monitor MQM

Il parametro MQMQMREFRESHINT PATHWAY per MQS-MQMSVR00 determina la frequenza con cui gli schermi del monitor relativi a canali e code sono aggiornati. La frequenza predefinita è ogni 30 secondi. Per impostare la frequenza su ogni 10 secondi, ad esempio, modificarla nel PATHWAY relativo al proprio Queue Manager:

```
ALTER SERVER MQS-MQMSVR00, PARAM MQMQMREFRESHINT 10
```

Configurazione di PATHWAY per un Queue Manager

Ecco un esempio di configurazione PATHWAY per un Queue Manager. Questo esempio è stato generato inviando una sequenza di comandi INFO agli oggetti nella configurazione predefinita di un Queue Manager.

Gestione di TS/MP

```
TCP MQS-TCP-01
  AUTORESTART 0
  CHECK-DIRECTORY OFF
  CODEAREALEN 80000
  CPUS 0:1
  DEBUG OFF
  DUMP OFF
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  INSPECT OFF
  MAXINPUTMSGLEN 6000
  MAXINPUTMSGS 0
  MAXPATHWAYS 0
  MAXREPLY 32000
  MAXSERVERCLASSES 1
  MAXSERVERPROCESSES 10
  MAXTERMDATA 500000
  MAXTERMS 10
  NONSTOP 0
  POWERONRECOVERY ON
  PRI 175
  PROGRAM \RAPTOR.$SYSTEM.SYSTEM.PATHTCP2
  SERVERPOOL 32000
  STATS OFF
  TCLPROG \RAPTOR.$DEV.ZMQSEXE.POBJ
  TERMBUF 1500
  TERMPPOOL 10000
```

Figura 3. Esempio di configurazione PATHWAY (Numero 1 di 13)

```
PROGRAM MQMC
  ERROR-ABORT OFF
  OWNER \RAPTOR.44,1
  SECURITY "N"
  TCP MQS-TCP-01
  TMF ON
  TYPE T16-6520 (BREAK OFF,ECHO ON,EXCLUSIVE OFF,INITIAL MAINC,IOPROTOCOL
0,MAXINPUTMSGS 0,TRAILINGBLANKS ON)
```

Figura 3. Esempio di configurazione PATHWAY (Numero 2 di 13)

```

SERVER MQS-CHANINIT00
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0,1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 175
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.RUNMQCHI
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D

```

Figura 3. Esempio di configurazione PATHWAY (Numero 3 di 13)

```

SERVER MQS-CMDSERV00
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0,1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 175
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.MQCMSVR
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D

```

Figura 3. Esempio di configurazione PATHWAY (Numero 4 di 13)

Gestione di TS/MP

```
SERVER MQS-EC00
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0,1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 175
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.MQEC
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D
```

Figura 3. Esempio di configurazione PATHWAY (Numero 5 di 13)

```
SERVER MQS-ECBOSS
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0,1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 175
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.MQECBOSS
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D
```

Figura 3. Esempio di configurazione PATHWAY (Numero 6 di 13)

```

SERVER MQS-MQMSVR00
PROCESSTYPE GUARDIAN
AUTORESTART 0
CPUS (0,1)
CREATEDELAY 1 MINS
DEBUG OFF
DELETEDELAY 10 MINS
HIGHPIN ON
HOMETERM \RAPTOR.$ZTN0.#PTY001C
LINKDEPTH 255
MAXSERVERS 1
NUMSTATIC 1
OUT \RAPTOR.$ZTN0.#PTY001C
OWNER \RAPTOR.44,1
PARAM MQQUEMGRNAME "p101"
PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
PARAM MQDEFAULTPREFIX "$DEV"
PARAM MQMPAGESTORETRIEVE "20"
PRI 175
PROGRAM \RAPTOR.$DEV.ZMQSEXE.MQMSVR
SECURITY "N"
TMF ON
VOLUME \RAPTOR.$DATA1.P101D

```

Figura 3. Esempio di configurazione PATHWAY (Numero 7 di 13)

```

SERVER MQS-STATUS00
PROCESSTYPE GUARDIAN
AUTORESTART 0
CPUS (0:1)
CREATEDELAY 1 MINS
DEBUG OFF
DELETEDELAY 10 MINS
HIGHPIN ON
HOMETERM \RAPTOR.$ZTN0.#PTY001C
LINKDEPTH 255
MAXSERVERS 1
NUMSTATIC 1
OUT \RAPTOR.$ZTN0.#PTY001C
OWNER \RAPTOR.44,1
PARAM MQQUEMGRNAME "p101"
PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
PARAM MQDEFAULTPREFIX "$DEV"
PRI 176
PROCESS $P01S
PROGRAM \RAPTOR.$DEV.ZMQSEXE.MQSTSVR
SECURITY "N"
TMF ON
VOLUME \RAPTOR.$DATA1.P101D

```

Figura 3. Esempio di configurazione PATHWAY (Numero 8 di 13)

Gestione di TS/MP

```
SERVER MQS-TCPLIS00
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0,1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 175
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.RUNMQLSR
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D
```

Figura 3. Esempio di configurazione PATHWAY (Numero 9 di 13)

```
SERVER MQS-TRIGMON00
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0,1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 175
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.RUNMQTRM
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D
```

Figura 3. Esempio di configurazione PATHWAY (Numero 10 di 13)

```

SERVER MQS-QUEUE00
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0:1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 176
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.MQSSVR
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D

```

Figura 3. Esempio di configurazione PATHWAY (Numero 11 di 13)

```

SERVER MQS-QMGRSVR
  PROCESSTYPE GUARDIAN
  AUTORESTART 10
  CPUS (0,1)
  CREATEDELAY 1 MINS
  DEBUG OFF
  DELETEDELAY 10 MINS
  HIGHPIN ON
  HOMETERM \RAPTOR.$ZTN0.#PTY001C
  LINKDEPTH 255
  MAXSERVERS 1
  NUMSTATIC 1
  OUT \RAPTOR.$ZTN0.#PTY001C
  OWNER \RAPTOR.44,1
  PARAM MQQUEMGRNAME "p101"
  PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
  PARAM MQDEFAULTPREFIX "$DEV"
  PRI 175
  PROGRAM \RAPTOR.$DEV.ZMQSEXE.QMGRSVR
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D

```

Figura 3. Esempio di configurazione PATHWAY (Numero 12 di 13)

Gestione di TS/MP

```
SERVER MQS-REPSVR00
PROCESSTYPE GUARDIAN
AUTORESTART 10
CPUS (0,1)
CREATEDELAY 1 MINS
DEBUG OFF
DELETEDELAY 10 MINS
HIGHPIN ON
HOMETERM \RAPTOR.$ZTN0.#PTY001C
LINKDEPTH 255
MAXSERVERS 1
NUMSTATIC 1
OUT \RAPTOR.$ZTN0.#PTY001C
OWNER \RAPTOR.44,1
PARAM MQQUEMGRNAME "p101"
PARAM MQMACHINIFILE "$DATA1.p101D.UMQSINI"
PARAM MQDEFAULTPREFIX "$DEV"
PARAM MQREPMANAGER "YES"
PRI 175
PROGRAM \RAPTOR.$DEV.ZMQSEXE.MQREPSVR
SECURITY "N"
TMF ON
VOLUME \RAPTOR.$DATA1.P101D
```

Figura 3. Esempio di configurazione PATHWAY (Numero 13 di 13)

Modifica dei parametri delle classi di server PATHWAY

Per modificare i parametri delle classi di server PATHWAY:

1. Arrestare il Queue Manager inviando il comando **endmqm**. Tale comando arresta anche il processo PATHMON.
2. Passare al volume secondario *Queue ManagerD*, che contiene il file PATHCTL.
Ad esempio:

```
>VOLUME $DATA2.MT01D
```

3. Avviare un PATHMON con lo stesso nome del PATHMON del Queue Manager e con l'opzione NOWAIT. Ad esempio:

```
>PATHMON /NAME $MT01, NOWAIT/
```

4. Avviare PATHCOM sul nuovo PATHMON. Ad esempio:

```
>PATHCOM $MT01
```

5. Caricare la configurazione PATHWAY esistente per il Queue Manager inviando il seguente comando dal prompt PATHCOM:

```
>START PATHWAY COOL
```

6. Apportare le modifiche necessarie utilizzando comandi PATHCOM.

7. Arrestare il sistema PATHWAY inviando il seguente comando:

```
>SHUTDOWN2
```

8. Avviare il Queue Manager mediante il comando **strmqm**.

Aggiunta di classi di server definite dall'utente a un PATHWAY di MQSeries

È possibile aggiungere le proprie definizioni di classe di server alla configurazione di PATHWAY di MQSeries mediante PATHCOM. Tuttavia, ciò è sconsigliato: i server *devono* essere ben configurati, altrimenti **endmqm** non funziona correttamente. Si noti inoltre che le definizioni di classe di server definite dall'utente vengono perse quando un Queue Manager viene eliminato. Per ridurre gli inconvenienti, si consiglia di creare uno script riutilizzabile.

Capitolo 4. Gestione dei Queue Manager

Questo capitolo descrive tutti gli aspetti della gestione dei Queue Manager di MQSeries.

Questo capitolo contiene le seguenti sezioni:

- "Introduzione"
- "Come creare un queue manager"
- "Modifica delle proprietà del queue manager" a pagina 48
- "Nome queue server predefinito" a pagina 59
- "Aggiunta e rimozione di queue server non predefiniti" a pagina 60
- "Struttura dei volumi" a pagina 61
- "Lavorare con i gestori code" a pagina 64
- "Creazione di un Queue Manager predefinito" a pagina 64
- "Creazione dei principal di MQSeries" a pagina 65
- "Esecuzione di cleanrdf per un queue manager abilitato a RDF" a pagina 66
- "Avvio di un queue manager" a pagina 66
- "Ripristino degli oggetti predefiniti e di sistema" a pagina 66
- "Analisi dei file di oggetto" a pagina 66
- "Interruzione di un queue manager" a pagina 67
- "Riavvio di un Queue Manager" a pagina 68
- "Rendere predefinito un Queue Manager esistente" a pagina 68
- "Eliminazione di un queue manager" a pagina 68
- "Utilizzo della funzione MQM (Message Queue Management)" a pagina 69

Introduzione

Prima di utilizzare messaggi e code, è necessario creare almeno un queue manager. Dopo aver completato il processo di installazione, è possibile utilizzare i comandi di controllo di MQSeries per creare un queue Manager, creare dei principal di queue manager ed avviare il queue manager.

Quando si crea un queue manager, gli oggetti di sistema e gli oggetti predefiniti richiesti vengono automaticamente creati. Gli oggetti predefiniti costituiscono la base di qualsiasi definizione di oggetto che si realizza; gli oggetti di sistema sono necessari alle operazioni del Queue Manager. Per ulteriori informazioni sul ripristino degli oggetti di sistema predefiniti, consultare: "Ripristino degli oggetti predefiniti e di sistema" a pagina 66.

Per ulteriori informazioni sui comandi che è possibile utilizzare con MQSeries for Compaq NSK ed i diversi metodi per richiamarli, consultare: "Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries" a pagina 29.

Come creare un queue manager

Un queue manager gestisce le risorse ad esso associate, quali ad esempio le code di sua proprietà. Un queue manager fornisce i servizi di accodamento alle applicazioni per le chiamate MQI (Message Queuing Interface) e i comandi per la creazione, la modifica, la visualizzazione e l'eliminazione di oggetti MQSeries. E' possibile creare un queue manager utilizzando il comando **crtmqm**. Viene qui di seguito riportato un elenco dei fattori da prendere in considerazione quando si crea un queue manager:

Creazione dei Queue Manager

- **Specificare un nome di queue manager unico.**

Quando si crea un Queue Manager bisogna fare attenzione a non assegnargli lo stesso nome di un altro Queue Manager presente nella rete. I nomi dei Queue Manager non vengono controllati al momento della loro creazione; se si utilizza più di una volta lo stesso nome, non sarà possibile creare canali per l'accodamento distribuito.

E' possibile garantire l'unicità di un nome aggiungendo come prefisso al nome di ciascun queue manager il relativo nome di nodo. Ad esempio, se un nodo è denominato `accounts`, si consiglia di denominare il Queue Manager `accounts.saturn.queue.manager`, in cui `saturn` identifica un determinato Queue Manager e `queue.manager` è l'estensione che può essere assegnata a qualsiasi Queue Manager. In alternativa, è possibile ignorare tale suggerimento, ma si noti che `accounts.saturn` e `accounts.saturn.queue.manager` sono nomi di Queue Manager *diversi*.

Nota: I nomi di queue manager nei comandi di controllo sono sensibili al maiuscolo/minuscolo. Ad esempio, è possibile creare due queue manager con i nomi `jupiter.queue.manager` e `JUPITER.queue.manager`.

- **Limitare il numero di queue manager.**

In MQSeries for Compaq NSK, è possibile creare tutti i queue manager consentiti dalle risorse. Tuttavia, poiché ogni Queue Manager richiede delle risorse proprie, è solitamente più efficace definire un queue manager con 100 code piuttosto che dieci Queue Manager con dieci code ciascuno. E' possibile eseguire molti nodi con un singolo Queue Manager; tuttavia, dei server di dimensioni maggiori possono essere eseguiti con più queue manager. E' possibile che speciali requisiti di prestazioni o funzionalità richiedano la presenza di più queue manager.

- **Specificare un queue manager predefinito.**

Ogni nodo deve avere un Queue Manager predefinito, anche se è possibile configurare MQSeries su un nodo che ne sia privo.

Il Queue Manager predefinito è il Queue Manager cui le applicazioni si connettono quando si specifica un nome di Queue Manager nella chiamata MQCONN. Inoltre il Queue Manager predefinito elabora i comandi MQSC quando si richiama il comando `runmqsc` senza specificare un nome di Queue Manager.

Quando si specifica un Queue Manager come predefinito, si *sostituisce* qualunque specificazione relativa al precedente Queue Manager predefinito per il nodo.

Cambiare il queue manager predefinito può avere effetto sugli altri utenti o sulle altre applicazioni. Questa modifica non ha però alcun effetto sulle applicazioni attualmente connesse, poiché esse nelle successive chiamate MQI utilizzano l'handle della connessione originaria. Questa connessione fa sì che le successive chiamate siano tutte dirette allo stesso Queue Manager. Invece, tutte le applicazioni che si connettono dopo la modifica, si conatteranno al nuovo Queue Manager.

Per creare un queue manager predefinito, specificare il flag `-q` nel comando `crtmqm`. Per una descrizione dettagliata di questo comando e dei relativi parametri, consultare la sezione "crtmqm (Crea Queue Manager)" a pagina 267.

- **Specificare una DLQ (dead-letter queue).**

La coda messaggi non recapitati è una coda locale a cui vengono inoltrati i messaggi che non riescono a raggiungere la loro corretta destinazione.

Attenzione: Bisogna disporre di una DLQ in ciascun queue manager presente nella rete. In caso contrario, potrebbero verificarsi degli errori di programma

nelle applicazioni che determinano la chiusura dei canali ed impediscono l'invio di risposte ai comandi di gestione. Ad esempio, se un'applicazione tenta di inviare un messaggio ad una coda relativa ad un altro Queue Manager, ma il nome del Queue Manager è errato, il canale viene arrestato e il messaggio rimane nella coda di trasmissione. Le altre applicazioni non potranno quindi utilizzare questo canale per i propri messaggi.

Se invece viene definita una coda messaggi non recapitati per ogni Queue Manager, i canali possono continuare le proprie attività. Il messaggio non recapitato viene inserito nella DLQ sul lato in ricezione del canale, lasciando così libero il canale e la relativa coda di trasmissione.

Pertanto, quando si crea un Queue Manager è necessario utilizzare il flag `-u` per specificare il nome della coda messaggi non recapitati. È inoltre possibile utilizzare i comandi MQSC per modificare gli attributi di un Queue Manager e specificare la coda messaggi non recapitati da utilizzare. Consultare la sezione "Modifica degli attributi di Queue Manager" a pagina 101 per un esempio del comando MQSC ALTER.

Per elaborare i messaggi che vengono rilevati nella coda messaggi non recapitati, è possibile utilizzare l'handler DLQ (Dead-Letter Queue), fornito con MQSeries. Per ulteriori informazioni sull'handler DLQ e su come ridurre il numero di messaggi altrimenti inseriti nella DLQ, consultare: "Capitolo 9. Handler di code messaggi non recapitati di MQSeries" a pagina 159.

- **Specificare una coda di trasmissione predefinita.**

Una coda di trasmissione è una coda locale nella quale i messaggi in transito per un Queue Manager remoto vengono accodati in attesa della trasmissione. La coda di trasmissione predefinita è la coda utilizzata tutte le volte che non è definita esplicitamente un'altra coda. Ad ogni Queue Manager può essere assegnata una coda di trasmissione predefinita.

Quando si crea un Queue Manager, utilizzare il flag `-d` per specificare il nome della coda di trasmissione predefinita. Il flag `-d` non crea effettivamente la coda, che bisognerà quindi creare successivamente. Consultare la sezione "Utilizzo delle code locali" a pagina 106 per ulteriori informazioni.

- **Eseguire il backup dei file di configurazione dopo aver creato un queue manager.**

Il file di configurazione di MQSeries (MQSINI) viene creato quando si esegue l'installazione di MQSeries. Questo file contiene un elenco dei queue manager che viene aggiornato ogni volta che si esegue la creazione o la cancellazione di un queue manager. Esiste un file MQSINI per ciascuna installazione. Per impostazione predefinita, il file MQSINI si trova in `$SYSTEM.ZMQSSYS`.

Un file di configurazione di queue manager (QMINI) viene automaticamente creato quando si esegue la creazione di un nuovo queue manager. Questo file contiene i parametri di configurazione per il queue manager.

È consigliabile eseguire una copia di riserva di questi file. Se si esegue la creazione di un queue manager che determina problemi, è possibile reinstallare le copie di backup dopo aver rimosso la causa del problema. Eseguire il backup dei file di configurazione ogni volta che si esegue la creazione di un nuovo queue manager.

Per ulteriori informazioni sui file di configurazione, consultare il "Capitolo 13. File di configurazione" a pagina 187.

Eseguire una copia di riserva dei file di configurazione dopo aver creato un Queue Manager

Due sono i file di configurazione di cui eseguire il backup, MQSINI e QMINI:

Creazione dei Queue Manager

1. Il file di configurazione di MQSeries (MQSINI) viene creato quando si esegue l'installazione di MQSeries. Questo file contiene un elenco dei queue manager che viene aggiornato ogni volta che si esegue la creazione o la cancellazione di un queue manager. Esiste un file MQSINI per ciascuna installazione. Per impostazione predefinita, il file MQSINI si trova in \$SYSTEM.ZMQSSYS.
2. Un file di configurazione di queue manager (QMINI) viene automaticamente creato quando si esegue la creazione di un nuovo queue manager. Questo file contiene i parametri di configurazione per il queue manager.

È consigliabile eseguire una copia di riserva di questi file. Se si esegue la creazione di un queue manager che determina problemi, è possibile reinstallare le copie di backup dopo aver rimosso la causa del problema. Eseguire il backup dei file di configurazione ogni volta che si esegue la creazione di un nuovo queue manager.

Modifica delle proprietà del queue manager

È possibile modificare molte delle proprietà di un queue manager in fase di creazione del queue manager. Ci sono altre proprietà che è possibile modificare dopo che è stata eseguita la creazione del queue manager ma per rendere effettive le modifiche apportate a questi specifici parametri bisogna arrestare e riavviare il queue manager.

Il resto di questa sezione contiene una descrizione di alcune proprietà del queue manager che un utente potrebbe voler modificare.

Volume iniziale del queue manager

Questo è il volume in cui vengono creati tutti i database, code incluse. (È tuttavia possibile, dopo aver eseguito le operazioni di creazione, spostare delle singole code in un diverso volume utilizzando il comando di controllo **altmqfls**, come descritto in: "altmqfls (Modifica attributi del file di coda)" a pagina 254.)

Il valore predefinito viene preso dalla voce `QMDefaultVolume` della stanza `AllQueueManagers` nel file MQSINI. Esso viene sostituito dal parametro `-p DefaultPrefix` del comando **crtmqm**, se specificato.

Il volume iniziale può essere specificato *solo* in fase di creazione di un queue manager. Non è possibile modificarlo dopo che è stata eseguita l'operazione di creazione.

Numero di processi EC per queue manager

Per impostazione predefinita, esiste un processo EC per ciascun queue manager. È possibile specificare il numero di processi EC per un queue manager utilizzando il parametro `-e NumECs` del comando **crtmqm**.

Ciascun EC è responsabile di un sottogruppo di processi server che eseguono le operazioni di messaggistica e di accodamento per le applicazioni ed i canali nella stessa CPU dell'EC. Si consiglia di disporre di 1 EC per CPU, a meno che il numero di applicazioni per EC sia considerevole, nel qual caso disporre di un EC aggiuntivo in esecuzione sulla CPU potrebbe risultare vantaggioso. Per installazioni di notevole entità, ad esempio, è meglio disporre di più processi EC (spesso distribuiti su più CPU) per consentire al sistema di gestire simultaneamente numerose applicazioni e molti canali.

Durante la creazione di un queue manager, viene creata una classe di server TS/MP per ciascun EC specificato nel comando **crtmqm**. Le classi di server EC

specificate vengono distribuite in modo continuo a tutte le CPU del sistema. Ad esempio, la specifica di otto processi EC in un sistema con quattro CPU determina per impostazione predefinita due processi EC per CPU.

Il campo `ExpectedNumECs` della stanza `ECBoss` nel file `QMINI` del queue manager viene impostato sul numero di processi EC specificato in fase di creazione. Questo valore deve essere sempre congruente con la configurazione `TS/MP`.

E' possibile modificare il numero di processi EC in un queue manager dopo la creazione aggiungendo o cancellando le classi di server `TS/MP` e modificando la voce `ExpectedNumECs` della stanza `ECBoss` nel file `QMINI`. Questa operazione può essere eseguita solo dopo aver arrestato il queue manager.

Ripartizione del carico di lavoro nell'ambito del sistema

L'EC Boss provvede alla ripartizione del carico di lavoro di un queue manager sui vari EC. Le attività di elaborazione di un queue manager possono essere distribuite su più CPU in un modo bilanciato se la configurazione degli EC è corretta.

Quando viene ricevuta una richiesta di nuova connessione da un'applicazione locale oppure quando bisogna avviare un canale, l'EC Boss assegna la richiesta all'EC col minor numero di `LQMA` e `MCA` attivi.

Terminale iniziale del queue manager

Tutti i processi Compaq NSK, compresi i processi server di queue manager, dispongono di un terminale iniziale. Il terminale *deve* esistere ed essere in pausa. Di norma, il terminale iniziale del queue manager non viene utilizzato per l'output. Il terminale iniziale può essere qualsiasi unità terminale valida, compreso il prodotto Compaq Virtual Hometerm Service (VHS).

Compaq NonStop Kernel consente un massimo di 255 opener principali di un terminale fisico. Si raccomanda pertanto un'attenta pianificazione per assicurare che non venga superato questo limite. Si raccomanda di utilizzare il prodotto VHS della Compaq se è richiesto il supporto per più di 255 opener.

Identificare un terminale iniziale del queue manager nel parametro `-o HomeTerminalName` del comando `crtmqm`. Non esiste un valore predefinito; questo parametro è obbligatorio.

Gli attributi `HOMETERM` e `OUT` di tutte le classi server `TS/MP` sono impostate sull'unità terminale specificata. E' possibile modificare questi attributi quando lo si desidera quando le classi server sono in uno stato di arresto, di norma quando il queue manager è in uno stato di arresto.

Anche la voce `HomeTerminalName` nella stanza `Configuration` nel file `QMINI` deve essere modificata per modificare il terminale iniziale di un queue manager.

Il nome di processo PATHMON per il queue manager

Ciascun queue manager viene eseguito sotto la propria configurazione `TS/MP` (Pathway). Il processo che controlla quest'operazione è il processo `PATHMON`. Bisogna specificare un nome unico per ciascun queue manager. Il nome deve pertanto essere unico nel sistema.

Specificare il nome del processo `PATHMON` nel parametro `-n PATHMONProcessName` del comando `crtmqm`. Non esiste un valore predefinito; questo parametro è obbligatorio.

Creazione dei Queue Manager

Specificare un nome di processo che sia unico nel sistema e che sia facile da associare al queue manager da esso controllato.

E' possibile modificare il nome del processo PATHMON per un queue manager nel seguente modo:

1. Arrestare il queue manager.
2. Impostare il volume ed il volume secondario predefiniti sulla posizione dei file di dati del queue manager (di norma <nome QMgr>D).
3. Modificare la voce PathmonProcName nel file QMINI del queue manager per specificare il nuovo nome di processo.
4. Avviare PATHMON per il queue manager, utilizzando il *nuovo* nome di processo.

Da TACL, eseguire il seguente comando:

```
PATHMON /nome $<nuovo nome>, nowait/
```

5. Eseguire PATHCOM sul PATHMON appena ridenominato.

Da TACL, eseguire il seguente comando:

```
PATHCOM $<nuovonome>
```

6. Caricare la configurazione Pathway del configuration manager e confermare la modifica del nome del processo PATHMON.

Da PATHCOM, eseguire il seguente comando:

```
START PATHWAY COOL
```

Durante il caricamento della configurazione, verrà visualizzato un messaggio di avvertenza che indica che il nome del nuovo processo PATHMON è diverso da quello memorizzato nel file di configurazione. Dopodiché, il programma richiederà di confermare se si desidera proseguire. Immettere y sul prompt; il caricamento della configurazione verrà completato.

7. Salvare nuovamente le nuove informazioni di configurazione di PATHWAY nel database.

Da PATHCOM, eseguire i seguenti comandi:

```
SHUTDOWN2  
EXIT
```

Bisogna inoltre modificare la voce PathmonProcName nella stanza Configuration del file QMINI.

La modifica del nome del processo PATHMON è adesso completa. Il prossimo **strmqm** avvierà il queue manager utilizzando il nuovo nome di processo PATHMON.

Il CCSID del queue manager

È il CCSID (Coded Character Set ID) del set di caratteri utilizzato dal queue manager per memorizzare le informazioni relative ai messaggi.

Specificare il CCSID nel parametro `-l CCSID` del comando `crtmqm`. Il valore predefinito è 819.

Il CCSID del queue manager può essere sempre modificato dopo che è stata eseguita la creazione del queue manager utilizzando `runmqsc`, la funzione MQM oppure i comandi PCF.

Controllo della conversione dei dati EBCDIC

È possibile controllare il modo in cui vengono gestiti i caratteri NL (New Line) EBCDIC durante la conversione utilizzando la voce `ConvEBCDICNewline` nella stanza `AllQueueManagers` del file di configurazione `MQSINI`.

`ConvEBCDICNewline=NL_TO_LF|TABLE|ISO`

EBCDIC code page contiene un carattere NL (New Line) non supportato dai code page ASCII; tuttavia, alcune varianti ISO del formato ASCII contengono un equivalente. Utilizzare l'attributo `ConvEBCDICNewline` per specificare il metodo che MQSeries deve utilizzare in fase di conversione del carattere NL EBCDIC in formato ASCII.

NL_TO_LF

Specificare `NL_TO_LF` se si desidera che il carattere NL EBCDIC (X'15') venga convertito nel carattere Line Feed ASCII, LF (X'0A'), per tutte le conversioni da EBCDIC a ASCII. `NL_TO_LF` è predefinita.

TABLE

Specificare `TABLE` se si desidera convertire il carattere EBCDIC NL in base alle tabelle di conversione utilizzate sulla piattaforma per tutte le conversioni EBCDIC in ASCII. Notare che l'effetto di questo tipo di conversione può variare da piattaforma a piattaforma e da linguaggio a linguaggio, mentre sulla medesima piattaforma può variare il carattere in base all'utilizzo di differenti CCSIDs.

ISO Specificare ISO se si desidera che:

- i CCSID ISO vengano convertiti utilizzando il metodo `TABLE`.
- Tutti gli altri CCSID vengano convertiti utilizzando il metodo `NL_TO_LF`.

Esempi di possibili ISO CCSIDs sono riportati nella Tabella 2.

Tabella 2. Elenco di possibili ISO CCSIDs

CCSID	Code Set
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

Creazione dei Queue Manager

Se il CCSID ASCII non è un sottoinsieme ISO, ConvEBCDICNewLine assume come valore predefinito NL_TO_LF.

Per ulteriori informazioni sulla conversione dei dati, consultare: *MQSeries Application Programming Guide*.

EMS Collector per il queue manager

Se richiesto, è possibile configurare il queue manager in modo che utilizzi un collector alternativo. Per impostazione predefinita, gli eventi EMS vengono inviati a \$0. La voce EMSCollectorName nella stanza Configuration nel file QMINI specifica il nome dell'EMS Collector per questo queue manager.

E' sempre possibile modificare l'EMS Collector modificando il valore di questa voce, anche se questa modifica diventa effettiva solo dopo il riavvio del queue manager.

Il pool di agenti tenuto pronto da ciascun EC nel queue manager

Per ciascuno di questi quattro tipi di agenti di base, un EC può mantenere un pool di processi agenti inattivi, pronti da assegnare a nuovi lavori. La dimensione di questi pool può essere configurata in modo da raggiungere un adeguato bilanciamento tra tempo di risposta al nuovo lavoro e inizializzazione delle risorse. I valori dei seguenti campi della stanza Configuration nel file QMINI possono essere modificati in modo da specificare un numero diverso di processi da tenere inattivi:

MinIdleMCALU62Responders

Specifica il numero minimo di MCA Responder SNA LU 6.2 da mantenere in uno stato inattivo. Il valore predefinito è 0.

MinIdleMCATCPResponders

Specifica il numero minimo di MCA Responder TCP/IP da mantenere in uno stato inattivo. Il valore predefinito è 0.

MinIdleMCACallers

Specifica il numero minimo di MCA caller da mantenere in uno stato inattivo. Il valore predefinito è 0.

MinIdleLQMAgents

Specifica il numero minimo di LQMA (local queue manager agent (LQMA)) da mantenere in uno stato inattivo. Il valore predefinito è 1.

Notare che il numero di processi specificato in questi campi è valido per ciascun EC e non per ciascun queue manager. Pertanto, per un queue manager con due EC, ci sono per impostazione predefinita un minimo di due LQMA inattivi.

E' sempre possibile modificare questi valori anche se le modifiche diventano effettive solo dopo il riavvio del queue manager.

Limite massimo di riutilizzo processi e agenti inattivi

Per impostazione predefinita, un queue manager consente un massimo di 10 processi agente di ciascun tipo inattivi. Questo valore è controllato dalla voce MaxIdleAgents nella stanza Configuration del file QMINI.

La voce `MaxIdleAgentReuse` determina il numero di volte per cui è possibile riutilizzare un processo agente prima che venga sostituito da un nuovo processo agente. Per impostazione predefinita, `MaxIdleAgentReuse` è impostato su 10.

Priorità dei processi queue manager

È possibile che occorra modificare le priorità per bilanciare le risorse tra `MQSeries` e le altre applicazioni. Le priorità di processo delle classi server `TS/MP` possono essere modificate modificando (`ALTER`) gli oggetti `TS/MP` quando il queue manager è in stato di arresto.

Le priorità dei processi agente possono essere modificate modificando i campi `MCAAgentPriority` e `LQMAgentPriority` della stanza `EC` del file `QMINI`.

Fatta eccezione per i status server ed i queue server, che hanno una priorità di processo predefinita di 176, i processi configurati `TS/NP` hanno tutti una priorità predefinita di 175. Per impostazione predefinita, sia gli `MCA` che gli `LQMA` hanno una priorità di processo di 165.

Controllare che i status server ed i queue server abbiano la più alta priorità, seguiti da `EC Boss` ed `EC`, che a loro volta devono avere una priorità di processo più alta degli `MCA` e degli `LQMA`.

Numero massimo di canali per il queue manager

Esiste un limite per il numero di canali che può essere controllato simultaneamente per un queue manager. Se il limite è troppo elevato, le prestazioni potrebbero risentirne negativamente poiché questo parametro indica la dimensione della tabella di stato dei canali, in cui vengono eseguite numerose operazioni di ricerca. Se il limite è troppo basso, è possibile che non si riesca a controllare un numero di canali sufficiente per l'applicazione. Il campo `MaxChannels` della stanza `Channels` nel file `QMINI` definisce il numero massimo di canali che può essere controllato simultaneamente.

Il valore predefinito alla creazione è 10. Non esiste alcun modo per modificare l'impostazione predefinita in fase di creazione.

La voce `MaxChannels` nella stanza `Channels` del file `QMINI` può sempre essere modificata, anche se le modifiche effettuate diventeranno effettive solo dopo il riavvio del queue manager.

Numero massimo di canali attivi per il queue manager

Esiste un limite per il numero di canali simultaneamente attivi (in esecuzione) in un queue manager. È possibile utilizzarlo per controllare il limite massimo di richieste di risorse di sistema da parte dei canali. La voce `MaxActiveChannels` nella stanza `Channels` stanza nel file `QMINI` definisce il numero massimo di canali attivi per il queue manager.

Il valore predefinito alla creazione è 10. Non esiste alcun modo per modificare l'impostazione predefinita in fase di creazione.

La voce `MaxActiveChannels` nel file `QMINI` può sempre essere modificata, anche se le modifiche effettuate diventeranno effettive solo dopo il riavvio del queue manager.

Creazione dei Queue Manager

ID di segmento Guardian utilizzati da MQSeries

MQSeries assegna dei segmenti di memoria Guardian sia nei propri processi che nel processo del programma applicativo. Gli ID di segmento Guardian utilizzati da MQSeries per questi segmenti sono assegnati da un intervallo definito nel file di configurazione MQSINI. La voce `NSKSegidRange` nella stanza `AllQueueManagers` del file di configurazione MQSINI definisce l'intervallo di ID di segmento utilizzato da MQSeries per l'assegnazione dei segmenti di memoria.

Quando viene eseguita l'installazione di MQSeries, quest'intervallo viene impostato automaticamente su `NSKSegidRange=10-20`.

Poiché questi segmenti di memoria vengono assegnati anche nel processo del programma applicativo, assicurare che le applicazioni non assegnino segmenti in questo intervallo. E' possibile forzare MQSeries ad assegnare i propri segmenti in un intervallo diverso modificando la voce `NSKSegidRange` nel file MQSINI. Dopo aver modificato questa voce, bisogna arrestare e riavviare tutti i queue manager perché il nuovo valore diventi effettivo.

L'intervallo definito dal parametro `NSKSegidRange` deve essere interamente contenuto nei limiti di ID di segmento Guardian di 0-1023. Il parametro `NSKSegidRange` deve definire un intervallo che contiene almeno 10 ID di segmento che possono essere utilizzati da MQSeries.

Porta TCP/IP predefinita

La voce `TCPPort` nella stanza `TCPConfig` in QMINI definisce il numero di porta predefinito per i canali in uscita. Per impostazione predefinita, viene utilizzato il numero di porta 1414. Questo valore viene sostituito dai valori di numero di porta specificati nel campo `CONNNAME` per un canale.

Porte TCP/IP su cui è in ascolto il queue manager

E' possibile configurare un queue manager con dei canali TCP/IP in modo che sia in ascolto per connessioni in entrata su una o più porte TCP/IP. `TCPLNumListenerPorts` e le voci nella stanza `TCPConfig` nel file QMINI definiscono il numero di porte su cui essere in ascolto ed il numero di porte assegnato a questo queue manager. Per esempi di voci QMINI, consultare: "Esempi di comunicazione TCP/IP" a pagina 393.

Su un singolo sistema possono esserci più queue manager. A ciascun queue manager su un sistema deve essere assegnato un gruppo di porte TCP/IP (senza sovrapposizioni) su cui essere in ascolto. Il gruppo di porte TCP/IP per ciascun queue manager può essere una sola porta, dove il numero di richieste di connessione TCP/IP in entrata è basso, oppure più di una porta per grosse configurazioni. La porta TCP/IP predefinita è 1414 e, per impostazione predefinita, viene creato un queue manager in ascolto solo su questa porta.

L'elenco di porte in ascolto può essere modificato modificando la stanza `TCPConfig` nel file QMINI e riavviando il queue manager. Per essere in ascolto su più di una porta, un queue manager deve essere inoltre configurato con ulteriori classi server di listener TCP/IP con TS/MP. Questa operazione viene eseguita manualmente utilizzando `PATHCOM`.

In alternativa, è possibile specificare un numero di porta nel comando `runmqtsr` (descritto in: "runmqtsr (Run listener)" a pagina 296). I valori `TCPLListenerPort`

vengono sostituiti da un programma di classe server Listener se il parametro MQLISTENPORTNUM è presente nell'ambiente di detto programma Listener.

Processo TCP/IP utilizzato dal queue manager

L'interfaccia del prodotto Compaq TCP/IP utilizza un processo server, noto come processo TCP/IP. Per impostazione predefinita, viene utilizzato il valore predefinito \$ZTC0,. Non esiste alcun modo per modificare l'impostazione predefinita in fase di creazione del queue manager. E' possibile configurare i canali del queue manager in modo che utilizzino uno specifico processo TCP/IP, se il valore predefinito di sistema non è sufficiente. La configurazione della classe server TS/MP del TCP/IP Listener deve essere modificata in modo manuale se il processo TCP/IP predefinito di sistema, \$ZTC0, non è sufficiente oppure non è corretto. Questa configurazione di classe server può anche essere modificata in modo manuale per abilitare un Listener ad essere in ascolto su una porta specifica e sostituire le porte definite nel file QMINI.

Alla classe server MQS-TCPLIS00 deve inoltre essere aggiunto DEFINE TCPIP^PROCESS^NAME per fare riferimento al nome di processo TCP/IP alternativo richiesto. Notare che, se di dispone di più EC, bisogna aggiornarli tutti. Per ulteriori informazioni, consultare i manuali del prodotto Compaq NSK TCP/IP.

Le modifiche alle classi server TS/MP possono essere apportate solo quando il queue manager è in uno stato di arresto.

MQSeries for Compaq NSK può supportare più processi TCP/IP per queue manager. Sono inoltre supportati più processi listener e più porte TCP/IP.

Riconfigurazione della classe server MQS-TCPLISnn per una porta ed un processo TCP/IP non predefiniti

Per riconfigurare la classe server MQS-TCPLISnn per una porta ed un processo non predefiniti, seguire le istruzioni generali contenute in: "Modifica dei parametri delle classi di server PATHWAY" a pagina 42.

Per specificare un processo TCP/IP non predefinito, utilizzare i seguenti comandi PATHCOM:

```
ALTER SERVER MQS-TCPLIS00, (DEFINE
=TCPIP^PROCESS^NAME, FILE $ZZZZ)
```

dove \$ZZZZ è il nome del processo TCP/IP richiesto.

Per specificare una specifica porta, utilizzare i seguenti comandi PATHCOM:

```
ALTER SERVER MQS-TCPLIS00, PARAM MQLISTENPORTNUM "nnnn"
```

dove nnnn è il numero della porta su cui porsi in ascolto.

Per i listener in esecuzione da TACL (**runmqldr**) che richiedono un nome di processo TCP/IP diverso da quello predefinito \$ZTC0, aggiungere quanto segue alla variabile TACL da sarà eseguito il listener:

Creazione dei Queue Manager

```
ADD DEFINE =TCPIP^PROCESS^NAME, FILE $ZZZZ)
```

dove \$ZZZZ è il nome del processo TCP/IP richiesto.

Nota: Se singole classi server di listener TCP/IP desiderano utilizzare diversi processi TCP/IP con ciascuna porta, bisogna che definiscano sia la porta che il nome di processo TCP/IP in ciascuna istanza di classe server, altrimenti non sarà possibile essere certi di quale processo TCP/IP utilizzerà una porta definita nel file QMINI.

Se il parametro MQLISTENPORTNUM non è definito nell'ambiente del programma listener, il Listener ottiene la porta dal file QMINI.

Assegnazione di spazio di swap

MQSeries for Compaq NSK assegna dello spazio di swap sulla base dei valori ExtPoolSize per i vari eseguibili nel file di configurazione QMINI. Pertanto, se il queue manager sta utilizzando il file QMINI predefinito, che assegna 300 KB a ciascun eseguibile per impostazione predefinita, e sta eseguendo 10 canali in uscita, 20 agenti e 10 TCP Responder, sono necessari aumento 12 MB (40 * 300 KB) di spazio di swap. Per ridurre questo requisito, è possibile ridurre i valori nel file QMINI. I valori ExtPoolSize sono la memoria aggiuntiva minima assegnata quando si esaurisce l'assegnazione di memoria iniziale. Questo valore non deve essere superiore alla dimensione di messaggio massima per il queue manager.

Un modo per controllare l'assegnazione dello spazio di swap degli eseguibili di MQSeries consiste nel modificare le classi server MQ Pathway aggiungendo:

```
DEFINE =_DEFAULTS, CLASS DEFAULTS, VOLUME volume.qmD  
subvolume, SWAP volume
```

Aggiungendo DEFINE, i processi agente creati dal server EC ereditano anche il volume di swap definito creando così una certa scalabilità dell'utilizzo della swap quando vengono utilizzate più classi server EC.

Nome status server predefinito

Bisogna specificare un nome di processo univo per il process pair status server predefinito quando si esegue la creazione di un queue manager. Specificare il nome status server predefinito nel parametro *-s StatusServerName* obbligatorio del comando **crtmqm**. Non esiste un valore predefinito per questo parametro. Specificare un nome di processo che sia unico nel sistema e facile da associare al queue manager cui appartiene.

E' possibile modificare il nome di processo status server predefinito per un queue manager nel seguente modo:

1. Arrestare il queue manager.
2. Impostare il volume ed il volume secondario predefiniti sulla posizione dei file di dati del queue manager (di norma <NomeQMgr>D).
3. Modificare la voce DefaultStatusServerName nella stanza Configuration del file MQNIN del queue manager per registrare il nuovo nome di processo.

Creazione dei Queue Manager

4. Eseguire PATHMON per il queue manager immettendo il seguente comando da TACL:

```
PATHMON /nome $<pmon>, nowait/
```

dove <pmon> è il nome del processo PATHMON per il queue manager.

5. Eseguire PATHCOM sul processo PATHMON immettendo il seguente comando da TACL:

```
PATHCOM $<pmon>
```

6. Caricare la configurazione Pathway del queue manager immettendo il seguente comando da PATHCOM:

```
START PATHWAY COOL
```

7. Modificare il valore MQS-STATUS00 del server e reimpostare il processo immettendo i seguenti comandi da PATHCOM:

```
ALTER SERVER MQS-STATUS00, DELETE PROCESS  
$<vecchionome>  
ALTER SERVER MQS-STATUS00, PROCESS $<nuovonome>
```

8. Salvare la configurazione modificata su disco immettendo i seguenti comandi da PATHCOM:

```
SHUTDOWN2  
EXIT
```

La modifica del nome di processo status server predefinito è adesso completa. Il prossimo comando **strmqm** avvierà il queue manager utilizzando il nome di processo status server predefinito.

Aggiunta e rimozione di status server non predefiniti

Per aggiungere altri status server, utilizzare lo status server predefinito esistente come modello. Non è necessario arrestare il queue manager per poter aggiungere un nuovo status server.

La seguente procedura aggiunge un nuovo status server:

Nota: Questa procedura presuppone che il queue manager sia in esecuzione. In caso contrario, bisogna avviare PATHMON e caricare la configurazione PATHWAY prima di avviare questa procedura. Bisogna inoltre omettere il passo 5 a pagina 60 e salvare la configurazione Pathway su disco alla fine utilizzando un comando SHUTDOWN2.

1. Eseguire PATHCOM sul processo PATHMON immettendo il seguente comando da TACL:

Creazione dei Queue Manager

```
PATHCOM $<pmon>
```

2. Creare un gruppo di attributi di lavoro basato sulla classe status server predefinita come un modello immettendo i seguenti comandi da PATHCOM:

```
RESET SERVER  
SET LIKE MQS-STATUS00
```

3. Modificare il gruppo di lavoro per il nuovo server immettendo i seguenti comandi da PATHCOM:

```
SET SERVER CPUS(n:m)  
RESET SERVER PROCESS $<nome processo status server  
predefinito>  
SET SERVER PROCESS $<nuovo nome status server>
```

4. Aggiungere un nuovo server, fornendogli un nuovo nome di classe server immettendo il seguente comando da PATHCOM:

```
ADD SERVER MQS-STATUSxx
```

5. Avviare il nuovo status server, in modo che venga utilizzato, ed uscire da PATHCOM immettendo i seguenti comandi:

```
START SERVER MQS-STATUSxx  
EXIT
```

Le convenzioni di denominazione consigliate per le classi server aggiuntive sono MQS-STATUS01, MQS-STATUS02 e così via. Tuttavia, non è obbligatorio utilizzare queste convenzioni. Posto che il nome della classe server inizi con la stringa di caratteri MQS-STATUS, la classe server verrà correttamente avviata da **strmqm**.

Dopo aver aggiunto ed avviato uno status server (in modo esplicito utilizzando PATHCOM o in modo implicito utilizzando **strmqm**), è possibile riassegnare gli oggetti al nuovo status server utilizzando **altmqfls**. Per ulteriori informazioni sulla riassegnazione degli oggetti, consultare: "Riassegnazione degli oggetti agli status server e ai queue server" a pagina 111.

Prima di rimuovere uno status server, controllare che tutti gli oggetti configurati per questo status server siano stati cancellati oppure riassegnati ad un altro status server. **Non** cancellare lo status server predefinito, altrimenti non sarà più possibile utilizzare il queue manager.

Nome queue server predefinito

Bisogna specificare un nome di processo unico per il process pair queue server predefinito quando si esegue la creazione di un queue manager. Specificare il nome queue server predefinito nel parametro `-v QueueServer` obbligatorio del comando **crtmqm**. Non esiste un valore predefinito per questo parametro. Specificare un nome di processo che sia unico nel sistema e facile da associare al queue manager cui appartiene.

E' possibile modificare il nome del processo queue server predefinito per un queue manager nel seguente modo:

1. Arrestare il queue manager.
2. Impostare il volume ed il volume secondario predefiniti sulla posizione dei file di dati del queue manager (di norma `<NomeQMGr>D`).
3. Modificare la voce `DefaultQueueServerName` nella stanza Configuration del file `MQNIN` del queue manager per registrare il nuovo nome di processo.
4. Eseguire `PATHMON` per il queue manager immettendo il seguente comando da `TACL`:

```
PATHMON /nome $<pmon>, nowait/
```

dove `<pmon>` è il nome del processo `PATHMON` per il queue manager.

5. Eseguire `PATHCOM` sul processo `PATHMON` immettendo il seguente comando da `TACL`:

```
PATHCOM $<pmon>
```

6. Caricare la configurazione Pathway del queue manager immettendo il seguente comando da `PATHCOM`:

```
START PATHWAY COOL
```

7. Modificare il valore `mq5-queuenn` del server e reimpostare il processo immettendo i seguenti comandi da `PATHCOM`:

```
ALTER SERVER MQ5-QUEUEnn, DELETE PROCESS
$<vecchionome>
ALTER SERVER MQ5-QUEUEnn, PROCESS $<nuovonome>
```

8. Salvare la configurazione modificata su disco immettendo i seguenti comandi da `PATHCOM`:

```
SHUTDOWN2
EXIT
```

Creazione dei Queue Manager

La modifica del nome di processo status server predefinito è adesso completa. Il prossimo comando **strmqm** avvierà il queue manager utilizzando il nome di processo status server predefinito.

Aggiunta e rimozione di queue server non predefiniti

Per aggiungere altri queue server, utilizzare il queue server predefinito esistente come modello. Non è necessario arrestare il queue manager per poter aggiungere un nuovo queue server.

La seguente procedura aggiunge un nuovo queue server:

Nota: Questa procedura presuppone che il queue manager sia in esecuzione. In caso contrario, bisogna avviare PATHMON e caricare la configurazione PATHWAY prima di avviare questa procedura. Bisogna inoltre omettere il passo 5 e salvare la configurazione PATHWAY su disco alla fine utilizzando un comando SHUTDOWN2.

1. Eseguire PATHCOM sul processo PATHMON immettendo il seguente comando da TACL:

```
PATHCOM $<pmon>
```

2. Creare un gruppo di attributi di lavoro basato sulla classe queue server predefinita come un modello immettendo i seguenti comandi da PATHCOM:

```
RESET SERVER  
SET LIKE MQS-QUEUEnn
```

3. Modificare il gruppo di lavoro per il nuovo server immettendo i seguenti comandi da PATHCOM:

```
SET SERVER CPUS(n:m)  
RESET SERVER PROCESS $<nome processo queue server  
predefinito>  
SET SERVER PROCESS $<nuovo nome queue server>
```

4. Aggiungere un nuovo server, fornendogli un nuovo nome di classe server immettendo il seguente comando da PATHCOM:

```
ADD SERVER MQS-QUEUEnn
```

5. Avviare il nuovo queue server, in modo che venga utilizzato, ed uscire da PATHCOM immettendo i seguenti comandi:

```
START SERVER MQS-QUEUEnn  
EXIT
```

Le convenzioni di denominazione consigliate per le classi server aggiuntive sono MQS-QUEUE01, MQS-QUEUE02 e così via. Tuttavia, non è obbligatorio utilizzare queste convenzioni. Posto che il nome della classe server inizi con la stringa di caratteri MQS-QUEUE, la classe server verrà correttamente avviata da **strmqm**.

Dopo aver aggiunto ed avviato uno queue server (in modo esplicito utilizzando PATHCOM o in modo implicito utilizzando **strmqm**), è possibile riassegnare gli oggetti al nuovo queue server utilizzando **altmqfls**. Per ulteriori informazioni sulla riassegnazione degli oggetti, consultare: "Riassegnazione degli oggetti agli status server e ai queue server" a pagina 111.

Prima di rimuovere uno queue server, controllare che tutti gli oggetti configurati per questo queue server siano stati cancellati oppure riassegnati ad un altro queue server. **Non** cancellare lo queue server predefinito, altrimenti non sarà più possibile utilizzare il queue manager.

Struttura dei volumi

I file per MQSeries for Compaq NSK sono distribuiti in più volumi secondari. Il volume in cui si trovano questi volumi secondari viene selezionato quando viene eseguita la creazione del queue manager: viene preso dal valore di volume predefinito in MQSINI oppure specificato nel parametro *-p DefaultPrefix* del comando **crtmqm**.

Esistono cinque volumi secondari per queue manager. Il contenuto dei volumi secondari è determinato dal carattere finale del nome del volume secondario. Ad esempio, per un queue manager denominato QMGR che si trova su un volume \$DATA, sono di norma presenti i seguenti volumi secondari:

\$DATA.QMGR	volume secondario FFST
\$DATA.QMGRD	volume secondario file di dati queue manager
\$DATA.QMGRL	volume secondario log errori queue manager
\$DATA.QMGRM	volume secondario coda messaggi
\$DATA.QMGRS	volume secondario sincronizzazione canali

Se il nome del queue manager ha una lunghezza superiore ai sette caratteri, i nomi dei volumi secondari vengono modificati oppure abbreviati. Le stanze del file MQSINI QMVolume e QMSubVolume per il queue manager sono utilizzate per registrare la posizione ed i nomi di questi volumi secondari.

Volume secondario FFST del queue manager

Il volume secondario FFST contiene i file di supporto delle principali condizioni di malfunzionamento. Questi file iniziano tutti con il prefisso FD. Indicano seri problemi con il sistema MQSeries, come ad esempio un'insufficienza delle risorse, degli errori MQSeries interni oppure dei problemi con il sistema Compaq NSK.

Volume secondario file di dati queue manager

AMQRFNxx	Sono i file Repository Cache Shared Memory per ciascuna CPU (numero <i>xx</i>).
CCHDEFS	E' il file di definizione dei canali di connessione.
CCSIDMEM	E' il file di memoria condivisa di sola lettura per il supporto CCSID di conversione dati.
CHDEFS	E' il file di definizione canali. Questo file contiene le informazioni di configurazione per i canali definiti per un queue manager.

Struttura dei volumi

Lxxxxxx	Sono i file di elenchi nomi. Lxxxxxx è derivato dal nome oggetto; altrimenti, è un valore generato.
OAMDB	E' il database delle autorizzazioni.
OBJCAT	E' il catalogo oggetti.
ABJCAT	E' il file di chiavi alternative per OBJCAT che contiene un indice per nome oggetto.
PATHCTL	E' il file di controllo PATHWAY.
PRIDB	E' il database principal (utenti) di MQSeries.
PRIDBA	E' il file di chiavi alternative per PRIDB.
QMINI	E' il file di inizializzazione del queue manager.
QMINIMEM	E' il file di memoria condivisa di sola lettura per le informazioni di configurazione statica generale per il queue manager.
REPMGR	E' un file utilizzato per coordinare l'avvio di più processi di server di archivio di memoria.
RDFPURGE	E' il database utilizzato dal queue manager per registrare i file cancellati logicamente che verranno eventualmente rimossi da cleanrdf .
RUNTIME	E' il file per la coordinazione del ripristino di ECBOSS e EC.
SHUTDOWN	E' il file che controlla endmqm .
STATABLE	E' il file di tabelle di stato di canale. Questo file contiene le informazioni dinamiche associate allo stato di canale. Utilizzato per salvare le informazioni sugli stati dei canali in fase di arresto e riavvio di un queue manager.
TRACEOPT	Il file TRACEOPT contiene le impostazioni di traccia correnti per un queue manager sotto forma di un record bit-map non formattato. I comandi di controllo strmqtrc e endmqtrc modificano il contenuto del file, utilizzando il meccanismo CONTROL 27 per notificare a tutti i processi l'aggiornamento.
UMQSINI	E' uno snapshot del file MQSINI non strutturato in fase di avvio del queue manager.

Volume secondario log errori queue manager

Il volume secondario log errori contiene i file di log di traccia e di errori. Il prefisso TR identifica i file di traccia. E' possibile modificare il prefisso modificando la voce TracePrefix nel file QMINI. I file di traccia contengono le informazioni di diagnostica, e sono creati solo se la funzione di traccia viene attivata utilizzando la funzione MQM oppure il comando di controllo **strmqtrc**.

I log di errori hanno nomi nel formato MQERRLGn, dove n è 1, 2 o 3. MQERRLG1 è sempre il log di errori corrente. Il suo contenuto viene spostato in MQERRLG2 quando MQERRLG1 è pieno; MQERRLG2 viene spostato in MQERRLG3 quando MQERRLG1 viene in seguito svuotato. Se necessario, MQERRLG3 viene sovrascritto. Non ci sono mai più di tre log di errori; è pertanto assegnare loro delle dimensioni corrette per evitare la perdita di utili informazioni sugli errori.

Volume secondario coda messaggi del queue manager

Il volume secondario di coda messaggi contiene i file associati alla memorizzazione dei messaggi nelle code locali. I nomi di file sono nel seguente formato:

Qxxxxxxx	Sono i file di coda stessi, che contengono i messaggi permanenti.
Txxxxxxx	Sono i touch file. Se un oggetto viene modificato, i touch file modificano la data-ora dell'oggetto. Txxxxxxx è derivato dal nome oggetto; altrimenti, è un valore generato.
Oxxxxxxx	E' il file di overflow di coda.
xxxxxxx	Può essere il nome coda se esso è un nome breve univoco; altrimenti, è un valore generato. (consultare la sezione "Trasformazione dei nomi di oggetto").

Oltre ai file sopra descritti, con specifiche lettere di prefisso dei nomi file, i file di overflow di messaggio vengono creati nel volume secondario delle code di messaggi per impostazione predefinita. I file di overflow di messaggio vengono creati per memorizzare messaggi di notevoli dimensioni e, anche se non hanno uno specifico prefisso di nome di file, possono essere facilmente identificati perché sono file non strutturati e non sottoposti a procedure di controllo. La posizione dei nuovi file di overflow di messaggio può essere modificata utilizzando **altmqfls** su ogni singola coda specificando il parametro `--msgofsubvol`. (consultare la sezione "dspmqfls (Visualizza gli attributi file MQSeries)" a pagina 278 per ulteriori informazioni).

Volume secondario sincronizzazione canali del queue manager

Il volume secondario sincronizzazione messaggi del queue manager contiene i database interni che registrano lo stato delle unità di lavoro (o batch di messaggi) trasmessi o ricevuti sui canali di proprietà del queue manager.

Dopo che i canali sono stati utilizzati su un queue manager, il volume secondario contiene i seguenti file:

SYNCHIDX

Il file di indice di sincronizzazione. Contiene una voce per ciascun file di sincronizzazione creato dal queue manager.

Sxxxxxxx

File di sincronizzazione individuali. Esiste un file per ciascuna combinazione unica di canale locale e remoto utilizzata nel queue manager. Questi file registrano le identità dei messaggi trasmessi o ricevuti in un batch di messaggi. Queste informazioni vengono utilizzate nelle risincronizzazione dei canali in seguito a condizioni di malfunzionamento e per la risoluzione di canali in stato dubbio.

Trasformazione dei nomi di oggetto

I nomi degli oggetti non sono necessariamente dei validi nomi di file system. Pertanto, potrebbe essere necessario modificare i nomi di oggetto. Il metodo utilizzato è diverso da quello utilizzato per i nomi di queue manager poiché, anche se è possibile che vi siano solo pochi nomi di queue manager per sistema, potrebbero esserci molti alti oggetti per ciascun queue manager. Solo le definizioni di processo, gli elenchi di nomi e le code sono rappresentati utilizzando file

Struttura dei volumi

separati nel file system; i canali e gli altri oggetti non sono influenzati da queste considerazioni poiché vengono memorizzati come record in database che contengono varie definizioni di oggetto.

Quando un nuovo nome viene generato dal processo di trasformazione, esso non presenta alcuna relazione con il nome originario. E' possibile utilizzare il comando **dspmqfls** per la conversione tra nomi di oggetto reali e trasformati: **dspmqfls** visualizza i nomi dei file principali associati ad un oggetto MQSeries.

Lavorare con i gestori code

MQSeries fornisce dei comandi di controllo per la creazione, l'avvio, la chiusura e la cancellazione di queue manager. E' anche possibile visualizzare gli attributi di un queue manager utilizzando il comando **MQSC DISPLAY QMGR** e modificandoli utilizzando **ALTER QMGR**. Consultare: "Visualizzazione degli attributi Queue Manager" a pagina 100 e: "Modifica degli attributi di Queue Manager" a pagina 101.

Controllare che la variabile di ambiente **PMSEARCHLIST** specifichi la posizione degli eseguibili MQSeries prima di tentare di utilizzare i comandi di controllo. Per ulteriori informazioni su questa variabile di ambiente, consultare: "Appendice C. Impostazione delle variabili di ambiente **TACL** per MQSeries for Compaq NSK" a pagina 325.

Creazione di un Queue Manager predefinito

Utilizzare il comando **crtmqm** per creare un Queue Manager predefinito. Il comando **crtmqm** specificato con un indicatore **-q**:

- Si crea un Queue Manager predefinito denominato **saturn.queue.manager**
- Si creano gli oggetti predefiniti e quelli di sistema
- Si specifica sia il nome della relativa coda di trasmissione che quello della relativa coda messaggi non recapitati.

```
crtmqm -q -d MY.DEF.XMITQ -u SYSTEM.DEAD.LETTER.QUEUE -n $PMON
-o $TRM01 -s $MQSS
-v $MQQS saturn.queue.manager
```

dove:

- q** Indica che questo è il Queue Manager predefinito.
- d MY.DEF.XMIT.Q**
E' il nome della coda di trasmissione predefinita.
- u SYSTEM.DEAD.LETTER.QUEUE**
Indica il nome della coda messaggi non recapitati.
- n \$PMON**
E' il nome di processo di **PATHMON** per il queue manager.
- o \$TRM01**
E' il nome del terminale iniziale (deve essere in pausa)
- s \$MQSS**
E' il nome di processo del status server predefinito.

-v \$MQQS

E' il nome di processo del queue server predefinito.

saturn.queue.manager

Indica il nome del Queue Manager. Per **crtmqm**, questo nome deve essere l'ultimo parametro nel comando.

Viene creato un queue manager con il nome e le opzioni specificati. Creando un queue manager, si creano automaticamente anche:

- Il principal predefinito di MQSeries per il creatore del queue manager
- Lo status server per il queue manager
- Il queue server per il queue manager
- Un file di configurazione per il queue manager

E' adesso possibile avviare il queue manager. Per ulteriori informazioni, consultare il "Avvio di un queue manager" a pagina 66.

E' possibile che l'utente desideri modificare alcuni degli attributi di un queue manager. E' possibile farlo utilizzando MQM oppure il comando di controllo **strmqm**. Per ulteriori informazioni, consultare Figura 5 a pagina 71 or "strmqm (Avvio Queue Manager)" a pagina 310.

Eseguire il backup dei due file di configurazione creati quando è stato creato il queue manager. Se si crea un altro queue manager che causa dei problemi, è possibile ripristinare le copie di backup. Per ulteriori informazioni, consultare il "Eseguire una copia di riserva dei file di configurazione dopo aver creato un Queue Manager" a pagina 47.

Creazione dei principal di MQSeries

Il comando **crtmqm** crea automaticamente un principal per l'utente che ha creato il queue manager. Questo principal (noto anche come principal predefinito) è anche detto **mqm** per compatibilità con altre implementazioni di MQSeries.

Dopo aver creato un queue manager, è possibile definire dei principal per altri utenti di MQSeries. Questa procedura può sempre essere eseguita (indipendentemente dal fatto che il queue manager sia stato avviato o meno). Se non sono richiesti altri utenti per il queue manager, è possibile omettere questa procedura.

Per creare un principal di MQSeries denominato **MQPRINCIPAL** corrispondente all'utente **Compaq NSK MQM.MQUSER**, immettere il comando:

```
altnqusr -m saturn.queue.manager -p MQPRINCIPAL -u MQM.MQUSER
```

Per visualizzare tutti i principal attualmente creati, immettere il comando:

```
dsmpnqusr -m saturn.queue.manager
```

Nota: se non si crea una voce principal per un utente, eventuali tentativi di accedere al queue manager eseguiti dall'utente in oggetto (indipendentemente dal fatto che OAM sia abilitato o meno) determineranno una condizione di errore di

Lavorare con i gestori code

autorizzazione. Questa modifica è stata introdotta in MQSeries for Compaq NSK Versione 2.2.0.1 ed è parte di tutte le versioni successive.

Esecuzione di cleanrdf per un queue manager abilitato a RDF

Se si sta eseguendo un queue manager nell'ambiente RDF e si è abilitata la modalità specifica RDF utilizzando il PARAM MQRDF, eseguire il programma di utilità **cleanrdf** su base periodica nel seguente modo:

- Dopo aver apportato le modifiche alla configurazione desiderate (quali creare o cancellare oggetti oppure apportare modifiche al file QMINI), eseguire **cleanrdf**.
- Se l'applicazione utilizzata crea e cancella oggetti come parte delle sue normali mansioni ed in particolar modo se utilizza delle code dinamiche, eseguire **cleanrdf** durante il normale utilizzo di detta applicazione con una frequenza basata sulla percentuale di cancellazione di oggetti. Utilizzare NetBatch oppure altro software per la pianificazione delle attività.

Avvio di un queue manager

Dopo aver creato un Queue Manager, perché questo possa elaborare comandi o chiamate MQI, è necessario avviarlo. Avviare il queue manager immettendo questo comando:

```
strmqm saturn.queue.manager
```

Il comando **strmqm** non restituisce il controllo fino a che il queue manager non sia stato avviato e sia pronto ad accettare le richieste di connessione.

Ripristino degli oggetti predefiniti e di sistema

Gli oggetti predefiniti e di sistema vengono creati automaticamente quando viene creato il queue manager; tuttavia, gli oggetti possono sempre essere sostituiti da altre definizioni di oggetto. Per ripristinare gli oggetti predefiniti e di sistema in un queue manager denominato saturn.queue.manager, immettere il comando **strmqm** con l'opzione **-c**:

```
strmqm -c saturn.queue.manager
```

Il queue manager viene avviato; gli oggetti predefiniti e di sistema creati in fase di creazione del queue manager vengono ripristinati; il queue manager viene quindi arrestato.

Analisi dei file di oggetto

Ciascun oggetto coda, queue manager o processo di MQSeries è rappresentato da un file. Poiché i nomi di questi oggetti non sono necessariamente nomi di file validi, il queue manager converte, se necessario, i nomi in nomi di file validi. Questa procedura è descritta in: "Trasformazione dei nomi di oggetto" a pagina 63.

Interruzione di un queue manager

Per arrestare un gestore code, utilizzare il comando **endmqm**. Ad esempio, per arrestare un queue manager denominato `saturn.queue.manager` utilizzare questo comando:

```
endmqm saturn.queue.manager
```

Per impostazione predefinita, questo comando esegue un arresto *controllato* o *ad attività completate* del Queue Manager specificato. Ciò può richiedere del tempo—una chiusura controllata attende la disconnessione di *tutte* le applicazioni collegate e l’arresto di tutti i canali in esecuzione.

“Chiusure immediate e prioritarie di un queue manager” descrive gli indicatori facoltativi per il comando **endmqm** che specificano come deve essere eseguito lo spegnimento.

In caso di problemi

I problemi di chiusura del Queue Manager sono spesso dovuti alle applicazioni. Ad esempio, quando le applicazioni:

- Non controllano in modo appropriato i codici di errore MQI.
- Non richiedono una notifica di un arresto.

Chiusure immediate e prioritarie di un queue manager

Se la chiusura di un queue manager è lenta, oppure se il queue manager non viene arrestato, è possibile interrompere il comando **endmqm** utilizzando **BREAK** seguito da **STOP**. Quindi, immettere nuovamente il comando **endmqm**, ma questa volta utilizzare un flag che specifichi o una chiusura immediata o una chiusura preventiva.

Per una *chiusura immediata*, le chiamate MQI correnti vengono completate, ma le nuove chiamate hanno esito negativo. Con questo tipo di chiusura, non viene atteso che le applicazioni vengano disconnesse dal gestore code. Per una chiusura immediata, il comando è:

```
endmqm -i saturn.queue.manager
```

Se una chiusura immediata non funziona, tentare una chiusura *prioritaria* specificando l’indicatore **-p**. Ad esempio:

```
endmqm -p saturn.queue.manager
```

Attenzione

Non utilizzare questo metodo fino a che non riescono tutti gli altri tentativi di interruzione del gestore code mediante il comando **endmqm**. Questo metodo potrebbe avere delle conseguenze imprevedibili sulle applicazioni connesse.

Lavorare con i gestori code

Se neanche questo metodo funziona, consultare: “Arresto manuale di queue manager” a pagina 333 per un’alternativa.

Per una descrizione dettagliata del comando **endmqm** e delle relative opzioni, consultare la sezione “endmqm (End queue manager)” a pagina 288.

Riavvio di un Queue Manager

Per riavviare un queue manager, utilizzare il comando:

```
strmqm saturn.queue.manager
```

Rendere predefinito un Queue Manager esistente

Quando si crea un Queue Manager predefinito, il nome viene inserito nella stanza *DefaultQueueManager* nel file di configurazione MQSeries (MQSINI). La voce e i relativi contenuti vengono creati automaticamente se non esistono già.

E’ possibile modificare questa stanza:

- Per rendere predefinito un queue manager esistente. Per eseguire quest’attività, bisogna modificare il nome di queue manager in questa stanza nel nome del nuovo queue manager predefinito. Bisogna eseguire questa operazione in modo manuale utilizzando un editor di testo.
- Se non si dispone di un queue manager predefinito nel nodo, e si desidera rendere predefinito un queue manager esistente. Per eseguire questa operazione, bisogna creare la stanza *DefaultQueueManager*—con il nome richiesto—manualmente.
- Se si imposta accidentalmente un altro queue manager come predefinito e si desidera reimpostare il queue manager predefinito originario. Per eseguire questa operazione, modificare la stanza *DefaultQueueManager* nel file di configurazione MQSeries, sostituendo il nome del queue manager predefinito indesiderato con quello desiderato.

Il queue manager predefinito viene impostato su quello specificato. Tutti i successivi tentativi di connessione in cui l’applicazione non specifica un queue manager, stabiliranno una connessione con il nuovo queue manager predefinito.

Consultare il “Capitolo 13. File di configurazione” a pagina 187 per informazioni relative ai file di configurazione.

Quando sono state immesse nella voce le informazioni richieste, arrestare il Queue Manager e riavviarlo.

Eliminazione di un queue manager

Per cancellare un queue manager, è necessario prima arrestarlo ed utilizzare quindi il seguente comando:

```
dltmqm saturn.queue.manager
```

Attenzione

Attenzione: cancellare un queue manager cancella anche tutte le risorse ad esso associate, incluso tutte le code ed i loro messaggi e tutte le definizioni di oggetto. Inoltre, è possibile che vengano eliminati tutti i dati contenuti nei volumi secondari del queue manager (anche se non sono stati creati da MQSeries).

Per una descrizione del comando **dltmqm** e delle relative opzioni, consultare la sezione “dltmqm (Delete queue manager)” a pagina 271. Assicurarsi che solo gli amministratori accreditati abbiano l’autorizzazione necessaria ad utilizzare questo comando.

Se i metodi normali per la cancellazione di un queue manager non funzionano, consultare: “Rimozione manuale dei queue manager” a pagina 333 per un’alternativa.

Utilizzo della funzione MQM (Message Queue Management)

La funzione MQM (Message Queue Management) di MQSeries for Compaq NSK, V5.1 viene eseguita come un requester PATHWAY SCOBOL sotto TCP (Terminal Control Process). Utilizza una classe di server MQM che richiama la API di linguaggio C.

Esiste un’istanza di MQM separata per ciascun queue manager configurato su un sistema, poiché ogni queue manager è controllato sotto la propria configurazione PATHWAY. Di conseguenza, MQM è limitata alla gestione del queue manager cui appartiene.

Nota: Per impostazione predefinita, un massimo di 10 utenti può utilizzare simultaneamente la funzione MQM. Per modificare questo limite in 20, ad esempio, immettere dal PATHWAY del queue manager:

```
ALTER TCP MQS-TCP-01, MAXTERMS 20
```

Per ulteriori informazioni, consultare il “Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries” a pagina 29.

Nota:

Per richiamare la MQM, immettere `run mqmc` dal prompt PATHCOM del queue manager.

Il Main Menu della MQM è il seguente:

Message Queue Management (MQM)

```
          IBM MQSeries for Compaq
NonStop Kernel Version 5.1

          ** Main Menu **

          Enter Choice:  _

          1. Queue Manager
          2. Queues
          3. Channels

F1 - Enter                               F16
- Return

(C) Copyright IBM Corp. 1993, 2001 All Rights Reserved.
```

Figura 4. Main Menu MQM di MQSeries for Compaq NSK

Dal Main Menu della MQM è possibile selezionare i seguenti menu secondari:

1. Queue Manager
2. Queues
3. Channels

Questi menu secondari sono descritti più avanti in questo capitolo. E' possibile ritornare al Main Menu della MQM in qualsiasi momento premendo Alt+F6. E' possibile ritornare al pannello precedente premendo il tasto per ritornare indietro di un pannello (F16). Quando selezionato dal Main Menu della MQM, F16 determina l'uscita dalla funzione MQM.

Utilizzo del Queue Manager Menu

Per selezionare l'opzione Queue Manager, immettere 1 nel campo Enter Choice del Main Menu della MQM e premere quindi il tasto di invio (F1). Viene visualizzato il pannello Queue Manager Menu:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1

** Queue Manager Menu **

Name          : MT01
Description   : _____

Command Level :      510 Trigger Interval : 999999999
Coded Char Set :      819 Platform      : NSK
Max Handles   :      256 Max Uncommitted Msg: 10000
Max Message   : 4194304 Max Priority    :      9
Dead Letter Queue Name :
SYSTEM.DEAD.LETTER.QUEUE _____
Command Input Queue Name :
SYSTEM.ADMIN.COMMAND.QUEUE _____
Default Xmit Queue Name :
_____

Authority Event Enabled Y/N? : N      Inhibit Event Enabled
Y/N?      : N
Local Event Enabled Y/N?    : N      Remote Event Enabled Y/N?
: N
Start/Stop Event Enabled Y/N?: N      Performance Event Enabled
Y/N?: N

FORCE Y/N? _

F1 - Modify   F2 - Trace   PGDN - Next Page
F16 - Return

```

Figura 5. Pannello Queue Manager Menu

Premere il tasto PGDN per visualizzare il secondo pannello di informazioni.

Message Queue Management (MQM)

```
          IBM MQSeries for Compaq
NonStop Kernel Version 5.1

                ** Queue Manager Menu **

Queue Manager Id      :
ROBERT_2001-01-15_10.28.40_____

Channel Auto Definition: N   Channel Auto Definition Events Enabled
Y/N?: N

Auto Definition Exit  : _____

Cluster Workload Data : _____
Cluster Workload Exit : _____

Cluster Workload Length:      100          Distribution List
Support: Y

Repository Name       :
_____

Repository Name List  :
_____

PGUP - Return
```

Figura 6. Pannello 2 di Queue Manager Menu

E' possibile utilizzare questo pannello per:

- Modificare alcuni attributi del queue manager
- Controllare la traccia degli oggetti MQSeries

Modifica degli attributi di Queue Manager

Sovrascrivere i valori che si desidera modificare nel pannello Queue Manager Menu e premere il tasto di modifica (F1). L'utente non potrà sovrascrivere i valori che non possono essere modificati.

Traccia degli oggetti di MQSeries

Premere il tasto per la traccia (F2) per visualizzare il QUEUE MANAGER TRACE MENU:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1

      ** QUEUE MANAGER TRACE MENU **

_ API      : MQI.
_ COMMS    : Communications networks processing flow.
_ CSFLOWS  : Common services processing flow.
_ LQMFLOWS : Local queue manager processing flow.
_ REMOTEFLOWS : Communications component processing flow.
_ ADMINFLOW : Administrative processing flow.
_ OTHERFLOWS : Other components processing flow.
_ CSDATA   : Common services data buffers.
_ LQMDATA  : Local queue manager internal data buffers.
_ REMOTEDATA : Communications component internal data buffers.
_ ADMINDATA : Administrative internal data buffers.
_ OTHERDATA : Other components internal data buffers.
_ VERSIONDATA : Output version of MQSeries running.
_ COMMENTARY : Output program comments in the MQSeries
components.

_ All      : Select all options.

F1-Start Trace  F2-Stop Trace
F16-Return

```

Figura 7. QUEUE MANAGER TRACE MENU

Sono disponibili le seguenti opzioni di traccia:

- API** Dati di output per i punti di traccia associati ai componenti MQI e al principale Queue Manager.
- COMMS** Dati di output per i punti di traccia associati ai dati trasmessi sulle reti di comunicazione.
- CSFLOWS** Dati di output per i punti di traccia associati al flusso di elaborazione nei servizi comuni.
- LQMFLOWS** Dati di output per i punti di traccia associati al flusso di elaborazione nel Queue Manager locale.
- REMOETFLOWS** Dati di output per i punti di traccia associati al flusso di elaborazione nei componenti di comunicazione.
- ADMINFLOW** Dati di output per i punti di traccia associati ai buffer di dati interni di gestione,
- OTHERFLOWS** I dati di output per i punti di traccia associati ad altri flussi di configurazione dei componenti.
- CSDATA** Dati di output per i punti di traccia associati ai buffer di dati interni nei servizi in comune.
- LQMDATA** Dati di output per i punti di traccia associati ai buffer di dati interni nel Queue Manager locale.
- REMOATEDATA** Dati di output per i punti di traccia associati ai buffer di dati interni nei componenti di comunicazione.

Message Queue Management (MQM)

ADMINDATA

Dati di output per i punti di traccia associati ai buffer di dati interni nei componenti di comunicazione.

OTHERDATA Dati di output per i punti di traccia associati agli altri buffer di dati interni delle applicazioni.

VERSIONDATA

I dati di output per i punti di traccia associati alla versione di MQSeries in esecuzione.

COMMENTARY

Dati di output per i punti di traccia associati ai commenti nei componenti MQSeries.

ALL I punti di traccia sono abilitati e viene generata una traccia completa.

Immettere un carattere sui nomi dei componenti per cui si desidera avviare (o arrestare) la traccia.

Per avviare la traccia per i componenti selezionati, premere il tasto di avvio traccia (F1). Per arrestare la traccia per i componenti selezionati, premere il tasto di arresto traccia (F2).

Utilizzo del menu Queues

Per selezionare l'opzione Queues, immettere 2 nel campo Enter Choice nel Main Menu della MQM e premere quindi il tasto di invio (F1). Viene visualizzato il pannello Search Criteria:

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1

** Search Criteria **

Queue Name: _____
Enter a queue name or part of one:

Queue Type: _
choose one or leave blank:      1. Local
                                2. Model
                                3. Remote
                                4. Alias

F1 - Enter                      F16 -
Return
```

Figura 8. Pannello Search Criteria (queue)

Nel campo Queue Name del pannello Search Criteria, immettere un nome di coda parziale o completo. E' anche possibile fornire un ID di Queue Type se si desidera limitare la ricerca alle code di un solo tipo. Premere il tasto di invio (F1). Viene

Message Queue Management (MQM)

visualizzato il Queue Menu, che è possibile utilizzare per visualizzare, modificare, creare, copiare, cancellare e controllare le code di MQSeries.

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Queue Menu **

Queue Name                                Type
--
ANNE.ET01.RQSD.LOCAL                      QLOCAL
ANNE.ET01.RQSD.REMOTE                     QREMOTE
ANNE.ET01.RQSV.LOCAL                      QLOCAL
ANNE.ET01.RQSV.REMOTE                     QREMOTE
ANNE.ET01.SDRCL.LOCAL                     QLOCAL
ANNE.ET01.SDRCL.REMOTE                    QREMOTE
ANNE.M401.RQSD.LOCAL                      QLOCAL
ANNE.M401.RQSD.REMOTE                     QREMOTE
ANNE.M401.RQSV.LOCAL                      QLOCAL
ANNE.M401.RQSV.REMOTE                     QREMOTE
ANNE.M401.SDRCL.LOCAL                     QLOCAL
ANNE.M401.SDRCL.REMOTE                    QREMOTE

F1 - Enter/Display/Modify      F2 - Create   F3 - Copy     F4
- Delete                        PGDN         PGUP          F16
F5 - Monitor                    - Return
```

Figura 9. Queue Menu

Nota: E' possibile creare, modificare e cancellare code solo sul queue manager associato al requester MQM che si sta utilizzando.

Utilizzare i tasti per scorrere le pagine verso l'alto e verso il basso per scorrere l'elenco di code.

Creazione di una coda

Dal Queue Menu, premere il pulsante per la creazione (F2) per visualizzare il pannello Create Queue:

Message Queue Management (MQM)

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Create Queue **

Queue Type: _ 1=Local, 2=Model, 3=Remote, 4=Alias,

Name:
_____

Replace [Y/N]: _

F1 - Enter
- Return

F16
```

Figura 10. Pannello Create Queue

Per creare una nuova definizione di coda:

1. Immettere 1 (per una coda locale), 2 (per una coda modello), 3 (per una coda remota) o 4 (per una coda alias) nel campo Queue Type.
2. Immettere il nome del queue manager nel campo Name.
3. Se la coda deve sostituire una coda esistente dello stesso nome e tipo, immettere Y nel campo Replace.
4. Premere il tasto di invio (F1).

Se si crea una coda locale, viene visualizzato il pannello Create Local Queue:

```

                IBM MQSeries for Compaq
NonStop Kernel Version 5.1
                ** Create Local Queue **

Queue Name : TEST
Description:
-----
Default Msg Priority : 0          Put Enabled [Y/N]
: _
Default Persistence : _          Get Enabled [Y/N]
: _

Retention Interval   :          0      Queue Definition Type   :
-----
Max Queue Depth      :          0      Priority/FIFO
[P/F]                : _
Max Message Length   :          0      Share [Y/N]
: _
Backout Threshold    :          0      Usage [N/X]
: _
Backout Requeue Name :

-----
Init. Queue          :

-----
Process Name         :

-----
Trigger Type [N/E/F/D]: _          Trigger/NoTrigger
[Y/N]                : _
Trigger Depth        :          0      Trigger Priority       : 0
Trig. Data           :

-----
Q Depth Max Event    : _          Q Serv. Int.
Event[H/O/N]: _
Q Depth High Limit   :          0      Q Depth High Event    : _
Q Depth Low Limit    :          0      Q Depth Low Event     : _
Q Service Interval   :          0      Scope                 :

-----
F1 - Enter          PGDN - Next Page
F16 - Return
    
```

Figura 11. Pannello Create Local Queue

Premere il tasto PGDN per visualizzare il secondo pannello di informazioni.

Message Queue Management (MQM)

```
          IBM MQSeries for Compaq
NonStop Kernel Version 5.1
          ** Create Local Queue **

Cluster Name      :
_____
Cluster Name List :
_____

Distribution List Y/N : _      Default Binding [0/N]   : _

PGUP - Return
```

Figura 12. Create Local Queue - pannello 2

Completare il pannello e premere Invio (F1).

Se si crea una coda remota, viene visualizzato il pannello Create Remote Queue:

```

      IBM MQSeries for Compaq
NonStop Kernel Version 5.1
      ** Create Remote Queue **

Queue Name : TEST_REMOTE
Description:
-----
Default Msg Priority : 0          Put Enabled (Y/N) : _
Default Persistence : _          Default Binding [O/N] : _

Scope :
Remote Queue Name :
-----
Remote Queue Manager :
-----
Transmit Queue Name :
-----
Cluster Name :
-----
Cluster Name List :
-----

F1 - Enter
- Return

F16
    
```

Figura 13. Pannello Create Remote Queue

Completare il pannello e premere Invio (F1).

Copia di una coda

Dal Queue Menu, premere il pulsante per la copia (F3) per definire una nuova coda copiando una definizione esistente. Viene visualizzato il pannello Copy Queue:

Message Queue Management (MQM)

```
          IBM MQSeries for Compaq
NonStop Kernel Version 5.1
          ** Copy Queue **

          Name:
ANNE.ET01.RQSD.LOCAL.2_____

          Replace [Y/N]: _

          Like Queue:
ANNE.ET01.RQSD.LOCAL_____

          Queue Type: QLOCAL__

F1 - Enter                               F16
- Return
```

Figura 14. Pannello Copy Queue

Immettere il nome della nuova definizione di canale nel campo Name; immettere Y nel campo Replace se la nuova coda deve sostituire una coda esistente con lo stesso nome e tipo; immettere il nome della definizione di cui si sta eseguendo la copia nel campo Like Queue; immettere il tipo di coda nel campo Queue Type. Premere il tasto di invio (F1).

Modifica di una coda

Dal Queue Menu, premere il pulsante per la modifica (F1) per visualizzare il pannello Display/Modify Local Queue:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Display/Modify Local Queue **
Queue Name : ANNE.ET01.RQSD.LOCAL
Description: Local queue ET01
receiver
-----
Default Msg Priority : 0          Put Enabled [Y/N]
: Y
Default Persistence : N          Get Enabled [Y/N]
: Y

Retention Interval : 999999999   Queue Definition Type :
PREDEFINED
Max Queue Depth : 5000          Priority/FIFO
[P/F] : P
Max Message Length : 1024       Share [Y/N]
: Y
Backout Threshold : 0           Usage [N/X]
: N
Backout Requeue Name :

-----
Init. Queue :

-----
Process Name :

-----
Trigger Type [N/E/F/D]: F       Trigger/NoTrigger
[Y/N] : N
Trigger Depth : 1              Trigger Priority : 0
Trig. Data :

-----
Q Depth Max Event : Y          Q Serv. Int.
Event[H/O/N]: N
Q Depth High Limit : 80        Q Depth High Event : N
Q Depth Low Limit : 20         Q Depth Low Event : N
Q Service Interval : 999999999 Scope :

QMGR
F1 - Modify PGDN - Next Page
F16 - Return

```

Figura 15. Pannello Display/Modify Local Queue

Sovrascrivere i valori che si desidera modificare e premere il pulsante per la modifica (F1). L'utente non potrà sovrascrivere i valori che non possono essere modificati.

Cancellazione di una coda

Nel Queue Menu, immettere un carattere sul nome della coda che si desidera cancellare. Premere il pulsante per la cancellazione (F4) e premere nuovamente F4 per confermare la cancellazione.

Controllo di una coda

Premere il pulsante per il controllo (F5) dal Queue Menu per visualizzare il pannello Monitor Local Queues:

Message Queue Management (MQM)

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Monitor Local Queues **
Queue          OPEN INPUT  OPEN OUTPUT
DEPTH
=====
ANNE_M401_RQSD_LOCAL
ANNE_M401_RQSV_LOCAL
ANNE_M401_SDRD_LOCAL
ANNE_MA02_RQSD_LOCAL
ANNE_MA02_RQSV_LOCAL
ANNE_MA02_SDRD_LOCAL
ANNE_MD01_RQSD_LOCAL
ANNE_MD01_RQSV_LOCAL
ANNE_MD01_SDRD_LOCAL
ANNE_MD01_SVRC_LOCAL
ANNE_ME02_RQSD_LOCAL
ANNE_ME02_RQSV_LOCAL
ANNE_ME02_SDRD_LOCAL 10
ANNE_ME02_SVRC_LOCAL

F12 - Refresh          PGDN          PGUP          F16
- Return
```

Figura 16. Pannello Monitor Local Queues

In quest'esempio, le code sono aperte nè per l'input nè per l'output. Una coda, ANNE_ME02_SDRD_LOCAL, contiene 10 messaggi.

Il parametro MQMQMREFRESHINT PATHWAY per MQS-MQMSVR00 determina la frequenza con cui gli schermi del monitor relativi a canali e code sono aggiornati. La frequenza predefinita è ogni 30 secondi. Per impostare la frequenza su ogni 10 secondi, ad esempio, modificarla nel PATHWAY relativo al proprio Queue Manager:

```
ALTER SERVER MQS-MQMSVR00, PARAM MQMQMREFRESHINT 10
```

Utilizzo del menu Channels

Per selezionare l'opzione Channels, immettere 3 nel campo Enter Choice nel Main Menu della MQM e premere quindi il pulsante di invio (F1). Viene visualizzato il pannello Channel di Search Criteria:

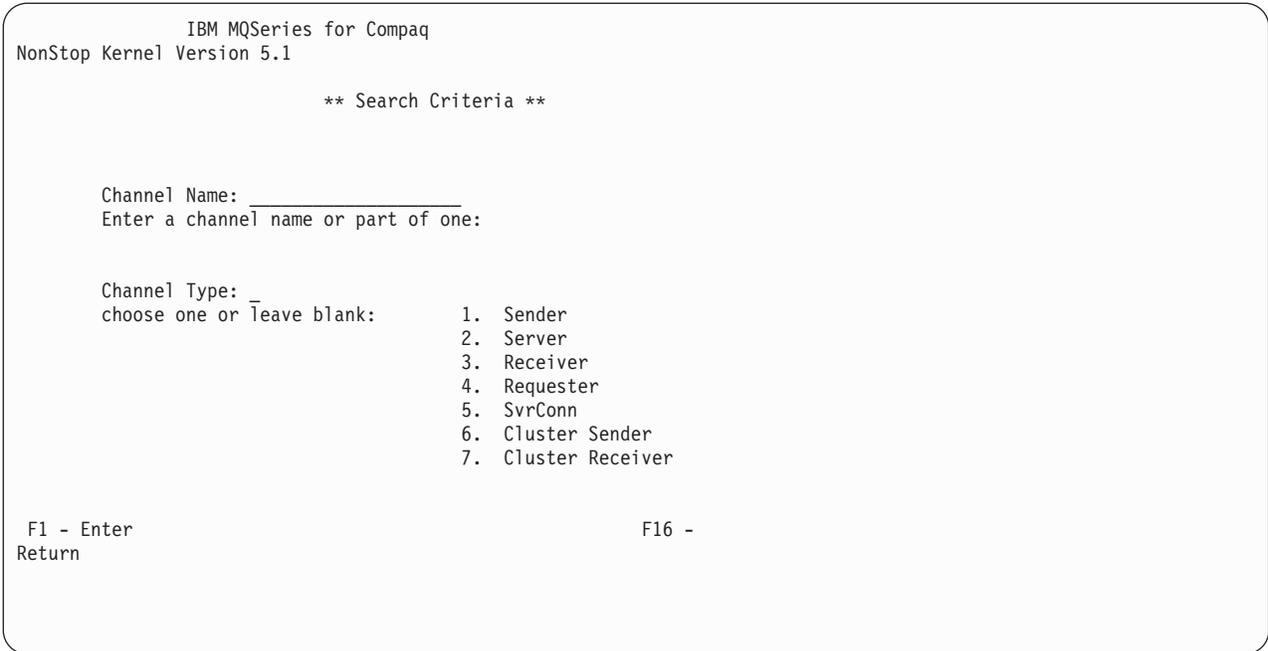


Figura 17. Pannello Search Criteria (channel)

Nel campo Channel Name, immettere un nome di canale parziale o completo. Nel campo Channel Type, è possibile immettere un numero compreso tra 1 e 5 per identificare il tipo di canale cui si è interessati. Premere il pulsante di invio (F1) per visualizzare il Channel Menu:

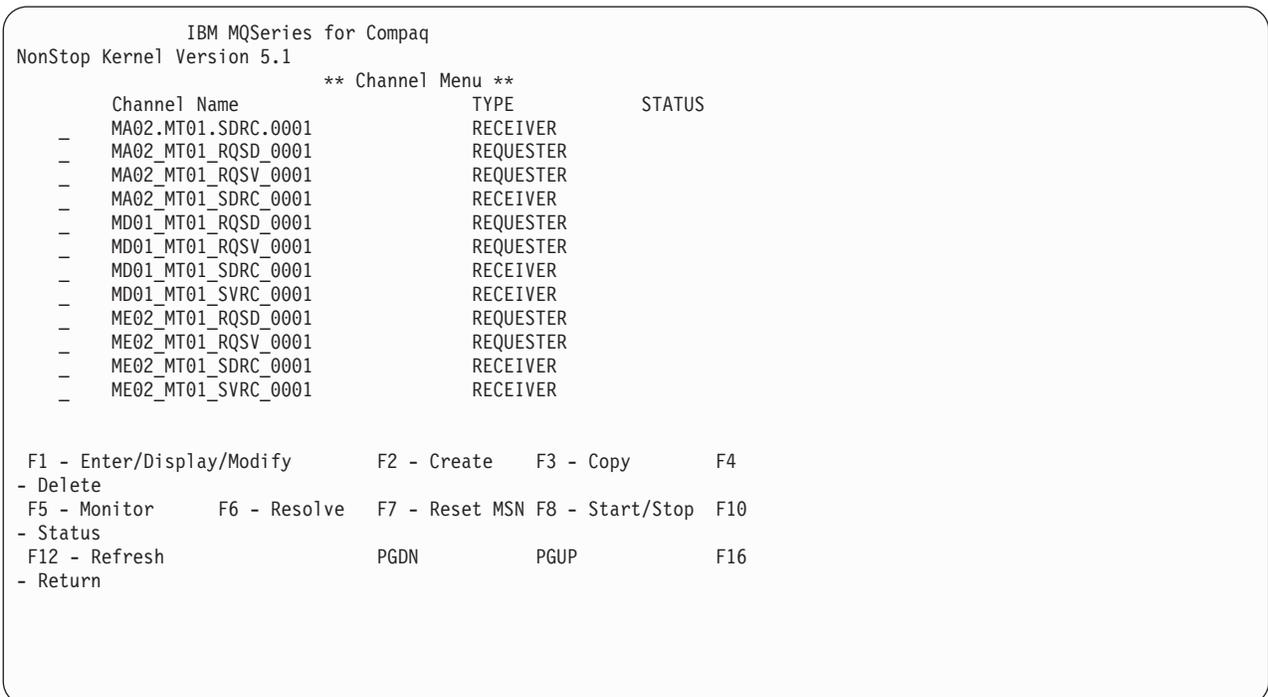


Figura 18. Channel Menu

Message Queue Management (MQM)

Il Channel Menu visualizza un elenco dei canali che corrispondono ai criteri di ricerca. Dal Channel Menu è possibile:

- Visualizzare e modificare lo stato di un canale.
- Creare una nuova definizione di canale.
- Copiare una definizione di canale.
- Cancellare una definizione di canale.
- Controllare lo stato di un canale.
- Risolvere un canale.
- Reimpostare un MSN (message sequence number).
- Avviare o arrestare un canale.

Modifica di un canale

Nel Channel Menu, immettere un carattere sul canale che si desidera modificare e premere il pulsante di invio/visualizzazione/modifica (F1). Viene visualizzato il corretto pannello. Ad esempio, se si seleziona un canale di invio, viene visualizzato il pannello Display/Modify Sender Channel:

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Display/Modify Sender Channel **
Channel Name      : MT01.M401.SDRC.0001_
Description       : Sender to M401_____
Xmit Queue Name   : _____
M401.TQ.SDRC.0001_____
Data Conversion Y/N: N   NonPersistent Msg Speed [FAST/NORMAL]:
FAST_
User Id           : _____ Password           :
MCA Name          : _____ MCA UserID         :
Batch Size        : _____ 50 Max Message Size :
4194304
MSN Wrap Count    : _____ 9999999 Disconnect Interval:
60
Short Retry Count : _____ 10 Short Timer       :
60
Long Retry Count  : _____ 9999999 Long Timer    :
1200
Heartbeat Interval : _____ 300 Batch Interval  :
0
Transport Protocol : 1 (1=Lu6.2/ 2=TCP/IP) TCP/IP Port Number :
TCP/IP Address    : _____
TCP/IP/SNA Process : $ZTC1_____
Local LU Name     : IYAHT080 Remote LU Name      :
IYAFT110
Local TP Name     : INTCRS6A_____ Mode Name      :
LU62PS_
Remote TP Name    : _____
F1 - Modify      PGDN - Exits
- Return
```

Figura 19. Pannello Display/Modify Sender Channel (1)

Premere il tasto PGDN per visualizzare il secondo pannello di informazioni:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Display/Modify Sender Channel **

Scrty Data: _____
Scrty Exit: _____

NOTE: RUNMQSC must be used to update the Send, Receive and Message
Data and Exit attributes of channels

Return

PGUP -
    
```

Figura 20. Pannello Display/Modify Sender Channel (2)

Sovrascrivere i valori che si desidera modificare e premere il pulsante per la modifica (F1). L'utente non potrà sovrascrivere i valori che non possono essere modificati.

Creazione di una definizione di canale

Dal Channel Menu, premere il pulsante di creazione (F2) per visualizzare il pannello Create Channel:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Create Channel **

Channel Type: 1 1=Sender, 2 = Server, 3=Receiver,
               4=Requester, 5 = Server Connection
               6 = Cluster Sender, 7 = Cluster Receiver

Name: _____

Replace [Y/N]: _

F1 - Enter
- Return

F16
    
```

Figura 21. Pannello Create Channel

Per creare una nuova definizione di canale:

Message Queue Management (MQM)

1. Immettere 1 (per un canale di invio), 2 (per un canale server), 3 (per un canale di ricezione), 4 (per un canale richiedente) o 5 (per una connessione server) nel campo Channel Type.
2. Immettere il nome della definizione di canale nel campo Name.
3. Premere il tasto di invio (F1).
4. Immettere Y nel campo Replace se la definizione deve sostituire una definizione esistente con lo stesso nome e tipo.

Se si immette un 1 nel campo Channel Type, viene visualizzato il pannello Create Sender Channel:

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Create Sender Channel **
Channel Name      : Compaq_TO_SOLARIS__
Description      : _____
Xmit Queue Name  : _____

Data Conversion Y/N: NonPersistent Msg Speed [FAST/NORMAL]:
FAST__
User Id          : _____ Password          :
MCA Name         : _____ MCA UserID        :
Batch Size      : 50 Max Message Size :
4194304
MSN Wrap Count  : 999999999 Disconnect Interval:
6000
Short Retry Count : 999999 Short Timer      :
60
Long Retry Count : 999999 Long Timer       :
1200
Heartbeat Interval : 300 Batch Interval   :
0
Transport Protocol : _ (1=Lu6.2/ 2=TCP/IP) TCP/IP Port Number :
TCP/IP Address    : _____
TCP/IP/SNA Process : $ZTC1_____
Local LU Name     : _____ Remote LU Name  :
Local TP Name     : _____ Mode Name      :
Remote TP Name    : _____

F1 - Enter      PGDN - Exits      F16
- Return
```

Figura 22. Pannello Create Sender Channel

Se si immette un 3 nel campo Channel Type, viene visualizzato il pannello Create Receiver Channel:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Create Receiver Channel **

Channel Name      : SOLARIS_TO_Cmpaq____
Description       : _____
                  : _____

Put Authority D/C : _ NonPersistent Msg Speed [FAST/NORMAL]:
FAST_
User Id          : _____ MCA UserID      :
Batch Size       :          50      Max Message Size :
4194304
Msg Retry Count  :          10      Msg Retry Interval :
1000
Heartbeat Interval :          300    MSN Wrap Count      :
999999999

Transport Protocol : _ (1=Lu6.2/ 2=TCP/IP)

F1 - Enter      PGDN - Exits
- Return
F16
    
```

Figura 23. Pannello Create Receiver Channel

Se si immette un 5 nel campo Channel Type, viene visualizzato il pannello Create Server Connection Channel:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Create Server Connection Channel **

Channel Name      : WINDOWS_CLIENT_____
Description       : _____
                  : _____

MCA UserID        : _____
Max Message Size  :          4194304
Heartbeat Interval :          300

Transport Protocol : _ (1=Lu6.2/ 2=TCP/IP)

F1 - Enter      PGDN - Exits
- Return
F16
    
```

Figura 24. Pannello Create Server Connection Channel

Message Queue Management (MQM)

Per creare una nuova definizione di canale, completare il pannello richiesto e premere il pulsante di invio (F1).

Se si immette un 6 nel campo Channel Type, viene visualizzato il pannello Create Cluster Sender Channel:

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Create Cluster Sender Channel **
Channel Name      : CLUSTER_SENDER_____
Description      : _____
Cluster Name     : _____
Cluster Name List : _____
Data Conversion Y/N: _ NonPersistent Msg Speed [FAST/NORMAL]:
FAST_
User Id          : _____ Password          :
MCA Name         : _____ MCA UserID        :
Batch Size      :          50 Max Message Size :
4194304
MSN Wrap Count  :          99999999 Disconnect Interval:
6000
Short Retry Count :          999999 Short Timer      :
60
Long Retry Count :          999999 Long Timer       :
1200
Heartbeat Interval :          300 Batch Interval    :
0
Transport Protocol : _ (1=Lu6.2/ 2=TCP/IP) TCP/IP Port Number :
TCP/IP Address   :
TCP/IP/SNA Process : _____
Local LU Name    : _____ Remote LU Name    :
Local TP Name    : _____ Mode Name         :
Remote TP Name   : _____
F1 - Enter      PGDN - Exits
- Return
```

Figura 25. Pannello Create Cluster Sender Channel

Se si immette un 7 nel campo Channel Type, viene visualizzato il pannello Create Cluster Receiver Channel:

```

IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Create Cluster Receiver Channel **
Channel Name      : CLUSTER_RECEIVER____
Description       : _____ Put
Authority D/C: _  _____ Network
Priority : _
Cluster Name      : _____
Cluster Name List : _____
Data Conversion Y/N: _ NonPersistent Msg Speed [FAST/NORMAL]:
FAST_
User Id           : _____ Password           :
MCA Name          : _____ MCA UserID         :
Batch Size        : _____ 50 Max Message Size :
4194304
MSN Wrap Count    : _____ 999999999 Disconnect Interval:
6000
Short Retry Count : _____ 999999 Short Timer      :
60
Long Retry Count  : _____ 999999 Long Timer       :
1200
Heartbeat Interval : _____ 300 Batch Interval    :
0
Msg Retry Count   : _____ 0 Msg Retry Interval  :
0
Transport Protocol : _ (1=Lu6.2/ 2=TCP/IP) TCP/IP Port Number :
TCP/IP Address    : _____
TCP/IP/SNA Process : _____ Local LU Name      :
Local TP Name     : _____ Remote LU Name     :
Remote TP Name    : _____ Mode Name         :
F1 - Enter      PGDN - Exits
- Return
F16

```

Figura 26. Pannello Create Cluster Receiver Channel

Controllo di un canale

Premere il pulsante per il controllo (F5) dal pannello Channel Menu per visualizzare il pannello Monitor Channels:

Message Queue Management (MQM)

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
** Monitor Channels **
Channel Name      Status   Curr MSN  Last MSN  MCA Status
Stop
=====
MT01.MH01.SDRC.0002  BINDING
NO
MT01.VM03.SDRC.0002  RUNNING      6266      6266  RUNNING
NO

F12 - Refresh          PGDN          PGUP          F16
- Return

Refreshing.....
```

Figura 27. Pannello Monitor Channels

Il parametro MQMQMREFRESHINT PATHWAY per MQS-MQMSVR00 determina la frequenza con cui gli schermi del monitor relativi a canali e code sono aggiornati. La frequenza predefinita è ogni 30 secondi. Per impostare la frequenza su ogni 10 secondi, ad esempio, modificarla nel PATHWAY relativo al proprio Queue Manager:

```
ALTER SERVER MQS-MQMSVR00, PARAM MQMQMREFRESHINT 10
```

Cancellazione di un canale

Nel Channel Menu, selezionare un canale da cancellare immettendo un carattere sul nome di canale. Premere il pulsante per la cancellazione (F4) per cancellare il canale e premere nuovamente F4 per confermare la richiesta di cancellazione.

Visualizzazione dello stato di un canale

Premere il pulsante per lo stato (F10) dal pannello Channel Menu per visualizzare il pannello Channel Status:

```

                IBM MQSeries for Compaq
NonStop Kernel Version 5.1
                Channel Status

Channel Name   : MT01.VM03.SDR0.0002_
Xmit Queue Name: VM03NCM.TQ.SDR0.0001_____
Connection Name:
$BP01.IYAHT080.IYCVM03_____

Channel Status      : RUNNING___      In Doubt          :
NO_
Start Date         : 2001-02-03      Start Time        :
15.07.14
Last Msg Date      : 2001-02-03      Last Msg Time     :
16.34.04

MCA Job Name       : 000069AA_____
Current LUW ID     : 03544240E28B0277
Last LUW ID        : 03544240E28B0277 Current Messages  :
0
MCA Status         : RUNNING___      Current Seq Num   :
6266
Stop Requested     : NO_              Last Seq Num      :
6266
Number of Batches  :          6       Number of Messages :
6
Number of Buffers Sent:          14   Number of Buffers Recvd:
7
Number of Bytes Sent :          3204   Number of Bytes Recvd :
196
Num of Long Retry Left:  9999999     Num of Short Retry Left:
10

F12 - Refresh                      F16
- Return
    
```

Figura 28. Pannello Channel Status

Avvio e arresto di un canale

Premere il pulsante per l'Avvio/Arresto (F8) dal Channel Menu per visualizzare il pannello Start/Stop Channel:

Message Queue Management (MQM)

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
      Start/Stop Channel

      Name: MT01_MA02_SDR0_0001_
Status:
Action: _  choose one of the following:
          1. Start Channel
          2. Stop Immediate
          3. Stop Quiesce

F1 - Enter
- Return                                F16
```

Figura 29. Pannello Start/Stop Channel

Immettere il nome del canale nel campo Name e immettere un numero compreso tra 1 e 3 nel campo Action. Premere il tasto di invio (F1).

Reimpostazione di un MSN (Message Sequence Number)

Dal Channel Menu, premere il pulsante per reimpostare l'MSN (F7) per visualizzare il pannello Reset Channel:

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1
Reset Channel

Name:
MT01_M401_RQSD_0001
Sequence Number:      1

F1 - Enter
- Return

F16
```

Figura 30. Pannello Reset Channel

L'MSN garantisce la non duplicazione dei messaggi ed assicura che i messaggi vengano memorizzati nello stesso ordine in cui sono stati trasmessi. Questo pannello consente, se necessario, di reimpostare il numero di sequenza di un canale.

Risoluzione di un canale

Dal Channel Menu, premere il pulsante per la risoluzione (F6) per visualizzare il pannello Resolve Channel.

Message Queue Management (MQM)

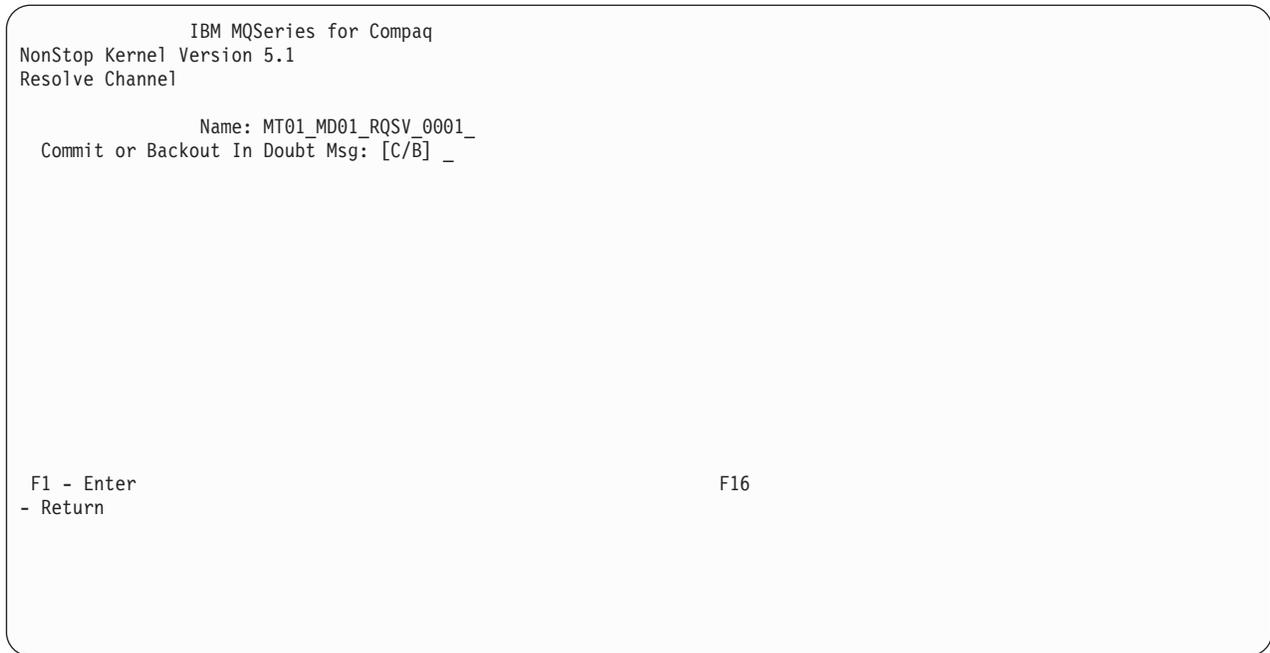


Figura 31. Pannello Resolve Channel

E' possibile:

- Eseguire il backout del batch di messaggi in uno stato dubbio (B)
- Eseguire il commit del batch di messaggi in uno stato dubbio (C)

Copia di un canale

Nel Channel Menu, premere il pulsante per la copia (F3). Viene visualizzato il pannello Copy Channel:

```
IBM MQSeries for Compaq
NonStop Kernel Version 5.1      ** Copy Channel **

Name: _____
Replace [Y/N]: _
Like Name: MT01_M401_RQSV_0001_
Channel Type: SERVER

F1 - Enter                      F16
- Return
```

Figura 32. Pannello Copy Channel

Immettere il nome del nuovo canale nel campo Name: immettere il nome della definizione di canale di cui si sta eseguendo la copia nel campo Like Name; immettere il tipo di canale nel campo Channel Type. Premere Invio (F1) per copiare la definizione di canale.

Capitolo 5. Gestione di oggetti MQSeries locali

Questo capitolo descrive come gestire gli oggetti locali di gestione MQSeries per supportare i programmi che utilizzano MQI (Message Queuing Interface). MQI consente ai programmi di accedere ai servizi di accodamento dei messaggi.

La gestione locale si verifica quando si creano, visualizzano, modificano, copiano e cancellano gli oggetti MQSeries.

Questo capitolo si compone delle seguenti sezioni:

- “Supporto dei programmi applicativi che utilizzano MQI”
- “Esecuzione delle attività di gestione locale utilizzando i comandi MQSC” a pagina 98
- “Esecuzione di comandi MQSC da file di testo” a pagina 102
- “Risoluzione dei problemi con MQSC” a pagina 105
- “Utilizzo delle code locali” a pagina 106
- “Utilizzo degli alias di coda” a pagina 115
- “Utilizzo delle code di esempio” a pagina 117
- “Gestione degli oggetti per triggering” a pagina 118

Supporto dei programmi applicativi che utilizzano MQI

I programmi MQI necessitano oggetti specifici prima di poter essere eseguiti regolarmente. Un'applicazione MQI può rimuovere i messaggi da una coda, elaborarli e inviare i risultati su un'altra coda dello stesso queue manager.

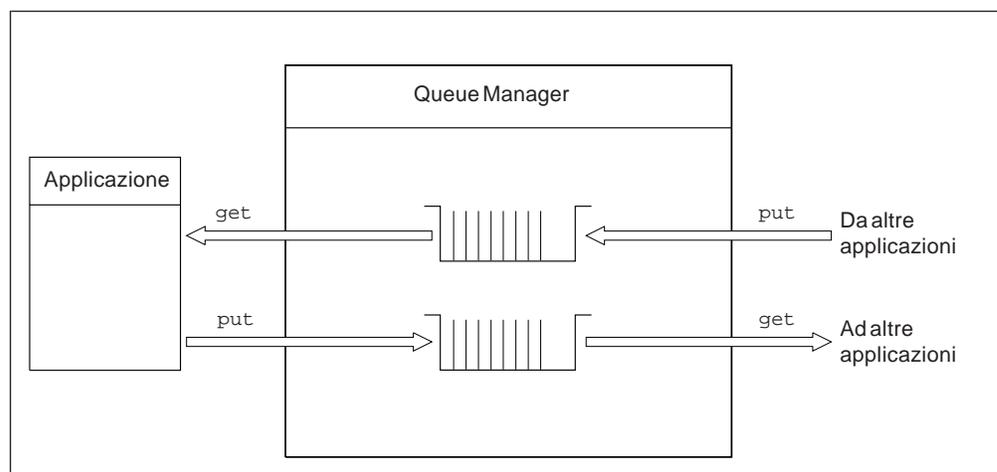


Figura 33. Code, messaggi e applicazioni

Le applicazioni possono inviare messaggi (utilizzando MQPUT) sia su code locali che remote, mentre possono riceverne solo da code locali (utilizzando MQGET).

Prima di poter eseguire questa applicazione, è necessario che le seguenti condizioni siano soddisfatte:

- Deve esistere un Queue Manager e deve essere in esecuzione.
- La prima coda dell'applicazione, quella da cui rimuovere i messaggi, deve essere definita.

Supporto delle applicazioni

- E' necessario inoltre definire la seconda coda in cui l'applicazione inserisce i messaggi (a meno che non sia una coda dinamica).
- L'applicazione deve potersi connettere al Queue Manager. Per eseguire questa attività, è necessario essere collegati al codice del prodotto. Consultare la sezione "Appendice I. Creazione ed esecuzione delle applicazioni" a pagina 349 per ulteriori informazioni.
- Le applicazioni che inviano i messaggi alla prima coda devono anche connettersi al Queue Manager. Se si tratta di code remote, occorre impostare anche le code di trasmissione e i canali.

Esecuzione delle attività di gestione locale utilizzando i comandi MQSC

I comandi MQSeries (MQSC) consentono di gestire gli oggetti MQSeries. E' possibile immettere i comandi utilizzando il comando **runmqsc** nel prompt dei comandi.

Per ulteriori informazioni sull'utilizzo di MQSC in ambiente MQSeries for Compaq NSK, consultare "Appendice G. MQSC supportato da MQSeries for Compaq NSK" a pagina 337.

E' possibile utilizzare i comandi script di MQSeries (MQSC) per gestire gli oggetti del queue manager, incluso lo stesso queue manager, i cluster, i canali, le code e i namelist e le definizioni di processo. Questa sezione si occupa di Queue Manager, di code e di definizioni di processo; per informazioni relative alla gestione di oggetti canale, consultare l'implementazione DQM nel manuale *MQSeries Intercommunication*.

È possibile immettere i comandi MQSC utilizzando il comando **runmqsc**, in modo interattivo, immettendo cioè i comandi dalla tastiera, oppure reindirizzando un input standard per eseguire una sequenza di comandi da un file di testo ASCII. In entrambi i casi, il formato dei comandi è lo stesso.

È possibile eseguire il comando **runmqsc** in tre modi, in funzione degli indicatori impostati nel comando:

- *Verification mode (Modo Verifica)*, in cui i comandi MQSC vengono verificati su un Queue Manager locale, ma non vengono eseguiti.
- *Direct mode (Modo Diretto)*, in cui i comandi MQSC vengono eseguiti su un Queue Manager locale.
- *Indirect mode (Modo Indiretto)*, in cui i comandi vengono eseguiti su un Queue Manager remoto.

Gli attributi degli oggetti specificati in MQSC vengono riportati in lettere maiuscole (ad esempio, RQMNAME), sebbene tale caratteristica non abbia rilevanza. I nomi degli attributi MQSC hanno una lunghezza massima di otto caratteri.

Prima di iniziare

Prima di iniziare, è necessario creare e riavviare il queue manager che esegue i comandi MQSC. Consultare la sezione "Creazione di un Queue Manager predefinito" a pagina 64 per ulteriori informazioni.

Nomi degli oggetti MQSeries

Negli esempi, vengono utilizzati dei nomi lunghi per gli oggetti. Ciò per semplificare l'identificazione degli oggetti che si utilizzano.

Immissione di comandi MQSC

Quando si immettono comandi MQSC, è necessario specificare solo il nome locale della coda. Negli esempi riportati, vengono utilizzati nomi quali `ORANGE.LOCAL.QUEUE`

La parte del nome `LOCAL.QUEUE` serve ad indicare che si tratta di una coda locale. *Non* è necessario inserirla nei nomi delle code locali.

Inoltre viene utilizzato il nome `saturn.queue.manager` come nome di un Queue Manager.

La parte del nome `queue.manager` serve ad indicare che l'oggetto in questione è un Queue Manager. *Non* è necessario inserirla nei nomi dei Queue Manager.

Non è obbligatorio l'utilizzo di questi nomi, ma se si ricorre ad altri nomi, è necessario modificare tutti i comandi degli esempi che si riferiscono a questi nomi.

Sensibilità al maiuscolo/minuscolo nei comandi MQSC

I comandi MQSC e i relativi attributi non sono sensibili al maiuscolo o minuscolo. I nomi di oggetto invece solo sensibili al maiuscolo e al minuscolo (`input-queue` e `INPUT-QUEUE` sono oggetti differenti). I nomi di oggetto nei comandi MQSC vengono convertiti in lettere maiuscole (quindi non c'è differenza tra `QUEUE` e `queue`) a meno che non vengano specificati tra virgolette. Se non si utilizzano virgolette, per i nomi di oggetto vengono utilizzati i caratteri maiuscoli. Per ulteriori informazioni, consultare *MQSeries Guida di riferimento per i comandi*.

Tuttavia, determinati argomenti del comando `runmqsc`, utilizzato per richiamare la funzione MQSC, sono sensibili al maiuscolo e minuscolo; consultare "Utilizzo dei comandi di controllo" a pagina 29.

Utilizzo interattivo della funzione MQSC

Per immettere i comandi in modo interattivo, aprire una sessione TACL e immettere:

```
runmqsc
```

In questo esempio, non è specificato un nome di queue manager, quindi i comandi MQSC vengono elaborati dal queue manager predefinito. E' possibile immettere qualsiasi comando MQSC. Ad esempio:

```
MQSC>DEFINE QLOCAL (ORANGE.LOCAL.QUEUE)
```

Per indicare che un comando si estende su più linee, devono essere utilizzati i caratteri di continuazione:

- Un segno meno (-) indica che il comando continua dall'inizio della linea seguente.
- Un segno più (+) indica che il comando continua dal primo carattere presente nella linea seguente.

L'input del comando termina con il carattere finale di una linea non vuota, purché non sia un carattere di continuazione.

Immissione di comandi MQSC

Il comando **runmqsc** inoltre supporta la cronologia standard Compaq NSK e risolve le funzioni del comando. Ad esempio:

- Digitando **history** o **h** si otterrà un elenco dei dieci comandi più recenti
- Digitando **!n**, dove **n** è il numero di comando, tale comando verrà eseguito nuovamente
- Digitando **h n** o **history n**, dove **n** è un numero, si otterrà un elenco degli **n** comandi più recenti
- Digitando **fc** sarà visualizzato l'ultimo comando immesso per la modifica. Digitando **fc n**, dove **n** è il numero del comando, sarà possibile modificare tale comando. Digitando **fc string**, dove **string** è la parte iniziale di un comando o l'intero comando precedentemente immesso, si otterrà l'ultima ripetizione di tale comando per la modifica. La sintassi segue lo standard NSK. Ad esempio, digitare **d** per eliminare un carattere, **i** per inserire un carattere e **r** per sostituire un carattere.

Feedback dei comandi MQSC

Quando si immettono i comandi MQSC, il queue manager fornisce i messaggi di errore e di conferma. Ad esempio:

```
AMQ8006: MQSeries queue created
.
.
.
AMQ8405: Syntax error detected at or near end of command segment below:-
```

Il primo messaggio conferma l'esito positivo della creazione di una coda. Il secondo messaggio indica la presenza di un errore di sintassi. Se non si è immesso il comando in modo corretto, appurarne la sintassi corretta consultando: *MQSeries Guida di riferimento per i comandi*.

Termine dell'input interattivo MQSC

Se si utilizza MQSC in modo interattivo, è possibile chiudere immettendo il carattere EOF CTRL+Y oppure digitando **exit** o **quit** o **end** e Invio.

Se si esegue il reindirizzamento dell'input da altre fonti, ad esempio un file di testo, MQSC terminerà quando viene raggiunta la fine del file.

Visualizzazione degli attributi Queue Manager

Per visualizzare gli attributi di Queue Manager specificati con il comando **runmqsc**, utilizzare il seguente comando MQSeries:

```
MQSC>DISPLAY QMGR ALL
```

Un tipico output viene visualizzato in Figura 34 a pagina 101:

```

1 : dis qmgr all
AMQ8408: Display Queue Manager details.
DESCR( )                                DEADQ(SYSTEM.DEAD.LETTER.QUEUE)
DEFXMITQ(MY.DEFAULT.XMIT.QUEUE)        CHADEXIT( )
CLWLEXIT( )                             CLWLDATA( )
REPOS( )                                 REPOSNL( )
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)   QMNAME(saturn.queue.manager)
CRDATE(2001-03-12)                      CRTIME(09.24.30)
ALTDATE(2001-03-12)                     ALTTIME(09.26.27)
QMID(SIMONW_2001-03-12_09.24.30)        TRIGINT(10000)
MAXHANDS(256)                           MAXUMSGS(10000)
AUTHOREV(ENABLED)                       INHIBTEV(ENABLED)
LOCALEV(ENABLED)                         REMOTEEV(ENABLED)
PERFMEV(ENABLED)                         STRSTPEV(ENABLED)
CHAD(DISABLED)                           CHADEV(ENABLED)
CLWLEN(100)                              MAXMSGL(100000000)
CCSID(819)                               MAXPRTY(9)
CMDLEVEL(510)                            PLATFORM(NSK)
SYNCPT                                    DISTL(YES)

```

Figura 34. Esempio output per QMGR ALL

Il parametro ALL del comando DISPLAY QMGR determina la visualizzazione di tutti gli attributi del Queue Manager. L'output indica il nome del Queue Manager (saturn.queue.manager), i nomi della coda messaggi non recapitati(SYSTEM.DEAD.LETTER.QUEUE) e la coda comandi (SYSTEM.ADMIN.COMMAND.QUEUE). Si osservi che, se non si specifica il nome della coda messaggi non recapitati nel comando **crtmqm**, è necessario modificare il Queue Manager in modo da associare la coda messaggi non recapitati al Queue Manager.

E' necessario confermare che la creazione di queste code è stata eseguita immettendo il comando:

```

DISPLAY QUEUE (*)

```

Utilizzo di un Queue Manager diverso dal predefinito

E' possibile specificare il nome del Queue Manager quando si esegue il comando **runmqsc** in modo da eseguire i comandi MQSC su un Queue Manager locale diverso da quello predefinito. Ad esempio, per eseguire i comandi MQSC sul Queue Manager denominato `jupiter.queue.manager`, utilizzare il comando:

```

runmqsc jupiter.queue.manager

```

Tutti i comandi MQSC immessi verranno elaborati da questo Queue Manager, purché il Queue Manager sia sullo stesso nodo e sia in esecuzione.

È inoltre possibile eseguire comandi MQSC su un Queue Manager remoto; consultare la sezione "Emissione remota di comandi MQSC" a pagina 131.

Modifica degli attributi di Queue Manager

Per modificare gli attributi del Queue Manager specificati con il comando **runmqsc**, utilizzare il comando MQSC ALTER QMGR, specificando gli attributi e i

Immissione di comandi MQSC

valori che si desidera modificare. Ad esempio, utilizzare i seguenti comandi per modificare gli attributi del Queue Manager `jupiter.queue.manager`:

```
runmqsc jupiter.queue.manager  
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

Il comando `ALTER QMGR` modifica la coda messaggi non recapitati utilizzata e abilita eventi inibitori.

Esecuzione di comandi MQSC da file di testo

L'esecuzione dei comandi MQSC in modo interattivo è appropriato per i controlli brevi; tuttavia, se è necessario immettere comandi molto lunghi o sequenze di comandi, è opportuno richiamare l'input da un file di testo.

Per eseguire questa attività, creare un file di testo contenente i comandi MQSC utilizzando l'editor di testo. Quando si utilizza il comando `runmqsc`, utilizzare gli operatori di reindirizzamento `TACL IN` e `OUT` oppure gli indicatori `-i` e `-o` su `runmqsc`. Ad esempio, il seguente comandi esegue una sequenza di comandi contenuta nel file di testo `mymqscin`:

```
runmqsc /IN mymqscin/
```

oppure

```
runmqsc -i mymqscin
```

In modo analogo, è possibile reindirizzare l'output su un file. Un file contenente i comandi MQSC per l'input viene denominato *file di comandi MQSC*. Il file di output che contiene le risposte del Queue Manager viene definito *file di notifica*.

Per reindirizzare l'input e l'output sul comando `runmqsc`, utilizzare il comando:

```
runmqsc /IN mymqscin, OUT mymqscou/
```

oppure

```
runmqsc -i mymqscin -o mymqscou
```

Questo comando richiama i comandi MQSC contenuti nel file `mymqscin`. Poiché non è specificato alcun nome di Queue manager, i comandi MQSC vengono eseguiti sul Queue Manager predefinito. L'output viene inviato al file di notifica `mymqscou`. Figura 35 a pagina 103 illustra un estratto del file di comandi MQSC `mymqscin` e Figura 36 a pagina 104 illustra l'estratto corrispondente dell'output in `mymqscou`.

Esecuzione di comandi MQSC

Per reindirizzare l'input e l'output nel comando `runmqsc` per un Queue Manager (saturn.queue.manager) non predefinito, utilizzare il comando:

```
runmqsc /IN mymqscin, OUT mymqscou/ saturn.queue.manager
```

oppure

```
runmqsc -i mymqscin -o mymqscou saturn.queue.manager
```

Utilizzo dei file di comandi MQSC

I file di comandi MQSC sono scritti come file EDIT (Compaq file type code 101). Figura 35 è un estratto di un file di comandi MQSC che illustra un comando MQSeries (DEFINE QLOCAL) con i relativi attributi. Il manuale *MQSeries Guida di riferimento per i comandi* contiene una descrizione di ciascun comando MQSC e della relativa sintassi.

```
.  
. .  
. .  
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +  
  DESCR(' ') +  
  PUT(ENABLED) +  
  DEFPRTY(0) +  
  DEFPSIST(NO) +  
  GET(ENABLED) +  
  MAXDEPTH(5000) +  
  MAXMSGL(1024) +  
  DEFSOPT(SHARED) +  
  NOHARDENBO +  
  USAGE(NORMAL) +  
  NOTRIGGER  
. . .
```

Figura 35. Estratto del file di comandi MQSC, mymqscin

E' necessario limitare il numero massimo di righe a 72 caratteri. Il segno più (+) indica che il comando continua sulla riga successiva. Il segno più deve essere preceduto da uno spazio.

Utilizzo dei prospetti MQSC

Il comando `runmqsc` restituisce un *prospetto* che viene inviata al flusso OUT corrente. Il prospetto contiene:

- Un'intestazione che identifica MQSC come origine del prospetto:

```
Starting MQSeries Commands.
```

- Un elenco numerato dei comandi MQSC immessi (opzionale). Per impostazione predefinita, il testo dell'input viene riportato all'output. Nell'output, ogni

Esecuzione di comandi MQSC

comando è ordinato in base ad una sequenza numerica, così come mostrato nella Figura 36. Tuttavia è possibile utilizzare l'indicatore `-e` nel comando `runmqsc` per eliminare l'output.

- Un messaggio di errore di sintassi per ogni comando in cui viene rilevato un errore.
- Un *messaggio per l'operatore* che indica il risultato dell'esecuzione di ogni comando. Ad esempio il messaggio per l'operatore in cui viene indicato il completamento con esito positivo di un comando `DEFINE QLOCAL` è:

```
AMQ8006: MQSeries queue created.
```

- Altri messaggi dovuti ad errori durante l'esecuzione del file di scrittura.
- Una chiara sintesi statistica della notifica che indica il numero di comandi letti, il numero di comandi con errori di sintassi e il numero di comandi che non è stato possibile elaborare.

Nota: il Queue Manager inizia l'elaborazione dei soli comandi che non presentano errori di sintassi.

```
Starting MQSeries Commands.
.
.
12:   DEFINE QLOCAL('RED.LOCAL.QUEUE') REPLACE +
:     DESCR(' ') +
:     PUT(ENABLED) +
:     DEFPRTY(0) +
:     DEFPSIST(NO) +
:     GET(ENABLED) +
:     MAXDEPTH(5000) +
:     MAXMSGL(1024) +
:     DEFSOFT(SHARED) +
:     USAGE(NORMAL) +
:     NOTRIGGER
AMQ8006: MQSeries queue created.
:
.
.
15 MQSC commands read.
0 commands have a syntax error.
0 commands cannot be processed.
```

Figura 36. Estratto del file di notifica MQSC, `mymqscou`

Esecuzione dei file di comandi MQSC forniti

Quando si installa MQSeries for Compaq NSK, viene fornito un file MQSC denominato `AMQSCOS0`. Il file contiene le definizioni degli oggetti utilizzati dai programmi di esempio. Il file è ubicato nel volume secondario modello, per impostazione predefinita `$SYSTEM.ZMQSSMPL`.

Utilizzo di `runmqsc` per la verifica dei comandi

È possibile utilizzare il comando `runmqsc` per verificare i comandi MQSC su un Queue Manager locale predefinito senza doverli necessariamente eseguire. Per

effettuare questa operazione, impostare l'indicatore `-v` sul comando `runmqsc`. Ad esempio:

```
runmqsc -i mymqscin -o mymqscou -v
```

Quando si richiama `runmqsc` con un file di comandi MQSC, il Queue Manager verifica ogni comando e restituisce una notifica senza eseguire i comandi MQSC. Questa operazione consente di controllare la sintassi di tutti i comandi del file di comandi. Questa operazione è importante nelle seguenti situazioni:

- Esecuzione di un numero elevato di comandi da una riga comandi.
- Utilizzo ripetuto di un file di comandi MQSC.

Questa notifica è simile a quella riportata nella Figura 36 a pagina 104.

Non è possibile utilizzare questo metodo per verificare i comandi MQSC da una postazione remota. Ad esempio, se si esegue questo comando:

```
runmqsc -i mymqscin -o mymqscou -w 30 -v jupiter.queue.manager
```

L'indicatore `-w`, che viene utilizzato per indicare che il queue manager è remoto, viene ignorato e il comando viene eseguito in locale in modalità verifica (verification mode).

Risoluzione dei problemi con MQSC

Se i comandi MQSC non vengono eseguiti regolarmente, utilizzare il seguente elenco di controllo per riscontrare l'eventuale presenza di uno di questi frequenti problemi.

Quando si utilizza il comando `runmqsc`:

- Controllare che `$SYSTEM.ZMQSEXE` sia in `PMSEARCH` in `TACLCSTM`.
- Utilizzare l'operatore `IN` o l'indicatore `-i` quando si reindirizza l'input da un file. In caso contrario, il Queue Manager interpreta il nome del file come un nome di Queue Manager e inoltra il seguente messaggio di errore:

```
AMQ8118: MQSeries queue manager does not exist.
```

- Se si reindirizza un output su un file, utilizzare l'operatore `OUT` o l'indicatore `-o`. Per impostazioni predefinite, il file di output viene creato utilizzando i valori predefiniti `TACL` nel momento in cui il comando viene immesso. Specificare un nome file completo per inviare l'output a un file specifico.
- Verificare di aver creato il Queue Manager che dovrà eseguire i comandi. Effettuare questo controllo nel file di configurazione `MQSINI` che per impostazione predefinita è ubicato nel volume secondario di installazione, `$SYSTEM.ZMQSSYS`. Questo file contiene i nomi dei Queue Manager e il nome del Queue Manager predefinito, se impostato.
- E' necessario che il Queue Manager sia in esecuzione; in caso contrario, avviarlo come descritto in "Avvio di un queue manager" a pagina 66. Se il Queue Manager è già stato avviato viene visualizzato un messaggio di errore.

Problemi con MQSC

- Specificare un nome di Queue Manager nel comando **runmqsc** se non è stato definito un Queue Manager predefinito, in caso contrario si verifiche questo errore:

```
AMQ8146: MQSeries queue manager not available.
```

Per informazioni relative alla correzione di questo tipo di problema, consultare “Rendere predefinito un Queue Manager esistente” a pagina 68.

- Non è possibile specificare un comando MQSC come parametro **runmqsc**. Ad esempio, il seguente comando non è valido:

```
runmqsc DEFINE QLOCAL(FRED)
```

- Non è possibile inserire i comandi MQSC da TACL prima di aver immesso il comando **runmqsc**. Ad esempio:

```
DEFINE QLOCAL(QUEUE1)
```

* Errore Nome variabile, integrato, o file necessario.

- Non è possibile eseguire comandi di controllo da **runmqsc**. Ad esempio, non è possibile avviare un Queue Manager una volta avviata interattivamente l'esecuzione MQSC:

```
runmqsc (C) Copyright IBM Corp. 1993, 2001. Tutti i diritti riservati
Starting MQSeries Commands.

strmqm saturn.queue.manager
  1 : strmqm saturn.queue.manager
AMQ8405: Syntax error detected at or near end of command segment below:
```

Consultare anche la sezione “Eventuali problemi nell'utilizzo remoto di MQSC” a pagina 132.

Utilizzo delle code locali

Questa sezione contiene esempi dell'utilizzo di alcuni comandi MQSC. Per una descrizione completa di questi comandi, fare riferimento al manuale *MQSeries Guida di riferimento per i comandi*.

Definizione di una coda locale

Per un'applicazione, il Queue Manager locale è quello a cui è connessa l'applicazione. Le code gestite dal Queue Manager locale vengono definite locali rispetto a tale Queue manager.

Utilizzare il comando MQSC **DEFINE QLOCAL** per creare una definizione di una coda locale e la struttura dei dati della coda. È possibile anche modificare le caratteristiche della coda dalle code locali predefinite.

In questo esempio, ORANGE.LOCAL.QUEUE è specificato con le seguenti caratteristiche:

- Abilitata al ricevimento, non abilitata all’invio e funzionante su base FIFO (First-In-First-Out).
- Si tratta di una coda ‘normale’. Nel senso che non è una coda di iniziazione o una coda di trasmissione e non genera messaggi trigger.
- La capacità massima della coda è 1000 messaggi; la dimensione massima dei messaggi è 2000 byte.

Il seguente comando MQSC esegue questa azione:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL)
```

Note:

1. Molti di questi attributi, così come forniti col prodotto, sono preimpostati. Tuttavia sono riportati in figura a scopo dimostrativo. È possibile omettere questi attributi se si è certi che i valori predefiniti corrispondano alle proprie esigenze e che non siano stati modificati.
2. USAGE (NORMAL) indica che la coda non è una coda di trasmissione.
3. Se sullo stesso Queue Manager già esiste una coda locale denominata ORANGE.LOCAL.QUEUE, questo comando avrà esito negativo. Utilizzare l’attributo REPLACE, se si desidera sostituire questa definizione di coda a quella già esistente, ma consultare anche la sezione “Modifica degli attributi di coda locale” a pagina 109.

Modifica della dimensione di file fisico per le code

Per impostazione predefinita, il Queue Manager crea i file dei dati della coda che supportano fino a 100 MB di dati. Se viene raggiunto questo limite, verrà restituito il codice di ritorno MQRC_Q_SPACE_NOT_AVAILABLE. Per modificare la memoria massima assegnata a una coda, identificare prima i file fisici che contengono i dati relativi alla coda utilizzando **dspmqls**, quindi utilizzare il comando **altmqfls** per regolare la dimensione dell’estensione principale e secondaria e l’estensione massima. Se il file è già suddiviso in partizioni, **altmqfls** ridimensionerà tutte le partizioni. Per ulteriori informazioni, consultare la sezione “altmqfls (Modifica attributi del file di coda)” a pagina 254. Se è necessaria memoria aggiuntiva, è possibile suddividere il file tra più volumi.

Definizione di una coda messaggi non recapitati

Ogni Queue Manager deve avere una coda locale da utilizzare come coda messaggi non recapitati, in modo che questi messaggi che non riescono a raggiungere la loro corretta destinazione, possano essere memorizzati per un futuro ripristino.

Utilizzo delle code locali

E' necessario comunicare al Queue Manager la coda dei messaggi non recapitati. Ciò è possibile utilizzando il comando **crtmqm** oppure il comando ALTER QMGR per specificarne una successivamente. La coda messaggi non recapitati deve essere specificata prima di poterla utilizzare.

Un coda messaggi non recapitati di esempio, denominata SYSTEM.DEAD.LETTER.QUEUE, è fornita con il prodotto. Questa coda viene creata automaticamente quando viene creato il Queue Manager. Se necessario, è possibile modificare la definizione. Non è necessario, tuttavia, modificare il nome.

Una coda dei messaggi non recapitati non presenta alcun requisito particolare, a eccezione dei seguenti:

- Deve essere una coda locale.
- Il relativo attributo MAXMSGL (Maximum Message Length) deve consentire la memorizzazione dei messaggi più estesi gestiti dal Queue Manager **più** lo spazio occupato dall'intestazione del messaggio non recapitato (MQDLH).

MQSeries fornisce un handler della coda messaggi non recapitati che consente di specificare come elaborare i messaggi rilevati sulla coda messaggi non recapitati. Per ulteriori informazioni, consultare il manuale "Capitolo 9. Handler di code messaggi non recapitati di MQSeries" a pagina 159.

Visualizzazione degli attributi predefiniti di oggetti

Quando si definisce un oggetto MQSeries, tutti gli attributi non specificati vengono copiati dal corrispondente oggetto predefinito. Ad esempio, quando si definisce una coda locale, essa eredita tutti gli attributi che vengono omessi nella definizione dalla coda locale predefinita, denominata SYSTEM.DEFAULT.LOCAL.QUEUE. Per esaminare questi attributi, utilizzare il seguente comando:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Nota: la sintassi di questo comando è diversa da quella del corrispondente comando DEFINE.

È possibile scegliere gli attributi da visualizzare specificandoli individualmente. Ad esempio:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
MAXDEPTH +  
MAXMSGL +  
CURDEPTH
```

Questo comando visualizza i tre attributi specificati come segue:

```
AMQ8409: Display Queue details.  
QUEUE(ORANGE.LOCAL.QUEUE)  
MAXDEPTH(1000)  
MAXMSGL(2000)  
CURDEPTH(0)
```

Visualizzazione degli attributi di oggetto predefiniti

CURDEPTH rappresenta la capacità della coda e, cioè, il numero massimo di messaggi che la coda può contenere. È utile visualizzare questo attributo poiché, controllando la capacità della coda, si può evitare il riempimento della coda.

Copia della definizione di una coda locale

È possibile copiare la definizione di una coda utilizzando l'attributo LIKE con il comando DEFINE. Ad esempio:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
  LIKE (ORANGE.LOCAL.QUEUE)
```

Questo comando crea una coda con gli stessi attributi della coda ORANGE.LOCAL.QUEUE, invece di quelli della coda locale predefinita dal sistema.

E' inoltre possibile utilizzare questo formato del comando DEFINE per copiare una definizione di coda, sostituendo una o più modifiche degli attributi dell'originale. Ad esempio:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
  LIKE (ORANGE.LOCAL.QUEUE) +  
  MAXMSGL(1024)
```

Questo comando copia gli attributi della coda ORANGE.LOCAL.QUEUE nella coda THIRD.QUEUE, ma specifica come capacità massima per i messaggi della nuova coda il valore di 1024 byte, invece di 2000.

Note:

1. Quando si utilizza l'attributo LIKE con il comando DEFINE, si esegue la copia dei soli attributi della coda. Non è in corso la copia dei messaggi presenti nella coda.
2. Se si definisce una coda locale, senza specificare LIKE, è come se si utilizzasse DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE).

Modifica degli attributi di coda locale

Gli attributi di coda possono essere modificati in due modi: utilizzando il comando ALTER QLOCAL o il comando DEFINE QLOCAL con l'attributo REPLACE. Nella sezione "Definizione di una coda locale" a pagina 106 viene definita la coda ORANGE.LOCAL.QUEUE. Si ipotizzi di voler aumentare la dimensione massima dei messaggi su questa coda fino a 10 000 byte; ciò è possibile:

- Utilizzo del comando ALTER:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Questo comando modifica un singolo attributo, quello della dimensione massima dei messaggi; tutti gli altri attributi rimangono invariati.

- Utilizzando il comando DEFINE con l'opzione REPLACE, ad esempio:

Modifica degli attributi di coda locale

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Questo comando modifica non solo la dimensione massima dei messaggi, ma anche tutti gli altri attributi, a cui vengono assegnati i valori predefiniti. La coda ora è abilitata all'invio, laddove prima tale attributo era di valore opposto. L'abilitazione all'invio è il valore predefinito, così come specificato dalla coda SYSTEM.DEFAULT.LOCAL.QUEUE, se non modificata.

Se si *diminuisce* la lunghezza massima dei messaggi su una coda esistente, la modifica non riguarderà i messaggi esistenti. Tutti i nuovi messaggi, invece, devono corrispondere alle nuove impostazioni.

Modifica del volume di una coda locale

Utilizzare il comando **altmqfls** per modificare il volume su cui è memorizzata una coda locale, predefinita. E' possibile che sia necessario suddividere gli I/O del disco tra i volumi per ottimizzare le prestazioni del sistema. Il Queue Manager deve essere avviato prima dell'immissione di questo comando e la coda non deve essere aperta. Su un comando **altmqfls** può essere specificata solo una coda. Per la sintassi del comando **altmqfls**, consultare "altmqfls (Modifica attributi del file di coda)" a pagina 254.

Modifica delle opzioni di una coda locale

Utilizzare il comando **altmqfls** per eseguire le seguenti modifiche:

- Se la coda viene caricata dal disco nella cache quando viene avviato il Queue Manager. Utilizzare --qsoptions S.
- Se la coda rimane nella memoria durante l'esecuzione del queue server dopo il caricamento. Se impostato, una volta caricato, il queue server conserva in memoria le strutture della coda e i dati. In caso contrario, la memoria associata alla coda può essere rimossa dallo spazio di indirizzo del queue server quando non si accede più a essa. Utilizzare --qsoptions L.
- Se viene eseguito il checkpoint dei messaggi non permanenti sul queue server di backup per prevenire problemi causati da malfunzionamento, anche se il carico di lavoro della CPU viene così aumentato per l'esecuzione delle operazioni necessarie per gestire il checkpoint supplementare, i messaggi IPC supplementari e la memoria supplementare richiesta per memorizzare i messaggi. Utilizzare --qsoptions C.

Nota: *Tutte* le --qsoptions SLC vengono impostate ogni volta che viene emesso il comando. Ad esempio --qsoptions S annullerà l'impostazione di L e C. L'opzione --qsoptions può essere specificata solo una volta su una riga comandi.

- Il numero massimo di byte di dati di ciascun messaggio permanente da conservare nella cache del queue server (e su disco). il numero specificato di byte di dati verrà anch'esso conservato in memoria e l'operazione di browse restituirà questi dati all'applicazione senza dover accedere al disco. Utilizzando questo parametro sarà possibile aumentare le risorse della memoria utilizzate dal queue server. Utilizzare il parametro --browse.
- La dimensione minima del messaggio utilizzata da un file di overflow messaggi per la memorizzazioni dei dati del messaggio. I messaggi permanenti di dimensione inferiore al limite vengono memorizzati nel file di overflow di coda.

Ricalcolo, aggiornamento e richiamo

I messaggi di dimensione superiore al limite, verranno parzialmente memorizzati in un apposito file di overflow di messaggi. Utilizzare il parametro `--msgofthresh`.

- Il volume secondario nel volume in cui si trova la coda dove il queue server crea nuovi file di overflow di messaggio. Utilizzare il parametro `--msgofsubvol`.
- Il nome di un Counter per le dimensioni (measure) che, se parte di una misurazione attiva, viene avviato a partire dalla profondità di coda corrente e quindi aumentato o ridotto dal responsabile del queue server relativo alla coda cui vengono aggiunti o rimossi i messaggi. Utilizzare il parametro `--meascount`.
- La dimensione dell'estensione principale e secondaria e il numero massimo delle estensioni relative al file di coda. Utilizzare il parametro `--qsize` (`primaryextent,secondaryextent,maxextents`). Con il parametro `--qsize`, è necessario specificare tutti i valori.
- La dimensione dell'estensione principale e secondaria e il numero massimo delle estensioni relative al file di overflow di coda. Utilizzare il parametro `--osize` (`primaryextent,secondaryextent,maxextents`). Con il parametro `--osize`, è necessario specificare tutti i valori.

Riassegnazione degli oggetti agli status server e ai queue server

Gli status server gestiscono tutti gli oggetti eccetto le code locali e modello. I queue server gestiscono tutti gli oggetti della coda locale e della coda modello.

Inizialmente, tutti gli oggetti vengono creati in modo da utilizzare il queue server o lo status server predefiniti, a seconda dell'oggetto. Utilizzando **altmqfls** dopo la creazione di un oggetto, è possibile configurare l'oggetto in modo che utilizzi un server appropriato diverso da quello predefinito. Prima di utilizzare l'oggetto, è necessario che la nuova classe server PATHWAY relativa al server sia stata configurata e avviata.

E' possibile specificare un nome processo o il termine DEFAULT nella riga comandi di **altmqfls**. Nel momento in cui l'oggetto viene riconfigurato non viene eseguito alcun controllo per verificare che il nuovo server sia attivo o configurato.

E' possibile utilizzare il comando **dspmqfls** per visualizzare status server corrente di un oggetto.

E' possibile impostare il queue server per le code modello e locali. Le code dinamiche ereditano il queue server dalla coda modello utilizzata per la relativa creazione.

Annullamento di una coda locale

Per eliminare tutti i messaggi presenti su una coda locale denominata MAGENTA.QUEUE, utilizzare il seguente comando:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

Non è possibile annullare la coda se:

- Alla coda sono stati inviati messaggi di cui non è stato eseguito il commit sottoposti a syncpoint.
- La coda è stata appena aperta da un'applicazione.


```

ApplOriginData : '      '
**** Message      ****

length - 36 bytes

00000000: 0000 0002 0000 0024 0000 0001 0000 0015 '.....$......'
00000010: 0000 0001 0000 0001 0000 0000 0000 0000 '.....'
00000020: 0000 0000                                '.....'

No more messages
MQCLOSE
MQDISC

```

Utilizzo degli alias di coda

Un alias di coda consente di reindirizzare le chiamate MQI. Un alias di coda non è una coda reale, ma una definizione che si riferisce ad una coda reale. La definizione di alias di coda contiene un nome di coda destinatario che viene specificato dall'attributo TARGQ (*BaseQName* in PCF). Quando un'applicazione specifica un alias di coda in una chiamata MQI, il Queue Manager, al momento dell'esecuzione, si riferisce al nome della coda reale.

Ad esempio, un'applicazione deve inviare un messaggio alla coda denominata MY.ALIAS.QUEUE. Essa specifica il nome di questa coda quando esegue una richiesta MQOPEN e, indirettamente, quando invia un messaggio a questa coda. L'applicazione non rileva che si tratta di un alias di coda. Per ciascuna chiamata MQI che utilizza questo alias, il Queue Manager si riferisce al nome della coda reale, che può essere sia una coda locale che remota rispetto a questo Queue Manager.

Modificando il valore dell'attributo TARGQ, è possibile reindirizzare le chiamate MQI verso un'altra coda, anche su un altro Queue Manager. Operazione utile per la gestione, la migrazione e il bilanciamento del carico.

Definizione di un alias di coda

Il seguente comando crea una coda alias:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (YELLOW.QUEUE)
```

Questo comando reindirizza le chiamate MQI che specificano MY.ALIAS.QUEUE, verso la coda YELLOW.QUEUE. Il comando non crea la coda di destinazione; le chiamate MQI hanno esito negativo se la coda YELLOW.QUEUE non esiste al momento dell'esecuzione.

Se si modifica la definizione dell'alias, è possibile reindirizzare le chiamate MQI verso un'altra coda. Ad esempio:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (MAGENTA.QUEUE) REPLACE
```

Questo comando reindirizza le chiamate MQI verso un'altra coda, MAGENTA.QUEUE.

Definizione di una coda alias

Inoltre è possibile utilizzare gli alias di coda per far sì che una singola coda sembri avere diversi attributi per diverse applicazioni. Per far ciò basta definire due alias, uno per ogni applicazione. Si ipotizzino le seguenti due applicazioni:

- L'applicazione ALPHA può inviare i messaggi alla coda YELLOW.QUEUE, ma non può riceverne da questa coda.
- L'applicazione BETA può ricevere messaggi dalla coda YELLOW.QUEUE, ma non può inviarvene.

E' possibile eseguire quest'azione utilizzando i seguenti comandi:

```
* This alias is put enabled and get disabled for application ALPHA

DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +
    TARGQ (YELLOW.QUEUE) +
    PUT (ENABLED) +
    GET (DISABLED)

* This alias is put disabled and get enabled for application BETA

DEFINE QALIAS (BETAS.ALIAS.QUEUE) +
    TARGQ (YELLOW.QUEUE) +
    PUT (DISABLED) +
    GET (ENABLED)
```

ALPHA utilizza il nome di coda ALPHAS.ALIAS.QUEUE nelle relative chiamate MQI; BETA utilizza il nome di coda BETAS.ALIAS.QUEUE. Entrambe hanno accesso alla stessa coda, ma in modi diversi.

È possibile utilizzare gli attributi LIKE e REPLACE quando si definiscono gli alias di code, nello stesso modo in cui si utilizzano tali attributi nella definizione di code locali.

Utilizzo di altri comandi con gli alias di coda

È possibile utilizzare gli appropriati comandi MQSC per visualizzare o modificare gli attributi dell'alias di coda o per eliminare gli oggetti dell'alias di coda. Ad esempio:

```
* Display the queue alias' attributes
* ALL = Display all attributes

DISPLAY QUEUE (ALPHAS.ALIAS.QUEUE) ALL

* ALTER the base queue name, to which the alias resolves.
* FORCE = Force the change even if the queue is open.

ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGQ(ORANGE.LOCAL.QUEUE) FORCE

* Delete this queue alias, if you can.

DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Non è possibile eliminare un'alias di coda se, ad esempio, un'applicazione ha aperto di recente la coda o una coda che richiama questa coda. Per ulteriori informazioni relative a questi o altri comandi alias della coda, consultare *MQSeries Guida di riferimento per i comandi*.

Utilizzo delle code di esempio

Un Queue Manager crea una *coda dinamica* se riceve una chiamata MQI da un'applicazione che specifica un nome di coda definito come coda di esempio. Il nome della nuova coda dinamica è generato dal Queue Manager al momento della sua creazione. Una *coda di esempio* è un modello che specifica gli attributi di ogni coda dinamica creata.

Le code di esempio forniscono un metodo utile alle applicazioni per creare code in base alle necessità.

Definizione di una coda di esempio

Per definire le code di esempio occorre utilizzare un gruppo di attributi, così come avviene per le code locali. Le code di esempio e le code locali hanno lo stesso gruppo di attributi; tuttavia con le code di esempio è possibile specificare se le code dinamiche create devono essere temporanee o permanenti. Le code permanenti vengono conservate quando il Queue Manager viene riavviato, quelle temporanee no. Ad esempio:

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
  DESCR('Queue for messages from application X') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL) +
  DEFTYPE (PERMDYN)
```

Questo comando crea una definizione di coda di esempio. In base all'attributo DEFTYPE, le code attualmente create da questo modello sono code dinamiche permanenti.

Nota: gli attributi non specificati vengono copiati automaticamente dalla coda predefinita SYSYTEM.DEFAULT.MODEL.QUEUE.

È possibile utilizzare gli attributi LIKE e REPLACE quando si definiscono code di esempio, nello stesso modo in cui si utilizzano tali attributi nella definizione di code locali.

Utilizzo degli altri comandi con le code di esempio

È possibile utilizzare i comandi MQSC appropriati per visualizzare, modificare gli attributi di una coda di esempio oppure per eliminare l'oggetto coda di esempio. Ad esempio:

Gestione degli oggetti per triggering

```
* Display the model queue's attributes
* ALL = Display all attributes

DISPLAY QUEUE (GREEN.MODEL.QUEUE) ALL

* ALTER the model to enable puts on any
* dynamic queue created from this model.

ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)

* Delete this model queue:

DELETE QMODEL (RED.MODEL.QUEUE)
```

Gestione degli oggetti per triggering

MQSeries prevede una funzione che consente l'avvio automatico di un'applicazione qualora si verificano determinate condizioni. Un esempio di tali condizioni si ha quando i messaggi presenti su una coda raggiungono un determinato numero. Questa funzione è denominata *triggering* ed è dettagliatamente descritta nel manuale *MQSeries Application Programming Guide*. Questa sezione descrive il modo come impostare gli oggetti in modo da supportare la funzione triggering in MQSeries for Compaq NSK.

Definizione di una coda di applicazione per triggering

Una coda di applicazione è una coda locale utilizzata da un'applicazione per la messaggistica tramite MQI. La funzione triggering richiede la definizione di alcuni attributi nella coda di applicazione. La stessa funzione triggering viene abilitata dall'attributo *Trigger* (TRIGGER in MQSC).

Nell'esempio che segue, viene generato un evento trigger quando nella coda locale MOTOR.INSURANCE.QUEUE sono presenti 100 messaggi di priorità 5 o superiore:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
MAXMSGL (2000) +
DEFPSIST (YES) +
INITQ (MOTOR.INS.INIT.QUEUE) +
TRIGGER +
TRIGTYPE (DEPTH) +
TRIGDPTH (100)+
TRIGPRI (5)
```

Dove:

QLOCAL (MOTOR.INSURANCE.QUEUE)

Specifica il nome della coda di applicazione di cui si esegue la definizione.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

Specifica il nome dell'applicazione che deve essere avviata da un programma di monitoraggio del trigger.

MAXMSGL (2000)

Specifica la dimensione massima dei messaggi nella coda.

DEFPSIST (YES)

Specifica che i messaggi sono permanenti su questa coda.

INITQ (MOTOR.INS.INIT.QUEUE)

Indica il nome della coda di iniziazione sulla quale il Queue Manager invia il messaggio trigger.

TRIGGER

È l'impostazione dell'attributo trigger.

TRIGTYPE (DEPTH)

Specifica che un evento trigger deve verificarsi qualora i messaggi della priorità indicata (TRIMPRI) raggiunga il numero specificato in TRIGDPH.

TRIGDPH (100)

Specifica il numero dei messaggi necessari per generare un evento trigger.

TRIMPRI (5)

Indica la priorità dei messaggi che il Queue Manager deve contare nel decidere di generare l'evento trigger. Vengono contati solo i messaggi di priorità 5 o superiore.

Definizione di una coda di iniziazione

Quando si verifica un evento trigger, il Queue Manager invia un messaggio trigger alla coda di iniziazione specificata nella definizione della coda di applicazione. Le code di iniziazione non hanno impostazioni particolari, ma è consigliabile utilizzare la definizione seguente della coda locale MOTOR.INS.INIT.QUEUE:

```
DEFINE QLOCAL(MOTOR.INS.INIT.QUEUE) +
  GET (ENABLED) +
  NOSHARE +
  NOTRIGGER +
  MAXMSGL (2000) +
  MAXDEPTH (10)
```

Creazione di una definizione di processo

Utilizzare il comando DEFINE PROCESS per creare una definizione di processo. Una definizione di processo associa una coda di applicazione all'applicazione che elabora i messaggi dalla coda. Ciò è possibile grazie all'utilizzo dell'attributo PROCESS nella coda di applicazione MOTOR.INSURANCE.QUEUE. Il seguente comando MQSC definisce il processo necessario, MOTOR.INSURANCE.QUOTE.PROCESS, indicato nell'esempio:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
  DESCR ('Insurance request message processing') +
  APPLTYPE (NSK) +
  APPLICID ('$DATA1.TEST.IRMP01') +
  USERDATA ('open, close, 235')
```

Dove:

MOTOR.INSURANCE.QUOTE.PROCESS

È il nome della definizione di processo.

DESCR ('Insurance request message processing')

Con questi caratteri di testo si indica il programma applicativo a cui si riferisce la definizione, in funzione delle parole chiave. Tale testo viene

Creazione di una definizione di processo

visualizzato utilizzando il comando DISPLAY PROCESS. Ciò può facilitare l'identificazione della funzione del processo. Se si utilizzano spazi nella stringa, sarà necessario racchiudere il testo tra virgolette.

APPLTYPE(NSK)

Indica il tipo di applicazione eseguita su Compaq NSK.

APPLICID ('\$DATA1.TEST.IRMPO1')

E' il nome del programma eseguibile applicativo sul sistema locale.

USERDATA ('open, close, 235')

Sono i dati definiti dall'utente, che verranno utilizzati dall'applicazione.

Visualizzazione della definizione di processo

Utilizzare il comando DISPLAY PROCESS, con la parola chiave ALL, per esaminare i risultati della definizione. Ad esempio:

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR (Insurance request message processing)
APPLICID ($DATA1.TEST.IRMPO1)
ENVRDATA ( )
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (NSK)
```

USERDATA è una stringa che rappresenta gli argomenti inoltrati all'applicazione con funzione trigger. Consultare i programmi modello AMQSTRG0 e AMQINQA (nel volume secondario ZMQSSMPL) per gli esempi relativi alle modalità di scrittura dei monitor trigger e delle applicazioni con funzione trigger.

E' inoltre possibile utilizzare MQSC ALTER PROCESS per modificare una definizione di processo esistente e DELETE PROCESS per cancellare una definizione di processo.

Capitolo 6. Automazione delle attività di gestione

Questo capitolo presume che l'utente abbia già avuto esperienza di gestione degli oggetti MQSeries.

Nel caso in cui si decida di migliorare l'installazione rendendo automatiche alcune attività di gestione e di controllo, è possibile utilizzare i comandi PCF (Programmable Command Format) per i Queue Manager sia locale che remoto.

Questo capitolo descrive:

- Come utilizzare i PCF per automatizzare le attività di amministrazione in esecuzione delle attività di gestione utilizzando comandi PCF.
- Come utilizzare il server di comando in "Gestione del server di comando per una gestione remota" a pagina 123.

Comandi PCF

La funzione dei comandi MQSeries PCF (Programmable Command Format) è di consentire la pianificazione delle attività di gestione in un programma di gestione. In questo modo è possibile creare code, definizioni di processi, canali ed elenchi nomi e modificare i queue manager da un programma.

Le funzioni MQSC sono realizzate in modo analogo dai comandi PCF.

Infatti, è possibile impostare un programma in modo che un singolo nodo può emettere comandi PCF verso qualsiasi Queue Manager della rete. In questo modo le attività di gestione vengono centralizzate e al tempo stesso automatizzate.

Ogni comando PCF è una struttura di dati che può essere integrato nella parte di un messaggio MQSeries relativa ai dati applicativi. Ogni comando viene inviato al Queue Manager di destinazione utilizzando la funzione MQI MQPUT così come avviene per ogni altro messaggio. Il server di comando del Queue Manager che riceve il messaggio lo riconosce come comando e lo esegue. Per ricevere le risposte, l'applicazione realizza una chiamata MQGET e i dati relativi alla risposta sono inviati in un'altra struttura di dati. A questo punto l'applicazione può analizzare la risposta e eseguirne il contenuto.

Nota: diversamente dai comandi MQSC, i comandi PCF e le relative risposte non sono in un formato testo leggibile.

In breve, queste sono note che il programmatore di applicazione deve specificare quando si crea un messaggio relativo ad un comando PCF:

Descrittore messaggio

Questo è un descrittore messaggio standard MQSeries, nel quale:
Tipo di messaggio (*MsgType*) è MQMT_REQUEST.
Formati di messaggio (*Format*) è MQFMT_ADMIN.

Dati applicativi

Contiene i messaggi PCF incluso l'intestazione PCF, nel quale:

Il tipo di messaggio PCF (*Type*) specifica MQCFT_COMMAND.

L'identificativo del comando specifica il comando, per esempio, *Cambia coda* (MQCMD_CHANGE_Q).

comandi PCF

Per una descrizione completa delle strutture dei dati PCF e sulla loro implementazione, consultare il manuale *MQSeries Programmable System Management*.

Attributi MQSC e PCF

Gli attributi degli oggetti specificati in MQSC sono riportati in questo manuale in lettere maiuscole (ad esempio RQMNAME), sebbene essi non siano sensibili al maiuscolo/minuscolo. I nomi degli attributi MQSC hanno una lunghezza massima di otto caratteri.

Gli attributi degli oggetti in PCF, il cui nome è formato da più di otto caratteri, è riportato in questo manuale con il tipo di carattere "italics". Ad esempio, il comando PCF equivalente a RQMNAME è *RemoteQMgrName*.

Escape PCF

Gli Escape PCF sono comandi PCF che contengono i comandi MQSC con il messaggio di testo. È possibile utilizzare i comandi PCF per inviare comandi ad un Queue Manager remoto. Per ulteriori informazioni relative all'utilizzo di escape PCF, consultare il manuale *MQSeries Programmable System Management*.

Utilizzo di MQAI per semplificare l'impiego di PCF

La MQAI è un'interfaccia di amministrazione per MQSeries adesso disponibile sulla piattaforma Compaq NSK.

Essa realizza attività di gestione su un Queue Manager grazie all'impiego di *data bags* (cartella dati). La cartella dati (Data bag) consente di gestire proprietà o parametri di oggetti in modo più semplice rispetto a PCF.

MQAI può essere utilizzato:

- **Per semplificare l'utilizzo dei messaggi PCF.** MQAI consente una facile gestione di MQSeries; non è necessario scrivere i messaggi PCF e questo evita i problemi che nascono dalla complessità delle strutture dei dati.
Per passare parametri in programmi scritti utilizzando le chiamate MQI, i messaggi PCF devono contenere il comando e i dettagli dei dati String o Integer. A questo scopo, per ogni struttura occorre fornire alcune specifiche al programma e destinare un apposito spazio nella memoria. Questa operazione è complessa ed impiega molto tempo.
Del resto, i programmi vengono compilati utilizzando i parametri MQAI pass nell'appropriata data bag (cartella dati) ed è richiesta una sola istruzione per ogni struttura. L'utilizzo della cartella dati MQAI esclude la necessità di fornire specifiche e assegnare spazio nella memoria e aumenta il grado di isolamento dai dettagli del PCF.
- **Per gestire in modo più semplice le condizioni di errore.** È difficile risalire al codice dai comandi MQSC, tuttavia MQAI rende più facile al programma la gestione quando si verifica un errore.

Dopo aver creato e inserito dati nella cartella dati (data bag), è possibile inviare un messaggio del comando di gestione al server di comando di un Queue Manager, utilizzando una chiamata mqExecute che attenderà un messaggio di risposta. La chiamata mqExecute gestisce lo scambio con il server di comandi e invia la risposta in una response bag (cartella risposte).

Per ulteriori informazioni sull'utilizzo della MQAI, consultare: *MQSeries Administration Interface Programming Guide and Reference*.

Per informazioni generali relative a PCFs, consultare il manuale *MQSeries Programmable System Management*.

Gestione del server di comando per una gestione remota

È possibile associare un server di comando a ciascun Queue Manager. Un server di comando elabora tutti i comandi inviati da una Queue Manager remota o i comandi PCF delle applicazioni. Esso fornisce i comandi al Queue Manager per l'elaborazione e restituisce un codice di completamento o un messaggio operatore in base all'origine del comando.

Un server di comando è necessario ad ogni gestione che utilizzano PCF, MQAI ed anche per la gestione in remoto.

Nota: per le gestioni in remoto, assicurarsi che il Queue Manager di destinazione sia in esecuzione. Altrimenti, i messaggi, che contengono i comandi, non possono lasciare il Queue Manager da cui sono inviati. Invece, questi messaggi sono accodati alla trasmissione locale che invia dati al Queue Manager remoto. Questa situazione deve essere evitata, se possibile.

Avvio del server di comando

Per avviare il server di comando utilizzare questo comando:

```
strmqcsv saturn.queue.manager
```

dove `saturn.queue.manager` indica il Queue Manager per il quale si avvia il server di comando.

Il server di comandi può essere avviato anche in PATHCOM sbloccandone ed avviandone la relativa classe server.

Visualizzazione dello stato del server di comando

Per le gestioni in remoto, assicurarsi che il server di comando del Queue Manager di destinazione sia in esecuzione. Diversamente i comandi in remoto non potranno essere elaborati. Tutti i messaggi che contengono comandi vengono accodati alla coda comandi del Queue Manager di destinazione.

Per visualizzare lo stato del server di comando relativo ad un Queue Manager, qui definito `saturn.queue.manager`, il comando è:

```
dspmqcsv saturn.queue.manager
```

È necessario inviare questo comando alla macchina di destinazione. Se il server di comando è in esecuzione viene inviato il seguente messaggio:

```
AMQ8027    MQSeries Command Server Status ...: Running
```

Gestione remota del server di comando

Arresto del server di comando

Per chiudere un server di comando, utilizzando l'esempio precedente, il comando è:

```
endmqsv saturn.queue.manager
```

È possibile arrestare il server di comando in due differenti modi:

- Per un arresto controllato, modo predefinito, utilizzare il comando **endmqsv** con il flag **-c**.
- Per un arresto immediato, utilizzare il comando **endmqsv** con il flag **-i**.

Nota: arrestando un Queue Manager si chiude anche il server di comando ad esso associato (se avviato).

Capitolo 7. Gestione di oggetti remoti di MQSeries

Questo capitolo illustra come gestire oggetti di MQSeries su un altro Queue Manager. In esso si descrive come utilizzare gli oggetti di coda remota per controllare la destinazione dei messaggi e dei messaggi di risposta.

Esso contiene le seguenti sezioni:

- “Gestione di un Queue Manager remoto” a pagina 127
- “Creazione di una definizione locale di una coda remota” a pagina 133
- “Utilizzo di definizioni di coda remota come alias” a pagina 136

Per ulteriori informazioni sui canali, i relativi attributi e le modalità di impostazione, consultare il manuale *MQSeries Intercommunication*.

Canali, cluster e accodamento remoto

Un Queue Manager comunica con un altro Queue Manager inviando un messaggio e, se necessario, ricevendo una risposta. Il Queue Manager ricevente potrebbe essere:

- Sullo stesso computer
- Su un altro computer nella stessa ubicazione o dall'altra parte del mondo
- In esecuzione sulla stessa piattaforma del Queue Manager locale
- In esecuzione su un'altra piattaforma supportata da MQSeries

Questi messaggi potrebbero essere originati da:

- Programmi applicativi scritti dall'utente, che trasferiscono dati da un nodo a un altro.
- Applicazioni di gestione scritte dall'utente, che utilizzano i PCF o MQAI.
- Queue Manager che inviano:
 - Messaggi di evento di strumentazione inviati a un altro Queue Manager.
 - Comandi MQSC emessi da un comando **runmqsc** in modo indiretto (in cui i comandi sono eseguiti su un altro Queue Manager).

Prima che sia possibile inviare un messaggio a un Queue Manager remoto, il Queue Manager locale necessita di un meccanismo per rilevare l'arrivo di messaggi e trasportarli consistenti in:

- Almeno un canale
- Una coda di trasmissione
- Un agente del canale di messaggio (MCA)
- Un listener di canale
- Un channel initiator

Un canale è una connessione a senso unico tra due Queue Manager e può trasportare messaggi destinati a un qualsiasi numero di code presso il Queue Manager remoto.

Ciascuna estremità del canale dispone di una distinta definizione. Ad esempio, se un'estremità è un mittente o un server, l'altra estremità dovrà essere un ricevente o un richiedente. Un semplice canale consiste in una *definizione di canale mittente* sull'estremità del Queue Manager e in una *definizione di canale ricevente*

gestione degli oggetti remoti

sull'estremità del Queue Manager remoto. Le due definizioni devono avere lo stesso nome e costituire assieme un unico canale.

Se si presume che il Queue Manager remoto risponda ai messaggi inviati dal Queue Manager locale, sarà necessario impostare un secondo canale per inviare risposte al Queue Manager locale.

I canali vengono definiti mediante il comando MQSC DEFINE CHANNEL. In questo capitolo, gli esempi relativi ai canali utilizzano gli attributi di canale predefiniti se non diversamente specificato.

In ciascuna estremità di un canale esiste un MCA (Message Channel Agent) che controlla l'invio e la ricezione di messaggi. È compito di MCA prelevare i messaggi dalla coda di trasmissione e inserirli nella connessione tra i Queue Manager. Inversamente, è compito dell'MCA ricevente estrarre i messaggi dalla connessione e inserirli nelle code di destinazione.

Una coda di trasmissione è una coda locale specializzata che contiene temporaneamente i messaggi prima che siano prelevati da MCA e inviati al Queue Manager remoto. Specificare il nome della coda di trasmissione su una *definizione di coda remota*.

La sezione "Preparazione di canali e code di trasmissione per la gestione remota" a pagina 128 illustra l'utilizzo di tali definizioni per impostare la gestione remota.

Per ulteriori informazioni sull'impostazione dell'accodamento distribuito in generale, consultare il manuale *MQSeries Intercommunication*.

Gestione remota mediante cluster

In una normale rete MQSeries che utilizza l'accodamento distribuito, tutti i Queue Manager sono indipendenti. Affinché un Queue Manager possa inviare un messaggio a un altro Queue Manager, occorre aver definito una coda di trasmissione, un canale diretto al Queue Manager remoto e una definizione di coda remota per ogni coda a cui inviare messaggi.

Un *cluster* è un gruppo di Queue Manager impostato in modo che i Queue Manager possano comunicare direttamente tra loro tramite un'unica rete, senza necessità di definizioni complesse di coda di trasmissione, canali e coda. È possibile impostare facilmente i cluster, che di norma contengono Queue Manager correlati in maniera logica e necessitano la condivisione dei dati o delle applicazioni.

Una volta creato un cluster, i Queue Manager al suo interno possono comunicare tra loro *senza la necessità di complicate definizioni di canale o di code remote*. Persino il cluster più piccolo ridurrà i rischi relativi alla gestione del sistema.

Stabilendo una rete di Queue Manager in un cluster si avranno meno definizioni rispetto allo stabilire un ambiente tradizionale di accodamento distribuito. Con un numero inferiore di definizioni da effettuare, sarà possibile impostare o modificare la propria rete più rapidamente e facilmente e si ridurrà il rischio di errori nelle definizioni.

Per impostare un cluster, di norma è necessario una definizione di mittente cluster (CLUSSDR) e una definizione di ricevente cluster (CLUSRCVR) per Queue Manager. Non saranno necessarie alcune definizioni di coda di trasmissione o di

coda remota. I principi della gestione remota sono gli stessi all'interno di un cluster, ma le definizioni stesse sono enormemente semplificate.

Per ulteriori informazioni sui cluster, i relativi attributi e le modalità di impostazione, consultare il manuale *MQSeries Queue Manager Clusters*.

Gestione di un Queue Manager remoto

Questa sezione indica come gestire un Queue Manager remoto da un Queue Manager locale. È possibile implementare la gestione remota da un nodo locale mediante:

- comandi MQSC
- comandi PCF

La preparazione delle code e dei canale è fondamentalmente identica per entrambi i metodi. In questo volume, gli esempi illustrano i comandi MQSC, poiché sono più semplici da comprendere. Tuttavia, è possibile convertire gli esempi in PCF, se lo si desidera. Per ulteriori informazioni sulla scrittura di programmi di gestione mediante PCF, consultare il manuale *MQSeries Programmable System Management*.

Con la gestione remota si inviano comandi MQSC a un Queue Manager remoto in maniera interattiva o da un file di testo contenente i comandi. Il Queue Manager remoto potrebbe risiedere sullo stesso computer o, più comunemente, su un computer diverso. È possibile gestire in remoto i Queue Manager in diversi ambienti di MQSeries, inclusi UNIX[®], Compaq NSK, AS/400[®], MVS/ESA, Windows[®] 2000 e OS/390.

Per implementare la gestione remota, occorre creare determinati oggetti. A meno che non siano necessari requisiti specializzati, i valori predefiniti dovrebbero risultare sufficienti (ad esempio, per la lunghezza del messaggio).

Preparazione dei Queue Manager alla gestione remota

Figura 37 a pagina 128 illustra la configurazione dei Queue Manager e dei canali necessari per la gestione remota. `source.queue.manager` è il Queue Manager *source* da cui è possibile inviare comandi MQSC e a cui vengono restituiti i risultati di tali comandi (messaggi per l'operatore), se possibile. `target.queue.manager` è il Queue Manager di destinazione, che elabora i comandi e genera qualsiasi messaggio per l'operatore.

Nota: `source.queue.manager` deve essere il Queue Manager predefinito sulla macchina utilizzata. Per ulteriori informazioni sulla creazione del Queue Manager, consultare la sezione "crtmqm (Crea Queue Manager)" a pagina 267.

Gestione di un Queue Manager remoto

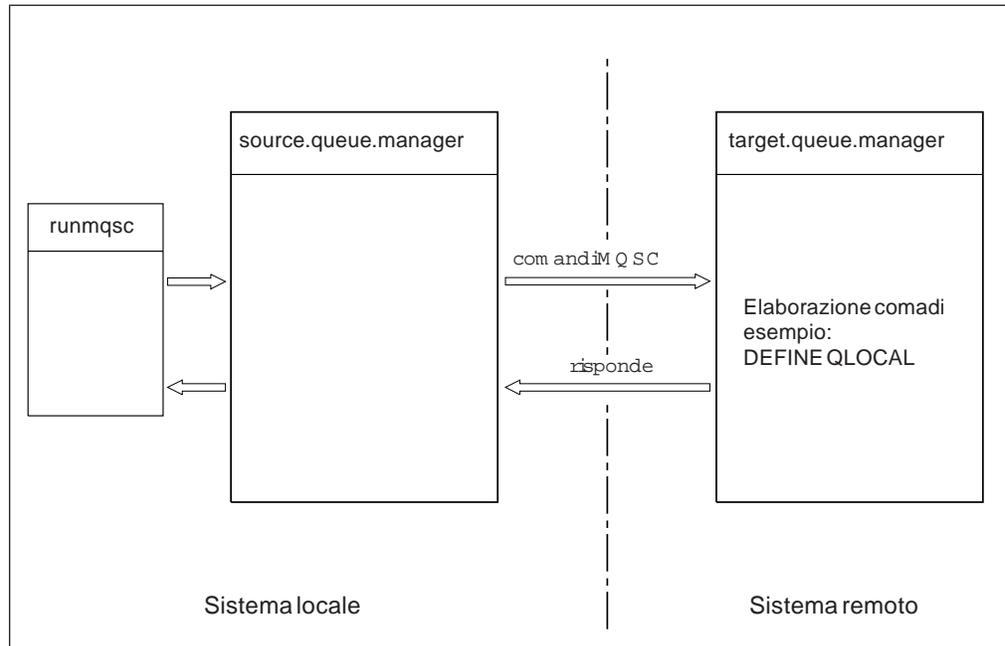


Figura 37. Gestione remota

Se non sono ancora state effettuate le seguenti operazioni, su entrambi i sistemi occorre:

- Creare il Queue Manager, mediante il comando **crtmqm**.
- Avviare il Queue Manager, mediante il comando **strmqm**.

Consultare la sezione “Ripristino degli oggetti predefiniti e di sistema” a pagina 66 per ulteriori informazioni su questa procedura. Occorre eseguire tali comandi localmente o su una rete, ad esempio Telnet.

Sul Queue Manager di destinazione:

- Deve essere presente la coda di comando, `SYSTEM.ADMIN.COMMAND.QUEUE`. Per impostazione predefinita, questa coda viene creata quando viene creato un queue manager.
- È necessario avviare il server di comandi mediante il comando **strmqcsv**.

Preparazione di canali e code di trasmissione per la gestione remota

Per eseguire in remoto i comandi MQSC, occorre impostare due canali, uno per ciascuna direzione e per le code di trasmissione associate. Tale esempio presume che si sta utilizzando il protocollo TCP/IP come tipo di trasporto e che si è al corrente dell'indirizzo TCP/IP.

Il canale `source.to.target` serve a inviare comandi MQSC dal Queue Manager di origine alla destinazione. Il relativo mittente è all'indirizzo `source.queue.manager` e il relativo ricevente è al Queue Manager `target.queue.manager`. Il canale `target.to.source` serve a restituire l'output dai comandi e qualsiasi messaggio per l'operatore generato per il Queue Manager di origine. Occorre inoltre definire una coda di trasmissione per ciascun mittente. Questa coda è una coda locale a cui viene assegnato il nome del Queue Manager ricevente. Per consentire la gestione in remoto, il nome `XMITQ` deve corrispondere al nome del queue manager remoto, a meno che per il queue manager non si stia utilizzando un alias. Figura 38 a pagina 129 riepiloga tale configurazione. Tuttavia, è necessario sapere che `SYSTEM.MQSC.REPLY.QUEUE` è il nome della coda modello utilizzata da MQSC

Gestione di un Queue Manager remoto

per sviluppare la propria coda di risposta dinamica. Il nome di tale coda varia ed è interno a MQSC.

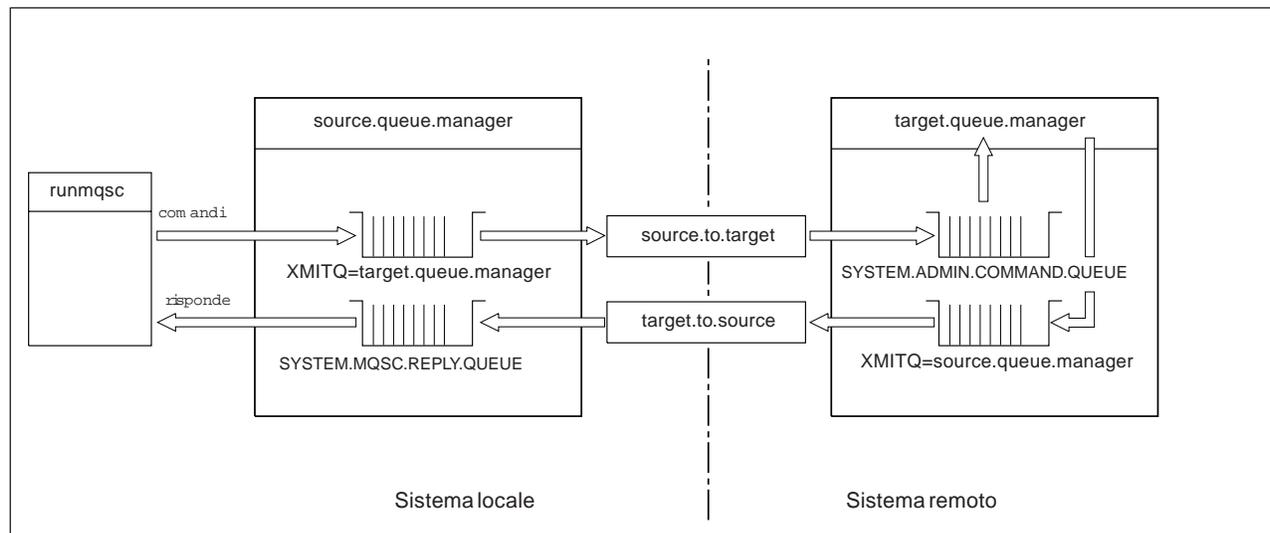


Figura 38. Impostazione di canali e code per la gestione remota

Consultare il manuale *MQSeries Intercommunication* per ulteriori informazioni sull'impostazione di canali remoti.

Definizione di canali e code di trasmissione

Sul Queue Manager di origine, emettere questi comandi MQSC per definire i canali e la coda di trasmissione:

```
* Define the sender channel at the source queue manager
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)

* Define the receiver channel at the source queue manager

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Define the transmission queue on the source

DEFINE QLOCAL ('target.queue.manager') +
  USAGE (XMITQ)
```

Emettere tali comandi sul Queue Manager di destinazione (`target.queue.manager`), per creare i canali e la coda di trasmissione:

Gestione di un Queue Manager remoto

```
* Define the sender channel on the destination queue manager

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ ('source.queue.manager') +
  TRPTYPE(TCP)

* Define the receiver channel on the destination queue manager

DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Define the transmission queue on the destination queue manager

DEFINE QLOCAL ('source.queue.manager') +
  USAGE (XMITQ)
```

Nota: i nomi della connessione TCP/IP specificati per l'attributo CONNAME nelle definizioni di canale mittente sono intesi solo a scopo illustrativo. Questo è l'indirizzo IP o il nome di rete dell'elaboratore all'altra estremità della connessione. Utilizzare i valori appropriati per la propria rete.

Avvio dei canali

La seguente descrizione presuppone che entrambe le estremità del canale siano in esecuzione su MQSeries for Compaq NSK. In caso contrario, fare riferimento alla relativa documentazione per l'estremità diversa da Compaq NSK del canale.

Per avviare i due canali, assicurarsi dapprima che il processo del listener TCP/IP di Compaq NSK sia stato configurato per MQSeries su entrambi i nodi e che essi siano in esecuzione su entrambe le estremità delle connessioni. Quindi, avviare i canali in **runmqsc**.

- Sul Queue Manager di origine, digitare:

```
start channel ('source.to.target')
```

- Sul Queue Manager di destinazione, digitare:

```
start channel ('target.to.source')
```

Definizione automatica dei canali

La definizione automatica dei canali si applica solo se il Queue Manager di destinazione è in esecuzione sui prodotti MQSeries Versione 5.1 o successive. Se viene ricevuta una richiesta di allegato in arrivo e non è possibile rilevare una definizione di un ricevente appropriato o di connessione al server nel file di definizione canale (CDF), MQSeries crea automaticamente una definizione e la aggiunge al CDF. Le definizioni automatiche si basano su due definizioni predefinite fornite con MQSeries: SYSTEM.AUTO.RECEIVER e SYSTEM.AUTO.SVRCONN.

Abilitare la definizione automatica delle definizioni di ricevente e connessione al server aggiornando l'oggetto Queue Manager mediante il comando MQSC, ALTER QMGR (o il comando PCF Change Queue Manager).

Per ulteriori informazioni sulla creazione automatica delle definizioni di canale, consultare il manuale *MQSeries Intercommunication*.

Per informazioni sulla definizione automatica di canali per i cluster, consultare il manuale *MQSeries Queue Manager Clusters*.

Emissione remota di comandi MQSC

Il server di comandi *deve* essere in esecuzione sul Queue Manager di destinazione, se sta per elaborare i comandi MQSC in remoto (ciò non è necessario sul Queue Manager di origine).

- Sul Queue Manager di destinazione, digitare:

```
strmqcsv target.queue.manager
```

- Sul Queue Manager di origine, è possibile eseguire MQSC interattivamente in modalità accodamento digitando:

```
runmqsc -w 30 target.queue.manager
```

Questo formato del comando **runmqsc**, con l'indicatore **-w**, esegue i comandi MQSC in modalità accodamento, in cui i comandi vengono inseriti (in formato modificato) nella coda di input del server di comandi ed eseguiti in ordine.

Quando si digita un comando MQSC, questo viene reindirizzato al Queue Manager remoto, in questo caso `target.queue.manager`. Il timeout è impostato su 30 secondi; se non viene ricevuta alcuna risposta entro 30 secondi, sarà generato il seguente messaggio sul Queue Manager locale (di origine):

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

Al termine della sessione di MQSC, il Queue Manager locale visualizza le eventuali risposte scadute giunte. Quando la sessione di MQSC è terminata, le eventuali ulteriori risposte vengono eliminate.

In modalità accodamento, è inoltre possibile eseguire un file di comandi MQSC su un Queue Manager remoto. Ad esempio:

Gestione di un Queue Manager remoto

```
runmqsc /IN mycmds, OUT report/ -w 60 target.queue.manager
```

dove mycmds è un file contenente comandi MQSC e report è il file di notifica.

Funzionamento dei Queue Manager su MVS/ESA

È possibile emettere comandi MQSC su un Queue Manager MVS/ESA da un Queue Manager di MQSeries for Compaq NSK. Tuttavia, a tale scopo, è necessario modificare il comando **runmqsc** e le definizioni di canale al mittente.

In particolare, si aggiunge l'indicatore -x al comando **runmqsc** su un nodo di Compaq NSK:

```
runmqsc -w 30 -x QMRI
```

La definizione di canale è la seguente:

```
* Define the sender channel at the source queue manager on Compaq NSK
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ (QMRI) +
  TRPTYPE(TCP) +
```

È necessario definire il canale ricevente e la coda di trasmissione presso il Queue Manager di origine come in precedenza. Ancora una volta, quest'esempio presuppone che il protocollo di trasmissione utilizzato sia TCP/IP.

Consigli per l'accodamento remoto

Quando si implementa l'accodamento remoto:

1. Inserire i comandi MQSC da eseguire sul sistema remoto in un file di comandi.
2. Verificare localmente i comandi MQSC, specificando il flag -v sul comando **runmqsc**.
Non sarà possibile utilizzare **runmqsc** per verificare i comandi MQSC su un altro Queue Manager.
3. Verificare, quando possibile, che il file di comandi sia eseguito localmente senza errori.
4. Infine, eseguire il file di comandi sul sistema remoto.

Eventuali problemi nell'utilizzo remoto di MQSC

Se si incontrano difficoltà nell'esecuzione remota di comandi MQSC, avvalersi del seguente elenco di controllo per verificare di aver:

- Avviato il server di comandi sul Queue Manager di destinazione.
- Definito una coda di trasmissione valida.
- Definito le due estremità dei canali di messaggio per:
 - Il canale assieme al quale si inviano i comandi.
 - Il canale assieme al quale si restituiscono le risposte.
- Specificato il nome di connessione corretto (CONNAME) nella definizione di canale.

Problemi nell'utilizzo remoto di MQSC

- Avviato i listener prima di aver avviato i canali di messaggio.
- Verificato che l'intervallo di disconnessione non sia scaduto, ad esempio, se un canale è stato avviato, ma arrestato dopo qualche tempo. Ciò è particolarmente importante se si avviano i canali manualmente.
- Controllato che non si stiano inviando richieste da un Queue Manager di origine che il Queue Manager di destinazione non può interpretare (ad esempio, richieste che includano nuovi parametri).

Consultare anche la sezione "Risoluzione dei problemi con MQSC" a pagina 105.

Creazione di una definizione locale di una coda remota

È possibile utilizzare una definizione di coda remota come definizione locale di una coda remota. Creare un oggetto coda remota sul Queue Manager locale per identificare una coda locale di un altro Queue Manager.

Analisi del funzionamento delle definizioni locali di code remote

Un'applicazione si connette a un Queue Manager locale e quindi emette una chiamata MQOPEN. Nella chiamata aperta, il nome di coda specificato è quello di una definizione di coda remota sul Queue Manager locale. La definizione di coda remota fornisce i nomi della coda di destinazione, il Queue Manager di destinazione e, facoltativamente, una coda di trasmissione. Per inserire un messaggio sulla coda remota, l'applicazione emette una chiamata MQPUT, specificando l'handle restituito dalla chiamata MQOPEN. Il Queue Manager aggiunge il nome di coda remota e il nome del Queue Manager remoto a un'intestazione di trasmissione nel messaggio. Queste informazioni sono utilizzate per indirizzare il messaggio alla destinazione corretta nella rete.

Come amministratore, l'utente può controllare la destinazione del messaggio modificando la definizione della coda remota.

Esempio

È necessario che un'applicazione inserisca un messaggio in una coda appartenente a un Queue Manager remoto.

Come funziona

L'applicazione si connette a un Queue Manager, ad esempio, saturn.queue.manager. La coda di destinazione appartiene a un altro Queue Manager.

Creazione di una definizione locale di una coda remota

Sulla chiamata MQOPEN, l'applicazione specifica questi campi:

Valore campo	Descrizione
<i>ObjectName</i> CYAN.REMOTE.QUEUE	Specifica il nome locale dell'oggetto coda remota. Ciò definisce la coda di destinazione e il Queue Manager di destinazione.
<i>ObjectType</i> (Coda)	Identifica quest'oggetto come una coda.
<i>ObjectQmgrName</i> Vuoto o saturn.queue.manager	Questo campo è facoltativo. Se lasciato vuoto, il nome del Queue Manager locale viene presupposto (cioè il Queue Manager su cui è stata effettuata la definizione di coda remota e al quale l'applicazione è connessa). Se non è vuoto, è necessario specificare il nome del Queue Manager locale.

Al termine, l'applicazione emette una chiamata MQPUT per inserire un messaggio su questa coda.

Sul Queue Manager locale, è possibile creare una definizione locale di una coda remota mediante i seguenti comandi MQSC:

```
DEFINE QREMOTE ('CYAN.REMOTE.QUEUE') +
  DESCR ('Queue for auto insurance requests from the branches') +
  RNAME ('AUTOMOBILE.INSURANCE.QUOTE.QUEUE') +
  RQMNAME ('jupiter.queue.manager') +
  XMITQ ('INQUOTE.XMIT.QUEUE')
```

Nell'esempio:

QREMOTE ('CYAN.REMOTE.QUEUE')

Indica il nome locale dell'oggetto coda remota. Questo è il nome che le applicazioni connesse a questo Queue Manager deve specificare nella chiamata MQOPEN per aprire la coda AUTOMOBILE.INSURANCE.QUOTE.QUEUE sul Queue Manager remoto jupiter.queue.manager.

DESCR ('Queue for auto insurance requests from the branches')

Testo aggiuntivo che descrive l'utilizzo della coda.

RNAME ('AUTOMOBILE.INSURANCE.QUOTE.QUEUE')

È il nome della coda di destinazione sul Queue Manager remoto. Questa è la coda di destinazione effettiva per i messaggi che vengono inviati dalle applicazioni che specificano il nome coda 'CYAN.REMOTE.QUEUE'. La coda 'AUTOMOBILE.INSURANCE.QUOTE.QUEUE' deve essere definita come una coda locale sul Queue Manager remoto.

RQMNAME ('jupiter.queue.manager')

È il nome del Queue Manager remoto a cui appartiene la coda di destinazione 'AUTOMOBILE.INSURANCE.QUOTE.QUEUE'

XMITQ ('INQUOTE.XMIT.QUEUE')

È il nome della coda di trasmissione. Questo è facoltativo; se il nome non è specificato, viene utilizzata una coda con lo stesso nome del Queue Manager remoto.

Creazione di una definizione locale di una coda remota

In entrambi i casi, occorre definire la coda di trasmissione appropriata come coda locale con un attributo *Usage* specificando che si tratta di una coda di trasmissione (USAGE(XMITQ) in MQSC).

Un modo alternativo per inserire messaggi in una coda remota

Utilizzare una definizione locale di una coda remota non è l'unico modo di inserire messaggi su una coda remota. Le applicazioni possono specificare il nome di coda completo, che include il nome del Queue Manager remoto, come parte della chiamata MQOPEN. In tal caso, non è necessaria una definizione locale di coda remota. Tuttavia, tale alternativa significa che le applicazioni devono conoscere o avere accesso al nome del Queue Manager remoto al runtime.

Utilizzo di altri comandi con code remote

È possibile utilizzare i comandi MQSC appropriati per visualizzare o modificare gli attributi di un oggetto coda remota o eliminare l'oggetto coda remota. Ad esempio:

```
* Display the remote queue's attributes.
* ALL = Display all attributes

DISPLAY QUEUE (CYAN.REMOTE.QUEUE) ALL

* ALTER the remote queue to enable puts.
* This does not affect the destination queue,
* only applications that specify this remote queue.

ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)

* Delete this remote queue
* This does not affect the destination queue
* only its local definition

DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

Nota: eliminando una coda remota, si elimina solo la rappresentazione locale della coda remota. Non eliminare la coda remota stessa o eventuali messaggi a essa appartenenti.

Creazione di una coda di trasmissione

Una coda di trasmissione è una coda locale che viene utilizzata quando un Queue Manager inoltra messaggi a un Queue Manager remoto tramite un canale di messaggi. Il canale fornisce un collegamento a senso unico al Queue Manager remoto. I messaggi vengono accodati nella coda di trasmissione finché il canale sarà in grado di accettarli. Nel definire un canale, è necessario specificare un nome di coda di trasmissione all'estremità mittente del canale di messaggi.

L'attributo *Usage* (USAGE in MQSC) definisce se una coda è una coda di trasmissione o una coda normale.

Code di trasmissione predefinite

Facoltativamente, è possibile specificare una coda di trasmissione in un oggetto coda remota, mediante l'attributo *XmitQName* (XMITQ in MQSC). Se non viene

Creazione di una coda di trasmissione

definita alcuna coda di trasmissione, sarà utilizzata una coda predefinita. Quando le applicazioni inseriscono messaggi su una coda remota, se esiste una coda di trasmissione con lo stesso nome della coda di destinazione, sarà utilizzata questa coda. Se questa coda non esistesse, sarà utilizzata la coda specificata dall'attributo *DefaultXmitQ* (DEFXMITQ in MQSC) sul Queue Manager locale.

Ad esempio, il seguente comando MQSC crea una coda di trasmissione predefinita su `source.queue.manager` per messaggi inviati a `target.queue.manager`:

```
DEFINE QLOCAL ('target.queue.manager') +
  DESCR ('Default transmission queue for target qm') +
  USAGE (XMITQ)
```

Le applicazioni possono inserire messaggi direttamente in una coda di trasmissione oppure è possibile inserirli direttamente, ad esempio, tramite una definizione di coda remota. Consultare anche la sezione “Creazione di una definizione locale di una coda remota” a pagina 133.

Utilizzo di definizioni di coda remota come alias

Oltre a individuare una coda su un altro Queue Manager, è anche possibile utilizzare una definizione locale di coda remota per entrambi:

- Alias di Queue Manager
- Alias di coda di risposta RTQ

Entrambi i tipi di alias vengono risolti tramite la definizione locale di una coda remota.

Come di norma nell'accodamento remoto, occorre impostare i canali appropriati se il messaggio deve giungere alla relativa destinazione.

Alias di Queue Manager

Un alias è il processo per il quale un nome del Queue Manager di destinazione come specificato in un messaggio è modificato da un Queue Manager sulla route del messaggio. Le alias di Queue Manager sono importanti perché è possibile utilizzarle per controllare la destinazione di messaggi all'interno di una rete di Queue Manager.

A tale scopo, modificare la definizione di coda remota sul Queue Manager al punto di controllo. L'applicazione mittente non è consapevole del fatto che il nome del Queue Manager specificato sia un alias.

Per ulteriori informazioni sugli alias del Queue Manager, consultare il manuale *MQSeries Intercommunication*.

Alias di coda di risposta RTQ

Facoltativamente, un'applicazione può specificare il nome di una coda di risposta RTQ quando inserisce un *messaggio di richiesta* su una coda. Se l'applicazione che elabora il messaggio estrae il nome della coda di risposta RTQ, sa dove inviare il *messaggio di risposta*, se necessario.

Un alias di coda di risposta RTQ è il processo per il quale una coda di risposta RTQ—come specificato in un messaggio di richiesta—è modificato da un Queue

Manager sulla route del messaggio. L'applicazione mittente non è consapevole del fatto che il nome della coda di risposta RTQ specificato sia un'alias.

Un'alias di una coda di risposta RTQ consente di modificare il nome della coda di risposta e facoltativamente il relativo Queue Manager. A sua volta, questo consente di controllare la route da utilizzare per i messaggi di risposta.

Per ulteriori informazioni su messaggi di richiesta, messaggi di risposta e code di risposta, consultare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*. Per ulteriori informazioni sugli alias delle code di risposta RTQ, consultare il manuale *MQSeries Intercommunication*.

Conversione dei dati

Il Queue Manager può convertire i dati dei messaggi in formati definiti da MQSeries (noti anche come formati incorporati) da un insieme di caratteri codificati a un altro, a condizione che entrambi gli insiemi di caratteri si riferiscano a un'unica lingua o a un gruppo di lingue simili.

Ad esempio, è supportata la conversione tra insiemi di caratteri codificati i cui identificativi (CCSID) sono 850 e 500, poiché entrambi si applicano alle lingue europee occidentali.

Per conversioni di caratteri NL (New Line) EBCDIC in ASCII, consultare la sezione "Contenuto del file di configurazione MQSeries" a pagina 187.

Le conversioni supportate sono definite nell'Appendice F. "Tabelle di conversione code page" nel manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

Quando un Queue Manager non è in grado di convertire messaggi in formati incorporati

Il Queue Manager non è in grado di convertire automaticamente i messaggi in formati incorporati se i relativi CCSID rappresentano diversi gruppi di lingue nazionali. Ad esempio, la conversione tra CCSID 850 e CCSID 1025 (che è un insieme di caratteri codificati EBCDIC per lingue che utilizzano script in cirillico) non è supportata perché molti dei caratteri in un insieme di caratteri codificati non possono essere rappresentati nell'altro. Se si dispone di una rete di Queue Manager funzionanti in diverse lingue nazionali e non è supportata la conversione di dati tra alcuni insiemi di caratteri codificati, sarà possibile abilitare una conversione predefinita. La conversione predefinita dei dati è descritta nella sezione "Conversione di dati predefinita".

File CCSID

Il CCSID del file specifica eventuali gruppi supplementari di codice e conversioni di dati predefiniti. È possibile aggiornare le informazioni registrate nel file CCSID; ciò potrebbe essere opportuno se, ad esempio, un futuro rilascio del proprio sistema operativo supportasse insiemi di caratteri codificati aggiuntivi. Per specificare insiemi di codici aggiuntivi, occorre modificare il file CCSID. A tale scopo, il file fornisce adeguate istruzioni.

Conversione di dati predefinita

Per implementare la conversione di dati predefinita, modificare il file CCSID per specificare un CCSID EBCDIC e un CCSID ASCII predefiniti e inoltre per specificare i CCSID predefiniti. Le istruzioni a tale scopo sono incluse nel file.

Alias

Se si aggiorna il file CCSID per implementare la conversione di dati predefinita, Queue Manager dovrà essere riavviato prima che le modifiche divengano effettive.

Il processo di conversione di dati predefinito è il seguente:

- Se la conversione tra i CCSID di origine e di destinazione non è supportata, ma i CCSID degli ambienti di origine e di destinazione sono entrambi EBCDIC o entrambi ASCII, i dati dei caratteri vengono spostati all'applicazione di destinazione senza conversione.
- Se un CCSID rappresenta un insieme di caratteri codificati ASCII e l'altro rappresenta un insieme di caratteri codificati EBCDIC, MQSeries convertirà i dati mediante i CCSID di conversione di dati predefinita indicati nel file CCSID.

Nota: l'utente dovrebbe provare a limitare i caratteri che vengono convertiti a quelli caratterizzati dagli stessi valori di codici nell'insieme di caratteri codificati specificato per il messaggio e nell'insieme di caratteri codificati predefinito. Se si utilizza solo l'insieme di caratteri valido per i nomi oggetto di MQSeries tale requisito sarà generalmente soddisfatto. Alcune eccezioni si verificano con i CCSID EBCDIC 290, 930, 1279 e 5026 utilizzati in Giappone, dove i caratteri minuscoli hanno codici diversi rispetto a quelli utilizzati in altri CCSID EBCDIC.

Conversione di messaggi in formati definiti dall'utente

Non è possibile convertire messaggi in formati definiti dall'utente da un insieme di caratteri codificati dal Queue Manager. Se i dati in un formato definito dall'utente richiedono la conversione, occorre fornire un'uscita della conversione dei dati per ciascun formato simile. L'utilizzo dei CCSID predefiniti per la conversione di dati di caratteri in formati definiti dall'utente è sconsigliato, sebbene sia possibile. Per ulteriori informazioni sulla conversione di dati in formati definiti dall'utente e sulla scrittura di uscite di conversione di dati, consultare il manuale *MQSeries Application Programming Guide*.

Modifica del CCSID del Queue Manager

Si consiglia di arrestare e riavviare il Queue Manager quando si modifica il CCSID del Queue Manager, utilizzando l'attributo CCSID del comando ALTER QMGR.

Ciò garantisce che tutte le applicazioni in esecuzione, inclusi il server di comandi e i programmi di canale, siano arrestate e riavviate.

Ciò è necessario, poiché qualsiasi applicazione, che sia in esecuzione quando viene modificato il CCSID del Queue Manager, continua a utilizzare il CCSID esistente.

Capitolo 8. Protezione di oggetti di MQSeries

Questo capitolo spiega le funzioni di controllo della sicurezza in MQSeries for Compaq NSK e ne descrive le modalità di implementazione.

Questo capitolo contiene tre sezioni:

- "ID utenti nel gruppo utenti MQM"
- "Perché bisogna proteggere le risorse di MQSeries"
- "Object Authority Manager (OAM)" a pagina 140
- "Utilizzo dei comandi OAM (Object Authority Manager)" a pagina 144
- "Autorizzazioni di accesso" a pagina 146
- "Comando di visualizzazione autorizzazioni" a pagina 147
- "Istruzioni per OAM (Object Authority Manager)" a pagina 147
- "Comprendere le tabelle di specifica delle autorizzazioni" a pagina 150
- "Comprendere i file di autorizzazione" a pagina 156

Perché bisogna proteggere le risorse di MQSeries

Poiché i Queue Manager di MQSeries gestiscono il trasferimento delle informazioni potenzialmente preziose, occorre salvarle mediante un sistema di autorizzazioni. Questa procedura garantisce che le risorse appartenenti a un Queue Manager e da questo gestite siano protette dall'accesso non autorizzato, che potrebbe condurre alla perdita o alla divulgazione delle informazioni. In un sistema protetto, è fondamentale che nessun utente o applicazione non autorizzati possano accedere a o modificare quanto segue:

- Connessioni a un Queue Manager.
- Accesso a oggetti di MQSeries, quali code, cluster, canali e processi.
- Comandi di gestione del Queue Manager, inclusi i comandi PCF e MQSC.
- Accesso a messaggi di MQSeries.
- Informazioni contestuali associate ai messaggi.

Occorre sviluppare i propri criteri per quanto riguarda i particolari utenti che accederanno a determinate risorse.

ID utenti nel gruppo utenti MQM

Tutte le risorse di queue manager vengono eseguite con l'ID di gruppo MQM.

Per poter accedere a MQSeries for Compaq NSK, l'ID dell'utente deve corrispondere ad un principal MQSeries. Inizialmente, solo l'ID utente che ha creato il queue manager ha il principal MQSeries mqm. Bisogna utilizzare il comando **altmqusr** per creare un principal per ciascun utente che accederà a MQSeries. Il principal e l'utente devono essere unici. Per visualizzare i principal e le loro proprietà, utilizzare il comando **dspmqusr**.

Se l'ID utente appartiene al gruppo MQM di MQSeries for Compaq NSK ed è stato creato per esso un principal, si dispone di tutte le autorizzazioni a tutte le risorse MQSeries. L'ID utente *deve* appartenere al gruppo MQM per poter utilizzare tutti i comandi di controllo MQSeries for Compaq NSK (tranne **crtmqcvx**). In particolare, questa autorizzazione è necessaria per:

- Utilizzare il programma di utilità **runmqsc** per eseguire i comandi MQSC.

ID utenti

- Gestire le autorizzazioni su MQSeries for Compaq NSK mediante il comando **setmqaut**.

Se si stanno inviando comandi di canale ai queue manager su un sistema Compaq NSK remoto, bisogna verificare che l'ID utente sia un membro del gruppo MQM di Compaq NSK sul sistema di destinazione. Per un elenco di comandi PCF e comandi di canale MQSC, consultare la sezione "Sicurezza del comando di canale" a pagina 150.

L'ID utente deve appartenere al gruppo MQM per poter emettere:

- comandi PCF, inclusi i PCF Escape, da un programma di gestione
- chiamate MQI da un programma applicativo

Nota: Le autorizzazioni per il principal mqm (ed i gruppi MQM) sono importanti poiché il principal mqm viene utilizzato dai componenti interni di queue manager stessi per accedere alle risorse protette. Se si rimuovono le autorizzazioni per il principal mqm o per il gruppo MQM dagli oggetti nel queue manager, oppure se si rimuove il principal mqm stesso, si potrebbe creare una situazione in cui non è più possibile gestire il queue manager o, nel peggiore dei casi, non è più possibile utilizzarlo.

Come ottenere ulteriori informazioni

Per ulteriori informazioni su:

- gruppi di comandi MQSeries for Compaq NSK, consultare: "Capitolo 3. Utilizzo dei gruppi di comandi di MQSeries" a pagina 29
- comandi di controllo MQSeries for Compaq NSK, consultare: "Capitolo 17. I comandi di controllo di MQSeries" a pagina 251
- Comandi PCF e Escape PCF, consultare il manuale *MQSeries Programmable System Management*.
- Chiamate MQI, consultare: *MQSeries Application Programming Guide e: MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*

Object Authority Manager (OAM)

Per impostazione predefinita, l'accesso alle risorse del Queue Manager viene controllato tramite un componente installabile del servizio di autorizzazione. Il componente di servizio di autorizzazioni fornito con MQSeries for Compaq NSK è denominato OAM e viene automaticamente installato ed abilitato per ciascun queue manager creato, se non diversamente specificato. In questo capitolo, il termine OAM è adoperato per denotare Object Authority Manager fornito con questo prodotto.

OAM è un *componente installabile* del servizio di autorizzazione. La possibilità di installare il componente OAM offre la flessibilità di:

- Sostituire OAM con il proprio componente di servizio di autorizzazione mediante l'interfaccia fornita.
- Aumentare le funzioni fornite da OAM con quelle del proprio componente di servizio di autorizzazione, usando sempre l'interfaccia fornita.
- Rimuovere o disabilitare OAM ed eseguire il programma senza alcun servizio di autorizzazione.

Per ulteriori informazioni sui servizi installabili, consultare il manuale *MQSeries Programmable System Management*.

OAM gestisce le autorizzazioni degli utenti per la manipolazione di oggetti MQSeries, inclusi code, definizioni di processo e canali. Esso fornisce inoltre un'interfaccia di comando tramite la quale è possibile concedere o revocare le autorizzazioni di accesso a un oggetto per uno specifico gruppo di utenti. OAM decide di concedere l'accesso a una risorsa e il Queue Manager si attiene a tale decisione. Se OAM non è in grado di prendere decisioni, il Queue Manager impedisce l'accesso a tale risorsa.

Come funziona OAM

L'OAM utilizza gli ID utente e di gruppo e le funzioni di sicurezza del sistema operativo Compaq NSK. Gli utenti possono accedere agli oggetti del Queue Manager solo se dispongono dell'autorizzazione richiesta.

Gestione dell'accesso mediante i gruppi utente

La gestione dei permessi di accesso per le risorse MQSeries è basata sui *gruppi* Compaq NSK. L'OAM mantiene le autorizzazioni a livello di gruppo.

Nelle interfacce comandi, invece degli ID utente si utilizzano i principal MQSeries. Ciò è dovuto al fatto che le autorizzazioni concesse ad un ID utente possono essere state concesse anche ad altre entità. Ad esempio, è possibile che delle autorizzazioni siano state concesse ad un programma applicativo che emette chiamate MQI oppure ad un programma di gestione che emette comandi PCF. In questi casi, il principal associato al programma non è necessariamente l'ID utente utilizzato quando è stato avviato il programma.

Gli ID utente Compaq NSK hanno il formato <gruppo>.<nome> dove sia il gruppo che il nome possono avere ciascuno una lunghezza massima di 8 caratteri, mentre i nomi di principal MQSeries possono avere una lunghezza massima di 12 caratteri. Inoltre, il carattere punto (.) è illegale negli ID utenti su alcune altre piattaforme. In MQSeries for Compaq NSK, il database dei principal contiene la messa in corrispondenza degli ID utente Compaq NSK e dei nomi principal MQSeries di una lunghezza pari o inferiore ai 12 caratteri.

Quando un utente appartiene a più di un gruppo utenti

L'autorizzazione di cui un utente dispone è il risultato dell'insieme delle autorizzazioni di tutti i gruppi cui appartiene e delle autorizzazioni predefinite per tutti gli utenti. E' possibile utilizzare il comando di controllo **setmqaut** per impostare le autorizzazioni per un gruppo specifico.

Nota: qualsiasi modifica apportata mediante il comando **setmqaut** implica un effetto immediato, a meno che l'oggetto non sia in uso. In questo caso, le modifiche diventano effettive quando l'oggetto viene nuovamente aperto.

Set di gruppi ed il gruppo principale

La gestione dei permessi di accesso alle risorse MQSeries è basata sui gruppi utenti Compaq NSK. Quando è in esecuzione SAFEGUARD, è possibile associare un ID utente Compaq NSK a più di un gruppo e, pertanto, il principal MQSeries corrispondente viene anch'esso associato a questi gruppi. Il gruppo principale è sempre il gruppo Compaq Administrative. I gruppi secondari sono configurati creando dei gruppi di condivisione file SAFEGUARD (File Sharing Group) ed associandovi un ID utente Compaq NSK.

L'OAM mantiene le autorizzazioni al livello dei gruppi piuttosto che dei singoli principal. La messa in corrispondenza dei principal con i nomi di gruppo viene eseguita nell'OAM utilizzando il database dei principal e le funzioni NSK e

Object authority manager

SAFEGUARD; le operazioni OAM vengono eseguite al livello dei gruppi. È possibile, tuttavia, visualizzare le autorizzazioni di un principal individuale.

Protezione delle risorse con l'OAM

Tramite OAM è possibile controllare:

- L'accesso a oggetti di MQSeries tramite MQI. Quando un programma applicativo tenta di accedere a un oggetto, OAM verifica se l'ID utente che sta effettuando la richiesta disponga dell'autorizzazione necessaria (tramite il relativo gruppo utenti) per l'operazione richiesta.
In particolare, ciò significa che le code e i messaggi in coda possono essere protetti dall'accesso non autorizzato.
- Il permesso di utilizzare i comandi MQSC; solo i membri del gruppo utenti mmq, o quelli autorizzati tramite **setmqaut**, possono eseguire i comandi di gestione del queue manager, ad esempio per la creazione di una coda.
- Il permesso di utilizzare i comandi di controllo; solo i membri del gruppo utenti mqmq possono eseguire i comandi di controllo, ad esempio la creazione di un queue manager o l'avvio di un command server.
- Il permesso di utilizzare comandi PCF.

È possibile concedere a diversi gruppi di utenti tipi differenti di autorizzazione di accesso allo stesso oggetto. Ad esempio, per una coda specifica, è possibile che un gruppo disponga dell'autorizzazione per eseguire sia operazioni di inserimento (put) che di estrazione (get), mentre un altro gruppo può disporre dell'autorizzazione per poter solo leggere la coda (MQGET con l'opzione browse). In modo analogo, è possibile che dei gruppi dispongano dell'autorizzazione per eseguire operazioni di inserimento e di estrazione ma non dell'autorizzazione richiesta per eseguire operazioni di modifica o cancellazione della coda.

Utilizzo dei gruppi per le autorizzazioni

L'utilizzo dei gruppi, piuttosto che dei principal individuali, per l'autorizzazione riduce la quantità di gestione richiesta. Di norma, è richiesto un tipo di accesso particolare da più principal. Ad esempio, si potrebbe definire un gruppo composto da utenti finali che desiderano eseguire un'applicazione particolare. I nuovi utenti potrebbero ottenere l'accesso aggiungendo il gruppo appropriato al proprio ID utente di Compaq NSK. Tranne nel caso in cui MQSeries sia installato su un sistema che utilizza SAFEGUARD per creare gruppi di condivisione dati, ciascun ID utente può essere associato solo ad un singolo gruppo principale.

Senza SAFEGUARD, esiste un limite di 255 principal per gruppo. Utilizzando la condivisione di file di SAFEGUARD, il limite è di 65535 principal per gruppo. L'utilizzo della condivisione di file SAFEGUARD consente inoltre ad un principal di disporre di autorizzazioni proprie. Ad esempio, se il gruppo Administrative di SAFEGUARD non dispone di alcuna autorizzazione, ciascun membro delle condivisione di file appartenente al gruppo può disporre di autorizzazioni proprie (Principal A, gruppo MQSEC membro TESTSEC1 ha solo l'autorizzazione PUT, Principal B gruppo MQSEC membro TESTSEC2 ha solo l'autorizzazione GET).

Senza SAFEGUARD, bisognerebbe creare due gruppi (due id utente Compaq NSK in gruppi diversi, ad esempio MQ.TEST (connect/put) e MQ1.TEST (connect/get)) per avere lo stesso risultato poiché le autorizzazioni per il gruppo vengono combinate con il principal (id utente Compaq NSK MQM.MANAGER gruppo MQM (connect) principal mqm e id utente Compaq NSK MQM.APPS gruppo MQM principal apps (get/put)). Il set di autorizzazioni per il gruppo (connect per mqm) e quello del principal (get/put per apps) vengono aggiunti l'uno all'altro

Object authority manager

(connect/get/put). La convalida delle autorizzazioni utente viene quindi eseguita sui dati combinati. Se un'applicazione tenta un'operazione MQGET mentre è collegata utilizzando l'ID utente MQ.TEST, l'operazione viene respinta con un codice di autorizzazione insufficiente MQRC_NOT_AUTHORIZED (2035). Se l'applicazione è collegata come MQ1.TEST, l'operazione ha esito positivo. Un utente collegato come un ID utente Compaq nel gruppo MQM può eseguire questa operazione con esito positivo. Un'autorizzazione concessa ad un principal viene concessa anche a tutti i principal che fanno parte del gruppo.

Principal Sicurezza	id utente Compaq	Gruppo	Tipo gruppo
mqm	MQM.MANAGER	MQM	connect
apps	MQM.APPS	MQM	get/put
inq	MQM.INQ	MQM	inq
mq	MQ.TEST	MQ	connect/put
mq1	MQ1.TEST	MQ1	connect/get
A	MQSEC.FRED	MQSEC	safeguard admin none
		TESTSEC1	safeguard file sharing
connect/put			
B	MQSEC.JOE	MQSEC	safeguard admin none
		TESTSEC2	safeguard file sharing
connect/get			
C	MQSEC.FRANK	MQSEC	safeguard admin none
		TESTSEC3	safeguard file sharing
none			

Figura 39. Utilizzo dei gruppi con SAFEGUARD per la concessione di autorizzazioni

In Figura 39 un utente collegato per il gruppo MQM dispone delle autorizzazioni connect, get, put e inq. L'utente MQ.TEST dispone delle autorizzazioni connect e put e l'utente MQ1.TEST delle autorizzazioni connect e get. Gli utenti nel gruppo dispongono delle autorizzazioni del membro safeguard file sharing concesse.

Provare a tenere più basso possibile il numero di gruppi. Ad esempio, è possibile dividere gli utenti in un gruppo per gli utenti delle applicazioni ed in un gruppo per gli amministratori.

Disabilitazione dell'OAM (Object Authority Manager)

Per impostazione predefinita, OAM è abilitato. E' possibile disabilitare l'OAM impostando la variabile di ambiente Compaq NSK MQSNOAUT prima di eseguire la creazione del queue manager nel seguente modo:

```
PARAM MQSNOAUT 1
```

Tuttavia, se si disabilita l'OAM per un queue manager, non è poi possibile riavviarlo successivamente. Si consiglia di mantenere OAM abilitato e assicurarsi che tutti gli utenti e le applicazioni dispongano dell'accesso tramite un ID utente appropriato. E' anche possibile disabilitare l'OAM a scopo di prova rimuovendo la stanza del servizio di autorizzazione (authorization service) nel file di configurazione del queue manager (QMINI) oppure disabilitando MQAUTH nella stanza Authority di QMINI, come descritto in: "File di configurazione Queue manager (QMINI)" a pagina 189.

Nota: Specificare PARAM MQSNOAUT 0 non abilita l'OAM. La variabile di ambiente non deve esistere nell'ambiente se bisogna riabilitare OAM.

Utilizzo dei comandi OAM (Object Authority Manager)

OAM fornisce un'interfaccia di comandi per concedere e revocare le autorizzazioni. Prima di poter utilizzare questi comandi bisogna essere autorizzati—il proprio ID utente deve appartenere al gruppo MQM di Compaq NSK. E' necessario che questo gruppo fosse già impostato prima dell'esecuzione dell'installazione di MQSeries for Compaq NSK.)

Se il proprio ID utente è un membro del gruppo MQM, si dispone di un'autorizzazione di 'super user' per il queue manager. Si è adesso autorizzati ad emettere qualsiasi richiesta MQI o comando di controllo dal proprio ID utente.

L'OAM fornisce quattro comandi che è possibile richiamare dal ACL per gestire le autorizzazioni degli utenti. Sono:

- **altmqusr** (Creazione, rimozione o modifica di un principal MQSeries)
- **dspmqusr** (Visualizzazione principal)
- **setmqaut** (Impostazione o reimpostazione autorizzazioni)
- **dspmqaut** (Visualizzazione autorizzazioni)

La verifica dell'autorizzazione di verifica nelle seguenti chiamate: MQCONN, MQOPEN, MQPUT1 e MQCLOSE. La verifica dell'autorizzazione viene eseguita solo alla prima istanza di una qualsiasi chiamata e l'autorizzazione non viene corretta finché l'oggetto non viene reimpostato (ovvero, chiuso e riaperto). Di conseguenza, qualsiasi modifica apportata all'autorizzazione di un oggetto che utilizza **setmqaut** non diverrà effettiva finché non si reimposta l'oggetto.

Cosa specificare quando si utilizzano i comandi OAM

I comandi OAM si applicano al Queue Manager specificato; se non si specifica un Queue Manager, sarà utilizzato il Queue Manager predefinito. Su tali comandi, è necessario specificare unicamente l'oggetto, cioè il nome e il tipo dell'oggetto. Occorre inoltre specificare il nome di utente o gruppo cui si applica l'autorizzazione.

Elenchi di autorizzazioni

Specificare un elenco di autorizzazioni con il comando **setmqaut**. Questo è un metodo rapido per specificare se un'autorizzazione deve essere concessa o revocata e a quali risorse si applica l'autorizzazione. Ciascuna autorizzazione nell'elenco è specificata come una parola chiave in caratteri minuscoli, preceduta da un segno più (+) o meno (-). E' possibile utilizzare un segno più per aggiungere l'autorizzazione specificata oppure un segno meno per rimuoverla. È possibile specificare qualsiasi numero di autorizzazioni in un unico comando. Ad esempio:

```
+browse -get +put
```

Utilizzo del comando **altmqusr**

Se si dispone dell'autorizzazione necessaria, è possibile utilizzare il comando **altmqusr** per creare un principal MQSeries ed associarlo ad un ID utente Compaq NSK (o alias SAFEGUARD). Il seguente esempio mostra come viene utilizzato il comando **altmqusr**:

```
altmqusr -m saturn.queue.manager -p MQPRINCIPAL -u MQM.MQUSER
```

In quest'esempio:

Questo termine...	Specifica...
saturn.queue.manager	Il nome del Queue Manager
MQPRINCIPAL	Il nome principal da creare
MQM.MQUSER	L'ID utente Compaq NSK

Per una descrizione di questo comando, consultare: "altmqsr (Modifica informazioni utente MQSeries)" a pagina 259.

E' anche possibile utilizzare il comando **altmqsr** per rimuovere un principal (e revocare pertanto tutti i diritti di accesso a MQSeries). Ad esempio:

```
altmqsr -m saturn.queue.manager -p MQPRINCIPAL -remove
```

In quest'esempio:

Questo termine...	Specifica...
saturn.queue.manager	Il nome del Queue Manager
MQPRINCIPAL	Il nome principal da rimuovere
-remove	L'istruzione di cancellare il principal

Utilizzo del comando dspmqsr

E' possibile visualizzare il contenuto del database dei principal, oltre ai gruppi di condivisione file (file sharing) e di gestione (administrative) di Compaq NSK cui appartiene l'ID utente corrispondente a ciascun principal MQSeries, utilizzando il comando **dspmqsr**. Il parametro -p limita le informazioni visualizzate al principal specificato. Ad esempio:

```
dspmqsr -m saturn.queue.manager -p MQPRINCIPAL
```

In quest'esempio:

Questo termine...	Specifica...
saturn.queue.manager	Il nome del Queue Manager
MQPRINCIPAL	Il nome principal da visualizzare

Per una descrizione di questo comando, consultare: "dspmqsr (visualizza informazioni utente MQSeries)" a pagina 283.

Utilizzo del comando setmqaut

Se si dispone dell'autorizzazione necessaria, è possibile utilizzare il comando **setmqaut** per concedere o revocare l'autorizzazione di un principal o di un gruppo utenti ad accedere a un particolare oggetto. Il seguente esempio illustra l'utilizzo del comando **setmqaut** :

Utilizzo dei comandi di OAM

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE -g  
GroupA +browse -get +put
```

In questo esempio:

Questo termine...	Specifica...
saturn.queue.manager	Il nome del Queue Manager
coda	Il tipo di oggetto
RED.LOCAL.QUEUE	Il nome di oggetto
GroupA	L'ID del gruppo cui concedere le autorizzazioni
+browse -get +put	Elenco di autorizzazioni per la coda specificata. Non devono esserci spazi tra i segni "+" o "-" e la parola chiave.

L'elenco di autorizzazioni specifica le autorizzazioni da concedere, dove:

Questo termine...	Specifica...
+browse	Aggiunge l'autorizzazione di lettura (MQGET con l'opzione browse)
-get	Rimuove l'autorizzazione di estrazione (MQGET) dei messaggi dalla coda.
+put	Aggiunge l'autorizzazione di inserimento (MQPUT) dei messaggi nella coda.

Le applicazioni avviate con gli ID utente appartenenti al gruppo utenti Compaq NSK GroupA dispongono di queste autorizzazioni.

Il seguente comando revoca l'autorizzazione sulla coda MyQueue ai gruppi GroupA e GroupB.

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -g GroupA  
-g GroupB -put
```

Per una definizione formale del comando e della relativa sintassi, consultare la sezione "setmqaut (Imposta/reimposta l'autorizzazione)" a pagina 302.

Comandi di autorizzazione e servizi installabili

Il comando **setmqaut** assume un parametro supplementare che specifica il nome del componente di servizio di autorizzazione a cui si applica l'aggiornamento. È necessario specificare tale parametro se si dispone di più componenti di autorizzazione in esecuzione simultaneamente. Per impostazione predefinita, non risulta tale situazione. Se il parametro viene omesso, l'aggiornamento viene effettuato sul primo componente di autorizzazione rilevato, se disponibile. Il servizio predefinito è OAM, che è incluso.

Autorizzazioni di accesso

Le autorizzazioni definite dall'elenco di autorizzazioni associato al comando **setmqaut** possono essere divise in categorie come segue:

- Autorizzazioni relative alle chiamate MQI
- Autorizzazioni relative ai comandi di gestione

- Autorizzazioni contestuali
- Autorizzazioni generali, ossia, per chiamate MQI, per i comandi o per entrambi

Ciascuna autorizzazione è specificata da una parola chiave utilizzata assieme ai comandi `setmqaut` e `dspmqaut`. Questi comandi sono descritti nella sezione “`setmqaut` (Imposta/reimposta l’autorizzazione)” a pagina 302.

Comando di visualizzazione autorizzazioni

È possibile utilizzare il comando `dspmqaut` per visualizzare le autorizzazioni che uno specifico principal o gruppo per un oggetto particolare. I flag possiedono lo stesso significato di quelli nel comando `setmqaut`. È possibile visualizzare l’autorizzazione per un solo gruppo o principal per volta. Consultare la sezione “`dspmqaut` (Display Authority)” a pagina 273 per una specifica formale di questo comando.

Ad esempio, il seguente comando visualizza le autorizzazioni concesse al gruppo `GpAdmin` per accedere a una definizione di processo denominata `Annuities` sul `Queue Manager QueueMan1`.

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

Le parole chiave visualizzate come risultato di questo comando identificano le autorizzazioni che sono attive.

Istruzioni per OAM (Object Authority Manager)

Alcune operazioni sono particolarmente riservate e dovrebbero essere limitate a utenti privilegiati. Ad esempio:

- Creazione, cancellazione, avvio e arresto dei queue manager
- Accesso di determinate code speciali, quali le code di trasmissione o la coda comandi `SYSTEM.ADMIN.COMMAND.QUEUE`
- Programmi che utilizzano opzioni contestuali MQI complete
- Creazione e copia delle code applicative

ID utente

Il gruppo speciale denominato `MQM` creato è dedicato solo agli amministratori del prodotto. Non deve mai essere disponibile agli utenti non privilegiati.

Volumi del queue manager

Il volume contenente code e altri dati del `Queue Manager` è per l’uso privato del prodotto. Gli oggetti presenti in questa directory dispongono di autorizzazioni utente di `Compaq NSK` correlate alle rispettive autorizzazioni di `OAM`. Non è possibile utilizzare i comandi standard di `Compaq NSK` per concedere o revocare autorizzazioni alle risorse MQI poiché:

- Gli oggetti di `MQSeries` non sono necessariamente identici al corrispondente nome dell’oggetto di sistema. Consultare la sezione “Struttura dei volumi” a pagina 61 per ulteriori informazioni in proposito.
- Gli oggetti `MQSeries` non corrispondono necessariamente alle impostazioni di sicurezza `NSK` dell’oggetto.

Istruzioni per OAM

Code

L'autorizzazione per l'accesso a una coda dinamica si basa su quella della coda modello da cui deriva, ma non è necessariamente uguale ad essa.

Per code di alias e code remote, l'autorizzazione è quella dell'oggetto stesso, non la coda in cui l'alias o la coda remota si trasforma. Di conseguenza, è possibile autorizzare un principal ad accedere ad un alias di coda risolta in una coda locale per cui il principal non dispone dei permessi di accesso.

Occorre limitare l'autorizzazione affinché crei code per utenti privilegiati. Se non si limita questa autorizzazione, gli utenti possono aggirare il normale controllo di accesso creando un alias.

Autorizzazione per utente alternativo

L'autorizzazione dell'utente alternativo controlla se un ID utente può utilizzare l'autorizzazione di un altro ID utente durante l'accesso a un oggetto di MQSeries. Questo metodo è essenziale quando un server riceve richieste da un programma ed il server deve garantire che il programma disponga delle autorizzazioni necessarie per la richiesta. Il server potrebbe disporre dell'autorizzazione necessaria, ma deve determinare se il programma dispone dell'autorizzazione per le azioni richieste.

Ad esempio:

- Un programma del server in esecuzione sotto l'ID utente PAYSERV recupera un messaggio di richiesta da una coda che era stata inserita in coda dall'ID utente USER1.
- Quando il programma del server riceve il messaggio di richiesta, elabora la richiesta e inserisce la risposta nella coda di risposta specificata con il messaggio di richiesta.
- Invece di utilizzare il proprio ID utente (PAYSERV) per autorizzare l'apertura della coda di risposta, il server potrà specificare alcuni altri ID utente, in tal caso, USER1. In questo esempio, è possibile utilizzare l'autorizzazione utente alternativo per controllare se a PAYSERV è concesso specificare USER1 come ID utente alternativo quando apre la coda di risposta.

L'ID utente alternativo è specificato nel campo *AlternateUserId* della descrizione dell'oggetto.

Perché l'autorizzazione venga concessa, è necessario che sia l'ID utente che gli ID utente alternativi vengano specificati come principal corrispondenti a voci nel database dei principal associate ad un ID utente Compaq NSK.

Nota: è possibile utilizzare ID utente alternativi su qualsiasi oggetto di MQSeries. L'utilizzo di un ID utente alternativo non influisce sull'ID utente utilizzato da altri gestori di risorse.

Autorizzazione contestuale

Il contesto è l'informazione applicata a un messaggio particolare ed è contenuta nella descrizione del messaggio, MQMD, che fa parte del messaggio. Le informazioni contestuali si dividono in due sezioni:

Sezione Identità

Questa parte specifica il mittente del messaggio. Consiste nei seguenti campi:

- *UserIdentifier*

- *AccountingToken*
- *ApplIdentityData*

Sezione Origine

Questa sezione specifica da dove proviene il messaggio e quando è stato inserito in coda. Consiste nei seguenti campi:

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

Le applicazioni possono specificare i dati contestuali quando viene eseguita una chiamata MQOPEN o MQPUT. Questi dati possono essere generati dall'applicazione, trasferiti da un altro messaggio o generati dal Queue Manager per impostazione predefinita. Ad esempio, i dati contestuali possono essere utilizzati dai programmi del server per verificare l'identità del richiedente, controllando se il messaggio proviene da un'applicazione, eseguita sotto un ID utente autorizzato.

Un programma di server può utilizzare *UserIdentifier* per determinare l'ID utente di un utente alternativo. *UserIdentifier* deve essere specificato come un principal corrispondente ad una voce nel database dei principal.

Utilizzare l'autorizzazione contestuale per controllare se l'utente può specificare qualsiasi opzione di contesto su qualsiasi chiamata MQOPEN o MQPUT1. Per informazioni sulle opzioni di contesto, consultare il manuale *MQSeries Application Programming Guide*. Per un'illustrazione dei campi di descrizione del messaggio relativi al contesto, consultare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

Considerazioni sulla sicurezza remota

Per la sicurezza remota, occorre considerare:

Autorizzazione put

Per la sicurezza in tutti i Queue Manager è possibile specificare l'autorizzazione put utilizzata quando un canale riceve un messaggio inviato da un altro Queue Manager.

Specificare l'attributo di canale PUTAUT come segue:

DEF ID utente predefinito. L'ID utente sotto cui è in esecuzione l'agente del canale di messaggio.

CTX L'ID utente nel contesto del messaggio.

In entrambi i casi, bisogna specificare l'ID utente come un principal corrispondente ad una voce nel database dei principal.

Code di trasmissione

I queue manager inseriscono automaticamente i messaggi remoti in una coda di trasmissione; non è richiesta alcuna autorizzazione speciale. Tuttavia, l'inserimento di un messaggio direttamente in una coda di trasmissione richiede un'autorizzazione speciale; consultare la Tabella 3 a pagina 151.

Uscite di canale

Le uscite dal canale possono essere utilizzate per una maggiore sicurezza.

Istruzioni per OAM

Per ulteriori informazioni, consultare il manuale *MQSeries Intercommunication*.

Sicurezza del comando di canale

I comandi di canale possono essere emessi come comandi PCF, tramite comandi MQSC, MQAI e comandi di controllo.

Comandi PCF

È possibile emettere comandi di canale PCF inviando un messaggio PCF a SYSTEM.ADMIN.COMMAND.QUEUE su un sistema Compaq NSK remoto. L'ID utente specificato nella descrizione di messaggio (message descriptor) del messaggio PCF deve essere specificato come un principal corrispondente ad una voce nel database dei principal associato ad un ID utente Compaq NSK appartenere al gruppo mqm sul sistema di destinazione. Tali comandi sono:

- ChangeChannel
- CopyChannel
- CreateChannel
- DeleteChannel
- PingChannel
- ResetChannel
- StartChannel
- StopChannel
- ResolveChannel

Consultare il manuale *MQSeries Programmable System Management* per i requisiti di sicurezza PCF.

Comandi canale MQSC

È possibile emettere comandi di canale MQSC a un sistema remoto Compaq NSK inviando il comando direttamente in un messaggio escape PCF oppure emettendo il comando mediante **runmqsc** in modo indiretto. L'ID utente specificato nella descrizione di messaggio (message descriptor) del messaggio PCF deve essere specificato come un principal corrispondente ad una voce nel database dei principal associato ad un ID utente Compaq NSK appartenere al gruppo mqm sul sistema di destinazione. (i comandi PCF sono impliciti nei comandi MQSC emessi da **runmqsc** in modo indiretto). Tali comandi sono:

- ALTER CHANNEL
- DEFINE CHANNEL
- DELETE CHANNEL
- PING CHANNEL
- RESET CHANNEL
- START CHANNEL
- START CHINIT
- STOP CHANNEL
- RESOLVE CHANNEL

Per i comandi MQSC emessi dal comando **runmqsc**, l'ID utente nel messaggio PCF è di norma quello dell'attuale utente.

Comprendere le tabelle di specifica delle autorizzazioni

Le tabelle di specifica delle autorizzazioni che iniziano a pagina 151 definiscono in maniera precisa il funzionamento delle autorizzazioni e le restrizioni applicate. Le tabelle si applicano alle seguenti situazioni:

- Applicazioni che emettono chiamate MQI.
- Programmi di gestione che emettono comandi MQSC come PCF escape.
- Programmi di gestione che emettono comandi PCF.

Tabelle di specifica delle autorizzazioni

In questa sezione, le informazioni sono presentate come un insieme di tabelle che specificano quanto segue:

Azione da eseguire	Opzione MQI, comando MQSC o comando PCF.
Oggetto di controllo accessi	Coda, processo o Queue Manager.
Autorizzazione richiesta	Espressa come una costante 'MQZAO_'.

Nelle tabelle, le costanti precedute dal prefisso MQZAO_ corrispondono alle parole chiave nell'elenco di autorizzazioni per il comando **setmqaut** relativo alla particolare entità. Ad esempio, MQZAO_BROWSE corrisponde alla parola chiave +browse ; similmente, la parola chiave MQZAO_SET_ALL_CONTEXT corrisponde alla parola chiave +setall e così via. Queste costanti sono definite nel file di intestazione CMQZCH nel volume secondario, fornito con il prodotto.

Autorizzazioni MQI

Un'applicazione può emettere determinate chiamate e opzioni MQI solo se all'identificativo utente sotto il quale è in esecuzione (o le cui autorizzazioni è in grado di assumere) è stata concessa la relativa autorizzazione.

Quattro chiamate MQI potrebbero richiedere verifiche di autorizzazione: MQCONN, MQOPEN, MQPUT1 e MQCLOSE.

Per MQOPEN e MQPUT1, la verifica di autorizzazione viene effettuata sul nome dell'oggetto aperto e non sul nome o i nomi, risultanti quando un nome è stato risolto. Ad esempio, è possibile concedere a un'applicazione l'autorizzazione per aprire un alias di coda senza che questa disponga dell'autorizzazione per aprire la coda base in cui si trasforma l'alias. Di regola la verifica viene effettuata sulla prima definizione incontrata durante il processo di risoluzione del nome che non sia un alias di Queue Manager, a meno che la relativa definizione non sia aperta direttamente; in altre parole, il nome appare nel campo *ObjectName* della descrizione dell'oggetto. L'autorizzazione è sempre necessaria per il particolare oggetto che viene aperto; in alcuni casi sono necessarie autorizzazioni supplementari indipendenti dalla coda che si ottengono tramite l'autorizzazione per l'accesso all'oggetto del Queue Manager.

La Tabella 3 riassume le autorizzazioni necessarie per ciascuna chiamata.

Tabella 3. Autorizzazione di sicurezza necessaria per chiamate MQI

Autorizzazione richiesta per:	Oggetto coda	Oggetto processo	Queue Manager	Namelist
Opzione MQCONN	Non applicabile	Non applicabile	MQZAO_CONNECT	Non applicabile
Opzione MQOPEN				
MQOO_INQUIRE	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)
MQOO_BROWSE	MQOO_BROWSE	Non applicabile	Nessuna verifica	Non applicabile
MQOO_INPUT_*	MQZAO_INPUT	Non applicabile	Nessuna verifica	Non applicabile
MQOO_SAVE_ALL_CONTEXT (3)	MQZAO_INPUT	Non applicabile	Nessuna verifica	Non applicabile
MQOO_OUTPUT (coda normale) (4)	MQOO_OUTPUT	Non applicabile	Nessuna verifica	Non applicabile
MQOO_PASS_IDENTITY_CONTEXT (5)	MQZAO_PASS_IDENTITY_CONTEXT	Non applicabile	Nessuna verifica	Non applicabile

Tablelle di specifica delle autorizzazioni

Tabella 3. Autorizzazione di sicurezza necessaria per chiamate MQI (Continua)

Autorizzazione richiesta per:	Oggetto coda	Oggetto processo	Queue Manager	Namelist
MQOO_PASS_ALL_CONTEXT (5, 6)	MQZAO_PASS_ALL_CONTEXT	Non applicabile	Nessuna verifica	Non applicabile
MQOO_SET_IDENTITY_CONTEXT (5, 6)	MQZAO_SET_IDENTITY_CONTEXT	Non applicabile	MQZAO_SET_IDENTITY_CONTEXT	Non applicabile
MQOO_SET_ALL_CONTEXT (5, 8)	MQZAO_SET_ALL_CONTEXT	Non applicabile	MQZAO_SET_ALL_CONTEXT (7)	Non applicabile
MQOO_OUTPUT (coda di trasmissione) 9)	MQZAO_SET_ALL_CONTEXT	Non applicabile	MQZAO_SET_ALL_CONTEXT (7)	Non applicabile
MQOO_SET	MQZAO_SET	Non applicabile	Nessuna verifica	Non applicabile
MQOO_ALTERNATE_USER_AUTHORITY	(10)	(10)	MQZAO_ALTERNATE_USER_AUTHORITY (10, 11)	Non applicabile
Opzione MQPUT1				
MQPMO_PASS_IDENTITY_CONTEXT	MQZAO_PASS_IDENTITY_CONTEXT (12)	Non applicabile	Nessuna verifica	Non applicabile
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT (12)	Non applicabile	Nessuna verifica	Non applicabile
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT (12)	Non applicabile	MQZAO_SET_IDENTITY_CONTEXT (7)	Non applicabile
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT (12)	Non applicabile	MQZAO_SET_ALL_CONTEXT (7)	Non applicabile
(Coda di trasmissione) (9)	MQZAO_SET_ALL_CONTEXT	Non applicabile	MQZAO_SET_ALL_CONTEXT (7)	Non applicabile
MQPMO_ALTERNATE_USER_AUTHORITY	(13)	Non applicabile	MQZAO_ALTERNATE_USER_AUTHORITY (11)	Non applicabile
Opzione MQCLOSE				
MQCO_DELETE	MQZAO_DELETE (14)	Non applicabile	Non applicabile	Non applicabile
MQCO_DELETE_PURGE	MQZAO_DELETE (14)	Non applicabile	Non applicabile	Non applicabile

Note specifiche:

- Se una coda modello viene aperta:
 - È necessaria l'autorizzazione MQZAO_DISPLAY per la coda modello, oltre a qualsiasi altra autorizzazione (anche per la coda modello) che sia richiesta per le opzioni aperte specificate.
 - L'autorizzazione MQZAO_CREATE non è necessaria per creare la coda dinamica.
 - All'identificativo utente utilizzato per aprire la coda modello sono automaticamente concesse tutte le autorizzazioni specifiche della coda (equivalenti a MQZAO_ALL) per la coda dinamica creata.

Tabelle di specifica delle autorizzazioni

2. Vengono verificati la coda, il processo, la namelist o il Queue Manager, a seconda del tipo di oggetto aperto.
3. Occorre specificare anche MQOO_INPUT_*. Ciò è valido per una coda locale, modello o di alias.
4. Questa verifica viene eseguita per tutti i casi di output, tranne il caso specificato nella nota 9.
5. Occorre specificare anche MQOO_OUTPUT.
6. Anche MQOO_PASS_IDENTITY_CONTEXT è implicato in questa opzione.
7. Quest'autorizzazione è richiesta sia dall'oggetto Queue Manager che dalla coda particolare.
8. Anche MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT e MQOO_SET_IDENTITY_CONTEXT sono implicati in questa opzione.
9. Questa verifica viene eseguita per una coda locale o modello che dispone di un attributo di coda *Usage* di MQUS_TRANSMISSION e viene aperta direttamente per l'output. Ciò non si applica se una coda remota viene aperta (specificando i nomi del Queue Manager remoto e della coda remota oppure specificando il nome di una definizione locale della coda remota).
10. Occorre inoltre specificare almeno uno fra MQOO_INQUIRE (per qualsiasi tipo di oggetto) o (per le code) MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT o MQOO_SET. La verifica eseguita, come per le altre opzioni specificate, utilizza l'identificativo utente alternativo fornito per l'autorizzazione all'oggetto dal nome specifico e l'autorizzazione dell'applicazione corrente per la verifica MQZAO_ALTERNATE_USER_IDENTIFIER.
11. Questa autorizzazione consente a qualsiasi *AlternateUserId* di essere specificato.
12. Viene eseguita anche una verifica MQZAO_OUTPUT, se la coda non dispone di un attributo di coda *Usage* di MQUS_TRANSMISSION.
13. La verifica eseguita, come per le altre opzioni specificate, utilizza l'identificativo utente alternativo fornito per l'autorizzazione alla coda dal nome specifico e l'autorizzazione della corrente applicazione per la verifica MQZAO_ALTERNATE_USER_IDENTIFIER.
14. La verifica viene eseguita solo se entrambi le seguenti affermazioni sono vere:
 - Si sta chiudendo ed eliminando una coda dinamica permanente.
 - La coda non è stata creata da MQOPEN che ha restituito l'handle dell'oggetto in uso.

In caso contrario, non vi sarà alcuna verifica.

Note generali:

1. L'autorizzazione speciale MQZAO_ALL_MQI include tutte le seguenti che sono relative al tipo di oggetto:
 - MQZAO_CONNECT
 - MQZAO_INQUIRE
 - MQZAO_SET
 - MQZAO_BROWSE
 - MQZAO_INPUT
 - MQZAO_OUTPUT
 - MQZAO_PASS_IDENTITY_CONTEXT
 - MQZAO_PASS_ALL_CONTEXT
 - MQZAO_SET_IDENTITY_CONTEXT
 - MQZAO_SET_ALL_CONTEXT
 - MQZAO_ALTERNATE_USER_AUTHORITY

Tabelle di specifica delle autorizzazioni

2. MQZAO_DELETE (vedere nota la 14 a pagina 153) e MQZAO_DISPLAY sono classificate come autorizzazioni di gestione. Esse non sono pertanto incluse in MQZAO_ALL_MQI.
3. 'Nessuna verifica' significa che non è stata eseguita alcuna verifica dell'autorizzazione.
4. 'Non applicabile' significa che la verifica dell'autorizzazione non è importante per questa operazione. Ad esempio, non è possibile emettere una chiamata MQPUT per un oggetto processo.

Autorizzazioni di gestione

Queste autorizzazioni consentono a un utente di emettere comandi di gestione, ovvero un comando MQSC come messaggio PCF escape o come comando PCF stesso. Tali metodi consentono a un programma di inviare un comando di gestione come messaggio a un Queue Manager, affinché lo esegua per conto di tale utente.

Autorizzazioni per comandi MQSC nei PCF escape

La Tabella 4 riassume le autorizzazioni necessarie per ciascun comando MQSC contenuto nel PCF Escape.

Tabella 4. Comandi MQSC e autorizzazioni di sicurezza necessari

(2) Autorizzazione richiesta per:	Oggetto coda	Oggetto processo	Oggetto Queue Manager	Namelist
ALTER object	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
CLEAR QLOCAL	MQZAO_CLEAR	Non applicabile	Non applicabile	Non applicabile
DEFINE object NOREPLACE (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Non applicabile	MQZAO_CREATE (4)
DEFINE object REPLACE (3, 5)	MQZAO_CHANGE	MQZAO_CHANGE	Non applicabile	MQZAO_CHANGE
DELETE object	MQZAO_DELETE	MQZAO_DELETE	Non applicabile	MQZAO_DELETE
DISPLAY object	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY

Note specifiche:

1. L'identificativo utente, sotto il quale il programma (ad esempio, **runmqsc**) che invia il comando è in esecuzione, deve disporre inoltre dell'autorizzazione MQZAO_CONNECT per accedere al Queue Manager.
2. Vengono verificati la coda, il processo, la namelist o il Queue Manager, a seconda del tipo di oggetto.
3. Per i comandi DEFINE, anche l'autorizzazione MQZAO_DISPLAY è necessaria per l'oggetto LIKE se specificato, o sull'oggetto SYSTEM.DEFAULT.xxx appropriato se LIKE viene omissso.
4. L'autorizzazione MQZAO_CREATE non è specifica di un particolare oggetto o tipo di oggetto. La creazione di autorizzazioni è concessa a tutti gli oggetti, per un Queue Manager specificato, specificando un tipo di oggetto di QMGR sul comando SETMQAUT.
5. Ciò si applica se l'oggetto da sostituire è già effettivamente esistente. In caso contrario, la verifica è la stessa di DEFINE object NOREPLACE.

Note generali:

1. Per eseguire qualsiasi comando PCF, è necessario disporre dell'autorizzazione DISPLAY sul Queue Manager.

Tabelle di specifica delle autorizzazioni

2. L'autorizzazione per eseguire un PCF escape dipende dal comando MQSC all'interno del testo del messaggio PCF escape.
3. 'Non applicabile' significa che la verifica dell'autorizzazione non è importante per questa operazione. Ad esempio, non è possibile emettere CLEAR QLOCAL su un oggetto Queue Manager.

Autorizzazioni per comandi PCF

La Tabella 5 riepiloga le autorizzazioni necessarie per ciascun comando PCF.

Tabella 5. Comandi PCF e autorizzazioni di sicurezza necessari

(2) Autorizzazione richiesta per:	Oggetto coda	Oggetto processo	Oggetto Queue Manager	Namelist
Modifica oggetto	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
Clear Queue	MQZAO_CLEAR	Non applicabile	Non applicabile	Non applicabile
Copia oggetto (senza sostituire) 3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Non applicabile	MQZAO_CREATE (4)
Copia oggetto (e sostituisci) 3, 6)	MQZAO_CHANGE	MQZAO_CHANGE	Non applicabile	MQZAO_CHANGE
Crea oggetto (senza sostituire) 5)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Non applicabile	MQZAO_CREATE (4)
Crea oggetto (e sostituisci) 5, 6)	MQZAO_CHANGE	MQZAO_CHANGE	Non applicabile	MQZAO_CHANGE
Elimina oggetto	MQZAO_DELETE	MQZAO_DELETE	Non applicabile	MQZAO_DELETE
Richiedi informazioni sull'oggetto	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY
Richiedi informazioni sui nomi dell'oggetto	Nessuna verifica	Nessuna verifica	Nessuna verifica	Nessuna verifica
Reimposta le statistiche di coda	MQZAO_DISPLAY e MQZAO_CHANGE	Non applicabile	Non applicabile	Non applicabile

Note specifiche:

1. Anche l'identificativo utente sotto il quale il programma che invia il comando è in esecuzione deve disporre dell'autorizzazione per connettersi al relativo gestore di code locali e per aprire la coda di comandi di gestione per l'output.
2. Vengono verificati la coda, il processo, o l'oggetto queue manager, a seconda del tipo di oggetto.
3. Per i comandi Copy, è necessaria l'autorizzazione MQZAO_DISPLAY anche per l'oggetto From.
4. L'autorizzazione MQZAO_CREATE non è specifica di un particolare oggetto o tipo di oggetto. La creazione di autorizzazioni è concessa a tutti gli oggetti, per un Queue Manager specificato, specificando un tipo di oggetto di QMGR sul comando SETMQAUT.
5. Per i comandi Create, è necessaria l'autorizzazione MQZAO_DISPLAY anche per l'appropriato oggetto SYSTEM.DEFAULT.*.
6. Ciò si applica se l'oggetto da sostituire è già esistente. In caso contrario, la verifica è la stessa di Copy or Create.

Note generali:

1. Per eseguire qualsiasi comando PCF, è necessario disporre dell'autorizzazione DISPLAY sul Queue Manager.

Tabelle di specifica delle autorizzazioni

2. L'autorizzazione speciale MQZAO_ALL_ADMIN include tutte le seguenti che sono relative al tipo di oggetto:
 - MQZAO_CHANGE
 - MQZAO_CLEAR
 - MQZAO_DELETE
 - MQZAO_DISPLAY

MQZAO_CREATE non è inclusa, perché non è specifica ad un particolare o tipo di oggetto.

3. 'Nessuna verifica' significa che non è stata eseguita alcuna verifica dell'autorizzazione.
4. 'Non applicabile' significa che la verifica dell'autorizzazione non è importante per questa operazione. Ad esempio, non è possibile utilizzare un comando Clear Queue su un oggetto di processo.

Comprendere i file di autorizzazione

Per MQSeries for Compaq NSK, V5.1, tutte le informazioni di autorizzazione sono memorizzate nei seguenti file controllati TM/MP in \$VOL.<QMgrSubVol>D:

OAMDB Il database OAM
PRIDB Il database dei principal
PRIDBA Il file di chiavi alternative del database dei principal

Il database dei principal

Ciascun record nel database dei principal mette in corrispondenza un ID utente Compaq NSK con un nome di principal. Il database dei principal è un file a sequenza di chiavi ENSCRIBE che fornisce una messa in corrispondenza tra i principal OAM e gli ID di collegamento Compaq NSK.

Principal OAM MQPRINCIPAL
ID di collegamento Compaq 0x2CFF

La chiave principale è il principal OAM (12 caratteri). La chiave alternativa è l'ID di collegamento Compaq (un valore numerico breve da 2 byte). Il principal OAM è sempre sensibile al maiuscolo/minuscolo. I byte del campo dell'ID di collegamento sono <gruppo>.<utente> ; l'esempio precedente è per l'ID Compaq NSK (44,255).

Il database OAM

Ciascun record nel nuovo database delle autorizzazioni OAM fa riferimento ad uno specifico oggetto Queue Manager o classe di oggetti. La chiave principale è il nome oggetto più il tipo di oggetto. I record sono di lunghezza variabile e la struttura di record è la seguente:

Nome Oggetto	Tipo	Numero voci aut	Voci aut
QUEUE.AUTH	1	2	PAYROLL 0x00000004, ADMIN 0xFFFFFFFF

Il campo Nome oggetto è il nome completo di oggetto senza spazi vuoti da 48 caratteri. Il campo Tipo (4 byte) distingue i tipi di oggetto MQSeries e le classi di oggetto richieste dall'OAM.

Il campo Tipo può avere i seguenti valori:

- 1 Nome coda
- 2 Nome processo

4 Il nome del Queue Manager
 128 Classe

Il campo Numero voci aut (4 byte) specifica il numero di singole autorizzazioni nel campo Voci aut in questo record. Ciascuna delle Voci aut specifica un nome di gruppo e l'autorizzazione per detto gruppo per quest'oggetto:

Nome gruppo PAYROLL
Aut 0x00000004

Il campo Nome gruppo ha una lunghezza di 12 byte e contiene un nome di gruppo di condivisione di file (File-sharing) SAFEGUARD o di gestione (Administrative) di Compaq NSK (solo i primi 12 caratteri). Il campo Aut è una bit mask da 4 byte (ULONG) che contiene l'autorizzazione per il gruppo. Un singolo record può contenere un massimo di 250 singole coppie gruppo-autorizzazione.

Se necessario, è possibile utilizzare più record per lo stesso oggetto per contenere le informazioni sulle autorizzazioni per più di 250 gruppi.

La specifica dell'autorizzazione è costituita dall'unione degli schemi di bit individuali basati sulle seguenti assegnazioni:

Parola chiave autorizzazione	Nome formale	Valore Esadecimale
connect	MQZAO_CONNECT	0x00000001
browse	MQZAO_BROWSE	0x00000002
get	MQZAO_INPUT	0x00000004
put	MQZAO_OUTPUT	0x00000008
inq	MQZAO_INQUIRE	0x00000010
set	MQZAO_SET	0x00000020
passid	MQZAO_PASS_IDENTITY_CONTEXT	0x00000040
passall	MQZAO_PASS_ALL_CONTEXT	0x00000080
setid	MQZAO_SET_IDENTITY_CONTEXT	0x00000100
setall	MQZAO_SET_ALL_CONTEXT	0x00000200
altusr	MQZAO_ALTERNATE_USER_AUTHORITY	0x00000400
allmqi	MQZAO_ALL_MQI	0x000007FF
crt	MQZAO_CREATE	0x00010000
dlt	MQZAO_DELETE	0x00020000
dsp	MQZAO_DISPLAY	0x00040000
chg	MQZAO_CHANGE	0x00080000
clr	MQZAO_CLEAR	0x00100000
chgaut	MQZAO_AUTHORIZE	0x00800000
alladm	MQZAO_ALL_ADMIN	0x009E0000
none	MQZAO_NONE	0x00000000
all	MQZAO_ALL	0x009E07FF

Queste definizioni sono effettuate nel file di intestazione cmqzc.h. Nel seguente esempio, a groupB sono state concesse le autorizzazioni basate sul numero esadecimale 0x40007. Ciò corrisponde a:

MQZAO_CONNECT	0x00000001
MQZAO_BROWSE	0x00000002
MQZAO_INPUT	0x00000004
MQZAO_DISPLAY	0x00040000
-----	-----
Authority is:	0x00040007

File di autorizzazione

Questi diritti di accesso significano che chiunque nel groupB può emettere le chiamate MQI:

- MQCONN
- MQGET (con browse)

Essi dispongono anche dell'autorizzazione DISPLAY per l'oggetto associato a tale file di autorizzazione.

Record di autorizzazione classe

I record di autorizzazione classe contengono le autorizzazioni correlate all'intera classe. I campi Nome oggetto e Tipo hanno questo schema:

Nome Oggetto	Tipo
@QMGRCLASS	0x80
@PROCESSCLASS	0x80
@QUEUECLASS	0x80

La voce MQZAO_CRT nel campo relativo all'autorizzazione concede l'autorizzazione a creare un oggetto nella classe. Questa è l'unica autorizzazione di classe.

Record di autorizzazione per tutte le classi

Il record di autorizzazione per tutte le classi contiene le autorizzazioni valide per un intero queue manager. I campi Nome oggetto e Tipo hanno questo schema:

Nome Oggetto	Tipo
@ALLCLASSES	0x80

Le seguenti autorizzazioni si applicano all'intero Queue Manager e sono contenute nel file di autorizzazione per tutte le classi.

Voce...	Fornisce l'autorizzazione per...
MQZAO_ALTUSER	Assumere l'identità di un altro utente durante l'interazione con oggetti MQSeries.
MQZAO_SET_ALL_CONTEXT	Impostare il contesto di un messaggio quando si emette MQPUT.
MQZAO_SET_IDENTITY_CONTEXT	Impostare il contesto di identità di un messaggio quando si emette MQPUT.

Capitolo 9. Handler di code messaggi non recapitati di MQSeries

MQSeries for Compaq NSK fornisce una DLQ (dead-letter queue), denominata anche undelivered-message queue, è una coda che contiene i messaggi che non riescono a raggiungere la loro coda di destinazione. Ogni queue manager in una rete dovrebbe essere associato ad una DLQ.

I messaggi sono inviati alla DLQ dai queue manager, dai MCA (message channel agent), e dalle applicazioni. Tutti i messaggi della DLQ devono avere la struttura *dead-letter header* MQDLH. I messaggi inviati alla DLQ da un Queue Manager o da un MCA hanno sempre una struttura header. Le applicazioni che inviano messaggi alla DLQ devono sempre fornire tale struttura. Il campo *Reason* della struttura MQDLH contiene un codice di errore che identifica il motivo per cui il messaggio è stato inviato alla DLQ.

E' necessario che sia impostata una routine da eseguire ad intervalli regolari per elaborare i messaggi presenti nella DLQ. MQSeries fornisce una routine predefinita denominata *dead-letter queue handler* (DLQ handler), che può essere richiamata utilizzando il comando **runmqdlq**.

Le istruzioni per l'elaborazione dei messaggi presenti nella DLQ vengono fornite al DLQ handler dai contenuti immessi dall'utente nella *tabella regole*. Il DLQ handler confronta i messaggi presenti nella coda messaggi non recapitati con le indicazioni della tabella regole. Quando rileva una corrispondenza tra un messaggio e un'indicazione, il DLQ handler esegue l'azione associata a quella determinata indicazione.

Questo capitolo contiene le seguenti sezioni:

- "Richiamo del DLQ handler"
- "Tabella regole DLQ handler" a pagina 160
- "Elaborazione della tabella regole" a pagina 166
- "Esempio di tabella regole per DLQ handler" a pagina 168

Richiamo del DLQ handler

Per richiamare il DLQ handler occorre utilizzare il comando **runmqdlq**. È possibile denominare la coda messaggi non recapitati che si desidera elaborare e il Queue Manager che si desidera utilizzare come segue:

- Dalla richiesta comandi utilizzando i parametri. Ad esempio:

```
runmqdlq /IN qrule/ ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER
```

- Nella tabella regole. Ad esempio:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Richiamo del DLQ handler

Gli esempi riportati si riferiscono alla coda DLQ denominata ABC1.DEAD.LETTER.QUEUE, di pertinenza del Queue Manager ABC1.QUEUE.MANAGER.

se non vengono specificati i nomi della DLQ e del Queue Manager come sopra riportato, viene utilizzato il Queue Manager predefinito per l'installazione e la coda DLQ di pertinenza.

Il comando **runmqdlq** legge le immissioni dalle tabelle, fornite con il file IN standard. Associare le tabelle al comando **runmqdlq** reindirizzando IN al file.

Per eseguire il DLQ handler, è necessaria l'autorizzazione ad accedere sia alla coda messaggi non recapitati, sia alle code cui erano destinati i messaggi. Pertanto, se si abilita il DLQ handler ad inviare i messaggi alle code con l'autorizzazione dell'ID utente nel contesto del messaggio, si dovrà avere l'autorizzazione ad assumere l'identità di altri utenti.

Per ulteriori informazioni sul comando **runmqdlq**, consultare la sezione "runmqdlq (Run dead-letter queue handler)" a pagina 295

Tabella regole DLQ handler

La tabella regole DLQ handler definisce il tipo di elaborazione dei messaggi che arrivano nella coda messaggi non recapitati. Vi sono due tipi di voci da poter immettere nella tabella:

- La prima, facoltativa, contiene *dati di controllo*.
- Tutte le altre sono *regole* che il DLQ handler deve seguire. Ogni regola è costituita di uno *schema* (un gruppo di caratteristiche del messaggio) cui un messaggio deve corrispondere e di un'*azione* che deve essere eseguita, quando nella coda messaggi non recapitati viene rilevato un messaggio che corrisponde alle caratteristiche dello schema. Deve esserci almeno una regola nella tabella.

Ogni voce della tabella comprende una o più parole chiave.

Per una descrizione delle regole di sintassi valide alle tabelle di regola, consultare la sezione "Convenzioni relative alla tabella regole" a pagina 165.

Dati di controllo

Questa sezione descrive le parole chiave che si possono inserire in una voce di dati di controllo nella tabella regole di un DLQ handler. Tenere presente le seguenti informazioni:

- L'impostazione predefinita dell'eventuale parola chiave è sottolineata.
- La linea verticale (|) separa le alternative, di cui una sola può essere specificata.
- Tutte le parole chiave sono facoltative.

INPUTQ (*QueueName* | ' _ ')

Tale parola chiave è il nome di DLQ a cui la tabella di regole viene applicata. Consente di denominare la coda messaggi non recapitati che si desidera elaborare:

1. Se viene specificato il parametro *QName* sul comando **runmqdlq**, esso ricopre qualsiasi valore INPUTQ nella tabella di regole.
2. Se non si specifica un parametro *QName* sul comando **runmqdlq**, ma si specifica un valore nella tabella regole, il valore INPUTQ viene utilizzato nella tabella regole.

3. Se non si specifica un DLQ o si specifica INPUTQ(' ') nella tabella regole, viene elaborato il DLQ che appartiene al queue manager, il cui nome viene fornito nel parametro *QMgrName* del comando **runmqdlq** o nella parola chiave INPUTQM della tabella regole.

INPUTQM (*QueueManagerName* | ' ')

Tale parola chiave è il nome del queue manager che gestisce il DLQ. Consente di denominare il Queue Manager che gestisce la coda DLQ denominata con la parola chiave INPUTQ:

1. Se si specifica un parametro *QMgrName* del comando **runmqdlq**, esso ricopre qualsiasi valore INPUTQM nella tabella regole.
2. Se non si specifica un parametro *QMgrName* del comando **runmqdlq**, il valore INPUTQM della tabella regole viene utilizzato.
3. Se non si specifica nessun Queue Manager o si specifica INPUTQM(' ') nella tabella regole, viene utilizzato il Queue Manager predefinito.

RETRYINT (*Interval* | 60)

Tale parola chiave indica, in secondi, l'intervallo di tempo che separa i tentativi di elaborazione da parte del DLQ handler di quei messaggi che non è riuscito ad elaborare al primo tentativo e per i quali è stato richiesto un nuovo tentativo. Per impostazione predefinita, l'intervallo di tempo è di 60 secondi.

WAIT (YES | NO | *nnn*)

Tale parola chiave indica se il DLQ handler deve attendere l'arrivo di ulteriori messaggi sulla coda DLQ quando rileva l'assenza di messaggi da elaborare.

YES Tale parola chiave determina un'attesa a tempo indeterminato.

NO Tale parola chiave determina l'arresto del DLQ handler quando rileva che non vi sono altri messaggi da elaborare sulla coda DLQ.

nnn Tale parola chiave determina un'attesa del DLQ handler di *nnn* secondi dall'elaborazione dell'ultimo messaggio prima dell'arresto.

Si consiglia di specificare WAIT (YES) per le code DLQ piene di messaggi e WAIT (NO) o WAIT (*nnn*) per la code DLQ con un basso livello di attività. Se il DLQ handler si arresta, è necessario richiamarlo con la funzione trigger.

I dati di controllo visualizzati in Figura 40 illustrano che le tabelle regola vengono applicate a DLQ che appartiene al queue manager QM1. Il segno più (+) situato all'estremità della riga 1 indica che i dati di controllo continuano dal primo carattere non vuoto della riga 2.

INPUTQ' ' + INPUTQM'QM1'

Figura 40. Esempio di dati di controllo

In alternativa all'immissione di dati di controllo nella tabella regole, è possibile inserire il nome della coda DLQ e del relativo Queue Manager come parametro di immissione del comando **runmqdlq**. Se viene specificato qualsiasi valore nella tabella regole e nell'immissione del comando **runmqdlq**, il valore specificato del comando **runmqdlq** ha la precedenza sul valore specificato.

Nota: se nella tabella regole è inclusa una voce dati di controllo, deve essere la prima voce della tabella.

Tabella regole

Regole (schemi e azioni)

La Figura 41 mostra un esempio di regole da una tabella regole DLQ handler. Questa regola ordina al DLQ handler di effettuare tre tentativi per inviare alla propria coda di destinazione ogni messaggio inviato alla coda DLQ in seguito all'inibizione di MQPUT e MQPUT1.

```
PERSIST(MQPER_PERSISTENT) REASON(MQRC_PUT_INHIBITED) +  
ACTION(RETRY) RETRY(3)
```

Figura 41. Esempio di regola. Il segno più (+) situato all'estremità della riga 1 indica che la regola continua dal primo carattere non vuoto sulla riga 2.

Tutte le parole chiave utilizzabili in una regola sono descritte nella rimanente parte di questa sezione. Tenere presente le seguenti informazioni:

- L'impostazione predefinita dell'eventuale parola chiave è sottolineata. Per molte parole chiave, il valore predefinito è * (asterisco), che comprende ogni valore.
- La linea verticale (|) separa le alternative, di cui una sola può essere specificata.
- Tutte le parole chiave sono facoltative, tranne ACTION.

Questa sezione descrive prima le parole chiave dello schema di confronto (quelle con le quali vengono confrontati i messaggi presenti nella coda DLQ) e poi quelle relative alle azioni (quelle che determinano il tipo di elaborazione dei messaggi da parte del DLQ handler).

Parole chiave dello schema di confronto

Di seguito vengono descritte le parole chiave dello schema di confronto utilizzate per specificare i valori con i quali confrontare i messaggi nella coda DLQ. Tutte le parole chiave sono facoltative.

APPLIDAT (*ApplIdentityData* | *)

Tale parola chiave indica il valore *ApplIdentityData* specificato nel descrittore di messaggio (MQMD) del messaggio presente nella coda DLQ.

APPLNAME (*PutApplName* | *)

Tale parola chiave indica il nome dell'applicazione che inoltra la chiamata MQPUT o MQPUT1, come specificato nel campo *PutApplName* del descrittore di messaggio MQMD, del messaggio nella coda DLQ.

APPLTYPE (*PutApplType* | *)

Tale parola chiave indica il valore *PutApplType* specificato nel descrittore del messaggio (MQMD) del messaggio presente nella coda DLQ.

DESTQ (*QueueName* | *)

Tale parola chiave indica il nome della coda destinataria del messaggio.

DESTQM (*QueueManagerName* | *)

Tale parola chiave indica il nome del Queue Manager che gestisce la coda destinataria del messaggio.

FEEDBACK (*Feedback* | *)

Se il valore *MsgType* è MQFB_REPORT, *Feedback* descrive la natura della notifica.

È possibile utilizzare nomi simbolici. Ad esempio, per identificare i messaggi nella coda DLQ che richiedono la conferma del proprio arrivo alla coda di destinazione, è possibile utilizzare il nome simbolico MQFB_COA.

FORMAT (*Format* | *)

Tale parola chiave indica il nome utilizzato dal mittente per descrivere il formato dei dati del messaggio inviato.

MSGTYPE (*MsgType* | *)

Tale parola chiave indica il tipo di messaggio presente nella coda DLQ.

È possibile utilizzare nomi simbolici. Ad esempio, per identificare i messaggi presenti nella coda DLQ che richiedono risposte, è possibile utilizzare il nome simbolico MQMT_REQUEST.

PERSIST (*Persistence* | *)

Tale parola chiave indica il valore di permanenza del messaggio. La permanenza determina la sopravvivenza o meno del messaggio al riavvio del Queue Manager.

È possibile utilizzare nomi simbolici. Ad esempio, per identificare i messaggi nella coda DLQ permanenti, è possibile utilizzare il nome MQPER_PERSISTENT.

REASON (*ReasonCode* | *)

Tale parola chiave indica il codice di errore che spiega il motivo dell'invio del messaggio alla coda DLQ.

È possibile utilizzare nomi simbolici. Ad esempio, per identificare i messaggi che sono stati inviati alla coda DLQ poiché la loro coda di destinazione era piena, è possibile utilizzare il nome simbolico MQRC_Q_FULL.

REPLYQ (*QueueName* | *)

Tale parola chiave indica il nome della coda di risposta specificata nel descrittore del messaggio (MQMD) del messaggio presente nella coda DLQ.

REPLYQM (*QueueManagerName* | *)

Tale parola chiave indica il nome del Queue Manager che gestisce la coda di risposta specificata nel descrittore di messaggio, MQMD, del messaggio presente nella coda DLQ.

USERID (*UserIdentifier* | *)

Tale parola chiave indica l'ID utente dell'utente specificato nel messaggio della DLQ e nel descrittore del messaggio (MQMD).

Parole chiave di azione

Di seguito vengono descritte le parole chiave che indicano il tipo di elaborazione dei messaggi, che rientrano nello schema di confronto.

ACTION (DISCARD | IGNORE | RETRY | FWD)

Tale parola chiave indica l'azione da eseguire per quei messaggi nella coda DLQ che rientrano nello schema definito in questa regola.

DISCARD Tale parola chiave determina l'eliminazione del messaggio dalla coda DLQ.

IGNORE Tale parola chiave determina la permanenza del messaggio nella coda DLQ.

RETRY Tale parola chiave determina un nuovo tentativo del DLQ handler di inviare il messaggio alla coda di destinazione.

FWD Tale parola chiave permette che il DLQ handler inoltri il messaggio alla coda denominata con la parola chiave FWDQ.

Tabella regole

E' necessario specificare la parola chiave ACTION. Il numero tentativi volti al compimento di un azione è determinato dalla parola chiave RETRY. L'intervallo di tempo tra i vari tentativi è indicato dalla parola chiave RETRYINT dei dati di controllo.

FWDQ (*QueueName* | **&DESTQ** | **&REPLYQ**)

Tale parola chiave indica il nome della coda a cui inoltrare il messaggio quando è impostato ACTION (FWD).

QueueName

Tale parola chiave indica il nome di una coda messaggi. FWDQ(' ') non è valido.

&DESTQ

Tale parola chiave determina l'assunzione del nome della coda dal campo *DestQName* nella struttura MQDLH.

&REPLYQ

Tale parola chiave determina l'assunzione del nome dal campo *ReplyToQ* nel descrittore di messaggio MQMD.

Per evitare messaggi di errore, che si verificherebbero qualora una regola che specifica FWDQ (&REPLYQ) riscontri un messaggio con il campo *ReplyToQ* vuoto, è consigliabile specificare REPLYQ (?*) nello schema messaggi.

FWDQM (*QueueManagerName* | **&DESTQM** | **&REPLYQM** | ' ')

Tale parola chiave identifica il Queue Manager che gestisce la coda cui deve essere inoltrato un messaggio.

QueueManagerName

Tale parola chiave indica il nome del Queue Manager che gestisce la coda cui inoltrare un messaggio quando è impostato ACTION (FWD).

&DESTQM

Tale parola chiave determina l'assunzione del nome del queue manager dal campo *DestQMgrName* nella struttura MQDLH.

&REPLYQM

Tale parola chiave determina l'assunzione del nome dal campo *ReplyToQMgr* nel descrittore di messaggi (MQMD).

' ' FWDQM(' '), valore predefinito, identifica il Queue Manager locale.

HEADER (YES | NO)

Tale chiave parola specifica se la struttura MQDLH debba rimanere su un messaggio per cui è stata specificata la parola chiave ACTION (FWD). Per impostazione predefinita, la struttura MQDLH rimane sul messaggio. La parola chiave HEADER non è valida per altre azioni diverse da FWD.

PUTAUT (DEF | CTX)

Tale parola chiave definisce l'autorizzazione in base alla quale il DLQ handler deve eseguire l'invio dei messaggi:

DEF Tale parola chiave determina l'invio dei messaggi con l'autorizzazione propria del DLQ handler.

CTX Tale parola chiave determina l'invio dei messaggi con l'autorizzazione dell'ID utente nel messaggio context. Se si specifica PUTAUT (CTX), è necessaria l'autorizzazione ad assumere l'identità di altri utenti.

RETRY (*RetryCount* | 1)

RETRY indica il numero di tentativi che possono essere realizzati per il

compimento di un'azione, in uno spazio tra 1-999 (l'intervallo di tempo è specificato con la parola chiave `RETRYINT` dei dati di controllo).

Nota: il conteggio dei tentativi fatti dal DLQ handler per completare una determinata azione è relativo all'istanza in corso; il conteggio non viene ripreso dopo un riavvio. Quando il DLQ handler viene riavviato, il conteggio dei tentativi eseguiti è azzerato.

Convenzioni relative alla tabella regole

La tabella regole deve rispettare le seguenti convenzioni relative alla sintassi, alla struttura e ai contenuti:

- Una tabella regole deve contenere almeno una regola.
- Le parole chiave possono ricorrere in qualsiasi ordine.
- Una parola chiave può essere inserita solo una volta in ciascuna regola.
- Le parole chiave non sono sensibili al minuscolo/maiuscolo.
- Una parola chiave e i relativi valori dei parametri devono essere separate dalle altre parole chiave con almeno uno spazio o una virgola.
- All'inizio e alla fine della regola, tra le parole chiave, la punteggiatura e i valori può essere inserito un qualsiasi numero di spazi vuoti.
- Ogni regola deve iniziare su una sua nuova linea.
- Per ragioni di flessibilità d'uso, la dimensione di una linea non deve superare i 72 caratteri.
- Utilizzare il segno più (+) come ultimo carattere di una linea per indicare che la regola prosegue con il primo carattere digitato nella linea successiva. Utilizzare il segno meno (-) come ultimo carattere di una linea per indicare che la regola prosegue dall'inizio della linea successiva. I caratteri di continuazione possono essere inseriti all'interno di parole chiave e parametri.
- Linee di commento, che iniziano con un asterisco (*), possono trovarsi in qualsiasi punto della tabella regole.
- Le linee vuote sono ignorate.
- Ogni voce della tabella regole del DLQ handler comprende una o più parole chiave e i relativi parametri associati. I parametri devono rispettare le seguenti regole di sintassi:
 - Ogni parametro deve includere almeno un carattere valido. I segni di delimitazione in cui vengono ricompresi i valori tra parentesi non sono considerati caratteri validi. Ad esempio, di seguito sono riportati dei caratteri validi:
 - FORMAT('ABC')** 3 caratteri validi
 - FORMAT(ABC)** 3 caratteri validi
 - FORMAT('A')** 1 carattere valido
 - FORMAT(A)** 1 carattere valido
 - FORMAT(' ')** 1 carattere valido

I seguenti parametri invece non sono validi, poiché non contengono caratteri validi:

```
FORMAT('')
FORMAT( )
FORMAT()
FORMAT
```

- I caratteri globali sono supportati: è possibile utilizzare il punto interrogativo (?) al posto di ogni singolo carattere, tranne gli spazi vuoti; è possibile

Convenzioni relative alla tabella regole

utilizzare l'asterisco (*) al posto dello zero o di altri caratteri simili. L'asterisco (*) e il punto interrogativo (?) sono *sempre* interpretati come caratteri globali nei valori di parametro.

- I caratteri globali non possono essere utilizzati nei parametri delle seguenti parole chiave: ACTION, HEADER, RETRY, FWDQ, FWDQM e PUTAUT.
- Gli spazi vuoti nell'impostazione dei parametri e dei corrispondenti campi nei messaggi presenti nella coda DLQ, non sono rilevanti nel confronto con i caratteri globali. Tuttavia, gli spazi vuoti inclusi tra le virgolette assumono rilevanza nel confronto con i caratteri globali.
- I parametri numerici non possono includere il punto interrogativo (?). L'asterisco (*) può invece essere utilizzato al posto di un intero parametro numerico, ma non può essere incluso come parte di un parametro numerico. Ad esempio, di seguito sono riportati parametri numerici validi:

MSGTYPE(2) Sono considerati solo i messaggi di risposta

MSGTYPE(*) Sono considerati tutti i tipi di messaggi.

MSGTYPE('*') Sono considerati tutti i tipi di messaggi.

Tuttavia, **MSGTYPE('2*')** non è valido, perché include l'asterisco (*) come una parte di un parametro numerico.

- I parametri numerici devono essere compresi tra 0–999. Se il valore è compreso in questa gamma viene accettato, anche se non è valido per la parola chiave a cui si riferisce. Possono essere utilizzati nomi simbolici per parametri numerici.
- Se l'impostazione di una stringa è più breve del campo nel MQDLH o MQDM cui la parola chiave si riferisce, al valore viene aggiunta una serie di spazi fino al termine del campo. Se invece l'impostazione, compresa di un asterisco, è più lunga del campo che deve ospitarla, viene diagnosticato un errore. Ad esempio, di seguito vengono riportate impostazioni valide per un campo di 8 caratteri:
'ABCDEFGH' 8 caratteri
'A*C*E*G*I' 5 caratteri escluso gli asterischi
'*A*C*E*G*I*K*M*O*'
8 caratteri esclusi gli asterischi
- Le stringhe che contengono spazi vuoti, caratteri minuscoli o caratteri speciali diversi dal punto (.), dallo slash inclinato in avanti (/), dai punti sospensivi (.) o dal segno percentuale (%) devono essere inclusi tra virgolette. I caratteri minuscoli non racchiusi tra virgolette vengono considerati maiuscoli. Se nella stringa sono previste le virgolette, queste dovranno essere racchiuse tra le virgolette doppie, per segnalare l'inizio e la fine del testo racchiuso nelle virgolette. Il carattere delle virgolette doppie viene contato come singolo carattere.

Elaborazione della tabella regole

Il DLQ handler ricerca una tabella regole il cui schema corrisponda alle caratteristiche del messaggio presente nella coda DLQ. La ricerca comincia dalla prima regola della tabella e prosegue in sequenza. Quando viene rilevata una regola con uno schema corrispondente alle caratteristiche del messaggio, viene tentata l'azione prevista da quella regola. Il DLQ handler incrementa il conteggio dei tentativi di uno, ogni volta che prova l'esecuzione di quella regola. Se il primo tentativo fallisce, se ne eseguiranno degli altri, fino a raggiungere il numero massimo stabilito con la parola chiave RETRY. Se tutti tentativi hanno esito negativo, il DLQ ricerca nella tabella la successiva regola corrispondente alle caratteristiche del messaggio.

Questo processo viene ripetuto in sequenza fino a quando un'azione avrà esito positivo. Quando tutte le regole corrispondenti alle caratteristiche del messaggio sono state eseguite, per il numero di volte specificato con la parola chiave `RETRY`, e dopo che tutte hanno avuto esito negativo viene utilizzato `ACTION (IGNORE)`. Tale parola chiave viene utilizzato anche nel caso non venga rilevata alcuna regola corrispondente alle caratteristiche del messaggio.

Note:

1. Gli schemi di confronto delle regole vengono ricercati solo per i messaggi che iniziano con un `MQDLH` presenti sulla coda `DLQ`. I messaggi che non hanno tale intestazione vengono identificati come messaggi affetti da errore e restano nella coda messaggi non recapitati per un tempo indefinito.
2. Per tutte le parole chiave dello schema possono essere utilizzate quelle predefinite, così come una regola può consistere di una sola azione. Si noti, tuttavia, che le regole composte di una sola azione verranno applicate a tutti i messaggi con `MQDDLH` che non sono stati già elaborati in funzione di altre regole della tabella.
3. All'avvio del `DLQ` handler viene rilevata la tabella regole e vengono segnalati gli errori. (I messaggi di errore emessi da `DLQ` handler sono descritti nel manuale *MQSeries Messages*.) È possibile effettuare modifiche in qualunque momento, ma tali modifiche non avranno effetto fino al successivo riavvio del `DLQ` handler.
4. Il `DLQ` handler non modifica il contenuto dei messaggi, del `MQDLH` o del descrittore dei messaggi. Il `DLQ` handler invia sempre i messaggi alle altre code con l'opzione di messaggio `MQPMO_PASS_ALL_CONTEXT`.
5. Il `DLQ` handler apre la coda messaggi non recapitati con l'opzione `MQOO_INPUT_AS_Q_DEF`.
6. È possibile eseguire più istanze del `DLQ` handler contemporaneamente verso la stessa coda, utilizzando la stessa tabella regole. Tuttavia, accade più spesso che venga elaborata un'unica istanza per volta.

Metodi per l'elaborazione di tutti i messaggi presenti nella coda `DLQ`

Il `DLQ` handler conserva un record di tutti i messaggi rilevati nella coda `DLQ` e non ancora rimossi. Se si utilizza il `DLQ` handler come filtro di estrazione dalla coda `DLQ` di un sottogruppo di messaggi, il `DLQ` handler conserverà ancora il record di quei messaggi della coda `DLQ` non elaborati. Inoltre, il `DLQ` handler non può garantire che i nuovi messaggi che arrivano nella coda `DLQ` saranno rilevati, anche se è stata definita l'impostazione `FIFO (First-In-First-Out)`. Perciò, a meno che la coda sia vuota, viene realizzata un'analisi periodica della coda `DLQ` per controllare tutti i messaggi. Per tali motivi, è necessario verificare che il `DLQ` contenga il minor numero di messaggi possibili. Se i messaggi che non possono essere eliminati o inoltrati ad altre code (per qualsiasi ragione) si accumulano, il carico di lavoro del `DLQ` handler aumenta e la stessa coda `DLQ` corre il rischio di riempirsi.

Esistono operazioni apposite per consentire al `DLQ` handler di svuotare la coda messaggi non recapitati. Ad esempio, si consiglia di evitare l'utilizzo del parametro `ACTION (IGNORE)`, poiché questo lascia accumulare i messaggi nella coda `DLQ`. Questo parametro è utilizzato per i messaggi che non hanno un'esplicita corrispondenza nella tabella regole. Al posto di questo parametro, per i messaggi che si desidera ignorare, utilizzare un'azione che sposti i messaggi verso un'altra coda. Ad esempio:

Elaborazione tabella regole

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Analogamente, l'ultima regola della tabella dovrebbe essere impostata in modo da corrispondere a tutti i messaggi; così i messaggi che non hanno trovato corrispondenza con le regole precedenti verranno elaborate in base all'ultima regola. Ad esempio, l'ultima regola della tabella potrebbe assomigliare a:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Questa azione fa sì che tutti i messaggi, che non sono stati elaborati in base alle precedenti regole, vengano inoltrati alla coda REALLY.DEAD.QUEUE, dove poi saranno elaborate manualmente. se non si imposta una regola simile a questa, i messaggi probabilmente rimarranno nella coda DLQ indefinitamente.

Esempio di tabella regole per DLQ handler

Di seguito viene riportato un esempio di tabella regole che contiene una singola voce dati di controllo e diverse regole:

```
*****
*           An example rules table for the runmqdlq command           *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never do things by half measures. If
```

Esempio di tabella regole

- * the queue manager BBBB.1 is unavailable, try to
- * send the message to BBBB.2

```
DESTQM(bbbb.1) +  
  action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)
```

- * The CCCC corporation considers itself very security
- * conscious, and believes that none of its messages
- * will ever end up on one of our DLQs.
- * Whenever we see a message from a CCCC queue manager on our
- * DLQ, we send it to a special destination in the CCCC organization
- * where the problem is investigated.

```
REPLYQM(CCCC.*) +  
  ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

- * Messages that are not persistent run the risk of being
- * lost when a queue manager terminates. If an application
- * is sending nonpersistent messages, it should be able
- * to cope with the message being lost, so we can afford to
- * discard the message.

```
PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
```

- * For performance and efficiency reasons, we like to keep
- * the number of messages on the DLQ small.
- * If we receive a message that has not been processed by
- * an earlier rule in the table, we assume that it
- * requires manual intervention to resolve the problem.
- * Some problems are best solved at the node where the
- * problem was detected, and others are best solved where
- * the message originated. We don't have the message origin,
- * but we can use the REPLYQM to identify a node that has
- * some interest in this message.
- * Attempt to put the message onto a manual intervention
- * queue at the appropriate node. If this fails,
- * put the message on the manual intervention queue at
- * this node.

```
REPLYQM('?*') +  
  ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

Esempio di tabella regole

Capitolo 10. Eventi EMS e di strumentazione

Questo capitolo:

- Fornisce una breve introduzione agli eventi di strumentazione MQSeries che è possibile utilizzare per controllare il funzionamento dei queue manager. Consultare la sezione “Eventi di strumentazione MQSeries”. Per ulteriori informazioni relative agli eventi di strumentazione, consultare il manuale *MQSeries Programmable System Management*.
- Descrive l’utilizzo degli eventi EMS (Event Management Service) da parte di MQSeries for Compaq NSK. Consultare la sezione “Eventi EMS (Event Management Service)” a pagina 173.

Eventi di strumentazione MQSeries

Gli eventi di strumentazione determinano la creazione di *messaggi di eventi* ogni volta che un queue manager rileva una serie di condizioni predefinite. Ad esempio, un evento *Coda Piena* deriva dalle seguenti condizioni:

- Gli eventi Coda Piena possono verificarsi solo per una determinata coda.
- Un’applicazione inoltra una chiamata MQPUT per inviare un messaggio ad una coda, ma la chiamata ha esito negativo poiché la coda è piena.

Altre condizioni che possono determinare un evento di strumentazione sono:

- Viene raggiunto il numero limite di messaggi su una coda.
- Una coda non viene servita nell’intervallo di tempo specificato.
- Viene avviata o arrestata un’istanza del canale.
- Un’applicazione tenta di aprire una coda specificando un>ID utente non autorizzato.

Con l’eccezione degli eventi canale, tutti gli eventi di strumentazione devono essere abilitati prima che si verifichino.

Il messaggio di evento contiene informazioni relative alle condizioni che determinano l’evento. Viene inserito in una *coda eventi*. Un’applicazione può così richiamare il messaggio di evento da questa coda per analizzarlo.

Se si definiscono le code di evento come code remote, è possibile inserire tutte le code di evento in un singolo Queue Manager (per i nodi che supportano gli eventi di strumentazione). Sarà così possibile utilizzare gli eventi prodotti per controllare una rete di Queue Manager da un singolo nodo.

Tipi di eventi

Esistono quattro tipi di eventi di strumentazione:

Eventi Queue Manager

Gli eventi del queue manager si riferiscono alle definizioni delle risorse all’interno i queue manager. Ad esempio, un evento del queue manager può essere generato quando un’applicazione tenta di inserire un messaggio in una coda che non esiste.

Eventi prestazioni

Gli eventi prestazioni rappresentano le notifiche indicanti che una risorsa

Eventi di strumentazione

ha raggiunto una soglia. Ad esempio, un evento di prestazioni può essere generato quando viene raggiunto un limite di capacità di una coda oppure, dopo una chiamata MQGET, se una coda non viene servita entro il limite di tempo prestabilito.

Eventi canale

Questi eventi di canale vengono riportati dai canali come il risultato delle condizioni rilevate durante la relativa operazione. Ad esempio, un evento di canale può essere generato quando si arresta un'istanza del canale.

Eventi trigger

Un evento trigger si verifica quando un rileva il verificarsi delle condizioni relative a un evento trigger. Ad esempio, una coda può essere configurata in modo da produrre un evento trigger tutte le volte che arriva un messaggio. Le condizioni per un evento trigger e quelle per un evento di strumentazione sono leggermente diverse.

Un evento trigger fa sì che un messaggio trigger sia inviato ad una coda di iniziazione e, facoltativamente, venga avviato un programma applicativo.

Notificazione di evento attraverso code di evento

Quando si verifica un evento, il Queue Manager inserisce un messaggio di evento nella coda di evento appropriata, se definita. Il messaggio di evento contiene informazioni relative all'evento che è possibile richiamare compilando un apposito programma applicativo MQI che:

- Estrae il messaggio dalla coda.
- Elabora il messaggio per estrarre i dati dell'evento. Per una descrizione dei formati dei messaggi di evento, consultare il manuale *MQSeries Programmable System Management*.

Ogni categoria di eventi ha la propria coda di evento. Tutti gli eventi di una categoria risultano in messaggio di evento che viene inviato alla stessa coda.

Questa coda di evento...	Contiene messaggi provenienti da...
SYSTEM.ADMIN.QMGR.EVENT	Eventi Queue Manager
SYSTEM.ADMIN.PERFM.EVENT	Eventi prestazioni
SYSTEM.ADMIN.CHANNEL.EVENT	Eventi Channel

Utilizzo delle code di evento con la funzione trigger

È possibile impostare le code di evento con la funzione trigger in modo tale che quando viene generato un evento, il messaggio di evento che viene inserito nella coda evento, avvia un'applicazione di controllo compilata dall'utente. Questa applicazione può elaborare i messaggi di evento ed avviare le dovute azioni. Ad esempio, alcuni eventi possono prevedere una notifica all'operatore, altri eventi possono avviare un'applicazione che esegue varie attività di gestione in modo automatico.

Abilitazione degli eventi di strumentazione

Le modalità di abilitazione di un evento di strumentazione dipendono dal tipo di evento:

- Gli eventi Queue Manager si abilitano impostando gli attributi del Queue Manager.

- Gli eventi prestazione devono essere abilitati integralmente sul queue manager. E' inoltre necessario abilitare specifici eventi prestazione impostando l'appropriato attributo coda e identificare le condizioni, ad esempio il limite superiore della capacità della coda, che determinerà l'evento.
- Gli eventi canale si verificano in modo automatico; non occorre abilitarli. Se si desidera escludere il controllo degli eventi canale, è possibile inibire la coda evento canale.

Abilitare e disabilitare la creazione degli eventi di strumentazione utilizzando i seguenti comandi:

- Comandi MQSC. Per ulteriori informazioni, consultare la sezione *MQSeries Guida di riferimento per i comandi*.
- Comandi PCF per i queue manager. Per ulteriori informazioni, consultare il manuale *MQSeries Programmable System Management*.
- Comandi MQAI. Per ulteriori informazioni, consultare la sezione *MQSeries Administration Interface Programming Guide and Reference*.

Messaggi di evento

I messaggi di evento contengono informazioni relative all'origine di un evento, incluso il tipo di evento, il nome dell'applicazione che ha determinato l'evento e, nel caso degli eventi prestazione, un breve riepilogo statistico della coda.

Il formato dei messaggi di evento è simile a quello dei messaggi di risposta PCF. I dati dei messaggi possono essere richiamati dai messaggi evento grazie ad un programma di gestione compilato dall'utente, utilizzando le strutture dei dati descritte nel manuale *MQSeries Programmable System Management*.

Eventi EMS (Event Management Service)

MQSeries for Compaq NSK genera messaggi di evento EMS (Event Management Service) che corrispondono agli eventi del queue-manager di MQSeries, agli eventi di canale e agli eventi prestazione. Possono inoltre essere generati dei messaggi EMS corrispondenti alle voci del messaggio di log di MQSeries e degli FFST. Tali messaggi di evento possono segnalare agli operatori o ai responsabili del sistema le condizioni software che potrebbero danneggiare il sistema operativo MQSeries.

I file modello EMS forniti con MQSeries for Compaq NSK

I seguenti file vengono forniti nel volume secondario ZMQSSYS:

ZMQSTMPL (file code 839)

Un file degli oggetti modello EMS contenenti la formattazione dei modelli relativi agli eventi EMS generati da MQSeries.

ZMQSDDL (file code 101)

Lo schema DDL (Data Definition Language) relativo agli eventi EMS generati da MQSeries.

ZMQSC (file code 101)

Output compilato (C) dal programma di compilazione DDL delle definizioni degli eventi EMS generati dal prodotto.

ZMQSCOB (file code 101)

Output compilato (COBOL) dal programma di compilazione DDL delle definizioni degli eventi EMS generati dal prodotto.

eventi EMS

ZMQSPAS (file code 101)

Output compilato (PASCAL) dal programma di compilazione DDL delle definizioni degli eventi EMS generati dal prodotto.

ZMQSTACL (file code 101)

Output compilato (TACL) dal programma di compilazione DDL delle definizioni degli eventi EMS generati dal prodotto.

ZMQSTAL (file code 101)

Output compilato (TAL) dal programma di compilazione DDL delle definizioni degli eventi EMS generati dal prodotto.

Il volume secondario ZMQSSYS contiene il file dei modelli EMS SMQSTMPL, da cui viene generato il file dei modelli ZMQSTMPL. Il file ZMQSTMPL è pronto per l'integrazione con i modelli di evento del sistema che utilizzano COUP e SYSGEN. L'origine dei modelli di evento viene fornita in modo da consentire la modifica della formattazione degli eventi, quando vengono utilizzati nel proprio ambiente.

Ad esempio, è possibile non visualizzare tutte le informazioni contenute in un evento oppure è possibile modificare o aggiungere del testo alle informazioni contenute nell'evento. Consultare la documentazione Compaq per una descrizione del linguaggio dell'origine dei modelli di evento EMS e per le procedure utilizzate per compilare le definizioni per la creazione di un file ZMQSTMPL alternativo.

Integrazione dei modelli di evento EMS MQSeries

E' necessario integrare il file oggetto del modello nei file modello EMS residenti e non residenti, in modo che i programmi, quali VIEWPOINT e EMSDIST, possano formattare e visualizzare gli eventi EMS MQSeries.

Una procedura per l'integrazione dei modelli EMS MQSeries con i modelli del sistema viene descritta nella parte finale di questa sezione. Tenere presente che è possibile utilizzare varie procedure nella propria installazione.

1. Determinare i nomi dei modelli del sistema corrente utilizzando il comando COUP INFO ALLPROCESSORS: si osservino i valori visualizzati per il parametro EMS^TEMPLATES. Ad esempio:

```
$DEV2 ZMQSSYS 425> coup
CONFIGURATION UTILITY PROGRAM - T9023D30 - (26MAY95)    SYSTEM
\RAPTOR
COPYRIGHT Compaq COMPUTERS INCORPORATED 1987-1994
CONFIG $SYSTEM.SYS06.OSCONFIG
1) info allprocessors
   EMS^TEMPLATES ( RESIDENT $SYSTEM.SYS01.RTMPLATE,
                   NONRESIDENT $SYSTEM.SYS01.RTMPLATE )
   SYSTEM^ID      ( NAME \RAPTOR, NUMBER 001 )
   SYSTEM^TIME    ( GMT^OFFSET -05:00, DST USA66 )
   DP2_UPSOPTION ( OFF )
2) exit
```

2. Determinare il nome del modello di sistema corrente per un sistema operativo G Series utilizzando il comando SCF ASSUME SUBSYS \$ZZKRN; INFO. Si osservino i valori visualizzati per NONRESIDENT_TEMPLATES e RESIDENT_TEMPLATES. Ad esempio:

```

SCF;ASSUME SUBSYS $ZZKRN;INFO
NONSTOP KERNEL - Info SUBSYS \HAWK.$ZZKRN
Current Settings
*DAYLIGHT_SAVING_TIME ..... USA66
*NONRESIDENT_TEMPLATES..... $SYSTEM.SYS01.TEMPLATE
*POWERFAIL_DELAY_TIME..... 30
*RESIDENT_TEMPLATES..... $SYSTEM.SYS01.RTMPLATE
SUPER_SUPER_IS_UNDENIABLE..... OFF
*SYSTEM_NAME..... \HAWK
*SYSTEM_NUMBER..... 2
SYSTEM_PROCESSOR_TYPE ..... NSR-W
*TIME_ZONE_OFFSET..... -05:00

Pending Changes (will take effect at next system load)
None Total Errors = 0 Total Warnings = 0

```

3. Eseguire il programma di compilazione TEMPLI per creare nuovi file di modello di sistema, combinando i modelli di sistema correnti con i nuovi modelli MQSeries. E' un processo suddiviso in due fasi:
 - a. Creare un file di testo contenente i seguenti comandi:

```

FILE <file di modello di sistema corrente NONRESIDENT>
FILE <volume di installazione MQSeries>.ZMQSSYS.ZMQSTMPL
EXIT

```

Ad esempio:

```

FILE $SYSTEM.SYS06.TEMPLATE
FILE $DEV2.ZMQSSYS.ZMQSTMPL
EXIT

```

- b. Eseguire il programma di compilazione TEMPLI specificando il nuovo file di testo come input:

```

TEMPLI /IN <file di comandi>/<nuovo file di modello residente>, <nuovo
file di modello non residente>

```

Ad esempio, se il file di comandi creato è denominato TEMGUIDE e vengono creati nuovi file di modello in \$SYSTEM.EMS:

```

TEMPLI /IN TEMGUIDE/$SYSTEM.EMS.NEWRES, $SYSTEM.EMS.NEWRES

```

La compilazione dei nuovi file di modelli può richiedere vari minuti, in quanto vengono elaborati tutti i modelli di evento EMS richiesti sul sistema.

4. Utilizzando il comando COUP, configurare il sistema in modo da utilizzare i nuovi modelli di evento EMS in sostituzione dei modelli correnti:

eventi EMS

```
ASSUME ALLPROCESSORS
ALTER EMS`TEMPLATES(RESIDENT <nuovo file di modello residente>,
                    NONRESIDENT <nuovo file di modello non residente>)
EXIT
```

Per G Series utilizzare i comandi SCF per configurare il sistema in modo da utilizzare i nuovi modelli di evento EMS:

```
ALTER $ZZKRN, RESIDENT_TEMPLATES $SYSTEM.SYS01.NEWRES
ALTER $ZZKRN, NONRESIDENT_TEMPLATES $SYSTEM.SYS01.NEWNRES
EXIT
```

Nota: Per rendere questa modifica permanente, è necessario aggiornare il sistema utilizzando SYSGEN.

Per ulteriori informazioni relative ai modelli EMS, consultare la pubblicazione *DSM Template Services Manual* di Compaq. In questo manuale viene inoltre descritto come utilizzare SYSGEN per eseguire questa attività.

Definizione di PARAM MQEMSEVENTS

Per completare l'abilitazione degli eventi EMS MQSeries, è necessario accertarsi che PARAM MQEMSEVENTS sia definito correttamente. Il valore è una stringa composta da quattro caratteri interpretato come bitmap, come segue:

Messaggio EMS	Voce Bit-map	Valore MQEMSEVENT
FFST	0x00000001	1
START / STOP	0x00000002	2
PERFORMANCE	0x00000004	4
CHANNEL	0x00000008	8
QUEUE MANAGER	0x00000010	16
MESSAGE	0x00000020	32
ERROR	0x00000040	64
ALL	0x0000007F	127

Ad esempio, per attivare tutti gli eventi EMS di MQSeries, è necessario definire il seguente PARAM in ambiente TAQL da cui vengono immessi i comandi di gestione:

```
PARAM MQEMSEVENTS 127
```

Questa definizione è inoltre obbligatoria nelle definizioni di tutte le classi del server di MQSeries. E' possibile configurare ciascuna classe di server con differenti opzioni. Consultare la sezione "Modifica dei parametri delle classi di server PATHWAY" a pagina 42 per ulteriori informazioni.

Per impostazione predefinita, non viene generato alcuno evento EMS (i PARAM quindi, non vengono definiti).

Utilizzo di un collector alternativo

Su un sistema Compaq NSK, il collector dell'evento EMS predefinito viene denominato \$0, ed è sempre presente. Tutti gli eventi EMS generati da un queue manager MQSeries vengono inviati al collector predefinito. Se si desidera un collector differente per raccogliere gli eventi EMS relativi a un queue manager, modificare la voce EMSCollector nella stanza Configuration nel file QMINI e riavviare il queue manager. E' possibile specificare un differente collector di evento EMS per ciascun queue manager.

Scrittura dei programmi per l'elaborazione degli eventi EMS MQSeries

E' possibile scrivere un'applicazione per controllare un queue manager MQSeries elaborando i messaggi di evento EMS. Un'applicazione di questo tipo potrebbe influire sul funzionamento del queue manager immettendo i comandi PCF in risposta ai messaggi di evento EMS generati.

I file ZMQSC, ZMQSTAL, ZMQSCOB, ZMQSPAS e ZMQSTACL forniti con MQSeries for Compaq NSK nel volume secondario ZMQSSYS definiscono i token contenuti nei messaggi di evento EMS di MQSeries in C, TAL, COBOL, PASCAL e TACL. Queste definizioni possono inoltre essere utilizzate da un programma di gestione per interpretare il formato dei messaggi.

Per ulteriori informazioni relative agli eventi EMS generati da MQSeries, consultare "Appendice P. Modello di evento EMS utilizzato da MQSeries for Compaq NSK" a pagina 405 .

Capitolo 11. Descrizione del supporto transazionale e della messaggistica

Le applicazioni che utilizzano la MQI (Message Queue Interface) consentono di eseguire operazioni di inserimento e di estrazione sotto il controllo di un syncpoint. In MQSeries for Compaq NSK, ci sono due operazioni transazionali:

- Commit — (o sincronizzazione); l'operazione di completamento di una transazione al fine di registrare e rendere permanenti le modifiche apportate al database. Le risorse protette vengono rilasciate dopo che è stato eseguito il commit di una transazione.
- Back out — un'operazione che inverte tutte le modifiche apportate durante la corrente unità di ripristino o unità di lavoro. Al termine dell'operazione, è possibile avviare una nuova unità di ripristino o unità di lavoro.

Le operazioni di commit e di backout vengono fornite come parte dell'ambiente di transazione TM/MP (o TMF) in Compaq NSK. In MQSeries for Compaq NSK, MQPUT, MQGET e MQPUT1 sono operazioni di cui viene eseguito il syncpoint per impostazione predefinita. Questo significa che, tranne nel caso in cui l'esecuzione del syncpoint è richiesta esplicitamente da un'applicazione, è *necessario* che sia in corso una transazione TMF altrimenti la chiamata MQI avrà esito negativo.

Un'applicazione avvia una transazione TM/MP utilizzando la procedura TM/MP BEGINTRANSACTION, esegue il commit della transazione utilizzando ENDTRANSACTION e può eseguire il backout della transazione utilizzando ABORTTRANSACTION.

MQSeries for Compaq NSK supporta inoltre i prodotti che forniscono l'operazione di syncpoint via TM/MP, ad esempio NonStop Tuxedo.

Utilizzo di NonStop TM/MP (Transaction Manager)

MQSeries for Compaq NSK V5.1 si serve delle funzioni di gestione delle transazioni di Compaq NonStop TM/MP per conservare l'integrità delle transazioni.

Il sistema di transazione NonStop TM/MP fornisce la simultaneità e la protezione transazionale e l'integrità dei messaggi e dei cataloghi di oggetto.

Le transazioni TM/MP possono coordinare le operazioni di inserimento e di estrazione di MQSeries con aggiornamenti ai file di database ENSCRIBE o NonStop SQL eseguite dall'applicazione.

MQSeries gestisce le transazioni TM/MP in modo trasparente. Se c'è una transazione TM/MP in corso quando viene richiamata una funzione MQI, le opzioni di inserimento e di estrazione con l'opzione syncpoint diventano parte della stessa transazione. Questo significa che gli aggiornamenti alle code si verificano quando viene eseguito il commit della transazione. In caso di malfunzionamento, TM/MP assicura che tutte le transazioni di cui è stato eseguito il commit vengano applicate ai file di database e che venga eseguito il backout delle transazioni per cui non è stato eseguito il commit. Un backout di una transazione riapplica le immagini precedenti l'apporto di modifiche correnti ai

Utilizzo di TM/MP

record di database per annullare gli effetti di una transazione annullata. Le modifiche vengono applicate solo dopo che è stata completata l'operazione di commit.

Se l'applicazione utente ha una transazione in corso e tenta una chiamata MQI fuori da un syncpoint, MQSeries sospende la transazione corrente, ereditata, ne avvia una propria, esegue la transazione della transazione in oggetto e ripristina la transazione originaria prima di restituire il controllo all'applicazione utente. Gli aggiornamenti alle code determinati dalle operazioni di inserimento e di estrazione vengono eseguiti immediatamente.

Le transazioni TM/MP vengono utilizzate per coordinare le operazioni di inserimento ed estrazione su messaggi non permanenti come pure su quelli permanenti. MQSeries for Compaq NSK utilizza un'interfaccia speciale per TM/MP chiamata *OpenTMF* per consentire tale coordinazione.

Limiti di Syncpoint

Il file system può limitare la quantità di dati di messaggi permanenti che è possibile immettere o ricevere entro una singola transazione limitando il numero di blocchi di record sui file fisici che conservano i dati dei messaggi permanenti.

Il limite blocco predefinito del file system per transazione è 5000 per volume di disco. E' possibile modificarlo utilizzando SCF. Ad esempio, il seguente comando modifica il limite in 10000 blocchi per transazione per volume di disco:

```
ALTER DISK $DISK01, MAXLOCKSPERTCP 10000
```

Per i messaggi memorizzati nei file overflow di coda (la loro dimensione è inferiore al limite per l'uso dei file overflow di messaggi) il numero e la dimensione dei messaggi è limitato.

E' preferibile impostare la dimensione limite overflow dei messaggi non oltre i 200 KB. Con questa dimensione di messaggi, il limite del blocco record predefinito può arrivare a circa 100 messaggi entro una singola transazione — sicuramente adeguata per più applicazioni. Inoltre, i vantaggi dell'utilizzo di file overflow di messaggi diventa rilevante con queste dimensioni di messaggi.

Notare inoltre che l'uso della traccia di controllo TM/MP è ampiamente ridotta con i file overflow di messaggi utilizzati al posto dei file overflow di coda.

Per ulteriori informazioni sulle differenze e i vantaggi dei file overflow di coda e dei messaggi, consultare "File di overflow messaggi" a pagina 224.

Operazioni diverse da syncpoint in messaggi permanenti

I messaggi permanenti richiedono che le transazioni TM/MP vengano avviate internamente dal queue server per aggiornare i file ENSCRIBE che conserva i dati dei messaggi. Esiste un limite imposto da NSK File System di 100 transazioni contemporanee avviate da qualsiasi processo. Quindi, un singolo queue server non può supportare più di 100 operazioni PUT o GET di messaggi permanenti diverse da syncpoint contemporanee. L'unico modo è se vari messaggi di grosse dimensioni (superiori a 32 KB) vengono messi in coda o tolti dalla coda allo stesso tempo attraverso lo stesso queue server.

Se si verifica tale situazione, MQPUT o MQGET vengono terminati dal codice ragione MQRC_SYNCPOINT_LIMIT_REACHED. Riassegnare le code per alternare i queue server per distribuire il carico su più processi, o modificare le applicazioni per usare varie code in queue server diversi.

Operazioni Syncpoint o messaggi non permanenti

Poiché i messaggi non permanenti vengono memorizzati non nei file del disco controllato, non richiedono spazio di controllo di traccia. MQSeries utilizza un'interfaccia interna di TM/MP per controllare la disponibilità di messaggi non permanenti messi in coda o tolti dalla coda in syncpoint. Ogni mescolanza di messaggi permanenti e non permanenti può essere inclusa in un'operazione syncpoint—MQSeries garantisce che le transazioni TM/MP vengano completate, verrà eseguito il commit delle operazioni su tutti i messaggi non permanenti, in base all'attuale risultato della transazione.

Requisiti della configurazione per TM/MP e MQSeries for Compaq NSK

Il Sistema NSK deve essere configurato con il controllo TMF (TM/MP) abilitato per tutti i volumi che contengono queue manager o code. Utilizzare il comando TMFCOM status datavols per determinare lo stato di controllo su ogni volume del sistema. (Notare che occorre SUPER.SUPER per utilizzare TMFCOM.) Inoltre, le tracce di controllo TMF configurate per i volumi di dati che supportano i queue manager devono essere abbastanza ampie per consentire il livello massimo e la dimensione dei messaggi prevista su tutti i queue manager che utilizzano tali volumi.

Poiché le applicazioni errate possono causare transazioni di lunga durata, il sistema TMF viene configurato automaticamente per cancellare tali transazioni. La dimensione della traccia di controllo e il tempo limite per le transazioni di lunga durata, sono parametri di ottimizzazione dipendenti dall'applicazione. La traccia di controllo configurata per MQSeries non deve essere configurata per l'esecuzione del dump su nastro.

Monitoring

Utilizzare l'interfaccia TMFCOM per controllare lo stato di TMF, con MQSeries in esecuzione. Utilizzare i comandi status tmf e status datavols per controllare lo stato generale di TMF e lo stato dei volumi dei dati individuali.

L'EMS (System event log) deve essere controllato per eventi principali TM/MP che indicano problemi potenziali futuri con TM/MP e che potrebbero danneggiare MQSeries o le applicazioni che lo utilizzano. TM/MP è una risorsa critica per MQSeries e deve operare continuamente affinché MQSeries funzioni correttamente.

Dimensione traccia di controllo

Una dimensione approssimativa della traccia di controllo TM/MP può essere calcolata utilizzando le seguenti istruzioni:

- Lo spazio per la traccia di controllo è richiesto solo per le operazioni dei messaggi permanenti (inserimento ed estrazione).
- Lo spazio per la traccia di controllo deve corrispondere alla dimensione totale dei messaggi circa più 1500 byte.
- Le operazioni dei messaggi permanenti per messaggi che superano la soglia di overflow dei messaggi richiedono 4 KB della traccia di controllo per le operazioni di estrazione o inserimento, indipendentemente dalle loro dimensioni effettive.

Configurazione gestore risorse

L'interfaccia interna di TM/MP deve essere configurata correttamente per il volume di transazioni che devono essere elaborate con MQSeries. I queue server MQSeries fungono da gestore risorse purché il sistema secondario TM/MP sia coinvolto ed esistono varie soglie e limiti in esso applicabili ai gestori risorse. La configurazione richiesta dipende dal numero di queue server in uso, la distribuzione di queue server attraverso i CPU e da come molte operazioni syncpoint simultanee siano in corso allo stesso tempo. Il comando ALTER BEGINTRANS di TMFCOM è utilizzato per modificare i valori nel modo seguente:

RMOPENPERCPU - deve essere almeno il doppio del numero massimo di queue server in esecuzione in qualsiasi CPU. Il valore predefinito 128 è di solito sufficiente.

BRANCHESPERRM - deve essere almeno il numero massimo di operazioni syncpoint simultanee gestibili da singoli queue server. Il valore predefinito 128 è di solito sufficiente, in caso contrario il parametro può essere modificato in 1024, o Queues può assegnare ad altri queue server per ridurre il numero massimo di operazioni syncpoint simultanee gestite da un queue server.

Perché i nuovi valori di tali parametri siano effettivi, il sistema secondario TM/MP deve essere arrestato e riavviato.

Risoluzione dei problemi

Gli eventi EMS o prospettati FFST che indicano che i comandi BEGINTRANSACTION sono stati disabilitati da TMF, indica che la traccia di controllo è completa. Ciò si può verificare poiché la traccia di controllo è troppo piccola, o perché un'applicazione non corretta ha conservato una transazione di lunga durata e TMF non è stato terminato in tempo.

In questa istanza:

- Aumentare la dimensione della traccia di controllo o
- identificare la causa della transazione di lunga durata e correggerla o
- riconfigurare TMF per terminare le transazioni di lunga durata dopo un breve periodo.

Gli eventi EMS e prospettati FFST che indicano che TMF non è in esecuzione indicano che esiste un problema di configurazione con TMF che deve essere corretto prima di eseguire queue manager di nuovo. In generale, MQSeries queue manager richiede che TMF sia in esecuzione correttamente per operare. Sebbene i messaggi non siano persi o danneggiati, il queue manager non può operare senza TMF.

Capitolo 12. Ripristino e riavvio

Un sistema di messaggistica garantisce che i messaggi immessi nel sistema siano consegnati alla relativa destinazione. Un sistema di messaggio deve fornire un metodo di verifica dei messaggi nel sistema e di ripristino dei messaggi se il sistema dovesse subire un guasto per un qualsiasi motivo.

MQSeries for Compaq NSK verifica che i messaggi permanenti non siano persi utilizzando Compaq NonStop Transaction Manager (TM/MP). TM/MP fornisce la protezione di transazioni, la consistenza tra file e coda e il ripristino tra file e coda.

MQSeries for Compaq NSK utilizza la tecnologia di coppia di processo NonStop per verificare che anche i messaggi non persistenti siano facili ad errori. I queue server che sono responsabili per la memorizzazione dei messaggi non persistenti di checkpoint sui relativi processi di backup sono in esecuzione in diversi CPU.

Il checkpoint di messaggi non persistenti è l'opzione per coda che è possibile configurare utilizzando **altnqfls**. Il checkpoint di messaggi non persistenti è impostato in modo predefinito.

Il sistema secondario TM/MP gestisce le complesse operazioni per le transazioni correnti e la consistenza del database, operazioni dell'utente e le operazioni di MQSeries, rendendo queste operazioni trasparenti sui programmi applicativi e dell'utente.

Il ripristino riporta Queue Manager allo stato in cui si trovava quando è stato arrestato. Su tutte le transazioni viene effettuato un roll back rimuovendo dalle code qualsiasi messaggio sui cui è stato eseguito il commit al momento in cui il Queue Manager è stato arrestato. Il ripristino recupera tutti i messaggi permanenti; i messaggi non permanenti vengono persi durante il processo.

La parte successiva di questo capitolo illustra i concetti di ripristino e riavvio più dettagliatamente e quindi indica le modalità di ripristino nell'eventualità si verificassero dei problemi. Tratta i seguenti argomenti:

- "Ripristino e tolleranza di guasti"
- "Back up e ripristino di MQSeries" a pagina 184

Ripristino e tolleranza di guasti

Se si configura il prodotto MQSeries Version 5.1 e il software di sistema Compaq NSK e l'hardware correttamente (ad esempio, tutti i componenti sono configurati come dispositivi con mirroring o coppie di processo come prescritto da Compaq), l'errore di qualsiasi componente hardware o software non risulta nella corruzione o duplicazione o perdita di dati o nella perdita permanente (che richiede un intervento esterno) di qualsiasi funzione del sistema. MQSeries for Compaq NSK V5.1 può ripristinare da un singolo errore durante la manutenzione dell'integrità dei dati come sopra riportato.

Errore consecutivo ripetuto (ad esempio, il loop di ripristino) dello stesso componente software viene intrappolato una volta che un numero massimo configurato di errore viene superato. In alcuni esempi o in caso di diversi errori, il prodotto MQSeries non può conservare l'integrità della coda.

Tolleranza guasti

Per ulteriori informazioni sull'impostazione di un queue manager per l'integrità e la disponibilità dei dati, consultare "Capitolo 16. Integrità e disponibilità dei dati" a pagina 231.

Back up e ripristino di MQSeries

Periodicamente, è possibile effettuare un backup dei dati del Queue Manager per proteggere il sistema da possibili danni dovuti a malfunzionamenti dell'hardware.

Esecuzione del backup di MQSeries

Per eseguire un backup di dati di un Queue Manager, è necessario:

1. Assicurarsi che il Queue Manager non sia in esecuzione.

Se il Queue Manager è in esecuzione, arrestarlo con il comando **endmqm**.

Nota: se si tenta di eseguire il backup di un Queue Manager in esecuzione, il backup può non essere congruente a causa degli aggiornamenti in corso quando è stata effettuata la copia dei file.

2. Individuare i volumi secondari e i volumi sotto cui il queue manager memorizza i propri dati.

Per individuare tali directory è possibile utilizzare le informazioni contenute nei file di configurazione. Per ulteriori informazioni, consultare il "Capitolo 13. File di configurazione" a pagina 187.

Nota: ciò avviene perché i nomi vengono trasformati allo scopo di assicurarne la compatibilità con la piattaforma su cui si utilizza MQSeries. Per ulteriori informazioni sulle trasformazioni del nome, consultare il "Struttura dei volumi" a pagina 61.

3. Effettuare copie di tutti i dati e delle directory di file di log di Queue Manager. Controllare che non vi siano file aperti.

Ripristino di MQSeries

Per ripristinare un backup di dati di un Queue Manager, è necessario:

1. Assicurarsi che il Queue Manager non sia in esecuzione.
2. Individuare i volumi secondari sotto cui il queue manager memorizza i propri dati. Queste informazioni si trovano nel file di configurazione.
3. Svuotare i volumi secondari nei quali si registreranno i dati del backup.
4. Copiare i dati di backup del Queue Manager nelle corrette posizioni.

Controllare la struttura della directory risultante per assicurarsi di disporre di tutte le directory necessarie.

Verificare che i file di configurazione di MQSeries e di Queue Manager siano congruenti in modo che MQSeries possa cercare i dati ripristinati nelle posizioni corrette.

Se il precedente backup e il successivo ripristino dei dati sono avvenuti correttamente, il Queue Manager si avvierà.

Ripristino e riavvio degli status server e dei queue server

Le elaborazioni di status server e del queue server sono coppie di processi Compaq NSK. Ciò significa che essi sono designati per continuare a fornire i relativi servizi nell'evento di un errore di una singola unità CPU o del processo stesso. In caso di un singolo errore (ad esempio, il CPU che contiene l'elaborazione primaria dello status server o del queue server ha esito negativo oppure lo stesso processo primario dello status server o del queue server ha esito negativo) il processo dello status server o del queue server di backup funziona come un nuovo processo primario senza interruzione di elaborazione del queue manager.

In normali situazioni di un singolo errore, non è richiesta alcuna azione di ripristino specifica allo status server o al queue server. Un messaggio viene collegato al terminale iniziale e al file di log di messaggi da uno status server o dal queue server quando il backup viene riavviato dal processo primario o dal backup che funziona come primario.

In caso di diversi errori gravi (ad esempio, un errore di ambiente che evita l'inizializzazione del server o di queue server primario o di backup) il processo TS/MP PATHMON tenta di riavviare lo status server o queue server fino a 10 volte.

Lo status server o il queue server accede solo ai database protetti da TM/MP, in modo tale che, in caso di errori, l'influenza di accesso ai dischi, la protezione fornita da TM/MP e dal sottosistema del disco DP2 può essere considerata.

Lo status server o il queue server può essere arrestato e riavviato individualmente utilizzando i comandi TS/MP PATHCOM se necessario, benché non sia richiesto normalmente. Il comando **strmqm** avvia tutte le classi di status server o di queue server in modo automatico; esse hanno nomi che iniziano con una stringa di carattere MQS-STATUS o MQS-QUEUE. Alla chiusura del queue manager, tutte le classi di status server o di queue server attive coordinano la propria chiusura e, quando sono chiuse tutte le connessioni del queue manager attivo, qualsiasi status server o queue server relativo a tali connessioni viene chiuso.

Compaq NSK interrompe le transazioni in caso di errori di un processo Primario di una coppia di processo NonStop. Ciò può provocare l'errore delle operazioni MQPUT o MQGET durante il verificarsi di un errore del processo Primario. Consultare la sezione "Capitolo 16. Integrità e disponibilità dei dati" a pagina 231 per ulteriori informazioni.

Ripristino mediante RDF

Le seguenti procedure dovrebbero essere utilizzare per fornire un queue manager sul sito di backup, se un errore grave si verifica sul sito primario:

1. Verificare che RDF abbia completato l'aggiornamento dei database.
2. Utilizzare FUP per impostare il flag di traccia per i seguenti file:
 - a. Tutti i file nel volume secondario <qmgr>M.
 - b. Tutti i file nel volume secondario <qmgr>D, eccetto QMINI, AMQRFNxx, CCSIDMEM, QMINIMEM, STABLE, UMQSINI, PATHCTL, TRACEOPT e SHUTDOWN.
 - c. Tutti i file nel volume secondario <qmgr>S.
3. Impostare il volume predefinito volume nel volume secondario <qmgr>D. Eseguire sino PATHMON manualmente, eseguire PATHCOM e caricare la configurazione PATHWAY per il queue manager.

Ripristino e riavvio degli status server e dei queue server

4. Eseguire un comando INFO su tutti gli oggetti.
 - a. Verificare che tutte le istanze del nome del nodo siano visualizzate come "`*`" o dispongano di un corretto nome del nodo per il sistema di backup.
 - b. Verificare che i numeri CPU assegnati alle classi del server siano ancora validi per questo sito di backup.
 - c. Verificare che i nomi del file Home Terminal e Out siano validi per il sito di backup. In caso contrario, modificarli.
 - d. Verificare che qualsiasi nome di processo TCP/IP alternativo sia valido per questo sistema.
 - e. Se il sito di backup non è configurato per l'operazione RDF, rimuovere qualsiasi impostazione PARAM MQRDF dalle definizioni di classe del server EC.
 - f. Una volta verificata la configurazione di PATHWAY, salvarla su disco mediante il comando shutdown2 ed uscire da PATHCOM. Se il nome del terminale iniziale è stato modificato, modificare il file QMINI del queue manager per corrispondere alla configurazione di PATHWAY. Se necessario, modificare la configurazione delle porte listener TCP/IP nel file QMINI.
 - g. Utilizzare **strmqm** per avviare il queue manager.
 - h. Utilizzando il comando **runmqsc**, verificare la configurazione di canale e regolarla, se necessario.
 - i. Se si tenta di attivare gli stessi canali come quando erano in esecuzione, la configurazione del canale sui queue manager remoti potrebbe essere modificata a meno che il sistema di backup possa essere riconfigurato per utilizzare lo stesso indirizzo IP del nome host, dei canali TCP/IP o delle stesse risorse SNAX/APC e ICE (ad esempio, nome del processo, nomi LU e così via) per i canali SNA.
 - j. Preparare la sincronizzazione di canale o errori di sequenza, in particolare modo se i canali del sito primario sono stati eseguiti nel momento in cui si è verificato l'errore grave. RDF non verifica che i database sul sito di backup siano aggiornati (in lockstep con il primario) in modo tale che i dati possano essere perduti come risultato di un completo disastro. Per ridurre tali possibilità, verificare che la configurazione RDF possa gestire il volume degli aggiornamenti del database associati al flusso di messaggi.

Capitolo 13. File di configurazione

MQSeries for Compaq NSK utilizza i *file di configurazione* per memorizzare le informazioni di base sulla configurazione del prodotto. Questo capitolo descrive i tipi di file di configurazione e il modo in cui utilizzarli per modificare alcune funzioni dei Queue manager. Contiene le seguenti funzioni:

- “Cosa sono i file di configurazione?”
- “File di configurazione MQSeries (MQSINI)”
- “File di configurazione Queue manager (QMINI)” a pagina 189
- “Modifica dei file di configurazione” a pagina 197

Cosa sono i file di configurazione?

I file di configurazione definiscono valori facoltativi per singoli queue manager e per MQSeries sul nodo come un unico valore. Questi file vengono indicati anche come file *ini* o *stanza*. Un file di configurazione contiene una o più stanze, che sono gruppi semplificati di linee nel file che insieme svolgono una comune funzione o definiscono una parte del sistema. Ad esempio, funzioni log, funzioni canale o servizi installabili.

I file di configurazione possono essere modificati automaticamente, utilizzando dei comandi che cambiano la configurazione dei queue manager sul nodo, oppure manualmente, editando tali comandi. In generale, comunque, i file di configurazione non dovrebbero essere modificati manualmente mentre i queue manager sono in esecuzione.

Esistono due tipi di file di configurazione:

- I file di configurazione di *MQSeries*, *MQSINI*, che specificano i valori per MQSeries sul nodo come un solo valore. Normalmente, esiste un solo file di configurazione MQSeries per ogni nodo.
- I file di configurazione di *Queue manager*, *QMINI*, che specificano i valori per determinati queue manager. Esiste un file di configurazione per ogni Queue Manager sul nodo.

File di configurazione MQSeries (MQSINI)

Il file di configurazione MQSeries, *MQSINI*, contiene informazioni rilevanti per tutti i Queue Manager sul nodo di installazione di MQSeries. Esso viene creato automaticamente durante l'installazione. In particolare, il file di configurazione MQSeries è utilizzato per cercare i dati associati a ciascun Queue Manager. Tale file è ubicato per impostazione predefinita nel volume secondario ZMQSSYS, \$SYSTEM.ZMQSSYS.MQSINI. Viene fornita una variabile di ambiente, *MQMACHINIFILE*, da utilizzare sui sistemi in cui per il file di configurazione MQSeries non è stato specificato alcun nome (o ubicazione) predefinito.

Contenuto del file di configurazione MQSeries

Il file *MQSINI* contiene i valori predefiniti per l'installazione, i nomi dei Queue Manager, il nome del Queue Manager predefinito e l'ubicazione dei file associati. Le seguenti stanze possono essere visualizzate in *MQSINI*:

file di configurazione MQSeries

AllQueueManagers

Specifica i valori per i volumi e le ubicazioni dei file di installazione.

DefaultQueueManager

Specifica il queue manager predefinito per l'installazione. Questo queue manager elabora ogni comando MQSC per il quale non è espressamente specificato il nome del Queue Manager. La stanza è aggiornata automaticamente se viene creato un nuovo Queue Manager predefinito. Se il nuovo Queue Manager viene creato involontariamente, e si desidera riabilitare il precedente, è necessario modificare questa stanza manualmente.

QueueManager

Esiste una stanza per ogni Queue Manager. La stanza QueueManager specifica il nome del queue manager e l'ubicazione dei file associati a questo queue manager. I nomi di questi file derivano dal nome del Queue Manager, ma vengono modificati se il nome del Queue Manager non è un valido nome di file.

La Figura 42 mostra un file MQSINI di esempio.

```
#####  
** Module Name: MQSINI                               **  
** Type:      MQSeries machine-wide ini file        **  
** Function:   Define configuration data for all queue managers **  
**                               **  
#####  
** Notes      :                                     **  
** 1) This file defines configuration data for all queue managers **  
**                               **  
#####  
AllQueueManagers:  
  MQSVolume=$DATA00                                /Volume for the installation  
  MQSExePath=$DATA00.ZMQSEXE                       /Location of product executables  
  QMDefaultVolume=$DATA00                          /Default volume for queue manager creation  
  ConvEBCDICNewline=NL_TO_LF                       /Data Conversion EBCDIC Newline  
  NSKSegidRange=10-20                              /Segment Id Range  
QueueManager:  
  Name=MT01                                         /A queue manager called MT01  
  QMVolume=$DATA00                                 /Volume of the queue manager  
  QMSubvolume=MT01                                 /Subvolume prefix for the queue manager  
DefaultQueueManager:  
  Name=MT01                                         /Name of the default queue manager (optional)
```

Figura 42. File di configurazione MQSeries di esempio (MQSINI). Il file MQSINI viene inizializzato durante l'installazione in base alle informazioni relative al volume principale e a quello secondario.

Nota: Dal momento che il file di configurazione MQSeries viene utilizzato per localizzare i dati associati ai Queue manager, un file di configurazione inesistente o non valido provocherebbe il mancato funzionamento di alcuni o tutti i comandi MQSeries. Inoltre, le applicazioni non potrebbero connettersi ad un Queue Manager non definito in un file di configurazione MQSeries.

File di configurazione Queue manager (QMINI)

Un file di configurazione Queue Manager, QMINI, contiene informazioni rilevanti per uno specifico Queue Manager. Esiste un file di configurazione per ogni Queue Manager. Viene creato automaticamente insieme al Queue Manager a cui è associato.

Questo file è ubicato nel volume secondario del queue manager. Ad esempio, il percorso ed il nome relativo ad un file di configurazione di Queue Manager definito QMNAME dovrebbe essere \$VOLUME.QMNAME.QMINI.

Nota: il nome di un Queue Manager può avere la lunghezza massima di 48 caratteri. Il nome del volume secondario viene creato basandosi sul nome del Queue Manager. Questo processo è definito *trasformazione del nome* e garantisce l'univocità e la validità del nome.

Contenuto del file di configurazione Queue manager

Le stanze che possono essere visualizzate in un file di configurazione Queue manager, QMINI, sono:

Configuration

Questa stanza definisce tutte le configurazioni per il Queue manager.

E' possibile modificare le seguenti voci:

```
DefaultQueueServerName
DefaultStatusServerName
HomeTerminalName
PathmonProcName
EMSCollectorName
MinIdleMCALU62Responders
MinIdleMCATCPResponders
MinIdleMCAcallers
MinIdleLQMAgents
```

Per ulteriori informazioni su queste voci, consultare la sezione "Modifica delle proprietà del queue manager" a pagina 48. Le altre voci della stanza non devono essere modificate.

DefaultProcess

Questa stanza definisce i valori predefiniti utilizzati per i processi MQSeries. Le voci nella stanza non devono essere modificate.

ECBoss

Questa stanza definisce la configurazione del processo MQSeries EC Boss. La voce ExpectedNumECs definisce il numero di processi EC per questo Queue manager. Questo valore deve corrispondere alla configurazione PATHWAY per il Queue manager. Per ulteriori informazioni, consultare il "Modifica delle proprietà del queue manager" a pagina 48. Le altre voci della stanza non devono essere modificate.

EC Le voci MCAAgentPriority e LQMAgentPriority della stanza EC, che controllano la priorità dei processi Agent, possono essere modificate. Per ulteriori informazioni, consultare la sezione "Modifica delle proprietà del queue manager" a pagina 48. Le altre voci della stanza non devono essere modificate.

File di configurazione Queue Manager

Le seguenti stanze definiscono parametri operativi specifici per ogni tipo di processo MQSeries. Di solito, non è necessario modificare i valori di questi parametri. Tuttavia, consultare la sezione "File di configurazione Queue manager di esempio" a pagina 191.

MCA Caller

MCATCP Responder

MCALU62 Responder

MQIServer

LQMAgent

Channel Initiator

TCPListener

Authority

Fornisce il metodo consigliato per abilitare e disabilitare OAM per un Queue manager. Impostare l'indicatore MQAUTH su On/Off per abilitare o disabilitare OAM senza aggiungere e rimuovere le stanze Service e Service Component.

Service

Specifica il nome di un servizio installabile e il numero di punti di accesso a quel servizio. Esiste una stanza per ogni servizio. Sono disponibili i seguenti servizi:

- Authorization Service
- Name Service

OAM (Object Authority Manager) viene abilitato per impostazione predefinita: la stanza Authorization Service e la stanza ServiceComponent ad essa associata risiedono in QMINI.

E' possibile disabilitare OAM impostando l'indicatore MQAUTH nella stanza Authority su Off e riavviando il Queue manager. Altrimenti, è possibile:

1. Eliminare il Queue manager (utilizzando il comando **dltmqm**)
2. Creare nuovamente il Queue manager (utilizzando il comando **crtmqm**) con la variabile di ambiente MQSNOAUT impostata.
3. Cancellare le stanze Authorization Service da QMINI.

La stanza Name Service deve essere aggiunta manualmente a QMINI, se si desidera abilitarla.

ServiceComponent

Queste stanze definiscono il componente di servizio associato ad un particolare servizio. È possibile che ci sia più di una stanza ServiceComponent per ogni servizio, ma ogni stanza ServiceComponent deve corrispondere alla relativa stanza Service. Consultare il manuale *MQSeries Programmable System Management* per ulteriori informazioni.

TuningParameters

Questa stanza definisce i parametri di regolazione interni utilizzati dagli agenti del Queue manager locale. Si consiglia di non modificare questi valori.

Channels

Questa stanza contiene informazioni relative ai canali. Oltre a indicare il

File di configurazione Queue Manager

numero massimo di canali (MaxChannels) che possono essere definiti per il Queue manager, una seconda voce (MaxActiveChannels) limita il numero di canali che possono essere attivi contemporaneamente. MaxActiveChannels non deve essere superiore a MaxChannels. La stanza Channels contiene anche una voce (ChanInitDiscInterval) che può essere utilizzata per regolare le prestazioni del Channel Initiator. Per ulteriori informazioni su queste voci, consultare la sezione “Modifica delle proprietà del queue manager” a pagina 48. Le altre voci della stanza non devono essere modificate.

Consultare il manuale *MQSeries Intercommunication* per ulteriori informazioni sui canali.

TCPCConfig

Specifica i parametri di configurazione del protocollo di rete. Queste stanze sovrascrivono i parametri predefiniti per i canali. Sono effettivamente presenti solo le stanze che rappresentano i valori predefiniti modificati.

I valori TCPListenerPort vengono sostituiti dal programma Listener se il parametro MQLISTENPORTNUM è presente nell’ambiente di detto processo listener.

Consultare il manuale *MQSeries Intercommunication* per ulteriori informazioni.

Per informazioni sulla modifica delle voci TCPPort, TCPNumListenerPorts e TCPListenerPort, consultare la sezione “Modifica delle proprietà del queue manager” a pagina 48.

File di configurazione Queue manager di esempio

La Figura 43 a pagina 192 mostra un file di configurazione Queue manager di esempio (QMINI).

File di configurazione Queue Manager

```
#####
#* Module Name: QMINI                                     *#
#* Type       : MQSeries queue manager configuration file *#
# Function    : Define the configuration of a single queue manager *#
#*           *#
#####
#* Notes      :                                           *#
#* 1) This file defines the configuration of the queue manager *#
#*           *#
#####
Configuration:
  PathmonProcName=$p01p
  DefaultStatusServerName=$p01s
  ServerClassName=MQS-ECBOSS
  EMSCollectorName=$0
  HomeTerminalName=$ZTN0.#PTY001C
  ShutdownFileName=SHUTDOWN
  TraceOptionsFileName=TRACEOPT
  RuntimeFileName=RUNTIME
  StatableFileName=STATABLE
  ChannelDefFileName=CHDEFS
  DefaultCCSID=819
  DefaultTraceOptions=0
  MaxIdleAgents=10
  MinIdleMCALU62Responders=0
  MinIdleMCATCPResponders=0
  MinIdleMCAcallers=0
  MinIdleLQMAgents=1
  MaxIdleAgentReuse=10
DefaultProcess:
  ExeFileName=DEFAULT
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=256000
  IniPoolSize=256000
  Priority=175
```

Figura 43. File di configurazione Queue manager di esempio (QMINI) (Numero 1 di 6)

File di configurazione Queue Manager

```
ECBoss:
  ExeFileName=MQECBOSS
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
  ExpectedNumECs=1
EC:
  ExeFileName=MQEC
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=256000
  IniPoolSize=256000
  Priority=175
  LQMAgentExe=MLQMAG
  MCACallerExe=MQMCACAL
  MCATCPResponderExe=MQTCPRES
  MCALU62ResponderExe=MLU6RES
  MCAAgentPriority=165
  LQMAgentPriority=165
  StopProcessTimer=3000
  IdleProcessTimer=3000
```

Figura 43. File di configurazione Queue manager di esempio (QMINI) (Numero 2 di 6)

File di configurazione Queue Manager

```
MCACaller:
  ExeFileName=MQMCACAL
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
MCATCPResponder:
  ExeFileName=MQTCPRES
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
MCALU62Responder:
  ExeFileName=MQLU6RES
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
```

Figura 43. File di configurazione Queue manager di esempio (QMINI) (Numero 3 di 6)

```
MQIServer:
  ExeFileName=MQMQISER
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
LQMAgent:
  ExeFileName=MQLQMAG
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=50
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=120000
  IniPoolSize=200000
  Priority=175
ChannelInitiator:
  ExeFileName=RUNMQCHI
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=256000
  IniPoolSize=256000
  Priority=175
TCPListener:
  ExeFileName=RUNMQLSR
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=256000
  IniPoolSize=256000
  Priority=175
```

Figura 43. File di configurazione Queue manager di esempio (QMINI) (Numero 4 di 6)

File di configurazione Queue Manager

```
Queue Manager Server:
  ExeFileName=MQMGRSVR
  TraceVolSubvol=$DATA01.MV1L
  TracePrefix=TR
  ErrorVolSubvol=$DATA01.MV1L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
Repository Server:
  ExeFileName=MQREPSVR
  TraceVolSubvol=$DATA01.MV1L
  TracePrefix=TR
  ErrorVolSubvol=$DATA01.MV1L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=256000
  IniPoolSize=256000
  Priority=175
```

Figura 43. File di configurazione Queue manager di esempio (QMINI) (Numero 5 di 6)

```
Authority:
  MQAUTH=On
Service:
  Service=AuthorizationService
  EntryPoints=9
ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.Compaq.auth.service
  Module=MQOAM
  ComponentDataSize=0
  ComponentID=0
TuningParameters:
  KernelMemSetSize=32000
  ObjCatMemSetSize=32000
  QueueMemSetSize=16000
  MQGETActiveQPoll=50
  MQGETInactiveQPoll=1000
Channels:
  RetryAll=1
  MaxChannels=10
  MaxActiveChannels=10
  MaxTries=3
  MaxTriesInterval=10
  ChanInitDiscInterval=10
  AdoptNewMCA=NO
  AdoptNewMCATimeout=60
  AdoptNewMCACheck=NAME,ADDRESS,QM
TCPConfig:
  TCPPort=1414
  TCPNumListenerPorts=1
  TCPListenerPort=1414
  TCPKeepAlive=1
```

Figura 43. File di configurazione Queue manager di esempio (QMINI) (Numero 6 di 6)

Modifica dei file di configurazione

E' possibile modificare i file di configurazione predefiniti per cambiare i valori predefiniti di sistema. Tuttavia, prima di modificare un file di configurazione, assicurarsi che esista una copia da poter eventualmente ripristinare, e che i Queue manager associati vengano arrestati.

E' necessario modificare i file di configurazione se, ad esempio:

- Si è perso un file di configurazione (ripristinarlo dalla copia di riserva se possibile).
- Occorre modificare la distribuzione del Queue manager tra le CPU.
- Si ha l'esigenza di modificare il Queue Manager predefinito; ciò può accadere se involontariamente si elimina il Queue Manager esistente.
- Il Centro di supporto IBM raccomanda questa operazione.

Per ulteriori informazioni, consultare il "Modifica delle proprietà del queue manager" a pagina 48.

Implementazione delle modifiche ai file di configurazione

Quando si modifica un file di configurazione, i cambiamenti non vengono implementati immediatamente dal Queue Manager. Le modifiche apportate al file di configurazione MQSeries (MQSINI) hanno valore solo quando vengono creati o avviati i Queue manager MQSeries. Le modifiche apportate ad un file di configurazione Queue manager (QMINI) vengono applicate nel momento in cui viene avviato il Queue manager. Se il Queue Manager è in esecuzione mentre vengono effettuate le modifiche, è necessario arrestare e poi riavviare il Queue Manager perché le modifiche vengano riconosciute dal sistema.

Suggerimenti relativi ai file di configurazione

Quando si crea un nuovo Queue Manager, è necessario:

- Eseguire il backup del file di configurazione MQSeries (MQSINI)
- Eseguire il backup del nuovo file di configurazione Queue manager (QMINI)

Modifica dei file di configurazione

Capitolo 14. Determinazione dei problemi

Questo capitolo fornisce le informazioni per la risoluzione dei problemi per MQSeries for Compaq NSK. Per determinare un problema, bisogna elencarne i sintomi e risalire quindi alla causa.

I problemi relativi alle prestazioni causati dai limiti dell'hardware non possono essere risolti immediatamente. Se si ha ragione di credere che il problema dipenda dal codice MQSeries, contattare il centro di supporto della IBM. Questo capitolo contiene le seguenti sezioni:

- "Controlli preliminari"
- "Errori di programmazione frequenti" a pagina 203
- "Come procedere" a pagina 204
- "Considerazioni sulla progettazione delle applicazioni" a pagina 207
- "Lunghezza del messaggio" a pagina 207
- "Log di errore" a pagina 212
- "Code messaggi non recapitati" a pagina 215
- "File di configurazione e determinazione dei problemi" a pagina 215
- "Utilizzo della traccia MQSeries" a pagina 215
- "FFST (First Failure Support Technology™)" a pagina 216

Controlli preliminari

La causa di un problema può dipendere da:

- MQSeries
- La propria rete
- Un'applicazione
- Il software del sistema Compaq

Le sezioni riportate di seguito forniscono domande ricorrenti. Rispondere a queste domande ed annotare qualsiasi informazione che potrebbe risultare rilevante per la risoluzione del problema.

Si è mai avuta in precedenza un'esecuzione con esito positivo di MQSeries?

Se MQSeries è stato eseguito correttamente in precedenza, non occorre reinstallarlo. Consultare il manuale *MQSeries for Compaq NSK Guida operativa* per verificare se tutti i passi sono stati eseguiti correttamente.

Vengono riportati messaggi di errore?

MQSeries utilizza log di errori per catturare messaggi relativi alle operazioni MQSeries, ai Queue Manager avviati e dati di errori provenienti dai canali utilizzati. Verificare le registrazioni degli errori associate al problema.

Consultare la sezione "Log di errore" a pagina 212 per informazioni relative ai contenuti dei log di errore e alle loro posizioni.

Vengono riportati codici di errori che spiegano il problema?

Se un'applicazione riceve un codice di errore in cui è indicato che una chiamata MQI (Message Queue Interface) ha avuto esito negativo, consultare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa* per una descrizione di questo codice di errore.

Il problema è riproducibile?

Se è possibile riprodurre il problema, fare attenzione alle seguenti condizioni:

- Il problema è determinato da un comando o un'altra attività di gestione?
L'operazione funziona se immessa utilizzando un altro metodo? Se il comando funziona solo immettendolo nella linea di comando e non in altro modo, verificare che il server di comando non sia stato arrestato e che non sia stata modificata la definizione di coda del SYSTEM.ADMIN.COMMAND.QUEUE.
- Il problema è determinato da un programma?
- È possibile individuare un'applicazione che sembra in esecuzione tutte le volte che si verifica il problema? In questo caso, esaminare l'applicazione per verificare la presenza di eventuali errori.
- Il volume in cui risiederà il database queue manager è protetto come TM/MP? Immettere TMFCOM; STATUS DATAVOLS per verificare che il volume sia protetto come TM/MP.
- Sono stati aggiunti i parametri TACL richiesti come descritto nella sezione "Appendice C. Impostazione delle variabili di ambiente TACL per MQSeries for Compaq NSK" a pagina 325? Immettere PARAM alla richiesta TACL per visualizzare i parametri correttamente impostati.
- La traccia di controllo TM/MP è in grado di gestire il caricamento? (consultare la sezione "Capitolo 11. Descrizione del supporto transazionale e della messaggistica" a pagina 179). Immettere TMFCOM; INFO AUDITTRAIL per visualizzare la configurazione della traccia di controllo.
- La funzione di sicurezza dei file Compaq consente l'accesso ai file database qmD? E' richiesto un privilegio minimo di accesso in lettura ai file per abilitare l'accesso agli utenti che non appartengono al gruppo MQM. Viene restituito il codice di ritorno 2035 (MQRC_NOT_AUTHORIZED) per una richiesta MQCONN se gli attributi di sicurezza dei file Compaq non consentono l'accesso.

Sono state effettuate modifiche dall'ultima esecuzione valida?

Nel riesaminare le eventuali modifiche apportate di recente, considerare il sistema MQSeries, i programmi con cui interagisce, l'hardware e le nuove applicazioni. Considerare inoltre che una nuova applicazione sconosciuta potrebbe essere stata eseguita sul sistema.

- Sono state aggiunte, modificate o eliminate definizioni di code?
- Sono stati aggiunte o modificate definizioni di canali? Potrebbero essere state effettuate modifiche alle definizioni di canali MQSeries o alle definizioni di comunicazioni di base richieste dal sistema.
- Le applicazioni sono occupate a gestire i codici di errore ricevuti in seguito a qualche modifica effettuata?
- Sono state apportate delle modifiche a PATHWAY MQSeries?
- Sono stati modificati i file di installazione di MQSeries, ad esempio cambiando gli attributi di sicurezza dei file?
- Sono stati modificati i file database Compaq NSK di MQSeries, ad esempio cambiando gli attributi di sicurezza dei file o la traccia di controllo TM/MP?
- Sono state applicate le modifiche al file Queue manager, QMINI?

Si è mai avuta in precedenza un'esecuzione con esito positivo dell'applicazione?

Se il problema sembra relativo ad una specifica applicazione, controllare che l'applicazione abbia avuto almeno per una volta esito positivo nell'esecuzione.

Esaminare le seguenti domande:

- Sono state apportate modifiche all'applicazione dopo l'ultima esecuzione valida?

In tal caso, è possibile che l'errore si trovi nella parte nuovo o modificata dell'applicazione. Controllare le modifiche e verificare l'eventuale presenza di motivazioni ovvie del problema. Provare ad utilizzare l'applicazione nello stato precedente la modifica.
- Sono state utilizzate tutte le funzioni dell'applicazione in precedenza?

Il problema si verifica quando viene utilizzata una parte dell'applicazione mai impiegata in precedenza? In questo caso, è probabile che l'errore risieda in quella parte dell'applicazione. Analizzare l'operazione compiuta dall'applicazione al momento del malfunzionamento e controllare in quella parte del programma il codice sorgente relativo agli errori.

Se il programma non ha mai determinato problemi durante le precedenti esecuzioni, controllare lo stato della coda e dei file elaborati al momento del verificarsi dell'errore. Può accadere che essi contengano dei valori inconsueti che provocano il richiamo di un percorso raramente utilizzato nel programma?
- L'applicazione controlla tutti i codici di errore?

È possibile che il sistema MQSeries sia stato modificato in modo che l'applicazione non controlli più i codici di errore che riceve. Ad esempio, l'applicazione è in grado di desumere che le code cui accede sono condivise? Se una coda è stata ridefinita esclusiva, l'applicazione elabora il codice che le indica l'impossibilità di accedere a quella coda?
- L'applicazione viene eseguita su altri sistemi MQSeries?

Il problema potrebbe dipendere dal tipo di impostazioni del sistema MQSeries? Ad esempio, sono state definite le stesse lunghezze e priorità di messaggi per le code?
- E' stato impostato PARAM SAVE-ENVIRONMENT ON?

In caso contrario, verrà ricevuto MQRC 2058 sulle chiamate MQCONN. Impostare PARAM nell'ambiente dell'applicazione, come descritto nella sezione "Appendice C. Impostazione delle variabili di ambiente TAQL per MQSeries for Compaq NSK" a pagina 325.

Se l'applicazione in precedenza non è stata eseguita con esito positivo

Se l'applicazione non ha mai avuto un'esecuzione valida, è necessario ricercare con cura la presenza di eventuali problemi.

Prima di controllare il codice e il linguaggio di programmazione in cui è stato scritto, esaminare l'output di traduzione, di compilazione e del redattore di collegamento, laddove possibile, per controllare se vengono riportati errori.

Se il malfunzionamento si verifica quando l'applicazione tenta di tradurre, compilare o redigere collegamenti nella libreria di caricamento, si verificherà di conseguenza anche durante i tentativi di esecuzione. Consultare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa* per ulteriori dettagli sulla creazione delle applicazioni.

Controlli preliminari

Se la documentazione dimostra che tutte queste operazioni sono state eseguite senza errori, il problema potrebbe dipendere dalla codificazione logica dell'applicazione. I sintomi del problema indicano la funzione in cui si verifica il malfunzionamento e, di conseguenza, il segmento di codice in errore? Consultare la sezione "Errori di programmazione frequenti" a pagina 203 per alcuni esempi di errori frequenti che provocano problemi con le applicazioni MQSeries.

Il problema è relativo a parti specifiche della rete?

Il problema potrebbe verificarsi solo in alcune parti della rete (code remote, ad esempio). Se il collegamento ad un Message Queue Manager remoto non è attivo, i messaggi non possono giungere ad una coda remota.

Verificare la disponibilità del collegamento tra i due sistemi e controllare che il componente di intercomunicazione di MQSeries sia in esecuzione.

Controllare che i messaggi ricevano la coda di trasmissione; controllare inoltre la definizione di coda locale della coda di trasmissione e ogni coda remota.

Sono state effettuate modifiche relative alla rete o nelle definizioni MQSeries tali da poter provocare problemi?

Controllare che il server Listener TCP PATHWAY di MQSeries, MQS-TCPLISxx, sia stato avviato. Se l'avvio del server non viene eseguito correttamente, verificare che non vi siano connessioni in chiusura (ad esempio, uno stato FIN-WAIT) per la porta che utilizza SCF (ad esempio, SCF; STATUS PROCESS \$ZTC0). Il partner remoto deve essere arrestato e riavviato per liberare la porta. Per SNA, controllare che il Listener SNA di MQSeries sia configurato per SNAX/APC o ICE PATHWAY, consultare "Processi responder LU 6.2" a pagina 378.

Verificare che vengano utilizzati il nome processo TCPIP, il nome host, l'indirizzo IP e la porta corretti per il nome connessione dei canali e che questi corrispondano alla definizione del canale remoto.

Il problema si verifica ad una determinata ora del giorno?

Se il problema si verifica ad una determinata ora del giorno potrebbe dipendere dall'eccessivo caricamento del sistema. Solitamente, il sistema riceve il massimo carico nelle fasce orarie di metà mattina e metà pomeriggio: è in questi orari che si verifica la maggior parte dei problemi dipendenti dal sovraccarico del sistema. Se la rete MQSeries è in funzione anche in altre fasce orarie, momenti di intenso carico del sistema potrebbero verificarsi anche in altri periodi del giorno.

Il problema si manifesta in modo discontinuo?

Un problema discontinuo può verificarsi se non si considera che i processi possono essere eseguiti in modo autonomo l'uno dall'altro. Ad esempio, un programma può inviare una chiamata MQGET, senza specificare un'opzione di attesa, prima che sia terminato un processo precedente. Un problema discontinuo può verificarsi anche nel caso in cui l'applicazione tenti di ricevere un messaggio da una coda, mentre la chiamata che invia il messaggio è sospesa (e ciò si verifica prima dell'esecuzione del commit o del back out).

Talvolta, gli errori PATHWAY possono essere registrati mentre viene eseguito ENDMQM. Il messaggio di errore *1018* SERVER FILE (6006) o *1018* SERVER

FILE (7006) può essere visualizzato mentre il Queue manager viene chiuso. Questi errori si verificano di solito mentre i server MQS-QMGRSVR00, MQS-STATUS00 e MQS-QUEUE00 vengono chiusi.

Sono stati eseguiti aggiornamenti del servizio?

Se sono stati eseguiti aggiornamenti del servizio ad MQSeries, controllare che le operazioni di aggiornamento siano completate con esito positivo e che non vengano riportati messaggi di errore.

- Esistono particolari istruzioni per l'aggiornamento?
- Sono state eseguite delle prove per verificare l'applicazione corretta e completa dell'aggiornamento?
- Il problema si verifica anche se si ripristina il precedente livello di servizio MQSeries?
- Nel caso di installazione con esito positivo, rivolgersi al Centro supporto della IBM per verificare l'esistenza di eventuali errori patch.
- Nel caso in cui venga applicato un patch ad un altro programma, considerare l'effetto che può avere sul modo in cui MQSeries interagisce con esso.
- Verificare il livello del servizio. Modificare ZMQSSYS.MEMOPTF di sola lettura. Si osservi la voce A) per CSD HISTORY. Alla richiesta TACL, immettere VPROC \$vol.ZMQSLIB.MQSRLLIB, dove \$vol indica il volume di installazione MQ. Deve corrispondere ai dati VPROC di MEMOPTF oppure può essere un volume nuovo se viene applicato efix. Fare riferimento al file MEMOEFIX fornito con efix per informazioni VPROC aggiornate.

Errori di programmazione frequenti

Gli errori riportati nel seguente elenco mostrano le cause più comuni dei problemi che si verificano durante l'esecuzione di programmi MQSeries. Il problema verificatosi al sistema MQSeries potrebbe essere dovuto ad uno o più di questi errori:

- Considerare condivisibili, code che invece sono state assegnate come esclusive.
- Utilizzare parametri non corretti in una chiamata MQI.
- Utilizzare parametri insufficienti in una chiamata MQI. Ciò comporta che MQI non può impostare il completamento ed i codici di errori relativi all'applicazione da elaborare.
- Errato controllo dei codici richiesti da MQI.
- Utilizzo di variabili di lunghezza non valida.
- Utilizzo di parametri in ordine errato.
- Inizializzazione errata di *MsgId* e *CorrelId*.
- Errore durante l'immissione di BEGINTRANSACTION, quando MQPMO_SYNCPOINT viene specificato nel comando MQPUT.

Problemi relativi ai comandi

L'immissione dei caratteri speciali, quali la barra retroversa (\) e le doppie virgolette ("), nel testo di alcuni comandi richiede una particolare attenzione. Se si utilizzano questi caratteri nel testo descrittivo, essi devono essere preceduti dal simbolo \. Quindi, immettere \\ o \" se si desidera includere il carattere \ o " nel testo.

Come procedere

Una volta stabilito che non sono state apportate modifiche al sistema e che non vi sono problemi relativi ai programmi applicativi, esaminare le seguenti opzioni e scegliere quella che meglio descrive i sintomi del problema.

- “Viene riportato un output non corretto?”
- “Non si è ottenuta alcuna risposta utilizzando un comando PCF?”
- “Il problema è relativo alle sole code remote?” a pagina 206
- “L’esecuzione dell’applicazione o di MQSeries for Compaq NSK procede lentamente?” a pagina 206

Viene riportato un output non corretto?

In questa pubblicazione, “output non corretto” si riferisce all’applicazione:

- Non viene riportato il messaggio atteso.
- Viene riportato un messaggio che contiene informazioni inattese o errate.
- Viene riportato un messaggio inatteso, come ad esempio un messaggio destinato ad un’applicazione diversa.
- La struttura utilizzata per visualizzare le informazioni è corretta? Ad esempio la struttura MQDLH, utilizzata per visualizzare i dati di intestazione della coda messaggi non recapitati, è stata aggiunta all’inizio del testo di un messaggio di questa coda?
- E’ riproducibile su una macchina MQSeries o su un’altra macchina con un sistema operativo (OS) dello stesso livello o di livello differente? Se non è riproducibile, è possibile che si sia verificato un errore OS corretto con un IPM (Interim Program Maintenance) di Compaq che deve essere applicato. Controllare il supporto Compaq.

In ogni caso, controllare che ogni coda o gli alias Queue Manager utilizzati dall’applicazione siano correttamente specificati e rivedere tutte le modifiche apportate alla rete.

Se viene riportato un messaggio di errore MQSeries, controllare nel log di errore tutto ciò che ha come prefisso le lettere “AMQ”. Per ulteriori informazioni, consultare la sezione “Log di errore” a pagina 212.

Non si è ottenuta alcuna risposta utilizzando un comando PCF?

Se, impostando un comando, non si riceve alcuna risposta, esaminare le seguenti considerazioni:

- Il server di comando è in esecuzione?
Utilizzare il comando **dspmqcsv** per verificare lo stato del server di comando. Se la risposta a questo comando segnala che il server non è in esecuzione, utilizzare il comando **strmqcsv** per avviarlo. Se invece la risposta segnala che SYSTEM.ADMIN.COMMAND.QUEUE non è abilitato alle richieste MQGET, abilitare la coda alle richieste MQGET.
- È stata inviata una risposta alla coda messaggi non recapitati?
La struttura di intestazione della coda messaggi non recapitati contiene un codice di errore o di feedback che descrive il problema. Consultare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa* per informazioni relative alla struttura di intestazione della coda messaggi non recapitati (MQDLH).

Se questa coda contiene messaggi, utilizzare l'applicazione di esempio browse (AMQSBCG) fornita per sfogliare i messaggi utilizzando la chiamata MQGET. L'applicazione di esempio guida attraverso tutti i messaggi di una data coda relativa ad un dato Queue Manager, visualizzando sia la descrizione, che i campi relativi al contesto di tutti i messaggi di una data coda.

- È stato inviato un messaggio al log di errore.

Per ulteriori informazioni, consultare la sezione "Log di errore" a pagina 212.

- Le code sono abilitate alle operazioni di inserimento ed estrazione?
- L'intervallo di tempo (*WaitInterval*) è abbastanza lungo?

Se una coda MQGET è scaduta, vengono riportati un codice di completamento di MQCC_FAILED e un codice di errore di MQRC_NO_MSG_AVAILABLE. Consultare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa* per informazioni relative campo *WaitInterval* e ai codici di completamento e di errore di MQGET.

- È necessario eseguire il commit di una transazione quando si utilizza il programma applicativo per inserire comandi nel SYSTEM.ADMIN.COMMAND.QUEUE?

Fin quando non si specifica l'esclusione dal syncpoint del messaggio di richiesta, è necessario eseguire il commit di una transazione prima di ricevere i messaggi di risposta.

- L'impostazione degli attributi delle code MAXDEPTH e MAXMSGL è abbastanza alta?
- I campi *CorrelId* e *MsgId* vengono utilizzati correttamente?

Impostare i valori di *MsgId* e *CorrelId* nell'applicazione in modo da poter ricevere tutti i messaggi dalla coda.

Per rispondere a qualsiasi messaggio di errore riprodotto, provare ad arrestare e poi riavviare il server di comando.

Se il sistema ancora non risponde, probabilmente il problema riguarda un Queue Manager. Arrestare Queue manager e riavviarlo. Se all'avvio il problema si verifica ancora, richiedere l'assistenza del centro di supporto IBM.

Si verificano malfunzionamenti in alcune code?

Se si ha motivo di ritenere che il problema riguardi solo un gruppo secondario di code, controllare la coda locale sospetta causa del problema:

1. Visualizzare le informazioni relative a ciascuna coda. A questo scopo, utilizzare il comando MQSC.
2. Utilizzare i dati visualizzati per eseguire i seguenti controlli:
 - Se CURDEPTH è a MAXDEPTH, indica che la coda non viene elaborata. Controllare che tutte le applicazioni vengano eseguite normalmente.
 - Se CURDEPTH non è a MAXDEPTH, controllare che i seguenti attributi di coda siano impostati correttamente:
 - Se viene utilizzata la funzione triggering:
 - Il trigger monitor è in esecuzione?
 - L'intensità di trigger è troppo elevata? In questo caso vengono generati eventi trigger troppo spesso?
 - Il nome del processo è corretto?
 - Il processo è disponibile e operativo?
 - La coda può essere condivisa? Se non può essere condivisa, un'altra applicazione potrebbe già averla impegnata per l'input.

Come procedere

- La coda è abilitata correttamente per l'inserimento e l'estrazione?
- Se non vi sono processi applicativi che ricevono messaggi dalla coda, occorre determinarne il motivo. Ciò potrebbe essere dovuto al fatto che l'applicazione deve essere avviata o al fatto che una connessione è stata eliminata o ancora al fatto che per motivi vari la chiamata MQOPEN ha avuto esito negativo.

Controllare gli attributi di coda IPPROCS e OPPROCS. Questi attributi indicano se la coda è stata aperta per l'input e l'output. Un valore pari a zero indica che tale operazione non è possibile. Si noti che i valori possono essere cambiati e perciò la coda può essere stata aperta e poi dopo chiusa.

È necessario esaminare lo stato della coda al momento in cui si prevede di inserire o estrarre un messaggio.

Se non si riesce a risolvere il problema, richiedere l'assistenza al centro di supporto IBM.

Il problema è relativo alle sole code remote?

Se il problema è relativo alle sole code remote, effettuare i seguenti controlli:

- Controllare che i canali richiesti siano avviati, che possono essere sottoposti al trigger e che siano in esecuzione tutti gli iniziatori richiesti.
- Controllare che i programmi che dovrebbero inserire messaggi nelle code remote non abbiano riportato problemi.
- Se si utilizza la funzione triggering per avviare il processo di accodamento distribuito, controllare che la coda di trasmissione è stata impostata per il triggering. Inoltre, controllare che il trigger monitor sia in esecuzione.
- Controllare l'eventuale presenza nei log degli errori di messaggi che indicano errori o problemi relativi al canale.
- Se necessario avviare il canale manualmente. Per informazioni relative a quest'operazione, consultare il manuale *MQSeries Intercommunication*.

Per informazioni relative alla definizione dei canali, consultare i manuali "Appendice M. Impostazione delle comunicazioni" a pagina 377 e *MQSeries Intercommunication*.

L'esecuzione dell'applicazione o di MQSeries for Compaq NSK procede lentamente?

Se l'esecuzione di un'applicazione procede lentamente, ciò potrebbe indicare che si trova in un loop o che è in attesa di una risorsa non disponibile.

La traccia MQSeries è stata abilitata utilizzando **strmqtrc** o il pannello MQMC del Queue manager. Ciò influirà in modo negativo sulle prestazioni. Controllare se vi sono file TR aperti nel volume secondario qmL. Disabilitare la traccia utilizzando **endmqtrc** o il pannello MQMC del Queue manager.

Questo errore potrebbe essere stato causato anche da un problema relativo alle prestazioni. Forse, perché il sistema ha quasi raggiunto i limiti delle proprie capacità. Questo tipo di problema peggiora quando il sistema è sovraccarico nelle ore di punta. Se la rete è in funzione anche in altre fasce orarie, momenti di intenso carico del sistema potrebbero verificarsi anche in altri periodi del giorno.

Controllare il livello di priorità dell'applicazione e dei processi del Queue manager utilizzando il comando STATUS. Un loop causa una graduale riduzione del livello di priorità dei processi fino allo zero da parte di NSK.

Controllare che ogni CPU sul sistema NSK venga completamente utilizzata. Se il carico di lavoro di alcuni processori risulta inconsistente, il sistema NSK deve eseguire la ripartizione di tale carico. Aggiungere gli EC ad altri processori per distribuire il carico di lavoro di MQSeries.

Un problema relativo alle prestazioni può essere causato anche da una limitazione dell'hardware.

Nota: Dopo una nuova installazione di MQSeries o un caricamento a sistema spento del sistema Compaq NSK, l'esecuzione dei programmi di MQSeries potrebbe impiegare un periodo di tempo più lungo del previsto. Ciò dipende dal sistema operativo NSK Compaq che passa attraverso la fase di "fixup", durante la quale verifica che tutte le dichiarazioni esterne vengano risolte.

Se il peggioramento delle prestazioni non dipende dal sovraccarico del sistema, è probabile che la causa sia da attribuire ad un programma non progettato in modo efficiente. Si potrebbe manifestare un problema che si verifica solo quando vengono accesse alcune code.

I seguenti sintomi indicano che l'esecuzione di MQSeries procede lentamente:

- Il sistema risponde in ritardo ai comandi MQSeries.
- Le immagini ripetute sulla capacità della coda indicano che l'elaborazione di tale coda procede lentamente per un'applicazione prevista con numerose attività.

Se le prestazioni del sistema risultano rallentate anche dopo aver esaminato tutte le cause possibile sopraelencate, il problema deve essere ricercato in MQSeries for Compaq NSK. In tal caso, contattare direttamente il Centro di supporto IBM.

Considerazioni sulla progettazione delle applicazioni

Sono numerose i casi in cui le prestazioni di un programma sono influenzate dai limiti della progettazione. Alle volte è difficile rilevare queste limitazioni, poiché il programma sembra avere un normale funzionamento, mentre invece danneggia le prestazioni di altre attività. Di seguito sono riportate alcuni dei problemi propri dei programmi che eseguono chiamate MQSeries.

Per ulteriori informazioni relative alla progettazione delle applicazioni, consultare il manuale *MQSeries Application Programming Guide*.

Lunghezza del messaggio

Sebbene MQSeries supporti messaggi che occupano fino a 100MB di dati, una notevole quantità di dati in un messaggio influisce sull'applicazione che lo elabora. Per ottenere le migliori prestazioni dall'applicazione, è opportuno inviare solo i dati essenziali in un messaggio; ad esempio, nella richiesta di addebito in conto corrente bancario, le uniche informazioni che dovrebbero essere inviate dal client all'applicazione server sono il numero del conto corrente e l'importo dell'addebito.

Ricerca di un particolare messaggio

La chiamata MQGET generalmente richiama il primo messaggio da una coda. Se si utilizzano gli identificativi di messaggio e di correlazione (MsgId e CorrelId) nel descrittore di messaggio per specificare un particolare messaggio, il Queue

Considerazioni sulla progettazione delle applicazioni

Manager deve iniziare una ricerca che termina solo quando rileva quel messaggio. Questo tipo di utilizzo della chiamata MQGET influisce sulle prestazioni dell'applicazione.

Code contenenti messaggi di diversa lunghezza

Se i messaggi in una coda hanno lunghezze diverse, per determinare la lunghezza di un messaggio l'applicazione potrebbe utilizzare la chiamata MQGET con il campo *BufferLength* impostato su zero, in modo che, se la chiamata dovesse avere esito negativo, verrebbe riportata la dimensione dei dati del messaggio. L'applicazione potrebbe allora ripetere la chiamata, specificando l'identificativo del messaggio misurato nella precedente chiamata e un buffer di dimensioni corrette. Tuttavia se vi sono altre applicazioni relative alla stessa coda, le prestazioni dell'applicazione saranno ridotte poiché la seconda chiamata MQGET impiegherà del tempo nella ricerca del messaggio che un'altra applicazione avrà richiamato nello spazio di tempo fra le due chiamate.

Se l'applicazione non può utilizzare messaggi di lunghezza fissa, un'altra soluzione potrebbe essere quella di utilizzare la chiamata MQINQ per trovare la dimensione massima di messaggi accettabile dalla coda, per poi utilizzare questo valore nella chiamata MQGET. La dimensione massima dei messaggi è memorizzata nell'attributo *MaxMsgLength* della coda. Tuttavia, questo metodo utilizza una gran quantità di memoria, poiché il valore di questo attributo può giungere fino a 100 MB, il massimo supportato da MQSeries for Compaq NSK.

Frequenza di syncpoint

I programmi che inviano numerose chiamate MQPUT in un syncpoint, senza eseguirne il commit, possono causare problemi di prestazioni. Le code interessate possono essere riempite di questi messaggi attualmente inaccessibili, mentre altre attività sono in attesa di ricevere tali messaggi. Questo ha implicazioni in termini di: memoria, utilizzo della traccia di controllo TMF e processi ad attività che tentano di ricevere messaggi.

Utilizzo della chiamata MQPUT1

Utilizzare la chiamata MQPUT1 solo quando si deve inviare un singolo messaggio ad una coda. Se invece si devono inviare più messaggi, utilizzare la chiamata MQOPEN, seguite da una serie di chiamate MQPUT e una singola MQCLOSE.

Output non corretto

Il termine "output non corretto" può avere diverse interpretazioni. L'accezione utilizzata in questo capitolo del manuale è quella riportata nella sezione "Viene riportato un output non corretto?" a pagina 204.

In questa sezione vengono illustrati due tipi di output non corretti:

- Il caso in cui non si trovano i messaggi attesi
- Il caso di messaggi che contengono informazioni errate o alterate

Inoltre vengono riportati anche ulteriori problemi relativi alle applicazioni tra cui l'utilizzo di code di distribuzione.

Messaggi assenti nella coda

Se non si trovano i messaggi attesi nella coda, effettuare i seguenti controlli:

- Il messaggio è stato inviato con esito positivo alla coda?

- La coda è stata definita in modo corretto? Ad esempio, MAXMSGL è abbastanza esteso?
- La coda è abilitata all'invio?
- La coda è già piena? Questo potrebbe rendere impossibile ad un applicazione l'invio di un messaggio alla coda.
- È possibile ricevere messaggi dalla coda?
- È necessario fissare un syncpoint?
Se sono stati inviati o richiamati messaggi all'interno di un syncpoint, non saranno disponibili per le altre operazioni finché non sarà eseguito il commit dell'unità di ripristino.
- L'intervallo di attesa è abbastanza lungo?
È possibile impostare come opzione per la chiamata MQGET l'intervallo di attesa. Occorre assicurarsi che il tempo di attesa sia lungo abbastanza per ottenere una risposta.
- Si è in attesa di un messaggio particolare, determinato da un identificativo di messaggio o di correlazione (*MsgId* o *CorrelId*)?
Controllare che gli identificativi *MsgId* o *CorrelId* siano corretti per il messaggio. Una chiamata MQGET con esito positivo imposta entrambi questi valori in base alla risposta del messaggio; per questo potrebbe essere necessario reimpostare questi valori per poter ricevere altri messaggi.
Controllare, inoltre, se è possibile ricevere altri messaggi dalla coda.
- Le altre applicazioni ricevono messaggi dalla coda?
- Il messaggio atteso era stato definito permanente?
In caso contrario, se MQSeries viene riavviato, il messaggio è perso.
- Esiste un'altra applicazione che ha accesso esclusivo alla coda?

Se si riesce ad individuare un problema relativo alla coda durante il funzionamento di MQSeries, effettuare i seguenti controlli sul processo che deve inviare il messaggio alla coda:

- L'applicazione è stata avviata?
Nel caso in cui venga sottoposta a trigger, controllare che siano state specificate le corrette opzioni.
- L'applicazione è stata arrestata?
- È in esecuzione un trigger monitor?
- Il processo trigger è stato definito correttamente?
- L'applicazione termina correttamente?
Verificare la presenza di una vistosa anomalia della chiusura nel log di lavoro.
- L'applicazione esegue il commit delle proprie modifiche o queste vengono ritirate (backed out)?

Se esistono transazioni multiple relative alla coda, esse possono trovarsi in conflitto tra loro. Si supponga, ad esempio, che una transazione invii una chiamata MQGET con un buffer di lunghezza zero per individuare la lunghezza del messaggio e poi invii una chiamata MQGET che specifica il *MsgId* di quel messaggio. Nello stesso tempo, un'altra transazione invia una chiamata MQGET relativa a quel messaggio che ha esito positivo; in questo caso la prima applicazione riceverà il codice di errore MQRC_NO_MSG_AVAILABLE. Le applicazioni che si desidera eseguire in ambiente multi-server devono essere progettate per adattarsi a queste situazioni.

Output non corretto

Si noti che il messaggio può essere stato ricevuto, anche se l'applicazione per qualche motivo non riesce ad elaborarlo. Ad esempio, un errore nel formato del messaggio atteso potrebbe determinare il rifiuto da parte del programma? In questo caso, consultare la sezione "Messaggi che contengono informazioni inattese o alterate".

Messaggi che contengono informazioni inattese o alterate

Se l'informazione contenuta nel messaggio non è ciò che l'applicazione attendeva o è stata alterata in qualche modo, prendere in esame le seguenti considerazioni:

- È stata modificata l'applicazione utilizzata o quella che invia il messaggio alla coda?

Assicurarsi che ogni modifica sia stata comunicata a tutti i sistemi che devono venire a conoscenza del cambiamento.

Ad esempio, se viene modificato il formato dei dati del messaggio, entrambe le applicazioni devono essere ricompilate per integrare le modifiche. Se un'applicazione non viene ricompilata, all'altra i dati sembreranno alterati.

- L'applicazione sta inviando messaggi alla coda sbagliata?

Controllare che i messaggi ricevuti dall'applicazione non siano in realtà destinati ad un'applicazione relativa ad una diversa coda. Se necessario, modificare le definizioni di sicurezza, in modo da impedire alle applicazioni non autorizzate di inviare messaggi alle code.

Se l'applicazione utilizza un alias di coda, controllare che l'alias si riferisca alla coda appropriata.

- Le informazioni del trigger relative a questa coda sono state specificate correttamente?

Controllare che l'applicazione da utilizzare sia stata avviata; è stata avviata un'applicazione diversa?

Se questi controlli non consentono di risolvere il problema, potrebbe essere necessario controllare la logica dell'applicazione, sia per quanto riguarda il programma che invia il messaggio, che per quello che lo riceve.

Problemi relativi ad output non corretti durante l'accodamento distribuito

Se l'applicazione utilizza l'accodamento distribuito, è consigliabile prendere in esame i seguenti controlli:

- MQSeries è stato correttamente installato sui sistemi di invio e di ricezione ed è stato correttamente configurato per l'accodamento distribuito?

- I collegamenti tra i due sistemi, sono disponibili?

Controllare la disponibilità dei sistemi e la loro connessione a MQSeries. Controllare che la connessione tra i due sistemi e i canali tra i due Queue Manager siano attivi.

- La funzione triggering è stata attivata nel sistema di invio?

- Il messaggio atteso è un messaggio di risposta da un sistema remoto?

Controllare che nel sistema remoto sia attiva la funzione triggering.

- La coda è già piena?

Questo potrebbe rendere impossibile ad un'applicazione l'invio di un messaggio alla coda. In questo caso, controllare se il messaggio è stato inviato ad una coda messaggi non recapitati.

L'intestazione della coda messaggi non recapitati contiene un codice di errore o di feedback che spiega il motivo del fallito invio alla coda di destinazione. Controllare il manuale *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa* per informazioni relative alla struttura dell'intestazione della coda messaggi non recapitati.

- Vi è un errore di corrispondenza tra il Queue Manager di invio e quello di ricezione?

Ad esempio, la dimensione del messaggio potrebbe essere superiore a quella che il Queue Manager ricevente può gestire.

- Le definizioni dei canali di invio sono compatibili con quelle dei canali di ricezione?

Ad esempio, una differenza nel numero di sequenza arresta il componente dell'accodamento distribuito. Consultare il manuale *MQSeries Intercommunication* per ulteriori informazioni relative all'accodamento distribuito.

- E' stato avviato un listener TCP/IP?

Se si sta utilizzando TCP/IP come protocollo di comunicazione tra MQSeries e Compaq, è possibile che un processo del listener TCP/IP sia in esecuzione. Consultare la sezione "Specificazione e controllo dei listener TCP/IP" a pagina 32 per ulteriori informazioni.

- Il listener TCP/IP è in ascolto sulla porta TCP/IP corretta?

Il listener TCP/IP è in ascolto su una porta definita nella voce `TCPListenerPort` della stanza `TCPCfg` del file `QMINI` per il proprio Queue manager. Consultare la sezione "Porte TCP/IP su cui è in ascolto il queue manager" a pagina 54 per ulteriori informazioni.

- Il nome del processo TCP/IP è corretto?

Se si sta utilizzando il protocollo di comunicazione TCP/IP, il sistema Compaq ha specificato il nome predefinito (`$ztc0`) per questo processo TCP/IP? In caso contrario, è necessario modificare alcune classi del server nel percorso di MQSeries per abilitare il nome processo corretto che deve essere utilizzato dai canali MQSeries. Consultare la sezione "Riconfigurazione della classe server `MQS-TCPLISnn` per una porta ed un processo TCP/IP non predefiniti" a pagina 55 per ulteriori informazioni.

- Il Listener SNA MQSeries è stato configurato?

Se:

- viene utilizzato SNA come protocollo di comunicazione per l'esecuzione dei canali MQSeries
- il tipo di canale su Compaq è in attesa di essere attivato da un sistema remoto MQSeries (ad esempio, RECEIVER)
- il sistema remoto sta riscontrando dei problemi nell'avvio del canale,

il Listener SNA `PATHWAY` di MQSeries potrebbe non essere in esecuzione per il proprio Queue manager. Verificare che il Listener SNA MQSeries sia stato configurato per il Queue manager `SNAX/APC` o `ICE PATHWAY`. Consultare la sezione "Processi responder LU 6.2" a pagina 378 per ulteriori informazioni.

- È necessaria la conversione dei dati? Se i formati dei dati differiscono tra l'applicazione di invio e quella di ricezione, è necessaria la conversione dei dati. La conversione automatica si verifica all'invio di `MQGET` se il formato viene riconosciuto come uno di quelli preinstallati.

Se il formato dei dati non è riconosciuto per il formato incorporato, il risultato della conversione viene utilizzato per consentire la conversione manuale. Verificare che la routine venga caricata correttamente.

Output non corretto

Consultare il manuale *MQSeries Application Programming Guide* per ulteriori informazioni sulla conversioni dei dati.

Log di errore

MQSeries for Compaq NSK utilizza diversi log di errore per registrare messaggi relativi alle operazioni dello stesso MQSeries, di un qualunque Queue Manager avviato e i dati di errori provenienti dai canali utilizzati.

L'associazione dell'errore ad un client o la conoscenza del nome del Queue Manager determinano la posizione dei log degli errori.

- Se il nome del Queue Manager è noto e il Queue Manager è disponibile:

```
<QMVOL>.<SUBVOL>.L.MQERRLG1
```

- Se il Queue Manager non è disponibile:

```
<MQSVOL>.ZMQSSYS.MQERRLG1
```

- FFST (First Failure Symptom Trap) in

```
<QMVOL>.<SUBVOL>.FDnnnnn
```

- Consultare la sezione "Esame dei FFST" a pagina 216.

File di log

Il volume secondario per la registrazione degli errori può contenere fino a tre file di log degli errori denominati:

- MQERRLG1
- MQERRLG2
- MQERRLG3

Dopo la creazione di un Queue Manager, in funzione delle sue necessità, vengono creati tre file di log degli errori. Questi file vengono denominati MQERRLG1, MQERRLG2 e MQERRLG3, e posizionati nel volume secondario di ogni Queue manager che viene creato.

Una volta prodotti, i messaggi di errore vengono memorizzati in MQERRLG1. Una volta compilato MQERRLG1, verrà copiato in MQERRLG2. Prima di questa operazione, MQERRLG2 viene copiato in MQERRLG3. L'eventuale contenuto di MQERRLG3 viene eliminato.

I messaggi di errore più recenti sono quelli memorizzati in MQERRLG1; gli altri file vengono utilizzati per conservare una storia dei messaggi di errore.

Tutti i messaggi relativi ai canali vengono inoltre memorizzati nei file di errore del proprio Queue Manager, a meno che il nome del Queue Manager è sconosciuto o il Queue Manager è indisponibile. In questi casi i messaggi relativi ai canali vengono memorizzati nel file di log degli errori di sistema (ZMQSSYS.MQERRLG1).

Per esaminare il contenuto dei file di log degli errori, utilizzare il comando fup copy o l'editor Compaq NSK in modalità di sola lettura. Se il file di log degli errori viene aperto in modalità aggiornamento, è possibile che i messaggi di errore vengano persi.

Errori precedenti

Esistono diverse situazioni particolari in cui possono verificarsi errori quando i log di errore di cui sopra non sono ancora stati creati. MQSeries cerca di registrare ogni errore in un log di errore. La posizione del log dipende da quanti Queue Manager vengono stabiliti.

Se non possono essere determinate informazioni sulla posizione, a causa, ad esempio, dell'alterazione del file di configurazione, gli errori vengono collegati ad un file di errori creato al momento dell'installazione nel volume secondario ZMQSSYS del file MQERRLG1.

Per ulteriori informazioni relative alla configurazione, consultare il "Capitolo 13. File di configurazione" a pagina 187.

Messaggi per l'operatore

In MQSeries for Compaq NSK, i messaggi per l'operatore indicano errori generalmente causati dall'utente che immette parametri non validi per un determinato comando. Questi messaggi vengono compilati nelle eventuali finestre associate ed anche in un file del volume secondario di Queue Manager.

Gli errori che possono essere associati ad un particolare Queue manager vengono registrati nel file MQERRLG1 del volume secondario di log del Queue manager. Gli errori che non possono essere collegati a un Queue manager definito e operativo vengono registrati nel file MQERRLG1 del volume secondario ZMQSSYS.

Decodifica dei numeri EC nel file MQERRLG

Il numero EC nei messaggi registrati nel file MQERRLG1 viene assegnato da ECBOSS per la traccia dei processi EC. Non esiste alcuna correlazione diretta tra il numero assegnato da ECBOSS nel file MQERRLG1 e quello utilizzato nel nome del server MQS-ECxx PATHWAY. Ad esempio, non è possibile assegnare a MQS-EC00 il numero EC 0. Il numero EC assegnato viene associato al nome del processo EC nel messaggio relativo al completamento dell'inizializzazione. Utilizzare il nome del processo EC per la determinazione del server MQS-ECxx PATHWAY e per l'analisi dei problemi.

Esempio di log degli errori

Questo esempio mostra parte di un log degli errori MQSeries for Compaq NSK:

Log di errore

```
...
02/01/01 11:41:56 AMQ8003: MQSeries queue manager started.
EXPLANATION: MQSeries queue manager Janet started.
ACTION: None.
-----
02/01/01 11:56:52 AMQ9002: Channel program started.
EXPLANATION: Channel program 'JANET' started.
ACTION: None.
-----
02/01/01 11:57:26 AMQ9208: Error on receive from host 'camelot
(9.20.12.34)'.
EXPLANATION: An error occurred receiving data from 'camelot
(9.20.12.34)' over TCP/IP. This may be due to a communications failure.
ACTION: Record the TCP/IP return code 232 (X'E8') and tell the
systems administrator.
-----
02/01/01 11:57:27 AMQ9999: Channel program ended abnormally.
EXPLANATION: Channel program 'JANET' ended abnormally.
ACTION: Look at previous error messages for channel program
'JANET' in the error files to determine the cause of the failure.
-----
02/01/01 14:28:57 AMQ8004: MQSeries queue manager ended.
EXPLANATION: MQSeries queue manager Janet ended.
ACTION: None.
-----
02/02/01 15:02:49 AMQ9002: Channel program started.
EXPLANATION: Channel program 'JANET' started.
ACTION: None.
-----
02/02/01 15:02:51 AMQ9001: Channel program ended normally.
EXPLANATION: Channel program 'JANET' ended normally.
ACTION: None.
-----
02/02/01 15:09:27 AMQ7030: Request to quiesce the queue manager
accepted. The queue manager will stop when there is no further
work for it to perform.
EXPLANATION: You have requested that the queue manager end when
there is no more work for it. In the meantime, it will refuse
new applications that attempt to start, although it allows those
already running to complete their work.
ACTION: None.
-----
02/02/01 15:09:32 AMQ8004: MQSeries queue manager ended.
EXPLANATION: MQSeries queue manager Janet ended.
ACTION: None.
...
```

Eventi EMS

Un evento EMS viene generato per ogni voce di errore scritta nel file MQERRLG1. Per ulteriori informazioni sugli eventi EMS, consultare la sezione “Eventi EMS (Event Management Service)” a pagina 173.

Code messaggi non recapitati

I messaggi che per qualsiasi motivo non vengono recapitati vengono inviati alla coda messaggi non recapitati. Immettendo il comando MQSC DISPLAY QUEUE è possibile controllare la presenza di messaggi nella coda. Se la coda contiene messaggi, utilizzare l'applicazione di esempio browse (MQSBCG0E) fornita per sfogliare i messaggi utilizzando la chiamata MQGET. L'applicazione di esempio guida attraverso tutti i messaggi di una data coda relativa ad un dato Queue Manager, visualizzando sia la descrizione, che i campi relativi al contesto di tutti i messaggi di una data coda.

È necessario decidere come disporre dei messaggi trovati nella coda messaggi non recapitati, in base al motivo che ha determinato questa loro destinazione.

Possono verificarsi dei problemi nel caso in cui non sia definita una coda messaggi non recapitati su ciascun Queue Manager utilizzato. Quando viene creata la coda messaggi non recapitati, è necessario modificare l'attributo DEADQ del Queue manager utilizzando **runmqsc**.

File di configurazione e determinazione dei problemi

Gli errori dei file di configurazione escludono la possibilità che vengano rilevati errori del tipo "queue manager unavailable".

Vi sono diversi controlli da eseguire sui file di configurazione:

- Verificare che i file di configurazione siano stati creati.
- Verificare che siano state assegnate le necessarie autorizzazioni.
- Verificare che il file di configurazione MQSeries si riferisca al Queue Manager e alle directory appropriate.

Utilizzo della traccia MQSeries

MQSeries for Compaq NSK utilizza i seguenti comandi per le funzioni di traccia:

- **strmqtrc** – consultare la sezione "strmqtrc (Avvio traccia di MQSeries)" a pagina 311
- **dspmqtrc** – consultare la sezione "dspmqtrc (Visualizza output di traccia formattato di MQSeries)" a pagina 282
- **endmqtrc** – consultare la sezione "endmqtrc (End MQSeries trace)" a pagina 291

La funzione di traccia utilizza un file per ogni entità di cui si esegue la traccia e registra le relative informazioni nel file appropriato.

Le opzioni di traccia vengono specificate nel file QMINI.

Nota: Con MQSeries for Compaq NSK, la funzione di traccia può essere controllata anche mediante il menu di Queue Manager della funzione MQM (Message Queue Management).

Nomi dei file di traccia

I nomi dei file di traccia sono strutturati nel seguente modo:

TRccpppp

ccpppp indica l'identificativo di processo (PID) relativo al processo che produce la traccia. PID è formato da:

Utilizzo della traccia MQSeries

cc, il numero della CPU.
pppp, il numero del processo.

Se il programma di utilità della funzione di traccia rileva un file di traccia con un identificativo di traccia identico, sostituisce l'ultimo carattere del numero del processo con una lettera, consentendo ai 26 processi con lo stesso PID di scrivere l'output. Ad esempio, per il primo file di traccia con il PID 00, 0315 sarà TR000315. Per un secondo processo avviato alla fine del processo 00, 0315 con lo stesso PID, il file di traccia sarà TR00031A.

Nota: A causa di questa limitazione, i file di traccia devono essere eliminati dal sistema subito dopo essere stati esaminati.

Esempio di dati traccia

L'esempio che segue rappresenta un estratto di traccia:

ID	ELAPSED_MSEC	DELTA_MSEC	APPL	SYSCALL	KERNEL	INTERRUPT
...						
30d	0 0	MQS CEI	Exit!	12484.1	xcsWaitEventSem	rc=10806020
30d	0 0	MQS CEI	Exit!	12484.1	zcpReceiveOnLink	rc=20805311
30d	0 0	MQS FNC	Entry	12484.1	xzcProcessChildren	
30d	0 0	MQS CEI	Entry	12484.1	xcsRequestMutexSem	
30d	1 0	MQS CEI	Entry..	12484.1	xcsHSHMEMBtoPTR	
30d	1 0	MQS CEI	Exit...	12484.1	xcsHSHMEMBtoPTR	rc=00000000
30d	1 0	MQS FNC	Entry..	12484.1	xllSemGetVal	
30d	1 0	MQS FNC	Exit...	12484.1	xllSemGetVal	rc=00000000
30d	1 0	MQS FNC	Entry..	12484.1	xllSemReq	
30d	1 0	MQS FNC	Exit...	12484.1	xllSemReq	rc=00000000
30d	1 0	MQS CEI	Exit..	12484.1	xcsRequestMutexSem	rc=00000000
30d	2 0	MQS CEI	Entry	12484.1	xcsReleaseMutexSem	
30d	2 0	MQS CEI	Entry..	12484.1	xcsHSHMEMBtoPTR	
30d	2 0	MQS CEI	Exit...	12484.1	xcsHSHMEMBtoPTR	rc=00000000
30d	2 0	MQS FNC	Entry..	12484.1	xllSemRel	
30d	2 0	MQS FNC	Exit...	12484.1	xllSemRel	rc=00000000
30d	2 0	MQS CEI	Exit..	12484.1	xcsReleaseMutexSem	rc=00000000
30d	2 0	MQS CEI	Entry	12484.1	xcsHSHMEMBtoPTR	
...						

Figura 44. Esempio di traccia

Note:

1. Nell'esempio viene riportata solo una sezione di dati. In una traccia completa vengono riportati per intero i nomi di funzione e i codici di errore.
2. I codici di errore vengono riportati in valori e non in lettere.

FFST (First Failure Support Technology™)

Gli errori FFST sono molto gravi e indicano un problema di configurazione del sistema o un problema interno di MQSeries. In molti casi, il Queue manager resta operativo, ma è probabile che si verifichi una breve interruzione di tutte o solo di alcune applicazioni. Gli errori FFST vengono riportati nel file ZMQSSYS.MQSYSLOG.

Esame dei FFST

I file vengono denominati FDnnnnn, dove:
nnnnn indica l'identificativo del processo che riporta l'errore

Quando un processo genera un prospetto FFST, viene creato anche un evento EMS.

Nella Figura 45 viene visualizzato un tipico prospetto FFST.

```

-----+-----
MQSeries First Failure Symptom Report
=====
Date/Time      :- February 6  12:23:26   2001
Host Name     :- \HURSLEY
PIDS          :- 5724A39
LVLS          :- 510
Product Long Name :- MQSeries for Compaq NonStop Kernel
Vendor        :- IBM
Probe Id      :- RM020011
Application Name :- MQM
Component     :- rrxOpenSync
Build Date    :- Feb 5 2001
Exe File Name :- \HURSLEY.$DATA0.ZMQSEX.E.MQMCACAL
UserID        :- MQM.MANAGER
Process File Name :- \HURSLEY.$Z734:15441941
Node number   :- 1
CPU           :- 0
PIN           :- 339
QueueManager  :- MT01
Major Errorcode :- xecF_E_UNEXPECTED_RC
Minor Errorcode :- Unknown(A)
Probe Type    :- MSGAMQ6118
Probe severity :- Severity 2: error
Probe Description :- AMQ6118: An internal MQSeries error has occurred.
Text          :- Error creating synch file

Arith1        :- 10 (0xa)
Comment1      :- error 0000000010 in function 0000000020
-----+-----

```

Figura 45. Prospetto FFST di esempio

Tuttavia, vi è un gruppo di problemi che potrebbero essere risolti dagli amministratori. Se FFST riporta la descrizione “out of resource” o “out of space on device”, è probabile che i limiti di sistema siano stati superati.

Per risolvere il problema, incrementare il limite appropriato e riavviare il Queue manager.

Capitolo 15. Scalabilità e prestazioni

capitolo illustra le tecniche per ottimizzare le prestazioni e la scalabilità dei programmi applicativi di MQSeries. Contiene le seguenti sezioni:

- "Introduzione"
- "Messaggi permanenti" a pagina 220
- "Messaggi non permanenti" a pagina 220
- "Server di code e file di code" a pagina 222
- "Assegnazione di CPU" a pagina 226
- "Programmi applicativi per binding FASTPATH" a pagina 228

Introduzione

L'ottimizzazione delle prestazioni e della scalabilità viene eseguita per ridurre l'uso di due risorse chiave: CPU ed il sottosistema disco. Le applicazioni che utilizzano una quantità minore di CPU e IE disco, si caratterizzano per migliori prestazioni e scalabilità. Ad esempio, esse possono essere configurate per elaborare carichi di lavoro sempre crescenti, utilizzando a pieno le risorse hardware e software.

In questo capitolo vengono illustrate le tecniche che consentono di migliorare le prestazioni dei programmi applicativi e di MQSeries. Nelle sezioni seguenti vengono riassunti i principi generali che consentono di migliorare le prestazioni delle applicazioni.

Progettazione di nuove applicazioni per prestazioni e scalabilità

Nella prima fase di progettazione di nuove applicazioni, è necessario considerare il modo in cui vengono utilizzati MQSeries ed altri sottosistemi. Le esigenze aziendali determinano quali funzioni di MQSeries sono necessarie o rilevanti per le singole applicazioni. Alcune funzioni di MQSeries, ad esempio la permanenza dei messaggi, richiedono una forte integrità e garanzia di consegna e, quindi, una grande quantità di CPU e di IE disco. Se tali garanzie non vengono richieste per una determinata applicazione, configurando MQSeries in base a tali esigenze, si ottengono significativi miglioramenti nelle prestazioni. E' opportuno fare tali valutazioni nelle fasi iniziali di progettazione, quando si esaminano le esigenze aziendali.

Progettazione per ridurre o eliminare l'utilizzo di risorse condivise

Il crescente traffico di messaggi richiede che il sistema hardware e software venga utilizzato completamente. Di solito, le risorse condivise producono un rallentamento quando il carico di lavoro aumenta. Tale rallentamento può verificarsi a causa di un aumento del grado di condivisione (ad esempio, più utenti condividono la stessa CPU) oppure perché viene esaurita la maggior parte delle risorse disponibili (ad esempio, ciascun utente esegue un numero maggiore di attività).

Dal punto di vista delle prestazioni e della scalabilità, la CPU e il sottosistema disco rappresentano le risorse condivise più di frequente e che, pertanto, richiedono maggiore attenzione.

Introduzione alla scalabilità e alle prestazioni

I processi MQSeries possono essere eseguiti su un numero infinito di CPU. Su un sistema con diverse CPU, distribuendo i processi MQSeries sulle CPU disponibili, si ottengono prestazioni migliori che utilizzando l'assegnazione predefinita della CPU. Allo stesso modo, per ottimizzare l'utilizzo del sottosistema disco, si consiglia di posizionare file coda diversi su diversi volumi disco, servite, se possibile, da processi disco Compaq NSK separati.

Ottimizzazione delle prestazioni iterativa

Per migliorare le prestazioni, è richiesto un metodo measure-tune-remeasure. Ciascun ciclo di ottimizzazione deve includere la modifica di un'unica variabile principale, in modo che l'effetto prodotto da quella variabile possa essere confrontato con quello prodotto dalle altre. In genere, è controproducente modificare più di un'impostazione per volta, poiché alcune modifiche possono produrre un miglioramento generale delle prestazioni più di altre. Mentre altre possono causare un peggioramento. Le prestazioni di un sistema di applicazioni sono generalmente determinate dai singoli rallentamenti. Modificando l'ottimizzazione di un sistema, vengono rallentate delle altre risorse. L'interazione tra le modifiche di questo tipo fa aumentare la necessità di utilizzare il metodo measure-tune-remeasure durante l'ottimizzazione delle prestazioni.

Ad esempio, una determinata applicazione può scrivere un gran numero di piccoli messaggi permanenti in una coda. Può essere utile verificare gli effetti prodotti nel rendere tutti questi messaggi non permanenti oppure nello scrivere la stessa quantità di dati in un numero minore di messaggi di grandi dimensioni. Entrambe le modifiche generalmente producono un miglioramento delle prestazioni. Tuttavia, senza un metodo measure-tune-remeasure distinto per ciascuna modifica, potrebbero non essere facilmente individuabili le modifiche che producono i miglioramenti più significativi.

Messaggi permanenti

I messaggi permanenti forniscono le garanzie migliori offerte da MQSeries in termini di consegna e recupero. I messaggi permanenti sono sempre memorizzati su supporti protetti, pertanto non vengono persi al riavvio di un queue manager. Nonstop TM/MP controlla i file di coda; in tal modo leggere o scrivere messaggi permanenti si traduce di un'attività del disco per il file di coda e per i file di controllo TM/MP. La registrazione delle verifiche TM/MP è richiesta per preservare l'integrità delle transazioni per i messaggi permanenti, anche nel caso di malfunzionamenti hardware o del sistema. La registrazione delle verifiche TM/MP associata ai messaggi permanenti deve essere considerata quando si valutano le prestazioni dei progetti di applicazioni MQSeries.

La permanenza è una proprietà del messaggio, non della coda in cui è memorizzato. Nelle code possono essere memorizzati messaggi permanenti e non permanenti. Tuttavia, l'amministratore può specificare se i nuovi messaggi sono permanenti quando non viene specificato dall'applicazione. (Fare riferimento all'attributo di coda DEFPSIST descritto in *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*).

Messaggi non permanenti

A differenza dei messaggi permanenti, i messaggi non permanenti non vengono protetti sul disco e vengono persi al riavvio di un queue manager. In base alle opzioni del server di coda per la coda, i messaggi non permanenti possono essere verificati sul server di code di backup. Il motivo principale per cui vengono

Messaggi non permanenti

utilizzati messaggi non permanenti riguarda le prestazioni. I messaggi permanenti richiedono forti garanzie per la consegna e il recupero. Pertanto, la loro lettura o scrittura richiede l'utilizzo del disco sia per i file di coda che per i file di controllo TM/MP. Tale utilizzo del disco riduce le prestazioni delle applicazioni che leggono o scrivono i messaggi permanenti e dei canali MQSeries che trasferiscono i messaggi permanenti su altri queue manager.

L'utilizzo dei messaggi non permanenti produce un notevole miglioramento delle prestazioni. Tali messaggi, infatti, non vengono protetti sul disco, bensì memorizzati nella cache gestita dai server di code Nonstop.

La permanenza è una proprietà del messaggio, non della coda in cui è memorizzato. Nelle code possono essere memorizzati messaggi permanenti e non permanenti. Tuttavia, l'amministratore può specificare se i nuovi messaggi sono permanenti quando non viene specificato dall'applicazione (fare riferimento all'attributo DEFPSIST).

Canali e messaggi non permanenti

I canali di messaggi utilizzano la registrazione nei log della sincronizzazione per l'invio e la ricezione per garantire un'unica consegna dei messaggi inviati sulla rete. Tale attività si aggiunge alla registrazione nei log dei controlli eseguiti da TM/MP (per il queue manager) quando i canali leggono e scrivono i messaggi sulle code. I canali di messaggi possono essere configurati per non eseguire la registrazione nei log sincronizzata durante l'invio e la ricezione di messaggi non permanenti. Per fare ciò, impostare l'attributo NPMSPEED su FAST. L'attributo di canale NPMSPEED controlla il funzionamento dei canali di invio e di ricezione che elaborano i messaggi non permanenti. Quando NPMSPEED è impostato su NORMAL per un canale, i messaggi non permanenti sono parte del batch di messaggi del canale (come definito dagli attributi BATCHINT e BATCHSZ) e, allo stesso modo dei messaggi permanenti, richiedono la registrazione nei log della sincronizzazione. Inoltre, quando NPMSPEED è impostato su NORMAL, il canale legge e scrive i messaggi non permanenti sotto il controllo del syncpoint. Ciò richiede un'attività limitata del file di controllo TM/MP all'inizio e alla fine di una transazione.

Quando NPMSPEED è impostato su FAST, i messaggi non permanenti non sono parte del batch del canale corrente e vengono letti e scritti sulle code al di fuori del controllo del syncpoint. Utilizzando NPMSPEED(FAST), vengono eliminate due origini di attività su disco dei canali. La registrazione nei log eseguita dal meccanismo di sincronizzazione batch del canale e la registrazione nei log dei controlli TM/MP che verrebbero altrimenti eseguite per la lettura e la scrittura dei messaggi nel syncpoint.

NPMSPEED(FAST) è un'opzione per le prestazioni che viene utilizzata per il recupero dei messaggi non permanenti in seguito ad un malfunzionamento per prestazioni particolarmente elevate. Utilizzando NPMSPEED(FAST), può accadere che i messaggi non permanenti vengano persi in caso di errori sul canale o sulla rete.

Il valore predefinito per NPMSPEED è FAST.

Server di code e file di code

I server di code sono processi MQSeries che mediano la lettura, la scrittura e la memorizzazione dei messaggi. Essi rappresentano un componente chiave di MQSeries. La configurazione del server di code può influire notevolmente sulle prestazioni dei sistemi MQSeries.

I server di code vengono utilizzati per la memorizzazione fisica dei messaggi presenti nelle code. E' utile esaminare la memorizzazione dei messaggi permanenti e non permanenti separatamente, poiché i server di code gestiscono tali messaggi in modi diversi.

Un server di code può gestire una o più code. Quando viene creata, una coda viene gestita dal server di code predefinito. Pertanto, per impostazione predefinita tutte le code vengono gestite dal server di code predefinito, a meno che esse non vengano assegnate ad altri server.

Per ciascuna coda gestita da un server di code, sono previsti i seguenti file:

- Un file di coda
- Un file overflow coda
- Un touch file

Può inoltre esistere un file di overflow messaggio per ciascun messaggio lungo nella coda (come definito dalla soglia di overflow messaggio della coda). I file di overflow messaggio verranno illustrati successivamente.

Memorizzazione di messaggi permanenti

I messaggi permanenti vengono protetti sempre sul disco. Il modo in cui vengono memorizzati i messaggi permanenti dipende soprattutto dalla loro dimensione:

Dimensione del messaggio	Tipo di memorizzazione
< 3000 byte (appross.)	Memorizzati nel file di coda.
> 3000 byte (appross.) <= Soglia di overflow messaggio	1 st 3000 byte (appross.) memorizzato nel file di coda, il resto nel file di overflow della coda
> Soglia di overflow messaggio	1 st 3000 byte (appross.) memorizzato nel file di coda, il resto nel file overflow messaggio

Memorizzazione dei messaggi non permanenti

I messaggi non permanenti vengono memorizzati in buffer di memoria nel processo del server di coda. Per questo motivo, il server di code non esegue attività IE disco durante la lettura e la scrittura di messaggi non permanenti.

Distribuzione della CPU del server di code

I server di code sono classi server PATHWAY, pertanto, vengono definiti nel PATHWAY del queue manager. Quando viene creato il queue manager, nel PATHWAY viene definito un server di code predefinito con il processo principale del server di code in esecuzione in CPU 0. Poiché CPU 0 normalmente contiene molti processi di sistema ad alta priorità, è preferibile non eseguire un server di code in CPU 0. Piuttosto, è necessario identificare un numero di CPU meno

utilizzate e creare un server di code in ciascuna di esse. Un nuovo server di code è in grado di gestire le code solo dopo che queste ultime sono state ad esso assegnate.

Riassegnazione delle code ai server di code utilizzando **altmqfls**

Quando si crea una coda (utilizzando **runmqsc**), MQSeries la assegna al server di code predefinito. A meno che non vengano creati nuovi server di code cui vengono assegnate delle code, tutte le code vengono gestite da un unico server di code predefinito.

In un sistema MQSeries con molti messaggi non è una scelta efficace né scalabile assegnare tutte le code ad un unico server di code. Per distribuire il carico della CPU del server di code sulle CPU disponibili, è opportuno creare nuovi server di code cui assegnare le code.

E' possibile utilizzare **altmqfls** per assegnare una coda ad un diverso server di code. Ad esempio:

```
altmqfls --qmgr QMGR --type QLOCAL --server $QS01 TEST.QUEUE
```

Per ulteriori informazioni sull'utilizzo di **altmqfls**, consultare "altmqfls (Modifica attributi del file di coda)" a pagina 254.

Coda di trasmissione cluster: **SYSTEM.CLUSTER.TRANSMIT.QUEUE**

MQSeries utilizza un'unica coda di trasmissione cluster per tutte le operazioni di cluster. Se il queue manager è parte di un cluster molto utilizzato, è necessario assegnare la coda ad un server di code dedicato per ottimizzare le prestazioni delle operazioni di cluster e per ridurre l'incidenza sulle altre applicazioni.

Modifica della posizione del file di code utilizzando **altmqfls**

Per impostazione predefinita, i tre file di coda principali (coda, overflow coda e touch file) vengono memorizzati nel volume secondario M del queue manager. I file di overflow messaggi vengono memorizzati nel volume secondario del server di code.

E' opportuno riposizionare i file di coda in altri volumi del disco per due motivi:

- Per distribuire il carico IE disco in modo equilibrato
- Per superare i limiti del sistema operativo in merito al numero di file aperti per volume del disco

E' possibile utilizzare **altmqfls** per spostare i file di coda esistenti su un altro volume del disco:

```
altmqfls --qmgr QMGR --type QLOCAL --volume $DATA01 TEST.QUEUE
```

Per spostare una coda utilizzando **altmqfls**, la coda non deve essere aperta.

Server di code e file di code

Non è possibile spostare i file di overflow messaggi esistenti. E' tuttavia possibile creare altrove i file di overflow messaggi utilizzando l'opzione `--msgofsubvol` di **altmqfls**:

L'utente può far sì che i nuovi file di overflow messaggi vengano creati in una nuova ubicazione:

```
altmqfls --qmgr QMGR --type QLOCAL --msgofsubvol $DATA01.TESTMOF TEST.QUEUE
```

Per ulteriori informazioni sull'utilizzo di **altmqfls**, consultare "altmqfls (Modifica attributi del file di coda)" a pagina 254.

Suddivisione dei file di coda

La suddivisione dei file consente di suddividere un file su più di un volume del disco. Il file quindi sarà costituito da più di una partizione logica su ciascun volume del disco.

Vantaggi della suddivisione di un file di coda:

- Il carico IE del disco per una singola coda viene ripartito su più volumi del disco, quindi su più percorsi IE
- Un file di coda suddiviso può avere dimensioni maggiori di quelle consentite per un file fisico su un volume del disco.

Per suddividere un file di coda esistente, utilizzare i comandi TACL standard (programma di utilità FUP). MQSeries fornisce uno script TACL denominato PARTIT in cui viene descritto l'utilizzo del programma FUP per la suddivisione di un file di coda esistente.

Dopo avere suddiviso un file di coda, il server di code tenta di distribuire equamente i nuovi messaggi su tutte le partizioni disponibili. In tal modo, vengono migliorati i vantaggi della suddivisione.

Nota: **altmqfls** non è in grado di spostare un file di coda suddiviso.

File di overflow messaggi

I file di overflow messaggi vengono creati per ciascun messaggio più lungo della soglia di overflow messaggi. I file di overflow messaggi sono file non strutturati che non vengono controllati da TM/MP. Per i messaggi molto lunghi è preferibile memorizzare la maggior parte di essi nel file di overflow messaggi (non controllato da TM/MP). Per i messaggi brevi è preferibile memorizzare l'intero messaggio nei file di coda controllati da TM/MP (file di coda e file di overflow coda). Il punto di incrocio viene determinato empiricamente intorno ai 200 KB.

La soglia di overflow messaggi predefinita è impostata su 200 KB quando viene creata una coda. Tale valore può essere modificato utilizzando l'opzione `--oflowsiz` di **altmqfls**.

E' possibile utilizzare il comando **altmqfls** per impostare la soglia di overflow messaggi sul nuovo valore:

```
altnmqfls --qmgr QMGR --type QLOCAL --oflowsize 400000 TEST.QUEUE
```

Buffer quando si sfogliano i messaggi

Il server di code può conservare in memoria i primi n byte di tutti i messaggi permanenti di una coda. Questa funzione può incidere notevolmente sulle prestazioni delle applicazioni che stanno sfogliando i messaggi permanenti contenuti in una coda. Se i messaggi sfogliati dall'applicazione sono più brevi della soglia prevista, per sfogliarli non è richiesta nessuna attività IE del disco.

E' inoltre possibile inserire delle informazioni nei primi 100 byte dei messaggi permanenti per identificarne il contenuto. L'utente può sfogliare i primi 100 byte di dati di ciascun messaggio per individuare quelli utili senza utilizzare le risorse IE del disco. Una volta trovato il messaggio, è possibile leggerne il contenuto e, se necessario, eliminarlo dalla coda.

L'utente può utilizzare l'opzione `--browse` di **altnmqfls** per impostare la soglia di browse di una coda. Per impostazione predefinita, i dati dei messaggi permanenti non vengono conservati in memoria (ad esempio, la soglia di browse è zero). Il valore massimo è 25000.

```
altnmqfls --qmgr QMGR --type QLOCAL --browse 100 TEST.QUEUE
```

Altre opzioni del server di code

Per le altre opzioni del server di code (ad esempio, se la coda viene caricata dal disco nella cache quando viene avviato il queue manager oppure se i messaggi non permanenti vengono controllati dal server di code di backup), viene utilizzato il parametro `--qsoptions` con le opzioni L, S e C. Tali opzioni possono essere utilizzate da sole o combinate per regolare l'affidabilità e la possibilità di richiamare la coda.

Nota: *Tutte* le `--qsoptions` SLC vengono impostate ogni volta che viene emesso il comando. Ad esempio `--qsoptions S` annullerà l'impostazione di L e C. L'opzione `--qsoptions` può essere specificata solo una volta su una riga comandi.

Caricamento all'avvio

Controllata dall'opzione **altnmqfls** `--qsoptions S`. Tale opzione consente al server di code di leggere i file di coda e di creare le strutture interne dei dati dei messaggi all'avvio del server di code (generalmente all'avvio del queue manager) piuttosto che alla prima apertura della coda.

```
altnmqfls --qmgr QMGR --type QLOCAL --qsoptions S TEST.QUEUE
```

L'utilizzo di tale opzione riduce l'attività della CPU e di IE del disco alla prima apertura di una coda, ma richiede una maggiore attività del server di code (CPU e IE del disco) all'avvio del queue manager.

Server di code e file di code

Conservazione nella cache

Controllato dall'opzione **altmqfls --qsoptions L**. Tale opzione consente al server di code di conservare in memoria le strutture dei dati e le catene associate ad una coda. Le strutture di dati in memoria della coda non vengono scaricate sul disco per lasciare spazio alle altre code. Il funzionamento predefinito prevede che le strutture dei dati della coda vengano scaricate sul disco quando è necessario.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions L TEST.QUEUE
```

L'utilizzo di tali opzioni facilita l'accesso alle strutture di dati in memoria di una coda a spese delle altre code.

Controllo NPM

Controllato dall'opzione **altmqfls --qsoptions C**. Tale opzione fa sì che il server di code verifichi i messaggi non permanenti con il suo processo di backup continuo. Per impostazione predefinita, vengono controllati i dati dei messaggi non permanenti.

L'utilizzo di tale opzione garantisce un più elevato grado di affidabilità per i messaggi non permanenti, ma prevede anche un maggiore traffico IPC ed un maggiore utilizzo di CPU e di memoria per i processi del server di code principale e di backup.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions C TEST.QUEUE
```

Measure Counter (calcoladimensioni)

Controllato dall'opzione **altmqfls --meascount**. Tale opzione fa sì che il server di code conservi un Measure Counter definito dall'utente con il CURDEPTH della coda. Il measure counter è utile quando si raccolgono i dati per valutare le prestazioni del sistema. Utilizzando questa opzione, è possibile collegare l'attività di lettura e di scrittura dei messaggi (MQGET e MQPUT) ad una coda con altre variabili di sistema, quali CPU e IE del disco.

```
altmqfls --qmgr QMGR --type QLOCAL --meascount TESTCOUNT1 TEST.QUEUE
```

Utilizzando questa opzione è possibile valutare accuratamente le prestazioni e la scalabilità di un sistema utilizzando MQSeries.

Assegnazione di CPU

Quando si avvia un queue manager, MQSeries for Compaq NSK crea un numero di processi. Alcuni di questi processi forniscono le operazioni di messaggistica principali mentre altre eseguono funzioni che supportano indirettamente tali operazioni. I processi che forniscono le funzioni di messaggistica principali di MQSeries vengono impegnate quando le applicazioni creano le chiamate di messaggi MQI (MQGET e MQPUT).

Processi MQSeries e Compaq NSK coinvolti nelle operazioni di messaggistica principali:

- LQMA (Local Queue Manager Agents)
- Server di code

- Status server
- Processi disco NSK

Processi MQSeries e NSK coinvolti nelle operazioni di accodamento:

- MCA
- Status Server predefinito
- Processi NSK TCPIP o SNA

Processi MQSeries coinvolti nelle operazioni di supporto o di gestione:

- Queue Manager Server
- Repository Server
- Execution Controller (EC e ECBoss)
- Channel Initiator
- Trigger Monitor
- Listener

Un sistema MQSeries con carichi di lavoro elevati generalmente si caratterizza per un elevato utilizzo della CPU per i processi nelle prime due categorie precedentemente illustrate (processi di messaggistica principali e processi di accodamento distribuiti).

Distribuire il carico di CPU di MQSeries significa distribuire LQMA del queue manager, i server di coda ed MCA su diverse CPU.

La configurazione predefinita per un nuovo queue manager prevede l'esecuzione di processi in CPU 0 (per coppie di processi Nonstop, i processi principali vengono eseguiti in CPU 0 ed il backup in CPU 1). Tale configurazione non è adeguata per un ambiente di produzione, pertanto può essere necessario riconfigurare il PATHWAY del queue manager per distribuire i processi su diverse CPU. CPU 0 generalmente contiene molti processi del sistema operativo ad alta priorità, pertanto è improbabile che possa essere utilizzata per i processi MQSeries utilizzati.

La distribuzione di CPU di EC, definita nel PATHWAY del queue manager, controlla l'esecuzione di LQMA e MCA. I server di coda sono anche definiti nel PATHWAY del queue manager.

Al di là dei processi MQSeries, i processi sul disco del sistema operativo NSK costituiscono una componente importante delle operazioni di messaggistica (in modo particolare, quando i server di coda gestiscono i messaggi permanente). Il grande utilizzo dell'accodamento distribuito sulla rete causa il consumo della CPU da parte dei corrispondenti processi TCPIP o SNA. Nel valutare il profilo complessivo delle prestazioni su un sistema MQSeries, è necessario considerare il numero di processi e l'ubicazione della CPU.

Per informazioni sull'assegnazione dei processi alle CPU, sulla configurazione dei processi del disco o sul TCPIP, consultare la documentazione sul sistema Compaq NSK. Per informazioni sulla configurazione dei processi SNA, consultare la documentazione SNAX o ICE. Per informazioni sulla configurazione dei processi TCPIP o SNA, consultare "Appendice M. Impostazione delle comunicazioni" a pagina 377.

Programmi applicativi per binding FASTPATH

Binding FASTPATH è una funzione di MQI progettata per migliorare l'efficacia delle applicazioni MQSeries. Binding FASTPATH può essere utilizzato per ridurre il carico di istruzioni MQI emesse dai programmi applicativi MQSeries. Le applicazioni che utilizzano Binding FASTPATH vengono definite applicazioni attendibili per la somiglianza del software del queue manager e della memoria con il software delle applicazioni del cliente. Eventuali errori nelle applicazioni attendibili possono danneggiare le strutture di dati MQSeries e possono compromettere l'integrità del queue manager.

Background

Quando un programma applicativo esegue un'istruzione MQCONN, MQSeries crea (o riutilizza) un processo specifico denominato LQMA (Local Queue Manager Agent). LQMA serve tutte le successive chiamate MQI eseguite dall'applicazione che utilizza quell'handle. In MQSeries for Compaq NSK, LQMA può utilizzare la stessa CPU o una CPU diversa da quella utilizzata dall'applicazione di connessione.

Poiché LQMA è un processo separato, il programma applicativo non ha accesso diretto alla memoria o ai file utilizzati da MQSeries. Un programma applicativo non può pertanto danneggiare LQMA. In tal modo, il software MQSeries e le strutture di dati importanti per le operazioni vengono isolati dai dati e dal software del cliente. Tale isolamento ha un prezzo. L'istruzione MQCONN non può essere completata finché non viene creato il nuovo processo LQMA (o viene riutilizzato un processo esistente). Il costo maggiore è dato dal fatto che le informazioni devono essere inviate a LQMA ogniqualvolta l'applicazione emette un'istruzione MQI.

Tali informazioni MQI vengono inoltrate a LQMA mediante un meccanismo IPC (Interprocess Communications). Le richieste IPC possono essere intra-CPU (se LQMA viene eseguito sulla stessa CPU delle applicazioni) oppure inter-CPU, che sono anche quelle più costose.

Riduzione del carico di MQI

I creatori di applicazioni possono utilizzare il Binding FASTPATH per eliminare il carico IPC delle applicazioni LQMA associato a ciascuna istruzione MQI (viene così evitata anche la creazione del processo LQMA). Quando si abilita il Binding FASTPATH per un'applicazione, non viene utilizzato nessun processo LQMA separato. Invece, i componenti di MQSeries normalmente contenuti in LQMA, vengono caricati sul processo dell'utente (ad esempio, il processo dell'applicazione di connessione).

Le successive istruzioni MQI emesse dall'applicazione non richiedono nessuna attività IPC con LQMA, poiché il software MQSeries e le strutture dei dati (normalmente memorizzate nel processo LQMA) vengono conservate localmente nel processo dell'applicazione. E' possibile che siano necessarie altre attività IPC quando il queue manager deve comunicare con altri processi MQSeries, ad esempio, i server di code. Binding FASTPATH non rimuove tutte le attività IPC, ma soltanto una fonte importante di tali attività.

Abilitazione del binding FASTPATH

Per utilizzare il binding FASTPATH, collegarsi al queue manager utilizzando l'istruzione MQCONN con l'opzione MQCNO_FASTPATH_BINDING. Il valore

Programmi applicativi per binding FASTPATH

(se presente) di MQCONNECTTYPE PARAM influisce sul funzionamento di MQCONNX. Se è presente MQCONNECTTYPE PARAM, il suo valore deve essere FASTPATH per consentire a MQCONNX di impostare una connessione FASTPATH. Dopo avere stabilito una connessione binding FASTPATH, il funzionamento di tutte le altre istruzioni MQI sarà quello previsto per una connessione di binding standard, con le eccezioni illustrate di seguito.

Limitazioni durante l'utilizzo del binding FASTPATH

Quando si utilizzano le connessioni di binding STANDARD, il software applicativo viene isolato dai dati interni di MQSeries. Tale isolamento viene meno quando un'applicazione affidabile stabilisce una connessione di binding FASTPATH. Eventuali errori nelle applicazioni affidabili possono quindi danneggiare le strutture di dati MQSeries e possono compromettere l'integrità del queue manager. Tale possibilità va considerata quando si valuta se utilizzare i binding FASTPATH per una determinata applicazione.

Per le applicazioni affidabili si considerino anche le seguenti valutazioni:

- Le applicazioni attendibili **devono** disconnettersi in modo esplicito dal queue manager (ad esempio, emettere MQDISC).
- le applicazioni attendibili devono essere chiuse prima che venga digitato il comando **endmqm**. Le applicazioni attendibili devono essere eseguite utilizzando l'ID utente dell'amministratore (l'ID utente corrispondente al principal mqm).
- Le applicazioni attendibili possono essere eseguite solo in CPU con un repository server MQSeries in esecuzione.

Programmi applicativi per binding FASTPATH

Capitolo 16. Integrità e disponibilità dei dati

Questo capitolo descrive i concetti di integrità e disponibilità dei dati e sul come questi importanti aspetti di un sistema si riflettono sulla gestione e la configurazione di MQSeries for Compaq NSK V5.1. Questo capitolo descrive i livelli di integrità e disponibilità dei dati offerti da MQSeries e le scelte di configurazione che possono influenzare questi livelli. Contiene le seguenti sezioni:

- “Integrità dei dati”
- “Disponibilità” a pagina 232
- “Dati permanenti e non permanenti” a pagina 233
- “Congruenza dei database” a pagina 236
- “Processi principali” a pagina 239
- “Cluster” a pagina 245
- “Considerazioni sulla configurazione per la disponibilità” a pagina 246
- “Considerazioni sulla configurazione per l’integrità dei dati” a pagina 246

Bisogna aver letto e compreso “Capitolo 2. Architettura di MQSeries for Compaq NSK V5.1” a pagina 21 per comprendere in modo corretto ed utilizzare le informazioni contenute in questo capitolo.

Integrità dei dati

Il concetto di integrità dei dati è meglio comprensibile valutando i seguenti auspicabili aspetti di memorizzazione e gestione dei dati, in particolare per le applicazioni di elaborazione delle transazioni on-line.

- Quando un record di dati viene scritto o letto da un record in un database, i dati non devono essere danneggiati, duplicati o persi senza un’indicazione dell’eventuale condizione di errore durante il trasferimento.
- Quando è necessario far accedere i dati in modo simultaneo da più processi, tali processi devono essere presentati con la stessa vista dei dati e i dati devono essere protetti da eventuali danni, duplicazione o perdita.
- Quando una serie di modifiche consistenti è richiesta per i dati presenti in più database, tutte le modifiche devono essere apportate o non deve essere apportata alcuna modifica.

Per MQSeries, i requisiti di integrità dei dati per la memorizzazione dei dati elencata precedentemente sono applicabili alle operazioni di messaggistica (ad esempio, MQPUT e MQGET) sulle code. Tenere presente che la consistenza delle molteplici modifiche del database devono essere preservate nelle e tra i database dell’applicazione e le code di MQSeries.

MQSeries for Compaq NSK V5.1 è stato progettato per conservare l’integrità dei dati per le operazioni sui dati persistenti attraverso qualsiasi punto singolo di errore (hardware o software). Infatti, l’integrità dei dati può essere mantenuta in diversi casi attraverso più punti di errore. Ciò non implica che i messaggi non persistenti non siano sicuri; l’architettura del server di code fornisce le funzioni per la creazione di messaggi non persistenti non sicuri come quelli persistenti, ad eccezione del caso di un errore grave di sistema. (Per ulteriori informazioni sull’architettura del server di code fare riferimento a “Capitolo 2. Architettura di MQSeries for Compaq NSK V5.1” a pagina 21.)

Integrità dei dati

Con l'integrità dei dati di MQSeries for Compaq NSK V5.1 viene fornita una combinazione di funzioni fondamentali del software e dell'hardware del sistema Compaq NSK e del software MQSeries stesso.

Esistono diversi modi in cui il livello di integrità dei dati può influire sulle selezioni nella configurazione di MQSeries:

- Selezione della persistenza dei messaggi dall'applicazione
- Selezione della tecnica di memorizzazione per i messaggi persistenti
- Selezione di opzioni di sintonizzazione messaggi non persistenti
- Selezione delle opzioni della configurazione del server di code
- Configurazione dell'hardware che supporta i file di code
- Utilizzo e configurazione di NonStop TM/MP
- Utilizzo e configurazione di Remote Database Facility (NonStop RDF).

Ciascuna selezione viene descritta successivamente in questo capitolo.

Disponibilità

La disponibilità in termini generali è una misura del tempo che un'applicazione oppure un servizio impiega per essere operativo e utilizzabile rispetto al tempo di scadenza. Quindi, *la disponibilità continua* esprime lo scopo finale di tali sistemi. Naturalmente, tali misure non significano niente senza un periodo di tempo corrispondente associato alla misurazione—poiché è facile dichiarare una disponibilità del 100% per un periodo di tempo breve.

In una situazione reale con un margine di tempo operativo ragionevole, un sistema subisce diversi tipi di sfida alla sua disponibilità:

- Errori hardware e software del system
- Errori all'interno del software di applicazione stesso
- La necessità di apportare modifiche a qualsiasi aspetto del sistema per la manutenzione preventiva
- Il carico di traffico o delle transazioni che eccede i limiti del progetto o delle risorse

Per quanto riguarda l'integrità dei dati, con MQSeries for Compaq NSK V5.1, la disponibilità viene fornita da una combinazione di funzioni fondamentali del software e dell'hardware di sistema di Compaq NSK e di MQSeries stesso.

Bisogna tenere presente che (almeno nella sua forma corrente) MQSeries for Compaq NSK non intende fornire un livello di disponibilità continuo equivalente a quello fornito dal software di sistema di Compaq NSK come ad esempio il file system. Ci sono infatti alcuni componenti del queue manager che forniscono questo livello di disponibilità ma il queue manager in sé no.

MQSeries for Compaq NSK, V5.1 è progettato per fornire un livello di disponibilità tale che qualora si verifici un malfunzionamento (hardware o software):

- Le connessioni del queue manager che vengono interrotte o che soffrono di una discontinuità nel servizio sono limitate a quelle con componenti direttamente influenzati dal malfunzionamento (ad esempio, nel caso di un malfunzionamento di una CPU, le connessioni interessate dal malfunzionamento saranno solo quelle che fornite dai processi LQMA che erano in esecuzione in detta CPU).

- Il queue manager rimane disponibile per nuovi tentativi di connessione senza che sia richiesto un intervento manuale da parte degli amministratori di sistema.
- L'accesso ad un oggetto queue manager (ad esempio una coda locale) non deve essere impedito da connessioni diverse da quelle direttamente interessate dal malfunzionamento.

Oltre a queste funzioni specifiche per Compaq NSK di MQSeries, ci sono varie altre funzioni comuni a tutte le piattaforme MQSeries Versione 5.1 che è possibile utilizzare per migliorare la disponibilità di MQSeries for Compaq NSK. Ci sono vari modi in cui è possibile, operando delle selezioni nella configurazione di MQSeries, di modificare il livello di disponibilità:

- Selezione del design delle applicazioni
- Selezione delle opzioni di configurazione PATHWAY
- Utilizzo e configurazione delle funzioni MQSeries standard (incluso i cluster)
- Selezione di hardware che supporta MQSeries
- Selezione delle opzioni di configurazione dei messaggi non permanenti
- Utilizzo e configurazione di NonStop TM/MP
- Utilizzo e configurazione di NonStop RDF.

Dati permanenti e non permanenti

Quando utilizzato per MQSeries, il termine *permanenza* implica varie qualità dei dati:

- Una modifica dei dati permanenti sopravvive il riavvio del queue manager
- I dati permanenti vengono memorizzati in supporti non temporanei
- I dati permanenti soddisfano i più elevati requisiti di integrità dei dati forniti da uno specifico ambiente operativo
- La velocità e l'utilizzo delle risorse delle operazioni sui dati permanenti influenzano questo livello superiore.

La maggior parte delle opzioni di gestione vengono eseguite sui dati permanenti, poiché i database di configurazione di MQSeries devono disporre del livello più elevato di integrità dei dati per ridurre al minimo il rischio che la disponibilità di MQSeries venga sensibilmente ridotto. La velocità o l'utilizzo delle risorse della maggior parte delle operazioni di gestione non sono fattori di primaria importanza in quanto non vengono eseguite molto spesso.

Esempi di operazioni di gestione permanenti:

- Modifica degli attributi di una coda
- Modifica degli attributi di un canale
- Creazione di una code, processi, namelist o canali nuovi.

Esempi di operazioni di gestione on permanenti:

- Avvio o arresto di un canale
- Esecuzione di inquiry sugli attributi di un oggetto queue manager.

MQSeries for Compaq NSK fornisce varie modalità di memorizzazione dei messaggi che variano sulla base della modalità di permanenza o non permanenza selezionata dall'applicazione in fase di accodamento di un messaggio e delle selezioni di configurazione del livello di coda operate dall'amministratore di sistema.

dati permanenti e non permanenti

Messaggi permanenti

I messaggi permanenti vengono sempre memorizzati su disco. Un amministratore di sistema può scegliere tra due tecniche di memorizzazione per i messaggi permanenti su una coda basate sulle code. La selezione del tipo di tecnica di memorizzazione da utilizzare è basata sulla dimensione dei messaggi poiché l'obiettivo primario per l'implementazione di diverse tecniche di memorizzazione consiste nel migliorare le prestazioni delle operazioni di messaggistica su messaggi di notevoli dimensioni.

Tutti i messaggi permanenti hanno un singolo record nel file di coda non controllato. Questo record contiene le intestazioni ed importanti informazioni di controllo del messaggio, oltre al numero massimo di dati del messaggio che può essere contenuto nella dimensione massima di record di 4096 byte.

Il meccanismo più rapido e più efficace per la memorizzazione di messaggi permanenti di piccole e medie dimensioni sotto il controllo TM/MP consiste nel memorizzare i dati di messaggio di overflow nel file di overflow di coda in più record, utilizzando la stessa tecnica di base utilizzata in MQSeries for Compaq NSK Versione 2.2.0.1. Tutti i dati per questi messaggi vengono registrati in TM/MP e sono pertanto, se necessario, completamente ripristinabili dalle tracce di controllo.

Per i messaggi di notevoli dimensioni (superiori ai 200 KB di dati), il meccanismo più efficiente consiste nell'utilizzo di un file di overflow di messaggio non controllato. I dati scritti nel file di overflow di messaggio non vengono pertanto scritti nella traccia di controllo TM/MP, salvando così CPU e I/E di disco e possono anche essere trasferiti in grossi blocchi.

Le differenze relative all'integrità dei dati, leggermente differenti tra i due meccanismi, sono descritti in: Tabella 6:

Tabella 6. Messa a confronto dell'overflow di coda con l'overflow di messaggio

Aspetto dell'integrità dei dati	Metodo di overflow di coda	Metodo di overflow di messaggio
Quantità di dati controllata da TM/MP	Vengono controllati tutti i messaggi	Vengono controllati solo i dati che rientrano nel record del file di coda
Ripristinabilità dei dati dalla traccia di controllo in caso di più malfunzionamenti che determinano una perdita totale nei volumi.	Tutto il messaggio è ripristinabile dalla traccia di controllo	Solo l'intestazione di messaggio e la prima parte dei dati è ripristinabile dalla traccia di controllo
Dimensione massima possibile dei messaggi	Limitata dal numero di blocchi di record per volume per transazione fissato da ENSCRIBE. Per un file senza partizioni, è pari a circa 20 MB per una configurazione ENSCRIBE predefinita. Il limite pratico può essere inferiore a quello qui indicato a causa di limiti nella memoria fisica.	Limitato solo dallo spazio su disco disponibile di 100 MB che è la dimensione di messaggio massima che è possibile consentire per MQSeries.

Tabella 6. Messa a confronto dell'overflow di coda con l'overflow di messaggio (Continua)

Aspetto dell'integrità dei dati	Metodo di overflow di coda	Metodo di overflow di messaggio
Compatibilità con Remote Database Facility (RDF)	Pienamente compatibile (specificare il parametro di ambiente MQRDF).	Incompatibile con RDF. Poiché i dati di messaggio non vengono controllati, non è possibile utilizzare RDF per trasmettere le operazioni di messaggio utilizzando dei file di messaggio al sistema di backup.
Capacità di recupero da condizioni di malfunzionamento in caso di problemi hardware ai dischi	Capacità di recupero piena per ciascun singolo punto di malfunzionamento se si utilizzano dei dischi di mirroring e, poiché vengono controllati tutti i dati, è possibile ripristinare i file nel caso in cui si verifichi un malfunzionamento di entrambi i dischi in una coppia con mirroring oppure se il singolo disco non ha alcun mirroring.	Capacità di recupero piena per un singolo punto di malfunzionamento se vengono utilizzati dischi con mirroring per conservare i file di overflow di messaggio. I dati di messaggio non possono essere ripristinati nel caso si verifichi un malfunzionamento totale dei volumi.

In sintesi, per i messaggi permanenti, entrambe le tecniche di memorizzazione consentono un elevatissimo livello di integrità dei dati, ma i file di overflow di messaggio presentano dei limiti poiché non vengono controllati tutti i dati di messaggio. Di norma, questi limiti valgono solo per messaggi di notevoli dimensioni e, in tali casi, è possibile superarli utilizzando la segmentazione (la divisione di un messaggio applicativo in vari messaggi fisici di dimensioni inferiori).

Messaggi non permanenti

I messaggi non permanenti (PM) vengono di norma registrati in memoria. Se un queue server sta gestendo una grossa quantità di dati non permanenti e raggiunge una soglia prossima all'esaurimento della memoria virtuale, il queue server forzerà i messaggi non permanenti verso il disco. In alcune occasioni in queue server copierà i messaggi non permanenti sui file di disco per alcune operazioni di gestione durante l'apporto di modifiche alla configurazione online. Quando le applicazioni accedono ai messaggi non permanenti, questi si trovano sempre in memoria.

Il queue server gestisce la memorizzazione di tutti i messaggi per le code di cui è responsabile e, per ciascuna cosa, fornisce un'opzione di configurazione per controllare il livello di integrità dei dati applicato ai messaggi non permanenti. Il parametro di configurazione controlla se il queue server esegue il checkpoint dei messaggi non permanenti nel processo di backup, per fare in modo che i messaggi non permanenti abbiano la stessa resistenza dei messaggi permanenti ad eventuali condizioni di malfunzionamento della CPU che contiene il processo di queue server principale.

Per utilizzare il checkpoint dei messaggi non permanenti, il queue server deve utilizzare più CPU (principale e di backup), trasferire più dati al backup durante

dati permanenti e non permanenti

L'esecuzione del checkpoint e consumare più memoria nel processo di backup poiché deve nuovamente memorizzare il messaggio.

Se il checkpoint dei messaggi non permanenti non è abilitato per una coda, ed il processo queue server principale termina in modo anomalo (per un malfunzionamento del processo o della CPU), i messaggi non permanenti presenti nella coda prima del verificarsi del malfunzionamento andranno persi poiché erano memorizzati solo nel processo principale.

L'opzione di messaggi non permanenti capaci di resistere a condizioni di malfunzionamento non è disponibile su altre piattaforme MQSeries, e per congruenza in tutta la linea di prodotti, il checkpoint dei messaggi non permanenti è abilitato per impostazione predefinita (utilizzare **altmqfls** per disabilitare questa funzione). Valutare attentamente se i messaggi non permanenti soddisfano i propri requisiti di integrità dei dati. In caso affermativo, si potrà usufruire dei vantaggi rappresentati dal notevole risparmio di utilizzo delle risorse e dalle prestazioni più elevate che essi consentono.

Congruenza dei database

La congruenza dei database dee essere conservata sia internamente da MQSeries, sia esternamente quando si utilizza l'opzione Syncpoint per le operazioni di messaggistica. Entrambe sono di fondamentale importanza per la disponibilità e l'integrità dei dati.

Congruenza dei database interni

Tutti i file di database più importanti nel queue manager sono controllati da TM/MP. I processi queue manager devono pertanto utilizzare le transazioni per apportarvi modifiche. Tutte le modifiche vengono registrate nella funzione di traccia TM/MP ed anche (se utilizzato) duplicate utilizzando RDF in uno o più sistemi di recupero da malfunzionamento. E' pertanto possibile assicurare il più elevato livello possibile di integrità dei dati per i database interni.

L'utilizzo di TM/MP per proteggere i database garantisce che in caso di malfunzionamenti di sistema o software l'integrità dei database più importanti non venga compromessa. Questo significa una maggiore disponibilità sarà praticamente garantita poiché il riavvio (automatico o manuale) dei processi o dei servizi che utilizzano questi database avrà quasi sicuramente esito positivo.

Congruenza dei database esterni

La coordinazione delle modifiche apportate e la congruenza dei database esterni con i database MQSeries è abilitata dall'utilizzo di TM/MP in MQSeries e dalle applicazioni. Le operazioni di messaggistica di MQSeries possono essere eseguite sotto controllo syncpoint, che richiede che l'applicazione abbia una transazione corrente attiva (ereditata da un altro processo o avviata utilizzando il servizio TM/MP BEGINTRANSACTION). Questa transazione viene ereditata dal queue manager e operazioni di aggiornamento dei database più importanti o di memorizzazione richieste vengono eseguite utilizzando questa transazione. TM/MP garantisce che le appropriate voci di controllo di traccia vengano registrate per tutti le operazioni di I/E di disco eseguite sotto la transazione in qualsiasi processo esegua l'operazione di aggiornamento.

TM/MP conserva inoltre una vista congruente degli aggiornamenti che sono stati eseguiti ma di cui non è ancora stato eseguito il commit conservando dei blocchi di record sui record interessati in tutti i file di database. Ad esempio, una riga in una

tabella SQL inserita sotto controllo TM/MP non può essere aggiornata o cancellata se non viene prima eseguito il commit della transazione.

Quando il queue manager risponde al processo applicativo (ed esegue la restituzione dalla MQI), l'applicazione può continuare ad eseguire altre operazioni sotto questa transazione—utilizzando MQSeries per inserire ed estrarre dalle code altri messaggi oppure eseguendo proprie operazioni di aggiornamento del database utilizzando ENSCRIBE o NonStop SQL. Quando l'applicazione è pronta ad eseguire le modifiche alle code MQSeries e ad altri database in modo permanente, viene richiamato il servizio ENDTRANSACTION di TM/MP, che esegue il commit per tutti i database, simultaneamente e su tutto il sistema.

Se l'applicazione determina che si è verificato un errore durante l'elaborazione della transazione, e sono già stati eseguiti degli aggiornamenti dei database, l'applicazione può richiamare ABORTTRANSACTION per eseguire il backout TM/MP delle modifiche per tutti i database, simultaneamente e su tutto il sistema. Questo può determinare, ad esempio, il reinserimento in una coda di un messaggio già estratto dalla coda in un'operazione Syncpoint MQGET oppure la rimozione di un precedente inserimento in una tabella di database SQL. Studiando attentamente il design delle applicazioni, è possibile gestire queste condizioni di errori in modo da conservare la congruenza e migliorare la disponibilità e l'integrità dei dati per le applicazioni.

Quando MQSeries esegue un'operazione di syncpoint MQPUT o MQGET, modifica la profondità di coda presupponendo che verrà eventualmente eseguito il commit della transazione. Pertanto la profondità di coda comprende, oltre ai messaggi di cui è stato eseguito il commit, i messaggi di cui non è stato eseguito il commit. Se si verifica un'interruzione della transazione (deliberata oppure dovuta ad un malfunzionamento), MQSeries modifica la profondità di coda per conservare un valore accurato. Questo è un miglioramento rispetto ai precedenti rilasci della V2.2 di MQSeries su Compaq NSK, dove il queue manager non poteva determinare l'esito delle transazioni e conservare così un accurato valore della profondità di coda in tutte le condizioni.

“OpenTMF” descrive il meccanismo che lo rende adesso possibile.

OpenTMF

OpenTMF è il nome informale per una nuova interfaccia interna per il prodotto NonStop TM/MP che MQSeries, con l'assistenza della Compaq, ha potuto utilizzare per determinare l'esito di una transazione utilizzata dal queue manager per eseguire le operazioni di messaggistica di syncpoint.

Questa nuova funzione di TM/MP è la base per l'introduzione della capacità di elaborazione delle transazioni eterogenea di NonStop TM/MP, introdotta con D42. OpenTMF consente a MQSeries di registrarsi come partecipante nelle transazioni che ha ereditato dalle applicazioni. TM/MP invia quindi a MQSeries una notifica dopo che la transazione è stata completata per indicare se la transazione è stata completata con esito positivo o se ne è stato eseguito il backout. Questa nuova notifica da TM/MP consente a MQSeries di:

- Conservare un corretto calcolo della profondità di coda in tutte le condizioni
- Conservare in modo accurato le altre informazioni sugli stati interni relativi alle code locali
- Controllare la disponibilità dei messaggi non permanenti interessati dalle operazioni di syncpoint.

Congruenza dei database

- Migliorare l'efficienza ed i tempi di risposta delle operazioni MQGET attese.

Dal punto di vista dell'amministratore di sistema, l'utilizzo di TMF è visibile solo in un modo: I processi MQSeries sono visibili in un elenco di resource manager che può essere prodotto utilizzando il comando STATUS RESOURCEMANAGER del programma di utilità TMFCOM. Tutti i processi MQSeries che utilizzano OpenTMF vengono visualizzati in quest'elenco come resource manager di tipo VOLATILE denominati automaticamente da TM/MP.

Non sono richieste speciali operazioni amministrative per questo nuovo utilizzo di TM/MP—MQSeries lo utilizza e lo gestisce automaticamente. Controllare che il parametro di configurazione RMOPENPERCPU (numero massimo di resource manager di tipo VOLATILE e RECOVERABLE per CPU) di TM/MP sia impostato su un valore non superiore al numero massimo di queue server e status server che è possibile eseguire in una singola CPU nel sistema. Nota: bisogna consentire i processi Backup poiché questi server sono NonStop Process Pair. Il valore predefinito di 128 è di norma sufficiente per la maggior parte delle installazioni. Il manuale *Compaq NSK NonStop TM/MP Configuration and Planning Guide* descrive i resource manager e l'elaborazione di transazioni eterogenee.

NonStop Tuxedo

MQSeries può coordinare le operazioni di messaggistica per le applicazioni OSS utilizzando NonStop Tuxedo, poiché questo prodotto è basato su NonStop TM/MP ed utilizza le stesse funzioni per l'elaborazione di transazioni eterogeneo di MQSeries.

Il manuale di Compaq NSK NonStop Tuxedo *System Application Development Guide* fornisce informazioni sull'utilizzo dell'ambiente per le transazioni NonStop Tuxedo e su come interagisce con TM/MP.

Transazioni applicative interleaf

Con MQSeries for Compaq NSK, le applicazioni possono usufruire di quest'unico ambiente per le transazioni in modi non possibili su altre piattaforme. Di norma, su Compaq NSK, un processo può gestire più transazioni simultaneamente. Un aggiornamento al database controllato viene sempre eseguito sotto il controllo della transazione corrente e l'applicazione può passare ad una delle altre transazioni attive prima di eseguire il commit di una qualsiasi di esse. Questo consente ad un'applicazione di eseguire simultaneamente più operazioni di messaggistica di syncpoint di MQSeries.

File dei database più importanti di MQSeries

I file dei database controllati più importanti per MQSeries sono descritti in: Tabella 7.

Tabella 7. File di database controllati principali

Nome descrittivo	Posizione/Nome	Utilizzo
Catalogo oggetti	Volume secondario Data/OBJCAT e ABJCAT	Conserva gli attributi per ciascun oggetto coda, processo e namelist ed il queue manager stesso.

Tabella 7. File di database controllati principali (Continua)

Nome descrittivo	Posizione/Nome	Utilizzo
Definizioni di canali non client	Volume secondario Data/CHDEFS	Conserva gli attributi di ciascun canale non client (tipi di canale SENDER, RECEIVER, SERVER, REQUESTER e CLUSTER)
Definizioni di canali client	Volume secondario Data/CCHDEFS	Conserva gli attributi di ciascun canale client (tipi di canale SVRCONN)
database OAM	Volume secondario Data/OAMDB	Conserva i permessi (diritti di accesso) per ciascun oggetto e principal OAM autorizzato ad accedere al queue manager.
Database dei principal	Volume secondario Data/PRIDB e PRIDBA	Conserva il nome di tutti i principal OAM autorizzati e degli ID utente Guardian cui corrispondono i principal.
file di controllo EC	Volume secondario Data/RUNTIME	Conserva le informazioni utilizzata da EC Boss e dagli EC per coordinare le operazioni di avvio e di ripristino.
Definizioni namelist	Volume secondario Data/Lxxxxxxx	Conserva il contenuto di ciascun oggetto namelist definito.
File di coda	Volume secondario Message/Qxxxxxxx	Conserva le informazioni dell'intestazione ed alcuni dati per tutti i messaggi permanenti su una coda locale.
File di overflow di coda	Volume secondario Message/Oxxxxxxx	Conserva i dati per tutti i messaggi permanenti di medie e grandi dimensioni su una coda locale.
Touch file oggetti	Volume secondari Data e Message/Txxxxxxx	Utilizzati per rilevare le modifiche di gestione agli attributi degli oggetti.
Sync file canali	Volume secondario Sync/Sxxxxxxx	Conserva le informazioni di sincronizzazione dei canali per un'istanza canale.

Processi principali

Tabella 8 a pagina 240 descrive i processi principali del queue manager e mostra come MQSeries è protetto e in che modo può essere ripristinato dai malfunzionamenti causati da cattivo funzionamento del software o del sistema. Nella tabella riportata di seguito, l'emergenza fa riferimento a casi di molteplici malfunzionamenti del sistema, o perdita totale del sistema.

Processi principali

Tabella 8. I metodi di protezione utilizzati per i processi più importanti

Processo	Metodi di protezione utilizzati	Processo di ripristino
Queue Server	NonStop process pair. Mantiene sempre uno stato accurato di code locali e di messaggi, tranne in casi di malfunzionamenti molto gravi.	Riavvia dai database controllati dopo che si è verificato un grave malfunzionamento. Non è richiesto alcun altro ripristino. I messaggi non permanenti andranno persi in caso di malfunzionamenti molto gravi.
Status Server	NonStop process pair. Mantiene sempre uno stato accurato degli oggetti di code non locali e degli stati dei canali, tranne nel caso in cui si verificano malfunzionamenti molto gravi.	Riavvia dai database controllati dopo che si è verificato un grave malfunzionamento. Non è richiesto alcun altro ripristino.
Local Queue Manager Agent (LQMA)	<p>La connessione viene contrassegnata come interrotta per l'applicazione (2009).</p> <p>La funzione di raccolta dei dati obsoleti del repository manager elimina i dati dalle aree di registrazione della cache dei cluster lasciati dai processi per cui si è verificato il malfunzionamento.</p> <p>EC, EC Boss, queue server and Status Server riconoscono immediatamente la condizione di malfunzionamento tramite la connessione NSK IPC e correggono/rettificano i dati di stato in modo appropriato.</p> <p>TM/MP interrompe le transazioni attive ed utilizzate dal processo quando si è verificato il malfunzionamento.</p>	Nessuno. Una LQMA serve una connessione. La connessione viene interrotta in caso di malfunzionamento e l'applicazione deve avviare una nuova connessione.

Tabella 8. I metodi di protezione utilizzati per i processi più importanti (Continua)

Processo	Metodi di protezione utilizzati	Processo di ripristino
Message Channel Agent	<p>Lo Status Server notifica immediatamente il malfunzionamento tramite la connessione NSK IPC e contrassegna in modo appropriato lo stato del canale.</p> <p>Lo Status Server assicura un nuovo tentativo per i canali in uscita per cui si verifica un malfunzionamento.</p> <p>La funzione Adopt MCA può essere utilizzata per consentire il riavvio del canale per cui si è verificato il malfunzionamento.</p> <p>I dati di sincronizzazione canali vengono controllati da TM/MP ed utilizzati da MQSeries per conservare l'integrità del canale. Nella rara eventualità che non sia possibile risolvere in modo automatico una situazione in dubbio, esistono delle funzioni amministrative standard che consentono di risolverla.</p> <p>Tutti gli altri metodi di protezione del processo LQMA.</p>	<p>Nessuno. Una MCA serve un canale. Il canale viene arrestato quando si verifica il malfunzionamento; le funzioni automatiche di MQSeries consentono di riavviarlo.</p> <p>MQSeries for Compaq NSK V5.1 introduce anche nuove funzioni come il clustering e l'heartbeat dei canali che migliorano la disponibilità dei canali.</p>

Processi principali

Tabella 8. I metodi di protezione utilizzati per i processi più importanti (Continua)

Processo	Metodi di protezione utilizzati	Processo di ripristino
Channel Initiator	<p>E' possibile configurare più Initiator per fornire una disponibilità più elevata diffondendo delle code di avvio canali in più Channel Initiator.</p> <p>Il Channel Initiator viene di norma eseguito come una classe server PATHWAY, configurato per un AUTOSTART in caso di malfunzionamento per un certo numero di volte (10, per impostazione predefinita) entro un intervallo di tempo fisso di 10 minuti.</p> <p>Le opzioni di configurazione PATHWAY standard possono essere utilizzate per configurare delle CPU alternative da utilizzare nel caso in cui si verifichi un malfunzionamento della CPU.</p>	<p>Nessuno. Lo Status Server predefinito conserva un accurato stato dei canali in tutte le condizioni ed è responsabile per i nuovi tentativi del canale. Channel Initiator utilizza le funzioni di triggering di MQSeries per determinare l'avvio.</p>
Command Server	<p>Il Command Server esegue comandi di gestione in syncpoint, in modo da preservare la congruenza.</p> <p>Funzioni PATHWAY standard come per il Channel Initiator.</p>	<p>Nessuno. Il riavvio determina una nuova connessione con il queue manager.</p>
EC Boss	<p>Le funzioni di protezione PATHWAY standard come sopra descritto.</p>	<p>EC Boss coordina il ripristino con gli EC utilizzando il file RUNTIME controllato. Gli EC eseguono nuovamente la registrazione con EC Boss e continuano le elaborazioni.</p>
EC	<p>Le funzioni di protezione PATHWAY standard come sopra descritto.</p>	<p>EC coordina il ripristino con EC Boss utilizzando il file RUNTIME controllato. L'EC esegue nuovamente la registrazione con EC Boss e continua le elaborazioni.</p>

Tabella 8. I metodi di protezione utilizzati per i processi più importanti (Continua)

Processo	Metodi di protezione utilizzati	Processo di ripristino
TCP/IP Listener	<p>E' possibile configurare più processi TCP/IP Listener per fornire una maggiore disponibilità diffondendo i canali su più porte, indirizzi IP o processi TCP/IP Server.</p> <p>Le funzioni di protezione PATHWAY standard come sopra descritto.</p>	<p>Nessuno. Il TCP/IP Listener tenta di stabilire una connessione con una delle porte configurate per il queue manager in fase di riavvio disponibili.</p>
Repository Manager o Repository Cache Manager	<p>In caso di malfunzionamento del repository manager o del repository cache manager, gli utenti correnti o nuovi della cache nella stessa CPU non subiscono alcuna interruzione nell'accesso alle operazioni in cluster o di cache.</p> <p>Il repository manager conserva sempre una versione congruente con harden della Cluster nella coda di archivio di memoria (Repository). Quando vengono eseguite delle modifiche, esse vengono apportate in syncpoint con l'MQGET del messaggio dal Cluster Command Queue che determina la modifica, conservando così la congruenza anche se si verifica un malfunzionamento del repository manager durante l'esecuzione dell'harden.</p> <p>Le funzioni di protezione PATHWAY standard come sopra descritto.</p>	<p>In fase di recupero, un processo Repository diventerà il repository manager se non ne esiste già uno nel queue manager. Altrimenti, assumerà il ruolo di un Repository Cache Server per la CPU in cui è in esecuzione.</p> <p>Un repository manager o un repository cache manager stabilisce nuovamente una connessione alla cache se esiste ancora in memoria. In caso contrario, la cache viene ricaricata dal disco, se presente .Se il file di disco non esiste, una nuova cache viene creata nella CPU ed avviata dalla coda di archivio di memoria (Repository).</p>
Queue Manager Server	<p>Il Queue Manager Server esegue l'estrazione dei messaggi scaduti e la generazione dei prospetti di scadenza in syncpoint per evitare che il malfunzionamento del server determini delle incongruenze.</p> <p>Le funzioni di protezione PATHWAY standard come sopra descritto.</p>	<p>Un queue manager riavviato eseguirà nuovamente in modo automatico la risincronizzazione con i queue server quando essi notificano il gruppo corrente di messaggi scaduti che richiedono prospetti ogni volta che eseguono le funzioni di manutenzione.</p>

Processi principali

Tabella 8. I metodi di protezione utilizzati per i processi più importanti (Continua)

Processo	Metodi di protezione utilizzati	Processo di ripristino
Applicazioni – collegate a STANDARD	<p>I malfunzionamenti in MQSeries vengono rilevati da restituzioni di condizioni di errore impreviste o di connessioni interrotte da parte di chiamate MQI. L'applicazione deve richiamare MQDISC e quindi nuovamente MQCONN per ristabilire la connessione con il Queue Manager.</p> <p>Il queue manager rileva il malfunzionamento del processo dell'applicazione immediatamente. Il queue manager esegue un MQDISC implicito per conto dell'applicazione determinando la chiusura ed il rilascio di tutte le risorse aperte.</p> <p>Le eventuali operazioni di syncpoint in corso vengono interrotte da TM/MP e MQSeries reagisce eseguendo le adeguate correzioni alle code locali.</p> <p>Il codice dell'applicazione che contiene gli errori non può danneggiare i database più importanti del queue manager oppure le risorse condivise poiché la sola memoria condivisa che si trova nello spazio di indirizzo dell'applicazione è di sola lettura e non esiste alcun accesso diretto ai file dei database più importanti.</p>	Nessuno. Le applicazioni ristabiliscono la connessione al queue manager.

Tabella 8. I metodi di protezione utilizzati per i processi più importanti (Continua)

Processo	Metodi di protezione utilizzati	Processo di ripristino
<p>Applicazioni – collegate a FASTPATH</p>	<p>I malfunzionamenti in MQSeries vengono rilevati da restituzioni di condizioni di errore impreviste da parte di chiamate MQI. L'applicazione deve richiamare MQDISC e quindi nuovamente MQCONNX per ristabilire la connessione con il queue manager.</p> <p>Il queue manager rileva il malfunzionamento del processo dell'applicazione immediatamente. Il queue manager esegue un MQDISC implicito per conto dell'applicazione determinando la chiusura ed il rilascio di tutte le risorse aperte.</p> <p>Le eventuali operazioni di syncpoint in corso vengono interrotte da TM/MP e MQSeries reagisce eseguendo le adeguate correzioni alle code locali.</p> <p>Il codice dell'applicazione che contiene gli errori può danneggiare i database più importanti del queue manager e le risorse condivise poiché le applicazioni dispongono di accesso alla Repository Cache di lettura/scrittura ed alle strutture interne del Queue Manager che hanno il potenziale di danneggiare l'Object Catalog. Nota: le strutture di coda ed i messaggi stessi sono al sicuro poiché solo i queue server vi accedono.</p>	<p>Nessuno. In fase di riavvio, le applicazioni stabiliscono nuovamente la connessione.</p>

Cluster

I cluster di MQSeries for Compaq NSK V5.1 servono a ridurre i requisiti di amministrazione di una rete MQSeries ed anche a migliorare disponibilità e scalabilità generali di MQSeries come un servizio distribuito.

I queue manager che appartengono ai cluster possono eseguire MQPUT nelle code note al cluster come se fossero code locali. La funzione di clustering di MQSeries

Cluster

svolge attività di amministrazione e di gestione di tutte le definizioni e di tutti i canali richiesti per trasferire il messaggio alla coda di destinazione.

Le code in cluster possono essere definite su più di un queue manager in un cluster. Questo crea più istanze di una coda nel cluster. Un'applicazione esegue inserimenti in una sola istanza di una coda come selezionato dal Cluster Workload Manager (CWLM), un componente di MQSeries. Questa scelta può essere fatta quando si apre la coda oppure dinamicamente per ogni operazione di inserimento. Il CWLM può determinare la migliore istanza di una coda in cluster da utilizzare sulla base di vari fattori, come ad esempio se il canale per l'istanza è aperto o meno e sulla base di alcuni altri fattori quali la priorità di rete e considerazioni applicative varie tramite la Cluster Workload Management Exit.

I cluster possono pertanto fornire un miglioramento della disponibilità a livello di rete di MQSeries. MQSeries su Compaq NSK è una buona scelta per svolgere mansioni di Full Repository per i cluster grazie all'affidabilità ed alla scalabilità delle sue operazioni.

Considerazioni sulla configurazione per la disponibilità

Questa sezione riepiloga le opzioni di configurazione che migliorano la disponibilità di MQSeries for Compaq NSK V5.1 e le sue applicazioni su Compaq NSK:

- Configurare PATHWAY con delle CPU alternative per tutte le classi server per una protezione dai malfunzionamenti delle CPU
- Considerare l'uso del clustering per una migliore disponibilità delle risorse di rete di MQSeries
- Considerare l'uso di messaggi non permanenti con la funzione di checkpoint abilitata per ottenere prestazioni migliori con elevate prestazioni per i tipi di messaggio adatti
- Assicurare che la configurazione di TM/MP sia sufficiente a gestire la domanda massima prevista di MQSeries e delle sue applicazioni
- Considerare l'uso dei file di overflow di messaggio per ridurre i requisiti di traccia di controllo per messaggi di notevoli dimensioni
- Se esiste un requisito di ripristino da gravi malfunzionamenti, considerare l'uso di RDF per la creazione e la conservazione di un sito di backup per MQSeries
- Controllare che le CPU siano in esecuzione con sufficiente memoria fisica per gestire la domanda massima di MQSeries e delle sue applicazioni
- Controllare di disporre di spazio di swap sufficiente per le CPU che contengono MQSeries e le sue applicazioni.

Considerazioni sulla configurazione per l'integrità dei dati

Questa sezione riassume le opzioni di configurazione che potenziano l'integrità dei dati di MQSeries for Compaq NSK V5.1 e le relative applicazioni in Compaq NSK:

- Determinare quali tipi di messaggio o transazione eseguiti da MQSeries richiedono un livello di integrità dei dati in base a quello fornito dai messaggi permanenti e non permanenti
- Determinare se i file di overflow messaggio costituiscono un meccanismo di memorizzazione compatibile con la memorizzazione di qualsiasi messaggio molto ampio che è necessario utilizzare

Considerazione sulla configurazione per l'integrità dei dati

- Determinare se i messaggi non permanenti richiedono l'esecuzione del checkpoint o se lo richiedono alcuni soltanto. I diversi tipi di messaggio devono essere collocati in code diverse per abilitare diverse opzioni di checkpoint da specificare
- Quando le applicazioni richiedono il livello massimo di integrità dei dati, accertarsi che le operazioni syncpoint utilizzino i messaggi permanenti
- Se si verifica una richiesta per ripristino d'emergenza, configurare ed utilizzare RDF per creare e mantenere una copia duplicata di backup dell'ambiente MQSeries.

Considerazione sulla configurazione per l'integrità dei dati

Parte 2. Riferimenti

Capitolo 17. I comandi di controllo di MQSeries	251
Riepilogo dei comandi di controllo	251
Utilizzo di nomi	251
Come leggere i diagrammi di sintassi	252
altmqfls (Modifica attributi del file di coda)	254
altmqusr (Modifica informazioni utente MQSeries)	259
cleanrdf (Eeguire la pulizia RDF)	261
cnvclchl (Converti le definizioni di canale client)	263
crtmqcvx (Conversione dei dati)	265
crtmqm (Crea Queue Manager)	267
dltmqm (Delete queue manager)	271
dspmqaout (Display Authority)	273
dspmqcsv (Display Command Server)	277
dspmqlfs (Visualizza gli attributi file MQSeries)	278
dspmqrtrc (Visualizza output di traccia formattato di MQSeries)	282
dspmquusr (visualizza informazioni utente MQSeries)	283
endmqcvs (End command server)	285
endmqm (End queue manager)	288
endmqtrc (End MQSeries trace)	291
instmqm (Installa MQSeries for Compaq NSK)	292
runmqchi (Run channel initiator)	293
runmqchl (Run channel)	294
runmqdlq (Run dead-letter queue handler)	295
runmqslr (Run listener)	296
runmqsc (Esegue comandi di MQSeries)	298
runmqtrm (Avvia un trigger monitor)	301
setmqaut (Imposta/reimposta l'autorizzazione)	302
strmqcvs (Avvia server di comandi)	309
strmqm (Avvio Queue Manager)	310
strmqtrc (Avvio traccia di MQSeries)	311
upgmqm (Aggiorna il gestore code V2.2.0.1)	314

Capitolo 17. I comandi di controllo di MQSeries

Questo capitolo contiene materiale di riferimento per i comandi di controllo utilizzati con MQSeries for Compaq NSK.

Riepilogo dei comandi di controllo

I seguenti comandi di controllo sono supportati da MQSeries for Compaq NSK tramite macro TAQL e programmi compilati:

- `altnqfls` (modifica attributi file coda)
- `altnqusr` (modifica informazioni utente MQSeries)
- `cleanrdf` (programma di utilità di pulizia RDF)
- `cnvclchl` (converte definizioni di canale client)
- `crtmqcvx` (conversione dati)
- `crtmqm` (crea queue manager)
- `dltmqm` (elimina il queue manager)
- `dspmqaut` (visualizza l'autorizzazione)
- `dspmqcsv` (visualizza command server)
- `dspmqfls` (visualizza gli attributi del file di MQSeries)
- `dspmqtrc` (visualizza output di traccia formattato MQSeries)
- `dspmqusr` (visualizza informazioni utente MQSeries)
- `endmqcsv` (termina command server)
- `endmqm` (termina queue manager)
- `endmqtrc` (termina traccia MQSeries)
- `instmqm` (installa MQSeries for Compaq NSK)
- `runmqchi` (esegue channel initiator)
- `runmqchl` (esegue channel)
- `runmqdlq` (esegue dead-letter queue handler)
- `runmqslr` (esegue TCP/IP listener)
- `runmqsc` (esegue comandi MQSeries)
- `runmqtrm` (avvia trigger monitor)
- `setmqaut` (imposta/reimposta l'autorizzazione)
- `strmqcsv` (avvia command server)
- `strmqm` (avvia queue manager)
- `strmqtrc` (avvia traccia MQSeries)
- `upgmqm` (aggiorna V2.2.0.1 queue manager)

Le descrizioni dettagliate di questi comandi sono fornite nel resto di questo capitolo.

Note:

1. I flag, che sono identificatori di singolo carattere preceduti da un trattino (ad esempio, `-v` sul comando `runmqsc`), devono essere specificati in carattere minuscolo.
2. L'utilizzo dei messaggi viene visualizzato se i comandi di controllo sono richiamati con `are -?`, `?` o con nessun parametro quando questo è in arrivo.

Utilizzo di nomi

I nomi dei seguenti oggetti MQSeries possono essere costituiti da un massimo di 48 caratteri:

- Queue manager
- Code

Nomi

- Definizioni di processo

La lunghezza massima dei nomi di canale è di 20 caratteri.

I caratteri che possono essere utilizzati per tutti i nomi di MQSeries sono:

- Maiuscolo A - Z
- Minuscolo a - z
- Numerici 0 - 9
- Punto (.)
- Sottolineatura (_)
- Barra (/)
- Simbolo di percentuale (%)

Note:

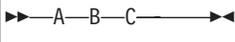
1. La barra e il simbolo di percentuale sono caratteri speciali. Se si utilizza uno di questi caratteri in un nome, questo dovrà essere racchiuso fra virgolette ogni qual volta viene utilizzato.
2. Non sono consentiti spazi vuoti iniziali o integrati.
3. Non sono consentiti caratteri di lingua nazionale.
4. È possibile racchiudere i nomi fra virgolette, ma ciò è fondamentale solo qualora nel nome siano inclusi caratteri speciali.

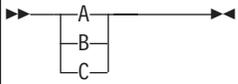
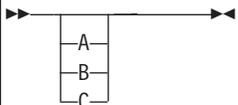
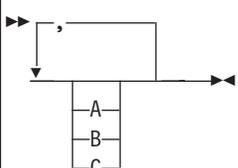
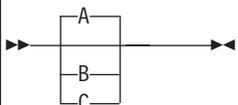
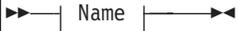
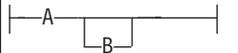
Come leggere i diagrammi di sintassi

Questo capitolo contiene diagrammi di sintassi (talvolta denominati diagrammi "railroad").

Tutti i diagrammi di sintassi iniziano con una doppia freccia a destra e terminano con una coppia di frecce a destra e sinistra. Le righe che iniziano con un'unica freccia a destra sono righe di continuazione. Il diagramma di sintassi va letto da sinistra a destra e dall'alto verso il basso, seguendo la direzione delle frecce.

Altre convenzioni utilizzate nei diagrammi di sintassi sono:

Convenzione	Significato
	Occorre specificare i valori A, B e C. I valori richiesti sono mostrati sulla riga principale di un diagramma di sintassi.
	È possibile specificare il valore A. I valori facoltativi sono illustrati sotto la riga principale di un diagramma di sintassi.

Convenzione	Significato
	<p>I valori A, B e C sono alternativi; occorre specificarne almeno uno.</p>
	<p>I valori A, B e C sono alternativi; è possibile specificarne almeno uno.</p>
	<p>È possibile specificare un o più valori A, B e C. Qualsiasi separatore necessario per valori multipli o ripetuti (in questo esempio, la virgola (,)) è mostrato sulla freccia.</p>
	<p>I valori A, B e C sono alternativi; è possibile specificarne almeno uno. Se non si specifica nessuno dei valori mostrati, verrà utilizzato il valore predefinito A (il valore mostrato sopra la riga principale).</p>
<p data-bbox="461 1157 699 1188">  </p> <p data-bbox="461 1234 548 1266">Name:</p> <p data-bbox="461 1297 699 1350">  </p>	<p>Il frammento di sintassi Name è mostrato separatamente dal diagramma di sintassi principale.</p>

altnmqfls (Modifica attributi del file di coda)

Scopo

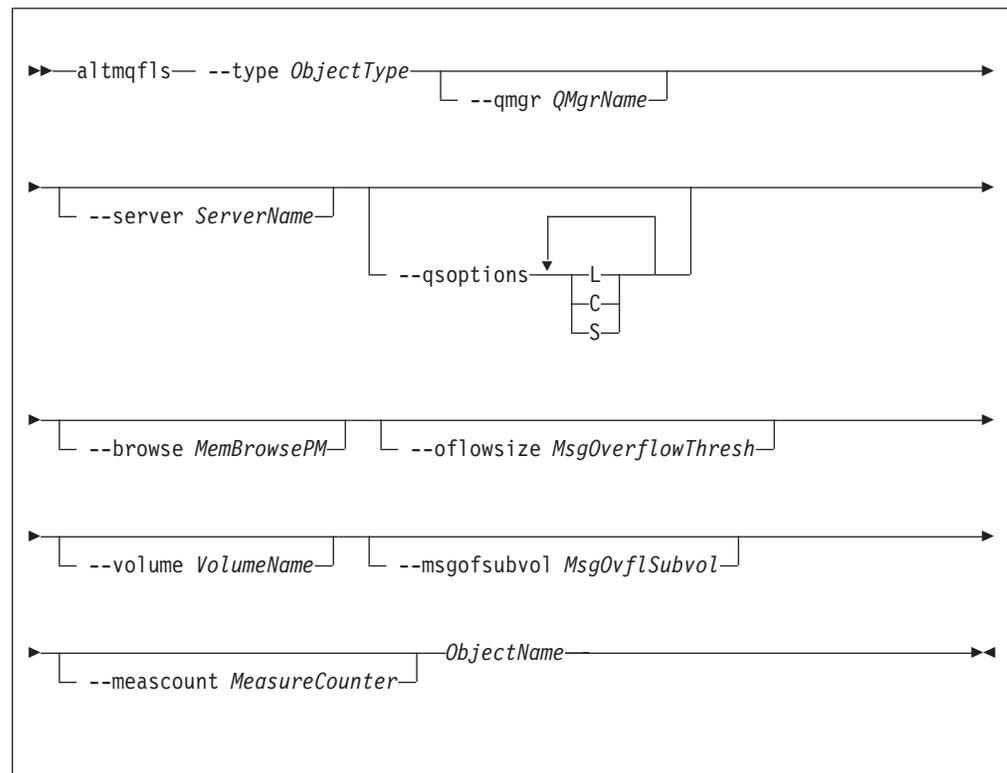
Il comando **altnmqfls** consente di modificare gli attributi del file di coda. Un singolo comando **altnmqfls** può eseguire solo uno dei seguenti tre gruppi di operazioni per volta:

- Spostare i file di messaggio appartenenti ad una coda locale predefinita in un diverso volume per distribuire I'/E di disco tra i vari volumi oppure
- Modificare la dimensione dei file di coda e di overflow associati ad una coda locale. Quest'operazione non può essere eseguita se la coda è aperta oppure
- Modificare le opzioni di queue server associate ad una coda locale. Queste opzioni consentono di ottimizzare il modo in cui il queue server gestisce la memoria associata alla coda e controlla l'esecuzione del checkpoint dei messaggi non permanenti. E' inoltre possibile modificare il queue server associato ad una coda locale oppure il status server associato ad un alias, coda remota o modello, oppure elaborare ed associare un calcoladimensioni ad una coda.

altnmqfls non consente il ricaricamento della coda mentre essa è in uso.

Nota: L'interfaccia utente e le opzioni di riga comandi per **altnmqfls** sono molto cambiate rispetto alla Versione 2.2.0.1. Queste modifiche riflettono le nuove funzioni aggiunte e forniscono dei nomi più significativi per le opzioni di riga comandi. Inoltre, tutti i parametri iniziano con '--' per conformità alle regole POSIX. Occorrerà aggiornare tutti gli script che richiamano **altnmqfls** per riflettere queste modifiche poiché MQSeries Versione 5.1 non accetterà le opzioni di riga comandi della Versione 2.2.0.1.

Sintassi



Parametri richiesti

ObjectName

é il nome della coda locale permanente per cui bisogna spostare i file di messaggi. La coda non deve essere aperta nè deve contenere messaggi non sincronizzati.

--type *ObjectType*

Identifica una coda permanente. *ObjectType* deve essere specificato e può essere uno dei seguenti:

ql o qlocal	Una coda locale
qa o qalias	Un alias di coda
qr o qremote	Una coda remota
qm o qmodel	Un modello di coda
proc o process	Un processo

Parametri facoltativi

--qmgr *QMgrName*

É il nome del queue manager cui appartiene la coda locale. É necessario che il queue manager sia stato avviato. Se non viene specificato alcun nome il queue manager, viene utilizzato il queue manager predefinito.

--server *ServerName*

É il nome del processo di status server o del processo di queue server che deve essere responsabile per i dati di stato per quest'oggetto. Se l'oggetto è una coda locale oppure una coda modello, utilizzare un nome di queue server. Se l'oggetto è diverso da una coda locale o da una coda modello, utilizzare il nome di status server.

altnmqfls

Quando un oggetto viene creato, il *ServerName* viene impostato su DEFAULT per indicare che il server predefinito è responsabile per la coda. Quando la responsabilità per una coda viene modificata, la coda non deve essere in uso e tutti i messaggi non permanenti vengono eliminati durante la modifica.

--volume *VolumeName*

E' un nome di volume Compaq NSK (ad esempio, \$DEV). Questo valore è richiesto se si sta utilizzando **altnmqfls** per spostare dei file di messaggio in un diverso volume. Questo parametro può essere specificato solo con il tipo ed il nome di oggetto. Non ne è consentito l'utilizzo in combinazione con le altre opzioni.

--qsoptions LSC

Utilizzare le opzioni L, S e C, da sole o in combinazione, per ottimizzare l'affidabilità e l'accesso della coda.

Nota: *Tutte* le --qsoptions SLC vengono impostate ogni volta che viene emesso il comando. Ad esempio --qsoptions S annullerà l'impostazione di L e C. L'opzione --qsoptions può essere specificata solo una volta su una riga comandi. Bisogna specificare almeno un'opzione, ma è possibile specificarne più di una.

C Specifica che viene eseguito il checkpoint dei messaggi non permanenti sul queue server di backup per prevenire problemi causati da malfunzionamento, anche se il carico di lavoro della CPU viene così aumentato per l'esecuzione delle operazioni necessarie per gestire il checkpoint supplementare, i messaggi IPC supplementari e la memoria supplementare richiesta per memorizzare i messaggi.

Utilizzare questa opzione se si desidera disporre di un elevato grado di recuperabilità per i messaggi non permanenti. Per impostazione predefinita, quando viene creata una coda, questo parametro deve essere impostato (checkpoint abilitato).

L Specifica che il queue server blocca in memoria le strutture ed i concatenamenti di dati associati ad una coda. Di norma, la memoria associata ad una coda può essere rimossa dallo spazio di indirizzo del queue server quando non si accede più ad essa. Utilizzare questa opzione per accedere in modo più rapido alle strutture di dati della memoria della coda ad eventuale discapito delle altre code.

Il valore predefinito per questo parametro quando viene creata una coda non è impostato (non è cioè bloccato in memoria).

S Specifica che il queue server carica la coda locale dal disco nella cache quando viene avviato il queue manager. Di norma, i messaggi per una coda vengono caricati quando un'applicazione fa per la prima volta riferimento ad essi. Se questa opzione è impostata, la coda viene caricata quando viene avviato il queue manager. Il valore predefinito per questo parametro quando viene creata una coda non è impostato (non è cioè caricato in fase di avvio).

Utilizzare questa opzione per ridurre l'attività della CPU e di I/E di disco quando una coda viene aperta per la prima volta, anche se questo comporta un aumento dell'attività del queue server (CPU e I/E di disco) in fase di avvio del queue manager.

None Specifica che non viene impostata alcuna opzione.

--browse *MemBrowsePM*

Specifica un numero massimo di byte di dati di ciascun messaggio permanente

da conservare nella cache del queue server (e su disco). Durante un'operazione di browse su un messaggio permanente, il queue manager di norma legge i dati per un messaggio dalla memoria su disco e li restituisce all'applicazione. Se questo parametro viene impostato su un valore diverso da zero, il numero specificato di byte di dati verrà anch'esso conservato in memoria e l'operazione di browse restituirà questi dati all'applicazione senza dover accedere al disco. Utilizzando questo parametro sarà possibile aumentare le risorse di memorie utilizzate dal queue server. Il valore minimo di questo parametro è pari a zero (0) byte, il valore massimo è 25,000 byte. Il valore predefinito per questo parametro quando viene creata una coda è zero.

--oflowsize *MsgOverflowThresh*

Specifica la dimensione di messaggio minima per l'utilizzo del file di overflow di messaggio per la memorizzazione dei dati di messaggio. I messaggi permanenti di dimensioni inferiori al limite qui fissato vengono memorizzati nel file di overflow di coda. I messaggi permanenti con una dimensione pari o superiore al limite qui fissato, verranno parzialmente memorizzati in un file di overflow di messaggi dedicato. Il valore predefinito per questo parametro quando viene creata una coda è 200,000 byte.

--msgofsubvol *MsgOvflSubvol*

Specifica un volume secondario nel volume dove si trova la coda dove il queue server crea dei nuovi file di overflow di messaggio. Tutte le code utilizzeranno inizialmente, per impostazione predefinita, il proprio volume secondario per i messaggi del queue manager.

--meascount *MeasureCounter*

Specifica il nome di un MEASURE Counter (calcoladimensioni) che, se parte di una misurazione attiva, viene avviato a partire dalla profondità di coda corrente e quindi aumentato o ridotto dal responsabile del queue server per la coda quando vengono aggiunti o rimossi messaggi.

Codici di ritorno

- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

Esempi

1. Nel seguente esempio, i file di messaggio appartenenti alla coda locale `flint.queue`, di proprietà del queue manager `target.queue.mgr`, vengono spostati nel volume `$DATA3`.

```
altnmqfls --qmgr target.queue.mgr --type ql --volume $DATA3 flint.queue
```

2. Nel seguente esempio, il nome di processo di queue server viene modificato per un oggetto:

```
altnmqfls --qmgr target.queue.mgr --type ql --server $TQS2 flint.queue
```

Questo comando determina il trasferimento della coda locale `flint.queue`, di proprietà di `target.queue.mgr`, in un queue server con un nome di processo di `$TQS2`.

altnqfls

Comandi correlati

dspmqfls

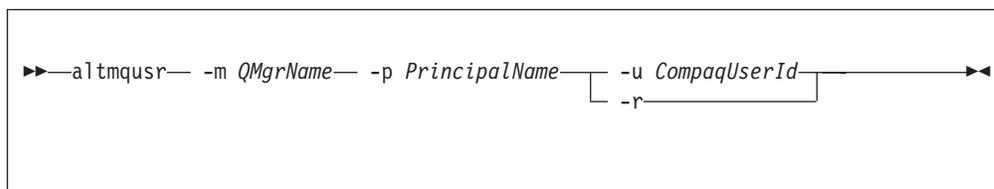
Visualizza file MQSeries

altmqusr (Modifica informazioni utente MQSeries)

Scopo

Utilizzare il comando **altmqusr** per definire o rimuovere un principal corrispondente ad un ID utente Compaq NSK che disporrà dell'accesso a MQSeries.

Sintassi



Descrizione

E' possibile utilizzare questo comando per:

- Creare un principal (per concedere cioè l'accesso ad un queue manager ad un ID utente Compaq NSK).
- Rimuovere un principal (revocare l'accesso ad un queue manager ad un ID utente Compaq).
- Modificare una definizione di principal.

Quando viene specificata *-u CompaqUserId*, **altmqusr** crea un principal se non esiste già uno oppure modifica la definizione esistenti. *CompaqUserId* può essere specificato come un ID utente di amministrazione Compaq oppure, se SAFEGUARD è in esecuzione, come un alias di SAFEGUARD.

Quando viene specificato *-r*, il principal viene cancellato dal database dei principal.

Bisogna specificare *-u CompaqUserId* oppure *-r*.

Parametri richiesti

-m *QMgrName*

E' il nome del queue manager cui appartiene il principal.

-p *PrincipalName*

E' il nome del principal da creare, modificare o rimuovere.

-u *CompaqUserId*

E' l'ID utente di amministrazione Compaq NSK oppure l'alias di SAFEGUARD da associare alla definizione di principal.

-r Specifica che la definizione di principal deve essere rimossa dal queue manager.

Codici di ritorno

- 0 operazione eseguita
- 36 Elemento immesso non valido
- 69 Memoria non disponibile
- 71 Errore imprevisto

altnqusr

Esempi

Per aggiungere un principal mquser1 in corrispondenza con un ID utente Compaq mqtest.fred:

```
altnqusr -m MT02 -p mquser1 -u mqtest.fred
```

Per aggiungere un principal mquser2 in corrispondenza con il gruppo group.user01:

```
altnqusr -m MT02 -p mquser2 -u group.user01
```

Per visualizzare i risultati di questi comandi, utilizzare il comando **dspmqusr**, come descritto in “dspmqusr (visualizza informazioni utente MQSeries)” a pagina 283.

Per rimuovere il principal mquser1:

```
altnqusr -m MT02 -p mquser1 -r
```

Comandi correlati

dspmqusr Visualizza le informazioni utente MQSeries

cleanrdf (Eeguire la pulizia RDF)

Scopo

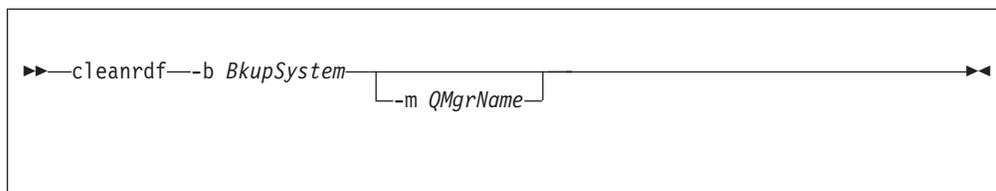
Utilizzare il programma di utilità **cleanrdf** per eseguire la pulizia di routine sul queue manager del sistema primario in ambiente RDF. Il programma di utilità **cleanrdf** completa la rimozione dei file che sono stati eliminati sui sistemi primario e di backup. Il programma di utilità duplica anche alcuni database non controllati sull'ubicazione corretta del sistema di backup.

Notare che il programma di utilità richiamato da **cleanrdf** sposta l'intero database di messaggio e catalogo di oggetto, in modo tale che alcune degradazioni di prestazioni sono simili a quelle che si verificano mentre è in esecuzione il programma di utilità.

I file del database non controllato duplicati sul sistema di backup dal programma di utilità sono:

- MQERRLG1
- MQSINI
- QMINI
- PATHCTL
- SHUTDOWN
- SYNCHIDX
- TRACEOPT
- UMQSINI

Sintassi



Parametri richiesti

-b *BkupSystem*

E' il nome del sistema Compaq NSK del sito di backup RDF per questo queue manager. *BkupSystem* viene specificato nel modulo `\name` (come è standard in ambiente Compaq NSK).

Parametri facoltativi

-m *QMgrName*

E' il nome del queue manager per cui **cleanrdf** deve essere eseguito. Se nessun nome del queue manager viene specificato, **cleanrdf** viene eseguito sul queue manager predefinito.

Codici di ritorno

- 0 Comando completato normalmente
- 20 Si è verificato un errore durante l'elaborazione

cleanrdf

Esempi

Nel seguente esempio, **cleanrdf** viene eseguito sul queue manager `test.queue.mgr`. Compaq NSK node \HAWK è stato configurato come sito RDF di backup per questo queue manager.

```
cleanrdf -b \HAWK -m test.queue.mgr
```

cnvclchl (Converti le definizioni di canale client)

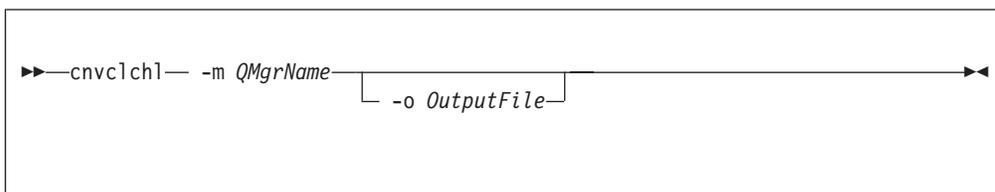
Scopo

Utilizzare il comando **cnvclchl** per convertire il file di definizione canale del client, creato per i canali CLNTCONN da MQSC, da un file Compaq strutturato in un formato non strutturato accettabile per i client MQSeries.

Nota:

I file di definizione Versione 2.2.0.1 non funzionano con i client Versione 5.1 poiché il formato dei record è stato modificato. Dopo aver installato MQSeries for Compaq NSK V5.1, è possibile eseguire nuovamente il comando **cnvclchl** per creare i file di definizione che funzionano con i client Versione 5.1.

Sintassi



Parametri richiesti

-m *QMgrName*
 Identifica il queue manager proprietario del file di definizioni canale (CCHDEFS) da convertire. Questo valore è richiesto.

Parametri facoltativi

-o *OutputFile*
 Identifica il file che conterrà le definizioni convertite. Il nome file predefinito è AMQCLCHL.

Esempi

Il seguente comando converte il file per la definizione del canale del client strutturato Compaq per queue manager MT01 in un file non strutturato. Due definizioni di canali di connessione del client sono contenuti nel file di emissione AMQCLCHL, SYSTEM.DEF.CLNTCONN e SOLARIS_TO_Cmpaq:

cnvclchl

```
$DATA01 SZMON 330> cnvclchl -m MV4
```

Conversione della tabella del canale del client MQSeries in corso

Apertura della tabella Compaq NSK v5.1 CLNTCONN in corso

Apertura della tabella Common v5.1 CLNTCONN AMQCLCHL per l'emissione in corso

Scrittura della voce della tabella Common v5.1 CLNTCONN per SYSTEM.DEF.CLNTCONN in corso

Chiusura della tabella Compaq NSK v5.1 CLNTCONN in corso

Chiusura della tabella Common v5.1 CLNTCONN in corso

Conversione della tabella del canale del client MQSeries completata

crtmqcvx (Conversione dei dati)

Scopo

Utilizzare il comando **crtmqcvx** per creare un frammento di codice che converte dati nelle strutture del tipo di dati. Il comando genera una funzione C che può essere utilizzato in un uscita per convertire le strutture C.

Il comando rileva un file di input che contiene una o più strutture da convertire. Quindi realizza un file di output che contiene un o più frammenti di codice per convertire queste strutture.

Per ulteriori informazione relative a questo comando e al suo utilizzo, consultare *MQSeries Application Programming Guide*.

Sintassi

```
▶▶—crtmqcvx—SourceFile—TargetFile—▶▶
```

Parametri richiesti

SourceFile

Specifica il file di input che contiene le strutture C da convertire.

TargetFile

Specifica il file di output che contiene i frammenti di codice creati per convertire le strutture.

Codici di ritorno

- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

Esempi

L'esempio che segue mostra l'effetto dell'utilizzo del comando di conversione dei dati in una struttura di origine C. Il comando inviato è:

```
crtmqcvx source target
```

Il file di input, source si presenta in modo simile al seguente:

crtmqcvx

```
/* This is a test C structure which can be converted by the */  
/* crtmqcvx utility */  
  
struct my_structure  
{  
    int    code;  
    MQLONG value;  
};
```

Il file di output, target, prodotto dal comando, viene qui di seguito mostrato. È possibile utilizzare questi frammenti di codice nell'applicazione per convertire la struttura dei dati. Tuttavia, procedendo in questo modo, si noterà che il frammento utilizza delle macro fornite nel file di intestazione MQSeries MQSVMHTH nel volume secondario ZMQSLIB.

```
MQLONG Convertmy_structure(  
    PMQBYTE *in_cursor,  
    PMQBYTE *out_cursor,  
    PMQBYTE in_lastbyte,  
    PMQBYTE out_lastbyte,  
    MQHCONN hConn,  
    MQLONG  opts,  
    MQLONG  MsgEncoding,  
    MQLONG  ReqEncoding,  
    MQLONG  MsgCCSID,  
    MQLONG  ReqCCSID,  
    MQLONG  CompCode,  
    MQLONG  Reason)  
{  
    MQLONG ReturnCode = MQRC_NONE;  
  
    ConvertLong(1); /* code */  
  
    AlignLong();  
    ConvertLong(1); /* value */  
  
Fail:  
    return(ReturnCode);  
}
```

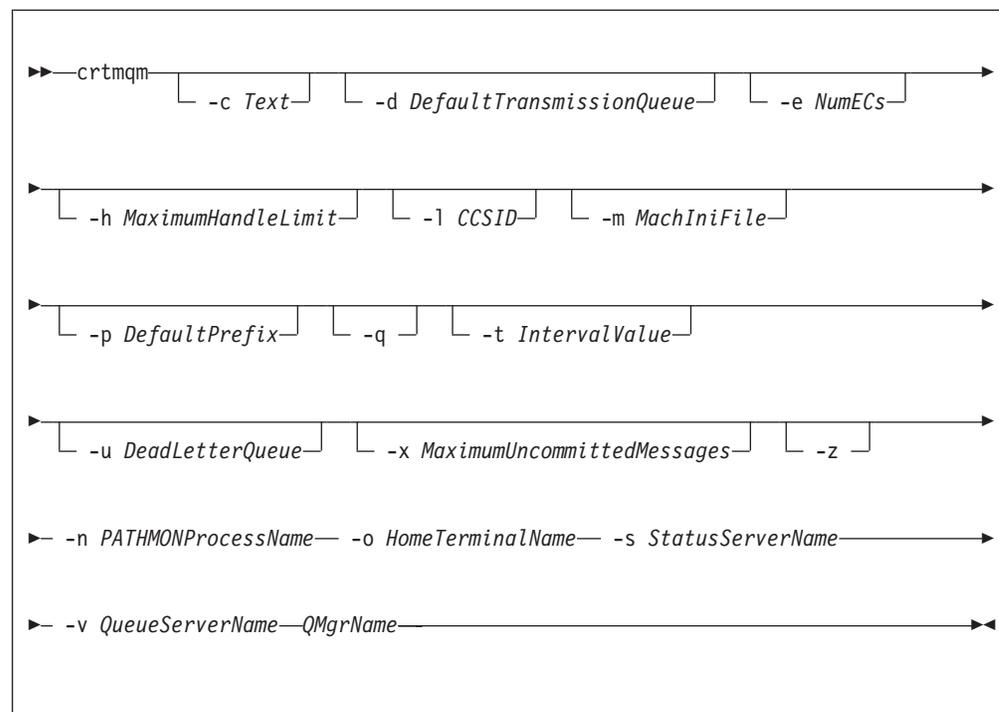
crtmqm (Crea Queue Manager)

Scopo

Utilizzo del comando **crtmqm** per creare un Queue Manager locale. Una volta creato un Queue Manager, utilizzare il comando **strmqm** per avviarlo.

La creazione automatica di un queue manager crea il sistema associato e gli oggetti predefiniti.

Sintassi



Parametri richiesti

- n** *PATHMONProcessName*
Il nome di processo del processo TS/MP PATHMON per queue manager. Tale nome processo deve essere univoco nel sistema.
- o** *HomeTerminalName*
Nome unità del terminale iniziale. (\$DDDD.#SS). Ad esempio, \$TRM1.#A.
- s** *StatusServerName*
Un nome di processo per il server di stato predefinito per queue manager. Il nome processo deve essere univoco nel sistema.
- v** *QueueServerName*
Specifica un nome processo univoco da assegnare al queue server predefinito per questo queue manager.

QMgrName

Il nome del queue manager da creare. Il nome può contenere fino a 48 caratteri. Questo deve essere l'ultimo parametro del comando.

Parametri facoltativi

-c *Text*

Parte del testo (fino a 64 caratteri) che descrive il queue manager. Per impostazione predefinita il campo è vuoto.

Se si desidera utilizzare alcuni caratteri speciali è necessario includere il testo tra virgolette.

-d *DefaultTransmissionQueue*

Il nome della coda di trasmissione locale su cui vengono posizionati i messaggi remoti, quando non viene loro specificamente assegnata come destinazione una coda di trasmissione. Non esistono valori predefiniti.

-e *NumECs*

Il numero di processi EC processes nel queue manager. Il valore predefinito è 1.

-h *MaximumHandleLimit*

In MQSeries for Compaq NSK, tale parametro è ignorato.

Il numero massimo di handle che ogni applicazione può avere allo stesso tempo. Specificare un valore compreso tra 1 e 999 999 999. Il valore predefinito è 256.

-l *CCSID*

Qmgr CCSID. Il valore predefinito è 819.

-m *MachIniFile*

Supera la ubicazione predefinita del file MQSINI specificata nella variabile di ambiente MQMACHINIFILE.

-p *DefaultPrefix*

Il volume per queue manager. Modifica la voce QMDefaultVolume nel file MQSINI.

-q

Specifica che questo Queue Manager viene reso predefinito. Il nuovo Queue Manager che ne sostituisce uno esistente diviene quello predefinito.

Se questo indicatore viene utilizzato accidentalmente e si desidera ripristinare come predefinito il precedente Queue Manager, è necessario digitare la voce DefaultQueueManager nel file di configurazione MQSeries.

-t *IntervalValue*

L'intervallo di tempo di trigger in millisecondi delle code controllate da questo Queue Manager. Questo valore specifica il tempo dopo la ricezione di un messaggio che genera un trigger-generating quando il triggering è sospeso. Di conseguenza, se la ricezione di un messaggio fa sì che la coda invii un messaggio di trigger alla coda di avvio, tutti i messaggi in arrivo sulla stessa coda all'interno dell'intervallo di tempo specificato non provocano un altro messaggio di trigger.

È possibile utilizzare l'intervallo di tempo di trigger per assicurare all'applicazione un sufficiente spazio di tempo per gestire una situazione relativa ad un trigger prima dell'avviso della necessità di gestirne un'altra sulla stessa coda. Potrebbe essere necessario visualizzare tutti gli eventi trigger che si verificano; a questo scopo impostare un valore pari o inferiore a zero in questo campo.

Specificare un valore tra 0 e 999 999 999. Il valore predefinito è 999 999 999 millisecondi, un tempo superiore a 11 giorni. Impostando il valore predefinito il triggering verrà disabilitato dopo la ricezione del primo messaggio di trigger.

Tuttavia potrà essere riabilitato grazie ad un applicazione servente la coda utilizzando un comando di modifica della coda che reimposta l'attributo trigger.

-u *DeadLetterQueue*

Il nome della coda locale che deve essere utilizzata come coda messaggi non recapitati. I messaggi vengono inviati a questa coda se non riescono a raggiungere la loro corretta destinazione.

Per impostazione predefinita, non esiste una coda di messaggi non recapitati.

-x *MaximumUncommittedMessages*

In MQSeries for Compaq NSK, tale parametro è ignorato.

Specifica il numero massimo di messaggi non assegnati relativi ad ogni syncpoint, cioè, la somma di:

- Il numero di messaggi che possono essere richiamati dalla coda.
- Il numero di messaggi che possono essere inviati alle code.
- Tutti i messaggi di trigger generati all'interno di quest'unità di elaborazione.

Questa limitazione non si applica ai messaggi richiamati o espulsi da un controllo syncpoint.

Specifica un valore tra 1 e 10 000. Il valore predefinito è 1000.

-z Sopprime i messaggi di errore.

Questo indicatore viene utilizzato solitamente in MQSeries per eliminare i messaggi non desiderati. Poiché l'utilizzo di questo indicatore comporta la perdita di informazioni, è preferibile non usarlo quando si immettono comandi.

Codici di ritorno

0	Queue Manager creato
8	Queue Manager già esistente
49	Queue Manager in chiusura
69	Memoria non disponibile
70	Spazio coda non disponibile
71	Errore imprevisto
72	errore relativo al nome del Queue Manager
111	Queue Manager creato. Tuttavia, si è verificato un problema durante l'elaborazione del Queue Manager predefinito nel file di configurazione del prodotto. La specifica del Queue Manager predefinito potrebbe essere errata.

Esempi

1. Questo comando crea un Queue Manager predefinito denominato `Paint.queue.manager`, a cui viene data la descrizione di `Paint Shop`:

```
crtmqm -c "Paint Shop" -n $PANT -o $TRM1.#A -s $PNT1 -v $PQS1 Paint.queue.manager
```

2. In questo esempio, viene creato un altro Queue Manager, `travel`. L'intervallo di trigger è definito in 5000 millisecondi (5 secondi) ed è specificata `SYSTEM.DEAD.LETTER.QUEUE`. come coda messaggi non recapitati,

crtmqm

```
crtmqm -t 5000 -u SYSTEM.DEAD.LETTER.QUEUE -n $TRAV -o $TRM1.#A -s $TRV1 -v $TQS1 travel
```

Una volta verificatosi un evento trigger, i successivi eventi vengono disabilitati per 5 secondi.

Comandi correlati

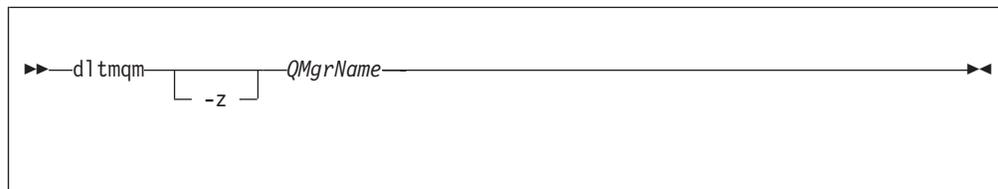
strmqm	Avvia un Queue Manager
endmqm	Arresta un Queue Manager
dltmqm	Elimina Queue Manager

dltmqm (Delete queue manager)

Scopo

Utilizzare il comando **dltmqm** eliminare il Queue Manager specificato. Con questo comando si eliminano anche tutti gli oggetti associati a questo Queue Manager. Per poter eliminare un Queue Manager è necessario prima chiuderlo con il comando **endmqm**.

Sintassi



Parametri richiesti

QMgrName

Specifica il nome del Queue Manager da eliminare.

Parametri facoltativi

-z Sopprime i messaggi di errore.

Codici di ritorno

0	Queue Manager eliminato
5	Queue Manager in esecuzione
16	Queue Manager inesistente
69	Memoria non disponibile
71	Errore imprevisto
72	errore relativo al nome del Queue Manager
112	Queue Manager eliminato. Tuttavia, si è verificato un problema durante l'elaborazione del Queue Manager predefinito nel file di configurazione del prodotto. La specifica del Queue Manager predefinito potrebbe essere errata.

Esempi

1. Il seguente comando cancella il queue manager saturn.queue.manager:

```
dltmqm saturn.queue.manager
```

2. Il seguente comando cancella il queue manager travel e sopprime inoltre gli eventuali messaggi causati dal comando:

```
dltmqm -z travel
```

dltmqm

Comandi correlati

crtmqm	Crea un Queue Manager
strmqm	Avvia un Queue Manager
endmqm	Arresta un Queue Manager

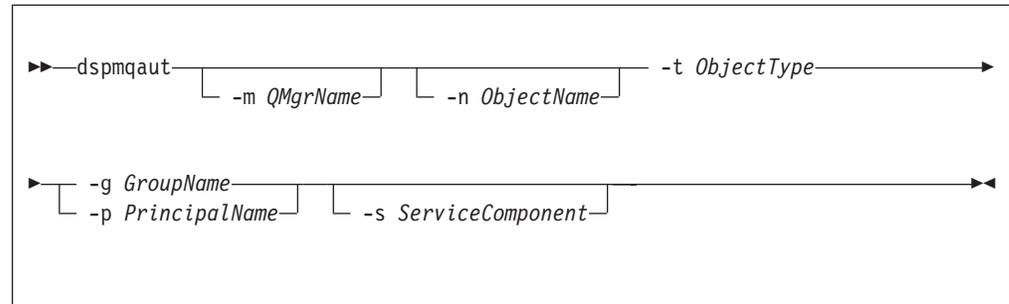
dspmqaout (Display Authority)

Scopo

Utilizzare il comando **dspmqaout** per visualizzare le correnti autorizzazioni relative ad un oggetto specificato. Può essere specificato un solo gruppo.

Se un ID utente è membro di più di un gruppo, controllare le autorizzazioni di ciascun gruppo per determinare tutte le autorizzazioni valide per l'ID utente.

Sintassi



Parametri richiesti

-g *GroupName*

Specifica il nome o il gruppo utente sul quale eseguire la richiesta. È possibile specificare solo *un* nome, che deve essere il nome di un gruppo utente esistente. Bisogna specificare -g *GroupName* oppure -p *PrincipalName*.

-p *PrincipalName*

Specifica il nome del principal per cui bisogna visualizzare le autorizzazioni ad uno specifico oggetto. Bisogna specificare -g *GroupName* oppure -p *PrincipalName*.

-t *ObjectType*

Specifica il tipo di oggetto sul quale deve essere fatta la richiesta. I valori possibili sono:

queue o q	Una o più code che corrispondano al parametro del tipo di oggetto.
qmgr	Un oggetto Queue Manager
process o prcs	Un processo
namelist o nl	Un namelist

Parametri facoltativi

-m *QMgrName*

Specifica il nome del Queue Manager sul quale eseguire la richiesta.

-n *ObjectName*

Specifica il nome dell'oggetto sul quale deve essere fatta la richiesta.

Questo parametro è necessario *a meno che* non sia lo stesso Queue Manager.

Occorre specificare il nome del Queue Manager, della coda o della definizione di processo.

dspmqaout

-s *ServiceComponent*

Questo parametro si applica solo se si utilizzano servizi installabili di autorizzazione, altrimenti viene ignorato.

Se tali servizi sono supportati, questo parametro specifica il nome dell'Authorization Service a cui si applicano le autorizzazioni. Questo parametro è facoltativo; se non viene specificato, l'aggiornamento dell'autorizzazione viene eseguito sul primo componente installabile per il servizio.

Parametri restituiti

Questo comando riporta un elenco di autorizzazione, che può contenere nessuno, uno o più parametri di autorizzazione. Ciascun parametro riportato indica che un ID utente nel gruppo specificato ha il potere di eseguire l'operazione indicata da quel parametro.

Tabella 9 mostra le autorizzazioni che possono essere date a ciascun tipo di oggetto.

Tabella 9. Autorizzazioni di sicurezza del comando dspmqaout

Autorizzazione	Coda	Processo	Qmgr	Namelist
tutto	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
chgaut	✓	✓	✓	
clr	✓			
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

L'elenco che segue riporta le autorizzazioni associati ai rispettivi parametri:

- tutto** Utilizza tutte le operazioni relative all'oggetto.
- alladm** Esegue tutte le operazioni di gestione relative all'oggetto.
- allmqi** Utilizza tutte le chiamate MQI relative all'oggetto.
- altusr** Specifica un ID utente alternativo su una chiamata MQI.

sfoglia	Richiama un messaggio da una coda inviando una chiamata MQGET con l'opzione BROWSE.
chg	Modifica gli attributi dell'oggetto specificato, utilizzando l'appropriato gruppo di comandi.
chgaut	Specificare le autorizzazioni per altri gruppi di utenti sull'oggetto, utilizzando il comando setmqaut .
clr	Ripulire una coda (comando PCF: Clear queue only).
connect	Collega l'applicazione al Queue Manager specificato inviando una chiamata MQCONN.
crt	Crea oggetti del tipo specificato, utilizzando l'appropriato gruppo di comandi.
dlt	Elimina l'oggetto specificato, utilizzando l'appropriato gruppo di comandi.
dsp	Visualizza gli attributi dell'oggetto specificato, utilizzando l'appropriato gruppo di comandi.
get	Richiama un messaggio da una coda inviando una chiamata MQGET.
inq	Effettua una richiesta su uno specifico Queue Manager inviando una chiamata MQINQ.
passall	Passa tutto il contesto.
passid	Passa il contesto identità.
put	Invia un messaggio su una specifica coda inviando una chiamata MQPUT.
set	Imposta attributi su una coda dal MQI inviando una chiamata MQSET.
setall	Imposta tutto il contesto su una coda.
setid	Imposta il contesto identità su una coda.

Le autorizzazioni per le operazioni di gestione si applicano, se supportate, a questi gruppi di comandi:

- Comandi di controllo
- comandi MQSC
- comandi PCF

Codici di ritorno

0	operazione eseguita
36	Elemento immesso non valido
40	Gestore code non disponibile
49	Queue Manager in chiusura
69	Memoria non disponibile
71	Errore imprevisto
72	errore relativo al nome del Queue Manager
133	Nome oggetto sconosciuto
145	Nome oggetto imprevisto
146	Nome oggetto assente
147	Tipo oggetto assente
148	Tipo oggetto invalido
149	Nome entità mancante

Esempi

L'esempio seguente mostra comando di visualizzazione delle autorizzazioni relative al Queue Manager saturn.queue.manager associato al gruppo utente staff:

dspmqaout

```
dspmqaout -m saturn.queue.manager -t qmgr -g staff
```

I risultati da questo comando sono:

```
Entity staff has the following authorizations for object :  
  get  
  browse  
  put  
  inq  
  set  
  connect  
  altusr  
  passid  
  passall  
  setid
```

Comandi correlati

setmqaout Imposta o reimposta l'autorizzazione (Set or reset authority)

dspmqsrv (Display Command Server)

Scopo

Utilizzare il comando **dspmqsrv** per visualizzare lo stato del server di comando relativo ad uno specifico Queue Manager.

Verrà visualizzato uno dei seguenti stati:

- Avvio
- Esecuzione
- Esecuzione con SYSTEM.ADMIN.COMMAND.QUEUE non abilitato per ricevere
- Chiusura
- Arresto

Sintassi

```
▶▶ dspmqsrv QMgrName ▶▶
```

Parametri richiesti

QMgrName

Specifica il nome del Queue Manager locale per il quale viene richiesto lo stato del server di comando.

Codici di ritorno

- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

Esempi

Il seguente comando visualizza lo stato del server di comando associato al `venus.q.mgr`:

```
dspmqsrv venus.q.mgr
```

Comandi correlati

- strmqsv** Avvia un server di comandi
- endmqsv** Chiude un server di comandi

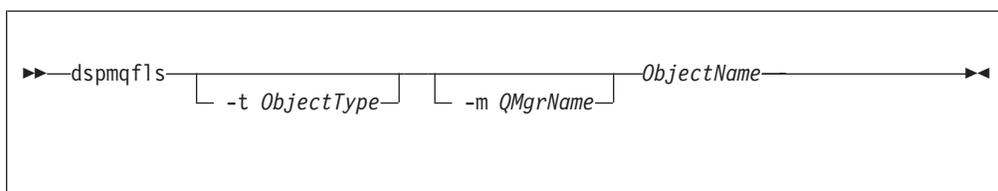
dspmqls (Visualizza gli attributi file MQSeries)

Scopo

Utilizzare il comando **dspmqls** per visualizzare il vero nome del file system per tutti gli oggetti MQSeries che rispondono ad un determinato parametro. È possibile utilizzare tale comando per identificare i file associati ad un particolare oggetto MQSeries. In questo modo è più agevole la realizzazione di copie di riserva di determinati oggetti. Consultare "Struttura dei volumi" a pagina 61 per ulteriori informazioni relative alla trasformazione del nome.

E' possibile utilizzare anche il comando **dspmqls** per visualizzare le impostazioni correnti per il nome-processo del server di code e per le opzioni di un oggetto.

Sintassi



Parametri richiesti

ObjectName

Specifica il nome dell'oggetto MQSeries. Il nome è una stringa priva di flag ed è un parametro obbligatorio. Se omissso, viene riportato un errore.

Questo parametro supporta il carattere globale * alla fine della stringa.

Parametri facoltativi

-m *QMgrName*

Specifica il nome del Queue Manager i cui file devono essere esaminati. Se il parametro viene omissso, il comando opera sul gestore code predefinito.

-t *ObjType*

Specifica il tipo di oggetto MQSeries. L'elenco che segue mostra i tipi di oggetto validi. I nomi abbreviati sono riportati seguiti dalla loro estensione.

* o all	Tutti i tipi di oggetto; questa è l'impostazione predefinita
q o queue	Una o più code corrispondenti al parametro nome oggetto
ql o qlocal	Una coda locale
qa o qalias	Un alias di coda
qr o qremote	Una coda remota
qm o qmodel	Un modello di coda
qmgr	Un oggetto Queue Manager
prcs o process	Un processo
ctlg o catalog	Un oggetto catalogo
nl o namelist	Un namelist

Nota: Il comando **dspmqls** visualizza i nomi di tutti i file per la coda.

Codici di ritorno

0 Comando completato normalmente
10 Comando completato con risultato inatteso

20 Si è verificato un errore durante l'elaborazione

Esempi

1. Il seguente comando visualizza i dettagli di tutti gli oggetti con i nomi che iniziano con SYSTEM.ADMIN che sono definiti sul gestore code predefinito:

```
dspmqls SYSTEM.ADMIN*
```

2. Il seguente comando visualizza i dettagli sul file per tutti i processi con i nomi che iniziano con PROC definiti sul gestore code RADIUS:

```
dspmqls -m RADIUS -t prcs PROC*
```

3. Il comando seguente visualizza le informazioni sul file per MY.LOCAL.QUEUE:

```
dspmqls -m MT02 -t q MY.LOCAL.QUEUE

MQSeries Display MQ Files

CONNECTING.
QLOCAL MY.LOCAL.QUEUE
  $DATA0.MT02M.QMYXLOCA
  $DATA0.MT02M.OMYXLOCA
  $DATA0.MT02M.TMYXLOCA

Queue/Status Server:          DEFAULT
Persistent message browse buffer: 0
Message overflow threshold:   200000
Queue Server Options:         None
Message overflow subvolume:   QMGR0010
Measure Counter:
```

- QMYXLOCA è il file di code, OMYXLOCA è il file di overflow per la coda e TMYXLOCA è il file touch nella directory dei dati del gestore code.
4. Il seguente esempio mostra un comando **altmqfls** che imposta il caricamento all'avvio e le opzioni del server di code NPM di verifica. Il comando **dspmqls** visualizza i risultati del comando **altmqfls**.

dspmqls

```
altmls --qmgr MT02 --type ql --qsoptions SC MY.LOCAL.QUEUE  
  
dspmqls -m MT02 -t q MY.LOCAL.QUEUE  
  
MQSeries Display MQ Files  
  
CONNECTING.  
QLOCAL MY.LOCAL.QUEUE  
  $DATA4.MT02M.QMYXLOCA  
  $DATA4.MT02M.OMYXLOCA  
  $DATA4.MT02M.TMYXLOCA  
  
Queue/Status Server:          DEFAULT  
Persistent message browse buffer: 0  
Message overflow threshold:    200000  
Queue Server Options:         SC  
Message overflow subvolume:    QMGR0010  
Measure Counter:
```

Comandi correlati

altmqfls Modifica del volume coda

dspmqtrc (Visualizza output di traccia formattato di MQSeries)

Scopo

Utilizzare il comando **dspmqtrc** per visualizzare l'output di traccia formattata di MQSeries. Per ulteriori informazioni sull'utilizzo della traccia di MQSeries, consultare: "Utilizzo della traccia MQSeries" a pagina 215.

Sintassi

```
▶▶—dspmqtrc— -t FormatTemplate —InputFileName————▶▶
```

Parametri richiesti

InputFileName

Specifica il nome del file contenente la traccia non formattata. Ad esempio
\$DATA.MQTRACE.AMQ12345..

-t *FormatTemplate*

Specifica il nome del file modello contenente dettagli su come visualizzare la traccia. Un file di modello di formato di traccia, AMQTRC, viene fornito nel volume secondario ZMQSSMPL.

Comandi correlati

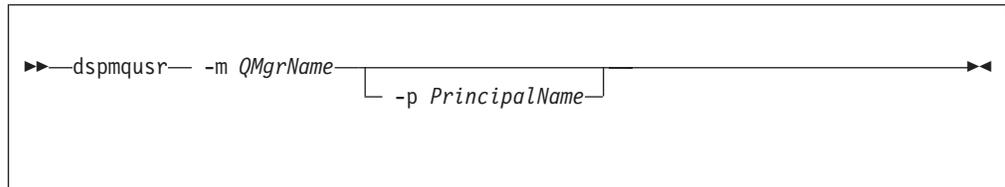
endmqtrc	Arresta traccia MQSeries (End MQSeries trace)
strmqtrc	Avvia traccia MQSeries (Start MQSeries trace)

dspmqusr (visualizza informazioni utenti MQSeries)

Scopo

Utilizzare il comando **dspmqusr** per visualizzare le informazioni sul principal specificato oppure tutti i principal per il queue manager.

Sintassi



Descrizione

E' possibile utilizzare questo comando per:

- Visualizzare tutti i principal oppure uno in particolare, definito per un queue manager.
- Visualizza Compaq NSK Administrative e i gruppi condivisione file SAFEGUARD corrispondenti all'ID utente Compaq NSK associati a ciascun principal.

Parametri richiesti

-m *QMGrName*

E' il nome del queue manager cui appartengono i principal.

Parametri facoltativi

-p *PrincipalName*

E' il nome del principal da visualizzare.

Codici di ritorno

- 0 operazione eseguita
- 36 Elemento immesso non valido
- 69 Memoria non disponibile
- 71 Errore imprevisto

Esempi

1. Tale esempio mostra **dspmqusr** per un queue manager appena creato:

```

dspmqusr -m MT02

```

Principale	IDutente	Nomeutente	Alias	Nomegruppo	Tipogruppo
	0.1				
NOBODY	0.0				
mqm	20.255	MQM.MANAGER	n	MQM	a

dspmqsusr

Il database principale contiene l'mqm principale, associato al nome utente dell'utente che ha creato il queue manager.

- Questo esempio mostra l'output da **dspmqsusr** dopo che i principal aggiuntivi sono stati aggiunti a **altmqusr**:

```
dspmqsusr -m MT02
```

Principale	IDutente	Nomeutente	Alias	Nomegruppo	Tipogruppo
	0.1				
NOBODY	0.0				
mqm	20.255	MQM.MANAGER	n	MQM	a
mquser1	50.3	MQTEST.FRED	n	MQTEST	a
				MQM	s
mquser2	1.1	GROUP.USER01	n	GROUP	a

Principal mquser1, che si associa all'utente Compaq ID MQTEST.FRED, è stato aggiunto. FRED è un membro del gruppo MQTEST e un membro del gruppo MQM che utilizza l'alias SAFEGUARD.

Principal mquser2 è in corrispondenza con l'ID utente Compaq GROUP.USER01.

Comandi correlati

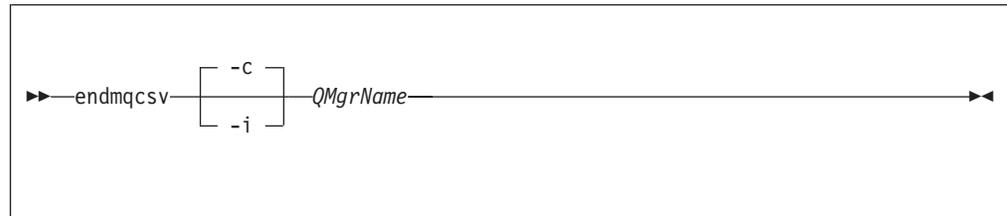
altmqusr Informazioni utente per Alter MQSeries

endmqcsv (End command server)

Scopo

Utilizzare il comando **endmqcsv** per arrestare il server di comandi relativo al Queue Manager specificato.

Sintassi



Parametri richiesti

QMgrName

Specifica il nome del Queue Manager il cui server di comandi deve essere arrestato.

Parametri facoltativi

-c Specifica che il server di comandi deve essere arrestato in modo controllato. Al server di comandi sarà così possibile completare le elaborazioni dei messaggi di comando già avviate. Nessuno nuovo messaggio viene letto dalla coda comandi.

Questo è il valore predefinito.

-i Specifica che il server di comandi deve essere arrestato immediatamente. Non verranno completate le azioni associate ad un messaggio di comando ancora in elaborazione.

Codici di ritorno

- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

Esempi

1. Il seguente comando arresta il server di comandi relativo al Queue Manager saturn.queue.manager:

```
endmqcsv -c saturn.queue.manager
```

Il server di comandi, prima dell'arresto, completa l'elaborazione dei comandi già avviati. I nuovi comandi ricevuti rimangono non elaborati nella coda comandi fino al nuovo riavvio del server di comandi.

2. Il seguente comando arresta immediatamente il server di comandi relativo al Queue Manager pluto:

endmqcsv

```
endmqcsv -i pluto
```

Comandi correlati

<code>strmqcsv</code>	Avvia un server di comandi
<code>dspmqcsv</code>	Visualizza lo stato di un server di comandi

endmqm (End queue manager)

Scopo

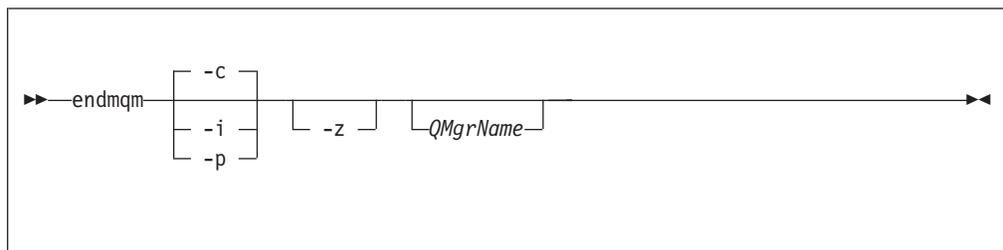
Utilizzare il comando **endmqm** per arrestare un Queue Manager locale specificato. Questo comando arresta il Queue Manager in uno dei seguenti tre modi:

- Chiusura normale o ad attività completate
- Chiusura immediata
- Chiusura preventiva

Gli attributi del Queue Manager e gli oggetti associati non vengono coinvolti. È possibile riavviare il Queue Manager utilizzando il comando **strmqm** (Start queue manager).

Per eliminare un Queue Manager, è necessario prima arrestarlo e poi utilizzare il comando **dltmqm** (Delete queue manager).

Sintassi



Parametri facoltativi

QMgrName

È il nome del queue manager dei messaggi da arrestare. Se non viene specificato alcun nome, viene arrestato il queue manager predefinito.

- c Chiusura controllata (o ad attività completate). Il Queue Manager viene chiuso solo dopo la disconnessione di tutte le applicazioni. Tutte le chiamate MQI in corso di elaborazione vengono completate. Questa è l'impostazione predefinita.
- i Chiusura immediata. Il Queue Manager viene chiuso dopo il completamento di tutte le chiamate MQI attualmente in elaborazione. Ogni richiesta di chiamata MQI successiva al comando ha esito negativo. Al successivo riavvio del Queue Manager viene eseguito il roll back di tutte le unità di elaborazione incomplete.
- p Chiusura preventiva.

Utilizzare questo tipo di chiusura solo in circostanze eccezionali. Ad esempio, qualora un Queue Manager non si arresta in seguito ad un normale comando **endmqm**.

Il Queue Manager viene chiuso senza attendere la disconnessione delle applicazioni o il completamento delle chiamate MQI. Questo metodo può avere conseguenze imprevedibili per le applicazioni MQI. Tutti i processi la cui chiusura ha esito negativo vengono arrestati 30 secondi dopo l'immissione del comando.

- z Sopprime i messaggi di errore relativi al comando

Codici di ritorno

- 0 Queue Manager arrestato
- 16 Queue Manager inesistente
- 36 Argomenti non validi
- 40 Gestore code non disponibile
- 69 Memoria non disponibile
- 71 Errore imprevisto
- 72 errore relativo al nome del Queue Manager

Esempi

Gli esempi seguenti mostrano comandi che arrestano il Queue Manager specificato:

1. Questo comando chiude il Queue Manager predefinito in modo controllato. Tutte le applicazioni attualmente connesse possono disconnettersi.

```
endmqm
```

2. Questo comando arresta il Queue Manager denominato saturn.queue.manager con il metodo immediato. Tutte le chiamate MQI in corso sono completate, ma non ne vengono accettate nuove.

```
endmqm -i saturn.queue.manager
```

endmqm

Comandi correlati

crtmqm	Crea un Queue Manager
strmqm	Avvia un Queue Manager
dltmqm	Elimina un Queue Manager

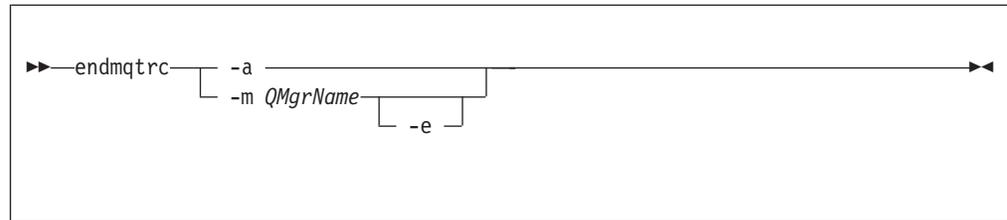
endmqtrc (End MQSeries trace)

Scopo

Utilizzare il comando **endmqtrc** per arrestare la traccia per uno specifico queue manager.

Per ulteriori informazioni sull'utilizzo della traccia di MQSeries, consultare: "Utilizzo della traccia MQSeries" a pagina 215.

Sintassi



Parametri richiesti

-m *QMgrName*

Indica il nome del Queue Manager la cui funzione di traccia deve essere arrestata.

Un nome di Queue Manager può essere specificato nello stesso comando come il flag **-e**.

-a Specificando questo flag, vengono arrestate tutte le tracce.

Questo flag *deve* essere specificato da solo.

Parametri facoltativi

-e Specificando tale flag, viene arrestata la traccia più recente su un dato queue manager.

Codici di ritorno

AMQ5611 Questo messaggio è emesso nel caso vengano forniti elementi non validi al comando.

Esempi

Questo comando arresta la funzione di traccia dei dati di un Queue Manager denominato QM1

```
endmqtrc -m QM1
```

Comandi correlati

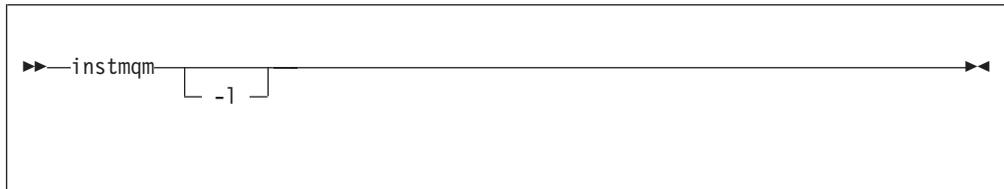
dspmqtrc Visualizza l'output di traccia formattato
strmqtrc Avvia traccia MQSeries (Start MQSeries trace)

instmqm (Installa MQSeries for Compaq NSK)

Scopo

Utilizzare il comando **instmqm** per installare MQSeries for Compaq NSK o aggiornare le informazioni sulla licenza.

Sintassi



Parametri facoltativi

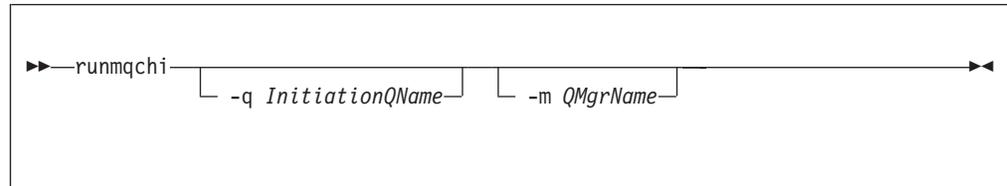
-l Richiama **instmqm** per gli aggiornamenti delle informazioni sulla licenza.

runmqchi (Run channel initiator)

Scopo

Utilizzare il comando **runmqchi** per eseguire un processo di inizializzazione di un canale. Per ulteriori informazioni relative all'utilizzo di questo comando, consultare il manuale *MQSeries Intercommunication*.

Sintassi



Parametri facoltativi

-q *InitiationQName*

Specifica il nome della coda da inizializzare elaborandola con channel initiator. Se non viene specificato alcun valore, viene utilizzato SYSTEM.CHANNEL.INITQ.

-m *QMgrName*

Specifica il nome del Queue Manager sul quale esiste la coda di iniziazione. Se viene omesso il nome, viene utilizzato Queue Manager predefinito.

Codici di ritorno

- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

Se si verificano gli errori riportati dai codici di errore 10 o 20 è necessario rivedere il log di errore Queue Manager associato al canale per i messaggi di errore. Inoltre è necessario rivedere il log di errore di sistema, poiché è lì che vengono registrati i problemi che si verificano prima dell'associazione del canale al Queue Manager. Per ulteriori informazioni relative ai log di errore, consultare "Log di errore" a pagina 212.

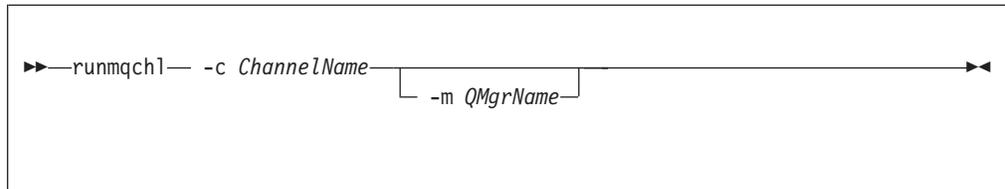
runmqchl (Run channel)

Scopo

Utilizzare il comando **runmqchl** per avviare un canale server completo, un Sender (SDR) o un Requester (RQSTR).

Il canale viene eseguito in modo asincrono. Per arrestare il canale, immettere il comando MQSC STOP CHANNEL.

Sintassi



Parametri richiesti

`-c ChannelName`
Specifica il nome del canale da avviare.

Parametri facoltativi

`-m QMgrName`
Specifica il nome del Queue Manager al quale è associato il canale. Se non viene specificato alcun nome, viene utilizzato il Queue Manager predefinito.

Codici di ritorno

- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

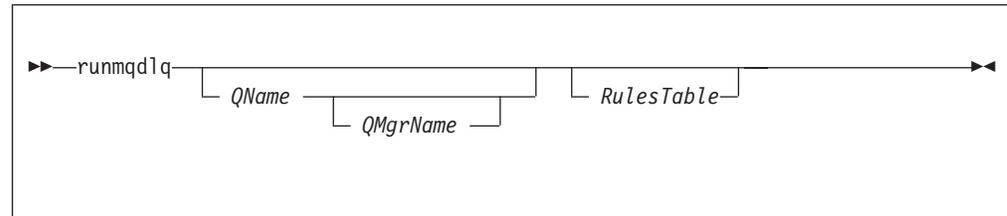
Se viene riportato il codice di errore 10 o 20, rivedere il log di errore del Queue Manager associato ai messaggi di errore. Inoltre è necessario controllare i log degli errori di sistema MQSeries poiché è lì che vengono registrati i problemi che si verificano prima dell'associazione del canale al Queue Manager.

runmqdlq (Run dead-letter queue handler)

Scopo

Utilizzare il comando **runmqdlq** per avviare il DLQ (Dead-Letter Queue) handler, un'utilità che elabora messaggi nelle code messaggi non recapitati.

Sintassi



Parametri facoltativi

QName

Indica il nome della coda messaggi non recapitati da elaborare.

Se si specifica un valore *QName*, questo modifica qualsiasi valore INPUTQ specificato in una tabella regole. Se non è specificato nessun nome nell'input per **runmqdlq** o nella tabella regole, la coda di messaggi non recapitati associata al queue manager definita nel parametro *QMgrName* viene elaborata.

QMgrName

Specifica il nome del Queue Manager che gestisce la coda da elaborare.

Se si specifica un valore *QMgrName*, questo modifica qualsiasi valore INPUTQM specificato in una tabella regole. Se non è specificato nessun nome nell'input per **runmqdlq** o nella tabella regole, la coda fa parte del queue manager predefinito.

RulesTable

È il nome del file contenente la tabella regole, che deve contenere almeno una regola.

Per impostazione predefinita, il comando **runmqdlq** riceve l'input dal file standard IN. Quando il comando viene elaborato, i risultati e un riepilogo vengono inseriti in un prospetto inviato al file standard OUT. Diversamente, reindirizzando l'input da un file, è possibile applicare una tabella regole alla coda specificata.

se non è specificata nessuna tabella regole per l'input per **runmqdlq**, le regole e le azioni devono essere specificate in modo interattivo.

In tal caso, DLQ handler:

- Legge i propri input dalla tastiera.
- Non avvia l'elaborazione delle code specificate se prima non riceve un carattere end_of_file (Ctrl-Y).

Le regole MQSC per le linee di commento e le linee di collegamento vengono applicate anche ai parametri di input DLQ handler.

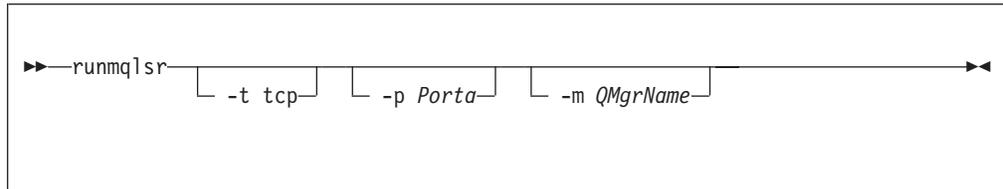
Per ulteriori informazioni sulle tabelle regole e sulla loro compilazione, consultare "Tabella regole DLQ handler" a pagina 160.

runmqlsr (Run listener)

Scopo

Il comando runmqlsr (Run listener) esegue un processo listener TCP/IP.

Sintassi



Descrizione

Quando viene eseguito da un prompt TACL, **runmqlsr** non restituisce il controllo al prompt TACL finché il listener non termina. Quindi **runmqlsr** rimane in attesa.

Il prompt TACL viene restituito solo se esiste un errore o il listener si interrompe. Se il terminale (TACL) viene arrestato prima del comando **runmqlsr**, il listener può accedere al relativo terminale principale o al file out. Prima che il comando **runmqlsr** venga richiamato, è necessario definire tutti i PARAM (ad esempio MQEMSEVENTS).

Per tali motivi, si consiglia di avviare e arrestare il listener dal PATHWAY del Queue Manager che fornisce un controllo di grado superiore.

Parametri facoltativi

-p *Port*

Numero di porta per TCP/IP. Se non è specificato alcun valore, verrà utilizzato il numero porta specificato nella voce TCPListenerPort della stanza TCPConfig del file QMINI. Il valore predefinito è 1414. Se in QMINI sono definite più porte di listener, verrà utilizzata la porta disponibile successiva.

Se è specificato il PARAM MPORTNUMBER nell'ambiente TACL o una definizione della classe del server PATHWAY relativa al programma, **runmqlsr** utilizzerà la porta specificata e non quella del file QMINI.

Se nessuna porta specificata in QMINI è libera oppure la porta specificata nel comando **runmqlsr** non è disponibile, il comando **runmqlsr** non verrà eseguito regolarmente.

-m *QMgrName*

specifica il nome del Queue Manager. Se non è specificato alcun nome, il comando viene eseguito sul Queue Manager predefinito.

-t *tcp*

Identifica TCP/IP come protocollo di trasmissione.

Se nell'ambiente TACL esiste DEFINE =TCPIP^PROCESS^NAME oppure una definizione della classe del server PATHWAY relativa al programma, **runmqlsr** utilizzerà il processo server Guardian TCP/IP anziché quello predefinito.

Se è specificato il PARAM MQPORTNUMBER o una definizione della classe del server PATHWAY relativa al programma, **runmqlsr** utilizzerà la porta specificata e non quella del file QMINI.

Questo è l'unico valore valido (e quello predefinito) inMQSeries for Compaq NSK.

Codici di ritorno

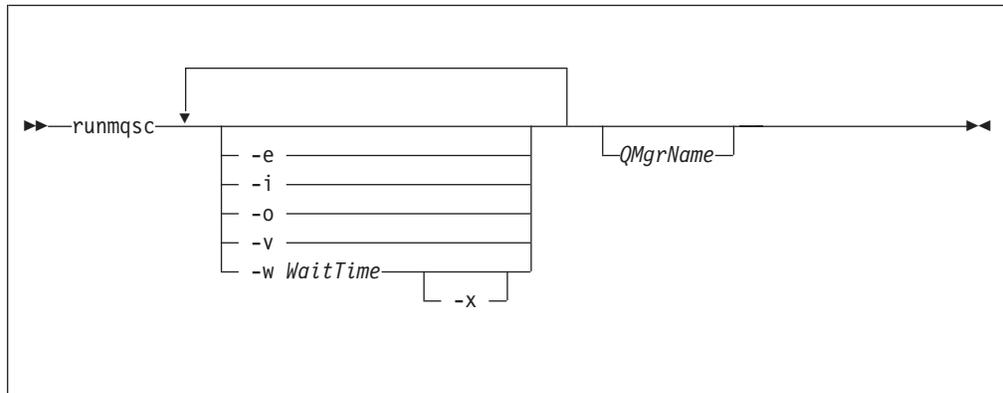
- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

runmqsc (Esegue comandi di MQSeries)

Scopo

Utilizzare il comando **runmqsc** per emettere comandi MQSC a un Queue Manager. I comandi MQSC consentono di eseguire attività amministrative, come ad esempio definire, alterare o eliminare un oggetto coda locale. I comandi MQSC e la loro sintassi sono descritti nel volume *MQSeries Guida di riferimento per i comandi*.

Sintassi



Descrizione

È possibile richiamare il comando **runmqsc** in tre modi:

Verify mode (modo verifica)

I comandi MQSC vengono verificati, ma non effettivamente eseguiti. Viene generato un prospetto di output che indica l'esito positivo o negativo di ciascun comando. Questo modo è disponibile solo su un Queue Manager locale.

Direct mode (modo diretto)

I comandi MQSC sono inviati direttamente al Queue Manager locale.

Indirect mode (modo indiretto)

I comandi MQSC sono eseguiti su un Queue Manager remoto. Questi comandi sono inseriti nella coda di comandi di un Queue Manager remoto e vengono eseguiti nell'ordine di accodamento. I prospetti dai comandi vengono restituiti al Queue Manager locale.

Il comando **runmqsc** riceve l'input dal file IN standard. Quando i comandi vengono elaborati, i risultati e un riepilogo vengono inseriti in un prospetto inviato al file standard OUT.

Attenendosi al file IN standard ed alla tastiera, sarà possibile immettere i comandi MQSC in maniera interattiva.

Reindirizzando l'input da un file sarà possibile eseguire una sequenza di comandi frequentemente utilizzati nel file. È anche possibile reindirizzare il prospetto di output a un file.

Nota: Per eseguire questo comando, è necessario che l'ID utente utilizzato appartenga al gruppo utenti MQM.

Parametri facoltativi

- e Impedisce che il testo d'origine relativo ai comandi MQSC sia copiato in un prospetto. Ciò è utile quando si immettono comandi in maniera interattiva.
- i nome file di input
- o nome file di output
- v Specifica il modo di verifica; ciò verifica i comandi specificati senza eseguire le azioni. Questa modalità è disponibile solo localmente. I flag -w e -x saranno ignorati se specificati simultaneamente.

-w *WaitTime*

Specifica il modo indiretto, e cioè che i comandi MQSC dovranno essere eseguiti su un altro Queue Manager. È necessario aver impostato a tale scopo le code di canale e trasmissione necessarie.

WaitTime

Specifica il tempo, in secondi, che **runmqsc** attende per le risposte. Qualsiasi risposta ricevuta dopo tale intervallo vengono eliminate, tuttavia i comandi MQSC saranno eseguiti ugualmente. Specificare un intervallo tra 1 e 999 999 secondi.

Ciascun comando viene inviato come PCF Escape alla coda di comando (SYSTEM.ADMIN.COMMAND.QUEUE) del Queue Manager di destinazione.

Le risposte saranno ricevute sulla coda SYSTEM.MQSC.REPLY.QUEUE e il risultato verrà aggiunto al prospetto. Ciò può essere definito come coda locale o coda modello.

Le operazioni in modo indiretto vengono eseguite tramite il Queue Manager predefinito.

Questo flag è ignorato se viene specificato il flag -v.

- x Indica che il queue manager di destinazione è eseguito in . Questo flag si applica solo in modo indiretto. Occorre specificare anche il flag -w. In modo indiretto, i comandi MQSC sono scritti in una forma adeguata alla coda di comandi di MQSeries per MVS/ESA.

QMgrName

Specifica il nome del Queue Manager di destinazione sul quale eseguire i comandi MQSC. Se omissso, i comandi MQSC saranno eseguiti sul Queue Manager predefinito.

Codici di ritorno

- 00 File di comandi MQSC elaborato con esito positivo.
- 10 File di comandi MQSC elaborato con errori - il prospetto contiene i motivi del malfunzionamento dei comandi.
- 20 Errore - file di comandi MQSC non eseguito.

Esempi

1. Immettere questo comandi alla richiesta comandi TAQL:

```
runmqsc
```

runmqsc

È possibile immettere i comandi MQSC direttamente. Non è stato specificato nessun nome di queue manager e pertanto i comandi MQSC saranno elaborati sul queue manager predefinito.

2. L'esempio seguente descrive come indicare che i comandi MQSC vengano solo verificati:

```
runmqsc -i $SYSTEM.CONFIG.MQSCIN -v BANK
```

Viene verificato il file di comando MQSC \$SYSTEM.CONFIG.MQSCIN. Il nome del Queue Manager è BANK. L'output è visualizzato nella finestra corrente.

3. Questo comando esegue un comando MQSC sul queue manager BANK:

```
runmqsc -i MQSCFILE -o $TEST.MQ.MQSCOUT BANK
```

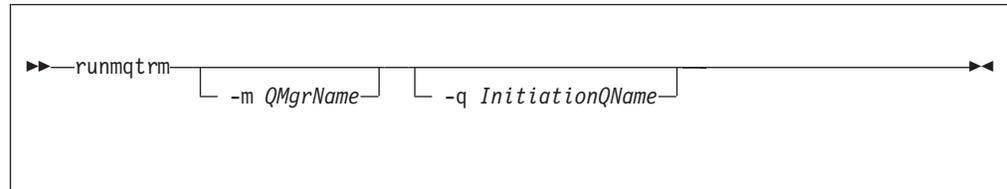
In questo esempio, l'output viene indirizzato al file \$TEST.MQ.MQSCOUT. Il file di input MQSCFILE nel volume secondario corrente.

runmqtrm (Avvia un trigger monitor)

Scopo

Utilizzare il comando **runmqtrm** per invocare un trigger monitor. Per ulteriori informazioni sull'utilizzo di trigger monitor, consultare *MQSeries Application Programming Guide*.

Sintassi



Parametri facoltativi

-m *QMgrName*

Specifica il nome del Queue Manager su cui opera il trigger monitor. Se questo parametro viene omissso, il trigger monitor opera sul Queue Manager predefinito.

-q *InitiationQName*

Specifica il nome della coda di avvio da elaborare. Se questo parametro viene omissso, viene utilizzato `SYSTEM.DEFAULT.INITIATION.QUEUE`.

Codici di ritorno

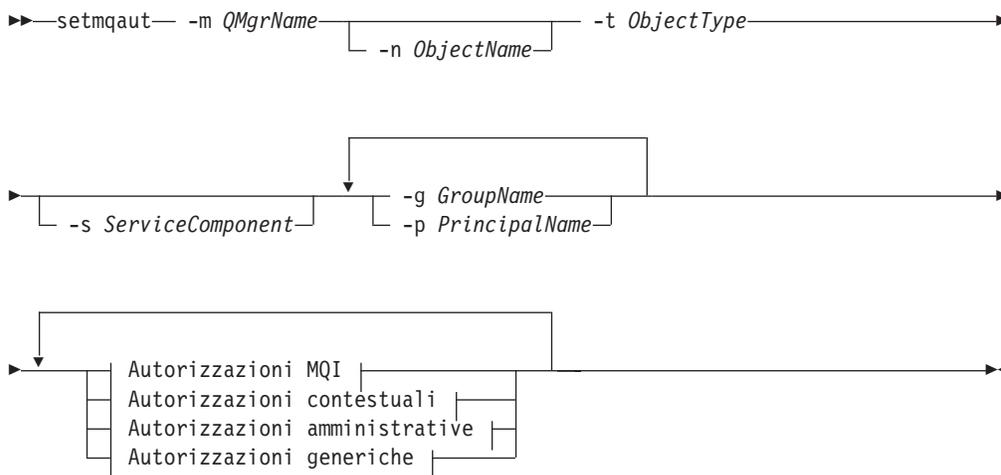
- 10 Trigger monitor interrotto per un errore.
- 20 Errore— trigger monitor non eseguito.

setmqaut (Imposta/reimposta l'autorizzazione)

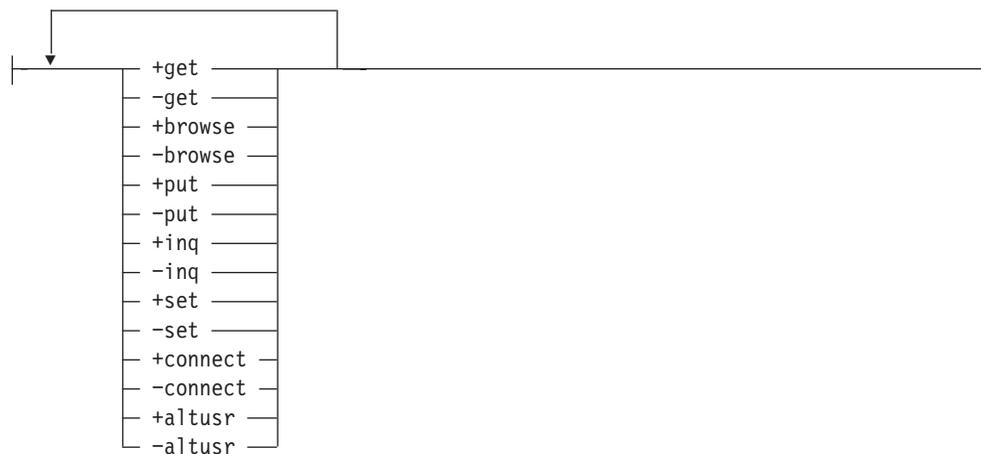
Scopo

Utilizzare il comando **setmqaut** per modificare le autorizzazioni di accesso a un oggetto o a una classe di oggetti. Le autorizzazioni possono essere concesse a, o revocate da, qualsiasi numero di principal o gruppi.

Sintassi



Autorizzazioni MQI:



Autorizzazioni contestuali:



Autorizzazioni amministrative:



Autorizzazioni generiche:



Descrizione

È possibile utilizzare questo comando sia per *impostare* un'autorizzazione, cioè accordare a un gruppo utente il permesso di eseguire un'operazione, che per *reimpostare* un'autorizzazione, cioè rimuovere il permesso di eseguire un'operazione. È necessario specificare i gruppi utente a cui applicare le autorizzazioni, oltre al Queue Manager, al tipo e al nome dell'oggetto. È possibile specificare qualsiasi numero di gruppi in un unico comando.

Le autorizzazioni che è possibile concedere si suddividono nelle seguenti categorie:

- Autorizzazioni di emissione di chiamate MQI
- Autorizzazioni per il contesto MQI
- Autorizzazioni di emissione di comandi per attività di amministrazione
- Autorizzazioni generiche

setmqaut

Ogni autorizzazione da modificare viene specificata in un elenco di autorizzazioni come parte del comando. Ogni voce dell'elenco è una stringa preceduta da '+' o '-'. Ad esempio, includendo +put nella lista delle autorizzazioni, si conferisce l'autorizzazione a emettere chiamate MQPUT verso una coda. In alternativa, includendo -put nell'elenco di autorizzazioni, si rimuove l'autorizzazione a emettere chiamate MQPUT.

È possibile specificare le autorizzazioni in qualsiasi ordine, a condizione che non entrino in conflitto. Ad esempio, la specificazione di allmqi insieme a set provoca un conflitto.

In un unico comando, è possibile specificare quanti gruppi o autorizzazioni sono necessari.

Se un ID utente è membro di più gruppi, le autorizzazioni applicate sono l'unione delle autorizzazioni di ogni gruppo a cui l'ID utente appartiene.

Parametri richiesti

-g *GroupName*

Specifica il nome del gruppo utente di cui si devono modificare le autorizzazioni. È possibile specificare più nomi di gruppo, ma ciascuno deve essere preceduto dal flag -g.

È necessario specificare almeno un principal o un gruppo.

-m *QMgrName*

Specifica il nome del Queue Manager dell'oggetto per il quale si devono modificare le autorizzazioni. Il nome può contenere fino a 48 caratteri.

-p *PrincipalName*

Specifica il nome del principal per il quale si devono modificare le autorizzazioni. È possibile specificare più nomi di principal, ma ciascuno deve essere preceduto dal flag -p.

È necessario specificare almeno un principal o un gruppo.

-t *ObjectType*

Specifica il tipo di oggetto per il quale si devono modificare le autorizzazioni.

I valori possibili sono:

- **q** o **coda**
- **prcs** o **processo**
- **qmgr**
- **nl** o **namelist**

Parametri facoltativi

-n *ObjectName*

Specifica il nome dell'oggetto per il quale si devono modificare le autorizzazioni.

Questo parametro è necessario *a meno che* non sia lo stesso Queue Manager. È necessario specificare un nome di Queue Manager, coda o processo, non utilizzando un nome generico.

-s *ServiceComponent*

Questo parametro si applica solo se si utilizzano servizi installabili di autorizzazione, altrimenti viene ignorato.

Se tali servizi sono supportati, questo parametro specifica il nome dell'Authorization Service a cui si applicano le autorizzazioni. Questo parametro è facoltativo; se non viene specificato, l'aggiornamento dell'autorizzazione viene eseguito sul primo componente installabile per il servizio.

Autorizzazioni

Specifica le autorizzazioni da concedere o rimuovere. Ogni voce dell'elenco è preceduta da un segno '+' che indica l'autorizzazione da concedere, o un segno '-', che indica l'autorizzazione da rimuovere. Ad esempio, per concedere l'autorizzazione per emettere una chiamata MQPUT dal MQI, specificare +put nell'elenco. Per rimuovere tale autorizzazione, specificare -put.

Tabella 10 mostra le autorizzazioni che si possono concedere ai diversi tipi di oggetto.

Tabella 10. Specificazione delle autorizzazioni per i diversi tipi di oggetto

Autorizzazione	Coda	Processo	Qmgr	Namelist
tutto	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
clr	✓			
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
put	✓			
inq	✓	✓	✓	✓
get	✓			
passall	✓			
passid	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

Autorizzazioni per le chiamate MQI

altusr Utilizza un ID utente alternativo in un messaggio.

Vedere *MQSeries Application Programming Guide* per ulteriori informazioni sugli ID utente alternativi.

sfoglia

Richiama un messaggio da una coda inviando una chiamata MQGET con l'opzione BROWSE.

connect

Collega l'applicazione al Queue Manager specificato inviando una chiamata MQCONN.

setmqaut

- get** Richiama un messaggio da una coda inviando una chiamata MQGET.
- inq** Effettua una richiesta su uno specifico Queue Manager inviando una chiamata MQINQ.
- put** Invia un messaggio su una specifica coda inviando una chiamata MQPUT.
- set** Imposta attributi su una coda dal MQI inviando una chiamata MQSET.

Nota: Se si apre un coda per più opzioni, occorre essere autorizzati per ognuna di esse.

Autorizzazioni per il contesto

- passall**
Sposta tutto il contesto sulla coda specificata. Tutti i campi del contesto vengono copiati dalla richiesta originale.
- passid** Passa il contesto d'identità sulla coda specificata. Il contesto d'identità è lo stesso della richiesta.
- setall** Imposta tutto il contesto sulla coda specificata. È utilizzato da speciali utilità di sistema.
- setid** Imposta il contesto d'identità sulla coda specificata. È utilizzato da speciali utilità di sistema.

Autorizzazioni per i comandi

- chg** Modifica gli attributi dell'oggetto specificato.
- clr** Cancella la coda specificata (solo comando PCF Clear queue).
- crt** Crea oggetti del tipo specificato.
- dlt** Cancella l'oggetto specificato.
- dsp** Visualizza gli attributi dell'oggetto specificato.

Autorizzazioni per operazioni generiche

- tutto** Utilizza tutte le operazioni applicabili all'oggetto.
- alladm**
Esegue tutte le operazioni di amministrazione applicabili all'oggetto.
- allmqi** Utilizza tutte le chiamate MQI applicabili all'oggetto.

Codici di ritorno

- 0 operazione eseguita
- 36 Elemento immesso non valido
- 40 Gestore code non disponibile
- 49 Queue Manager in chiusura
- 69 Memoria non disponibile
- 71 Errore imprevisto
- 72 errore relativo al nome del Queue Manager
- 133 Nome oggetto sconosciuto
- 145 Nome oggetto imprevisto
- 146 Nome oggetto assente
- 147 Tipo oggetto assente
- 148 Tipo oggetto invalido
- 149 Nome entità mancante
- 150 Specificazione di autorizzazione assente
- 151 Specificazione di autorizzazione non valida

Esempi

1. Questo esempio mostra un comando che specifica che l'oggetto a cui sono state concesse le autorizzazioni è la coda orange.queue sul Queue Manager saturn.queue.manager.

```
setmqaut -m saturn.queue.manager -n orange.queue -t queue -g tango +inq +alladm
```

Le autorizzazioni sono state concesse al gruppo utente tango e l'elenco autorizzazioni associato specifica che il gruppo utente tango:

- Può emettere chiamate MQINQ.
 - Dispone dell'autorizzazione a eseguire tutte le operazioni di amministrazione su quell'oggetto.
2. In questo esempio, l'elenco autorizzazioni specifica che il gruppo utente foxy:
 - Non può emettere alcuna chiamata da MQI alla coda specificata.
 - Dispone dell'autorizzazione a eseguire tutte le operazioni di amministrazione sulla coda specificata.

```
setmqaut -m saturn.queue.manager -n orange.queue -t queue -g foxy -allmqi +alladm
```

setmqaut

Comandi correlati

dspmqaut Visualizza autorizzazione (Display authority)

strmqcsv (Avvia server di comandi)

Scopo

Utilizzare il comando **strmqcsv** per avviare il server di comandi per il Queue Manager specificato. Ciò abilita MQSeries all'elaborazione dei comandi inviati alla coda di comando.

Sintassi

```
▶▶ strmqcsv QMgrName ▶▶
```

Parametri richiesti

QMgrName

Specifica il nome del Queue Manager per il quale avviare il server di comandi.

Codici di ritorno

- 0 Comando completato normalmente
- 10 Comando completato con risultato inatteso
- 20 Si è verificato un errore durante l'elaborazione

Esempi

Il seguente comando avvia un server di comandi per il Queue Manager earth:

```
strmqcsv earth
```

Comandi correlati

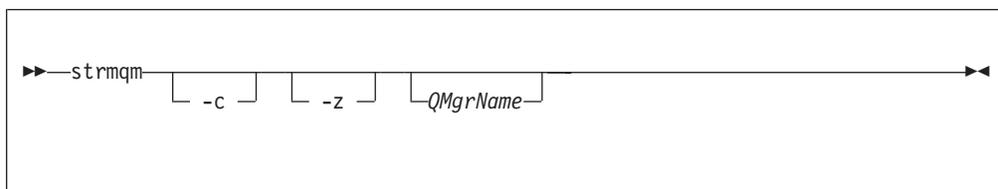
- endmqcsv** Chiude un server di comandi
- dspmqcsv** Visualizza lo stato di un server di comandi

strmqm (Avvio Queue Manager)

Scopo

Utilizzare il comando **strmqm** per avviare un Queue Manager locale. Solo dopo il Queue Manager è disponibile per elaborare connessioni o altre richieste, il comando **strmqm** ritornerà alla riga comandi.

Sintassi



Parametri facoltativi

-c Avvia il Queue Manager, ridefinisce gli oggetti di sistema predefiniti, quindi arresta il Queue Manager (gli oggetti di sistema predefiniti per un Queue Manager vengono creati inizialmente dal comando `crmqm`). Qualsiasi oggetto di sistema predefinito appartenente al Queue Manager viene sostituito se si specifica questo flag.

QMgrName

Specifica il nome di un Queue Manager locale da avviare. Se non si specifica un nome, viene avviato il Queue Manager predefinito.

-z Sopprime i messaggi di errore.

Questo flag viene utilizzato in MQSeries per eliminare i messaggi di errore indesiderati. Poiché l'utilizzo di questo flag comporta la perdita di dati, si consiglia di non utilizzarlo quando si immettono comandi in una riga comandi.

Codici di ritorno

- 0 Queue Manager avviato
- 3 Creazione di Queue Manager
- 5 Queue Manager in esecuzione
- 16 Queue Manager inesistente
- 49 Queue Manager in chiusura
- 69 Memoria non disponibile
- 71 Errore imprevisto
- 72 errore relativo al nome del Queue Manager

Esempi

Il seguente comando avvia il queue manager account:

```
strmqm account
```

Comandi correlati

- crmqm** Crea un Queue Manager
- dltmqm** Elimina un Queue Manager
- endmqm** Terminare un Queue Manager

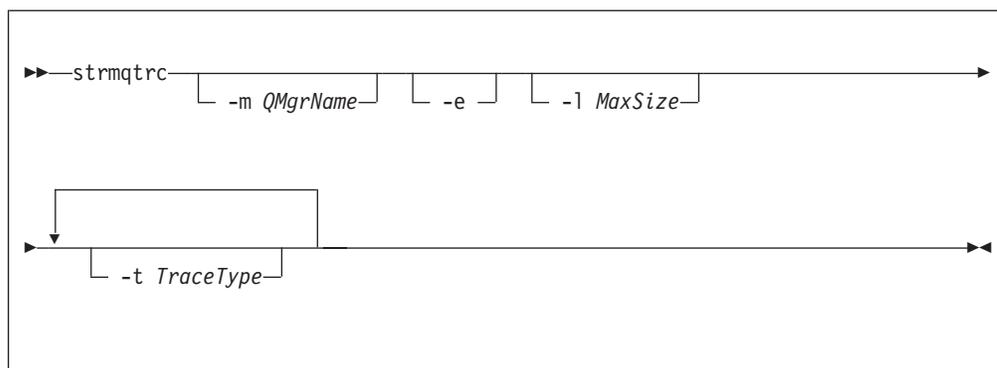
strmqtrc (Avvio traccia di MQSeries)

Scopo

Utilizzare il comando **strmqtrc** per abilitare la traccia. L'esecuzione di tale comando è possibile sia in caso di traccia abilitata che non abilitata. Se la traccia è già abilitata, le opzioni di traccia vengono infatti modificate in quelle specificate nel più recente richiamo del comando.

Per ulteriori informazioni sull'utilizzo della traccia di MQSeries, consultare: "Utilizzo della traccia MQSeries" a pagina 215.

Sintassi



Parametri facoltativi

-m *QMgrName*

È il nome del Queue Manager da tracciare. Se non viene specificato alcun nome, viene utilizzato il Queue Manager predefinito.

Il queue manager specificato non deve essere in esecuzione o neppure esistere. Di conseguenza, è possibile tracciare la creazione o l'avvio di un Queue Manager.

Un nome di Queue Manager può essere specificato nello stesso comando come il flag **-e**. Se a una data entità da tracciare si applica più di una specifica di traccia, la traccia include tutte le opzioni specificate.

- e** Se questo flag è specificato, è necessaria una traccia iniziale. Ciò implica la scrittura dell'informazione di traccia prima che i processi riconoscano a quale componente MQSeries appartengono. Qualsiasi processo, appartenente a qualsiasi componente di qualsiasi Queue Manager, traccia la propria elaborazione iniziale se tale flag è specificato. Per impostazione predefinita, se il flag non è specificato, non sarà eseguita la traccia iniziale.

-l *MaxSize*

Il valore di *MaxSize* indica la massima dimensione in milioni di byte di un file di traccia (AMQnnnn.TRC). Ad esempio, specificando per *MaxSize* il valore 1, la dimensione della traccia è limitata a un milione di byte.

Quando un file di traccia raggiunge la massima dimensione specificata, viene rinominato da AQnnnn.TRC a AMQnnnn.TRS e viene avviato un nuovo file AMQnnnn.TRC. Tutti i file di traccia vengono riavviati quando è raggiunta la massima dimensione. Verrà cancellata ogni copia preesistente del file AMQnnnn.TRS.

strmqtrc

-t *TraceType*

Definisce quali punti possono essere tracciati durante l'elaborazione. E' possibile fornire uno o più delle seguenti opzioni:

tutto	Dati di output per ogni punto di traccia nel sistema. Questo è anche il valore predefinito se il flag -t non è specificato.
api	Dati di output per i punti di traccia associati ai componenti MQI e al principale Queue Manager.
comms	Dati di output per i punti di traccia associati ai dati trasmessi sulle reti di comunicazione.
csflows	Dati di output per i punti di traccia associati al flusso di elaborazione nei servizi comuni.
lqmfloWS	Dati di output per i punti di traccia associati al flusso di elaborazione nel Queue Manager locale.
remotefloWS	Dati di output per i punti di traccia associati al flusso di elaborazione nei componenti di comunicazione.
otherfloWS	Dati di output per i punti di traccia associati al flusso di elaborazione in altri componenti.
csdata	Dati di output per i punti di traccia associati ai buffer di dati interni nei servizi in comune.
lqmdata	Dati di output per i punti di traccia associati ai buffer di dati interni nel Queue Manager locale.
remotedata	Dati di output per i punti di traccia associati ai buffer di dati interni nei componenti di comunicazione.
otherdata	Dati di output per i punti di traccia associati ai buffer di dati interni in altri componenti.
versiondata	Dati di output per i punti di traccia associati alla versione di MQSeries in esecuzione.
commentary	Dati di output per i punti di traccia associati ai commenti nei componenti MQSeries.

L'omissione di questo flag comporta l'abilitazione di tutti i punti di traccia e la generazione di una traccia completa.

Nota: Se sono disponibili più tipi di traccia, ciascuna *deve* disporre del proprio flag -t. È possibile specificare qualsiasi quantità di flag -t, a condizione che a ciascuno sia associato un tipo di traccia valido.

Non è un errore specificare lo stesso tipo di traccia su molteplici flag -t.

Codici di ritorno

AMQ7024

Questo messaggio è emesso nel caso vengano forniti elementi invalidi al comando.

AMQ8304

Il numero massimo di nove tracce simultanee è già in esecuzione.

Esempi

Questo comando abilita la traccia di dati dai servizi comuni e dal Queue Manager locale, per un Queue Manager denominato QM1.

```
strmqtrc -m QM1 -t csdata -t lqmdata
```

Comandi correlati

dspmqrtrc	Visualizza l'output di traccia formattato
endmqtrc	Arresta traccia MQSeries (End MQSeries trace)

upgmqm (Aggiorna il gestore code V2.2.0.1)

Scopo

Questo comando aggiorna un gestore code Versione 2.2.0.1 in modo che possa essere utilizzato con MQSeries for Compaq NSK V5.1. Il programma di utilità richiamato da **upgmqm** invia dei messaggi che indicano lo stato di avanzamento dell'operazione al terminale da cui è stato richiamato. Al termine dell'operazione di aggiornamento da parte del programma di utilità, il gestore code è pronto per essere utilizzato con MQSeries for Compaq NSK V5.1. Gli attributi del gestore code nuovi nella Versione 5.1 sono impostati sui loro valori predefiniti. E' possibile modificarlo seguendo la normale procedura.

Poiché la funzionalità del server di stato della nuova Versione 5.1 è diversa da quella del Server MQSS della Versione 2.2.0.1, quest'aggiornamento cancella tutte le classi del server *mm* MQS-Status esistenti dalla configurazione PATHWAY esistente e le sostituisce con una classe di server MQS-Status00 predefinita ed una classe di server MQS-Queue00 predefinita. Questo significa che dopo aver eseguito l'aggiornamento del gestore code, gli eventuali oggetti assegnati agli attuali processi di server MQSS dovranno essere riassegnati ai nuovi processi di server di stato o processi di server di coda, come appropriato. (Nella Versione 5.1, solo le code locali sono assegnate ai server di coda e tutti gli altri oggetti sono assegnati ai server di stato). E' anche possibile che occorra modificare gli eventuali script esistenti che fanno riferimento alle classi di server di stato in PATHWAY.

Se non si desidera eseguire l'eliminazione dei dati obsoleti durante il processo di aggiornamento, è possibile cancellare i seguenti file quando lo si ritiene più opportuno.

Tabella 11. Esempi di file che possono essere cancellati dopo un aggiornamento

Posizione	File	Esempio
Volume secondario indicato dall'opzione -p in upgmqm	Tutti i file	\$VOL.scratch
Volume secondario 'S' dei gestori code	Z*	\$VOL.myv2201S

Questi esempi sono basati sull'assunto che il gestore code esistente si trova sui volumi secondari \$VOL.myv2201?.

Nota: Se un gestore code Versione 2.2.0.1 non viene aggiornato utilizzando **upgmqm**, i relativi comandi di controllo non funzioneranno. Ciò vale anche per **dltmqm**, che avrà esito negativo con durante il tentativo di apertura del database principale. Un gestore code della Versione 2.2.0.1 non più necessario nella Versione 5.1, deve essere rimosso prima dell'esecuzione dell'aggiornamento del codice, oppure aggiornato utilizzando **upgmqm** e quindi rimosso.

Sintassi

```

▶—upgmqm—-m QMgrName—-v DefaultQueueServer—-p SubvolumePath—————▶
▶—-s DefaultStatusServer—————▶◀

```

Parametri richiesti

- m *QMgrName*
Specifica il nome del gestore code cui deve essere applicato il programma di utilità **upgmqm**.
- v *DefaultQueueServer*
Un nome di processo unico per il server di coda predefinito per il gestore code.
- p *SubvolumePath*
Un percorso di volume secondario (\$VOL.SUBVOL) che può essere utilizzato dal programma di utilità di aggiornamento per i file operativi. Questo volume secondario **deve** trovarsi sullo stesso volume del gestore code. Specificare solo la parte relativa al volume secondario del percorso; non specificare il nome del volume. Il comando **upgmqm** controlla che il volume secondario non sia già esistente prima di accettarlo come valido.
- s *DefaultStatusServer*
Un nome di processo unico per il server di stato predefinito per il gestore code.

Esempi

Questo esempio aggiorna un queue manager della Versione 2.2.0.1 Myv2201qm con un nome di queue server predefinito di \$MYQS, un nome di status server di \$MYSS ed utilizza il volume secondario \$VOL.scratch per i file di lavoro (dove \$VOL è il volume su cui risiede il queue manager):

```
upgmqm -m Myv2201qm -p scratch -s $MYSS -v $MYQS
```

Parte 3. Appendici

Appendice A. MQSeries for Compaq NSK al primo sguardo

Numero programma e parte

- 5724-A39 MQSeries for Compaq NSK, Versione 5 Rilascio 1, numero parte 0791003

Requisiti hardware

I requisiti hardware minimi sono:

- Una delle gamme di macchine Compaq NSK supportate da Guardian D45 o successive D4x, G06 o successive G0x.
- Hardware specifico a supporto dei protocolli di trasporto di rete selezionati dall'utente.

Si consiglia inoltre di disporre di uno o più dischi di dati con mirroring con i requisiti di spazio specificati per il controllo di traccia TMF e per il database di MQSeries.

Requisiti software

I requisiti software minimi sono:

- Compaq NSK Guardian D45 o successive D4x (hardware serie K) sistemi operativi G06 o successive G0x (hardware serie S), compreso TM/MP (TMF), ENSCRIBE e EMS.
- TS/MP (PATHWAY) corrispondente al sistema operativo.
- SCF per la configurazione, il comando ed il controllo dei trasporti di rete TCP e SNA.

Per le connessioni SNA:

- SNAX/APC e SNAX/XF o SNAX/APN corrispondente al sistema operativo

oppure

- Insession ICE Versione 3.2 o successive

Per connessioni TCP/IP:

- TCP/IP corrispondente al sistema operativo.

Per utilizzare le parti basate su OSS di MQSeries (binding MQI, applicazioni OSS, binding Java) è necessario disporre di una versione del prodotto OSS compatibile con il sistema operativo.

La registrazione delle transazioni è ancora valida nel prodotto Compaq TM/MP (TMF).

Sicurezza

MQSeries for Compaq NSK utilizza le funzioni di sicurezza del file system NSK, che forniscono un controllo dell'accesso a livello di file a UTENTI e GRUPPI per operazioni di lettura, scrittura, esecuzione ed eliminazione. SAFEGUARD non è richiesto per l'utilizzo di MQSeries for Compaq NSK; il prodotto è tuttavia compatibile con un ambiente SAFEGUARD.

Tutte le risorse MQSeries appartengono ad un singolo ID utente nel gruppo MQM. Per gestire MQSeries con i menu SCOBOL o con **runmqsc**, bisogna essere collegati con un ID utente assegnato o collegato al gruppo MQM.

Funzioni di manutenzione

Funzioni di MQSeries con:

- La funzione Message Queue Management (MQM) utilizzando gli schermi di configurazione del richiedente SCOBOL in un ambiente PATHWAY.
 - L'interfaccia della riga comandi **runmqsc**.
 - La funzione SCF per la configurazione, il comando ed il controllo degli ambienti TCP/IP e SNA per le offerte di protocolli di rete Compaq.
 - I programmi di utilità ICE forniti con detto prodotto per il controllo dell'interfaccia ICE LU 6.2.
 - MQSeries Explorer (non fornito con MQSeries for Compaq NSK).
 - Eventuali altri prodotti o programmi di utilità che utilizzando i comandi PCF standard per la gestione remota.
-

Compatibilità

La MQI per MQSeries for Compaq NSK V5.1 è compatibile con le applicazioni esistenti che eseguono MQSeries per Tandem NonStop Kernel V2.2.0.1, con fix di manutenzione PTF U473441.

Compilatori supportati

MQSeries for Compaq NSK V5.1 è stato creato utilizzando il CRE (Common Runtime Environment) per collegare tutti gli oggetti. Questo metodo impone i seguenti requisiti agli utenti delle versioni di MQI precedenti la Versione 2.2.0.1:

1. Tutto il codice oggetti precedente D45 COBOL e C deve essere ricompilato con il compilatore D45 (o successive) per integrare i collegamenti CRE.
2. Tutto il codice oggetto precedente D45 TAL deve essere ricompilato con un compilatore D45 (o successive) e bisogna assicurarsi che il programma TAL sia conforme ai criteri di programmazione speciali specificati nel manuale *Common Run-time Environment Programmer's Guide*. Ulteriori informazioni su questi criteri di programmazione sono disponibili nel manuale *TAL Programmer's Guide*.
3. Per il codice oggetto prodotto con i compilatori nativi su D45, viene fornito un binding separato.
4. I programmi C devono utilizzare il modello di memoria WIDE (valori numerici interi a 32-bit).
5. I programmi COBOL devono essere conformi ai requisiti di CRE.
6. Nei programmi TAL, tutti i valori numerici passati alle funzioni MQI devono essere di 32 bit (oppure essere convertiti in un formato a 32 bit utilizzando la macro \$INT32()).

Gli stessi programmi MQSeries sono compilati e collegati utilizzando gli strumenti di modalità originale per Guardian NSK. Le applicazioni di modalità originale di norma stabiliscono collegamenti con il gestore code SRL direttamente a meno che l'applicazioni non utilizzi già un SRL privato. In questo caso, poiché le applicazioni possono utilizzare solo un SRL privato, l'applicazione deve stabilire un collegamento con la libreria di binding MQI statica oppure il codice presente nell'SRL privato dell'applicazione deve essere combinato con l'SRL di MQSeries in un nuovo SRL privato.

Gestione delle licenze

Bisogna immettere il tipo di sistema per definire i diritti di cui dispone il programma. Questo parametro può essere immesso in fase di installazione o successivamente nel caso in cui l'utente acquisti un aggiornamento della licenza. In fase di avvio, questo valore viene messo a confronto con la configurazione della macchina Compaq fisica. Se la registrazione della licenza ed i diritti del programma sono insufficienti, viene emesso un messaggio di avvertenza.

Selezione della lingua

Un file di testo del messaggio fornito viene codificato in un insieme di caratteri a 7 bit che è originale per il sistema operativo Compaq NSK. MQSeries for Compaq NSK consente di specificare la lingua in fase di installazione del prodotto. La lingua dei messaggi è per impostazione predefinita l'inglese US.

Internazionalizzazione

MQSeries for Compaq NSK consente di specificare il CCSID quando viene creato il gestore code (anche se è possibile modificarlo successivamente). Il valore predefinito dei CCSID del Queue Manager è 819. MQSeries for Compaq NSK supporta la conversione dell'insieme di caratteri nel CCSID configurato del Queue Manager. Per ulteriori informazioni sui CCSID che è possibile specificare per il gestore code di MQSeries for Compaq NSK, compresi quelli che forniscono il supporto per il simbolo dell'Euro, consultare: *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

Appendice B. Impostazioni predefinite del sistema

Quando si crea un Queue Manager utilizzando il comando di controllo **crtmqm**, vengono automaticamente creati gli oggetti del sistema e gli oggetti predefiniti.

- Gli oggetti del sistema sono quelli di cui MQSeries necessita per il funzionamento di un Queue Manager o di un canale.
- Gli oggetti predefiniti definiscono tutti gli attributi di un oggetto. Quando si crea un oggetto, come ad esempio una coda locale, qualunque attributo omissivo verrà sostituito dall'oggetto predefinito.

Tabella 12. Oggetti del sistema e oggetti predefiniti per le code

Nome Oggetto	Descrizione
SYSTEM.DEFAULT.ALIAS.QUEUE	Alias di coda predefinito
SYSTEM.DEFAULT.LOCAL.QUEUE	Coda locale predefinita
SYSTEM.DEFAULT.MODEL.QUEUE	Modello di coda predefinito.
SYSTEM.DEFAULT.REMOTE.QUEUE	Coda remota predefinita.
SYSTEM.DEAD.LETTER.QUEUE	Esempio di coda messaggi non recapitati (DLQ)
SYSTEM.DEFAULT.INITIATION.QUEUE	Coda di avvio predefinita.
SYSTEM.CICS.INITIATION.QUEUE	Coda di avvio CICS® predefinita
SYSTEM.ADMIN.COMMAND.QUEUE	Coda di comando gestione. Utilizzata per comandi MQSC remoti e comandi PCF.
SYSTEM.MQSC.REPLY.QUEUE	Coda di risposta RTQ (Reply-To-Queue) MQSC. Questo è un modello di coda che crea una coda dinamica temporanea per rispondere ai comandi MQSC remoti.
SYSTEM.ADMIN.QMGR.EVENT	Coda evento per eventi Queue Manager.
SYSTEM.ADMIN.PERFM.EVENT	Coda evento per eventi delle prestazioni.
SYSTEM.ADMIN.CHANNEL.EVENT	Coda evento per eventi canale.
SYSTEM.CHANNEL.INITQ	Coda di avvio canale.
SYSTEM.CHANNEL.SYNCQ	La coda che contiene la sincronizzazione dei dati per i canali. (Tale oggetto viene creato ma non utilizzato in MQSeries for Compaq NSK. Le informazioni sulla sincronizzazione dei canali sono memorizzate nei database protetti TM/MP)
SYSTEM.CLUSTER.COMMAND.QUEUE	La coda utilizzata per trasmettere messaggi all'archivio Queue Manager.
SYSTEM.CLUSTER.REPOSITORY.QUEUE	La coda utilizzata per memorizzare tutte le informazioni dell'archivio di memoria.
SYSTEM.CLUSTER.TRANSMIT.QUEUE	La coda di trasmissione per tutti messaggi verso i cluster.

Tabella 13. Oggetti del sistema e oggetti predefiniti per i canali

Nome Oggetto	Descrizione
SYSTEM.DEF.SENDER	Canale di invio predefinito

Valori predefiniti di sistema

Tabella 13. Oggetti del sistema e oggetti predefiniti per i canali (Continua)

Nome Oggetto	Descrizione
SYSTEM.DEF.SERVER	Canale server predefinito
SYSTEM.DEF.RECEIVER	Canale di ricezione predefinito
SYSTEM.DEF.REQUESTER	Canale di richiesta predefinito
SYSTEM.DEF.SVRCONN	Canale predefinito per connessione server
SYSTEM.DEF.CLNTCONN	Canale predefinito per connessione client
SYSTEM.AUTO.RECEIVER	Canale di ricezione dinamico
SYSTEM.AUTO.SVRCONN	Canale dinamico per connessione al server
SYSTEM.DEF.CLUSRCVR	Canale predefinito di ricezione per il cluster utilizzato per sostituire le impostazioni predefinite relative a qualsiasi attributo non specificato durante la creazione di un canale CLUSRCVR su un Queue Manager in un cluster.
SYSTEM.DEF.CLUSSDR	Canale predefinito di invio per il cluster utilizzato per sostituire le impostazioni predefinite relative a qualsiasi attributo non specificato durante la creazione di un canale CLUSSDR su un Queue Manager nel cluster.

Tabella 14. Oggetti del sistema e oggetti predefiniti per elenchi di nomi

Nome Oggetto	Descrizione
SYSTEM.DEFAULT.NAMELIST	namelist predefinito

Tabella 15. Oggetti del sistema e oggetti predefiniti per processi.

Nome Oggetto	Descrizione
SYSTEM.DEFAULT.PROCESS	Processo predefinito di definizione

Appendice C. Impostazione delle variabili di ambiente TACL per MQSeries for Compaq NSK

MQSeries crea ed utilizza una serie di variabili di ambiente Compaq NSK, o PARAM. Quando si impostano questi PARAM, valutare che:

- Il PARAM MQDEFAULTPREFIX deve essere presente nell'ambiente di tutti i programmi. La configurazione TS/MP (Pathway) stabilita automaticamente dal comando **crtmqm** assicura che questi PARAM siano impostati correttamente per tutti i processi dei server dei gestori code. Gli utenti delle applicazioni MQSeries e dei comandi di controllo devono assicurare che le TACL e le configurazioni TS/MP utilizzate specifichino anch'esse queste variabili.
- Si consiglia di inserire le istruzioni PARAM nei file TACL CSTM in modo che, quando ci si collega, questi PARAM vengano creati correttamente e gli eventuali programmi eseguiti dalla TACL permutino i valori corretti. E' inoltre necessario modificare la seguente variabile di ambiente per consentire il rilevamento della posizione degli eseguibili MQSeries:

```
#SET #PMSEARCH $SYSTEM.ZMQSEXE [#PMSEARCH]
```

MQCONNECTTYPE

Questo PARAM, se presente, può essere utilizzato per disabilitare la capacità di utilizzare le connessioni FASTPATH delle applicazioni. Se questo PARAM viene impostato sul valore STANDARD, le applicazioni possono utilizzare solo le connessioni STANDARD anche se richiedono quelle FASTPATH. Qualsiasi altro valore viene elaborato come se non fosse stato specificato il PARAM (e quindi le applicazioni, se lo richiedono, possono utilizzare le connessioni FASTPATH).

MQDEFAULTPREFIX

Il nome del volume che contiene il volume secondario installato, ZMQSSYS. Questo PARAM deve essere definito correttamente in tutti gli ambienti.

Ad esempio:

```
PARAM MQDEFAULTPREFIX $data00
```

MQEMSEVENTS

Questo PARAM abilita gli eventi MQSeries EMS. Ad esempio, per attivare tutti gli eventi EMS per MQSeries, impostare il PARAM MQEMSEVENTS nel seguente modo:

```
PARAM MQEMSEVENTS 127
```

MQMACHINIFILE

La posizione del file MQSINI per l'installazione. Il valore predefinito è *MQDEFAULTPREFIX.ZMQSSYS.MQSINI*. Questo PARAM è richiesto solo se è richiesta una posizione non predefinita.

variabili di ambiente TACL

Ad esempio:

```
PARAM MQMACHINIFILE $data00.altinst.mqsini
```

MQRDF

Se questo PARAM è impostato su ON, MQSeries modifica il comportamento dell'operazione di cancellazione in modo che funzioni con RDF per i file sottoposti a controllo. Se questo PARAM non è definito oppure è impostato su un valore diverso da ON, l'operazione di cancellazione di MQSeries funziona in modo normale. Se utilizzato, questo PARAM deve essere impostato nell'ambiente TACL di tutti gli utenti che eseguono programmi di gestione e nell'ambiente di tutte le classi server TS/MP configurate nel gestore code.

MQRDFUPPROCESSNAME

Questo PARAM è interpretato solo dal programma di utilità **cleanrdf**. Esso viene utilizzato per specificare un nome di processo Guardian che verrà assegnato al processo server FUP creato dal programma di utilità **cleanrdf**. Se questo PARAM non è definito, il nome di processo FUP viene assegnato dal sistema operativo.

MQRDFUPPROGNAME

Questo PARAM è interpretato solo dal programma di utilità **cleanrdf**. Viene utilizzato per specificare il nome completo del file eseguibile FUP che deve essere utilizzato dal programma di utilità. Il valore predefinito è <nome sistema predefinito>.\$SYSTEM.SYS00.FUP .

MQSNOAUT

Se questo PARAM viene impostato su 1 quando viene eseguito **crtmqm**, il nuovo gestore code viene creato con OAM disabilitato.

Ad esempio:

```
PARAM MQSNOAUT 1
```

MQLISTENPORTNUM

Se questo PARAM viene impostato, il processo Listener TCP/IP lo utilizza per rilevare la porta su cui attivarsi. Se il PARAM non viene specificato, la porta viene determinata dalla stanza TCP/IP del file QMINI.

SAVE-ENVIRONMENT ON

Richiesto quando si eseguono programmi applicazioni per assicurare che il CRE (Common Run-Time Environment) passi i PARAM dall'ambiente al programma applicativo.

Ad esempio:

```
PARAM SAVE-ENVIRONMENT ON
```

Se questo PARAM non viene impostato, le applicazioni ricevono il codice di ritorno 2058, che indica un errore nel nome del gestore code.

Questo PARAM è richiesto solo per le applicazioni TAL e COBOL, in esecuzione come programmi non nativi.

Parametri di regolazione dei server di coda

I seguenti PARAM se definiti nell'ambiente TACL di un server di coda, possono essere utilizzati per sostituire i valori predefiniti integrati del server di coda per varie operazioni di manutenzione.

MQQSHKEEPINT

Se questo PARAM viene impostato, è possibile specificare un valore numerico, in secondi, per sostituire l'intervallo di manutenzione predefinito (60s) del server di coda. L'intervallo di manutenzione controlla la frequenza con cui il server di coda controlla le code per rilevare la presenza di eventuali messaggi scaduti ed esamina l'utilizzo della memoria al fine di ottimizzare le operazioni.

MQQSSIGTIMEOUT

Se questo PARAM viene impostato, è possibile impostare un valore numerico, in secondi, per sostituire il timeout predefinito (60s) per il recapito di un IPC di segnale ad un'applicazione che ha avviato un MQGET con l'opzione MQGMO_SET_SIGNAL. Se un server di coda non è in grado di recapitare il segnale entro questo timeout (dopo che sono state soddisfatte le condizioni richieste per generare il messaggio) il server di coda registra questo evento ed annulla quindi il segnale.

MQQSMAXBATCHEXP

Se questo PARAM viene impostato, è possibile specificare un valore numerico per sostituire il numero massimo predefinito di messaggi permanenti scaduti (100) che verranno eliminati in una singola transazione durante l'esecuzione delle operazioni di manutenzione da parte di un server di coda. Quando i messaggi permanenti scadono, bisogna rimuoverli fisicamente dai database delle code, operazione che richiede una transazione TM/MP interna. Questo PARAM consente di controllare il numero massimo di messaggi che verranno rimossi in una singola transazione TM/MP.

MQQSMAXMSGSEXPIRE

Se questo PARAM viene impostato, è possibile specificare un valore numerico per sostituire il numero massimo predefinito di messaggi scaduti (300) che verranno rilevati ed eliminati nel corso di una singola istanza di manutenzione di un server di coda.

variabili di ambiente TACL

Appendice D. Confronto tra gruppi di comandi

Le sezioni comprese tra le sezioni Tabella 16 e Tabella 21 a pagina 331 riportano il confronto tra le funzioni proprie di ciascun gruppo di comandi di gestione:

Nota: Sono riportati solo i comandi MQSC validi per MQSeries for Compaq NSK.

Tabella 16. Comandi per la gestione di Queue Manager

PCF	MQSC	Controllo
Modifica Queue Manager (Change Queue Manager)	ALTER QMGR	–
Crea Queue Manager (Create queue manager)*	CREATE QMGR	crtmqm
Elimina Queue Manager (Delete queue manager)*	DELETE QMGR	dltmqm
Interroga Queue Manager (Inquire Queue Manager)	DISPLAY QMGR	–
Arresta Queue Manager (Stop queue manager)*	STOP QMGR	endmqm
Ping Queue Manager	PING QMGR	–
Avvia Queue Manager (Start queue manager)*	START QMGR	strmqm
Nota: * non disponibile tra i comandi PCF		

Tabella 17. Comandi per la gestione di server di comandi

Descrizione	Controllo
Visualizza server di comandi (Display command server)	display command server
Arresta server di comandi (Stop command server)	stop command server
Avvia server di comandi (Start command server)	start command server
Nota: Come alternativa dei comandi di controllo, utilizzare i comandi PATHCOM, come descritto nella sezione "Gestione di TS/MP (PATHWAY)" a pagina 32. In questo gruppo non esistono equivalenti MQSC o PCF dei comandi.	

Confronto tra gruppi di comandi

Tabella 18. Comandi per la gestione di code

PCF	MQSC
Change Queue	ALTER QLOCAL ALTER QALIAS ALTER QMODEL ALTER QREMOTE
Clear Queue	CLEAR QLOCAL
Copy Queue	DEFINE QLOCAL(x) LIKE(y) DEFINE QALIAS(x) LIKE(y) DEFINE QMODEL(x) LIKE(y) DEFINE QREMOTE(x) LIKE(y)
Create Queue	DEFINE QLOCAL DEFINE QALIAS DEFINE QMODEL DEFINE QREMOTE
Delete Queue	DELETE QLOCAL DELETE QALIAS DELETE QMODEL DELETE QREMOTE
Inquire Queue	DISPLAY QUEUE
Inquire Queue Names	DISPLAY QUEUE
Nota: Non ci sono comandi di controllo per queste funzioni.	

Tabella 19. Comandi per la gestione di processi

PCF	MQSC
Change Process	ALTER PROCESS
Copy Process	DEFINE PROCESS(x) LIKE(y)
Create Process	DEFINE PROCESS
Delete Process	DELETE PROCESS
Inquire Process	DISPLAY PROCESS
Inquire Process Names	DISPLAY PROCESS
Nota: Non ci sono comandi di controllo per queste funzioni.	

Confronto tra gruppi di comandi

Tabella 20. Comandi per la gestione di canali

PCF	MQSC	Controllo
Change Channel	ALTER CHANNEL	–
Copy Channel	DEFINE CHANNEL(x) LIKE(y)	–
Create Channel	DEFINE CHANNEL	–
Delete Channel	DELETE CHANNEL	–
Inquire Channel	DISPLAY CHANNEL	–
Inquire Channel Names	DISPLAY CHANNEL	–
Inquire Channel Status	DISPLAY CHSTATUS	–
Ping Channel	PING CHANNEL	–
Reset Channel	RESET CHANNEL	–
Resolve Channel	RESOLVE CHANNEL	–
Start Channel	START CHANNEL	runmqchl
Start Channel Initiator	–	runmqchi
Start Channel Listener	–	runmqslr
Stop Channel	STOP CHANNEL	–

Nota: In MQSeries for Compaq NSK, utilizzare TS/MP o il comando **runmqslr** per avviare i listener del canale TCP/IP. Per ulteriori informazioni, consultare “Specificazione e controllo dei listener TCP/IP” a pagina 32 e “runmqslr (Run listener)” a pagina 296.

Tabella 21. Altri comandi di controllo

Descrizione	Controllo
Modifica il volume coda, le opzioni di memorizzazione messaggi del queue server	chtagfl
Aggiunge, cancella o modifica dei principal MQSeries	altmqusr
Utility di manutenzione RDF	cleanrdf
Converte la tabella di definizione canale del client	cvclchl
Crea un'uscita di conversione MQSeries (Create MQSeries conversion exit)	convexit
Visualizza autorizzazione (Display authority)	dspmqaat
Visualizza i file utilizzati dagli oggetti; opzioni di memorizzazione messaggi configurati per un oggetto	dspmqfls
Visualizza l'output di traccia formattato MQSeries (Display MQSeries formatted trace output)	dspmqtrc
Visualizza i principal MQSeries	dspmqusr
Arresta traccia MQSeries (End MQSeries trace)	endmqtrc
Installa MQSeries for Compaq NSK	instmqm
Esegue l'handler di coda messaggi non recapitati	runmqdlq
Esegui comandi MQSC (Run MQSC commands)	runmqsc
Esegue trigger monitor (Run trigger monitor)	runmqtrm
Imposta o reimposta l'autorizzazione (Set or reset authority)	setmqaut
Avvia traccia MQSeries (Start MQSeries trace)	strmqtrc
Aggiorna il queue manager V2201	upgmqm

Nota: Alternativa del comando di controllo **runmqtrm**, è possibile utilizzare i comandi PATHCOM, come descritto nella sezione “Gestione di TS/MP (PATHWAY)” a pagina 32. In questo gruppo non esistono equivalenti MQSC o PCF dei comandi.

Confronto tra gruppi di comandi

Appendice E. Arresto e rimozione manuale dei queue manager

Se i metodi normali per l'arresto e la rimozione dei queue manager non riescono, è possibile ricorrere a metodi più drastici descritti di seguito.

Arresto manuale di queue manager

Il metodo normale di arresto dei queue manager, utilizzando il comando **endmqm** funziona anche in caso di malfunzionamenti nell'ambito del queue manager. In circostanze eccezionali, se questo metodo di arresto del queue manager non riesce, utilizzare la seguente procedura per arrestarlo manualmente:

1. Ricercare gli ID di processo dei programmi del queue manager ancora in esecuzione.
2. FUP LISTOPENS nel file TRACEOPT nel volume secondario di dati del queue manager restituisce CPU, PIN di processi appartenenti al queue manager.
3. Terminare i processi queue manager ancora in esecuzione. Utilizzare il comando **STOP**, con gli ID di processo illustrati nella fase precedente.

Terminare i processi nel seguente ordine:

- a. MQECBOSS – EC Boss
- b. MQEC – ECs
- c. Ogni altro processo ancora in esecuzione

Nota: L'arresto manuale del queue manager può risultare nei FFST presi e la produzione di file FD. Questo non può essere considerato un problema del queue manager.

Il queue manager deve essere riavviato normalmente, anche se è stato terminato utilizzando il metodo precedente.

Se si desidera cancellare il queue manager dopo averlo arrestato manualmente, utilizzare il comando **dltmqm** regolare. Se, per altre ragioni, il comando non riesce a cancellare il queue manager, può essere utilizzato il processo manuale illustrato in "Rimozione manuale dei queue manager".

Rimozione manuale dei queue manager

Per rimuovere i queue manager manualmente:

1. Accertarsi che non vi siano processi del queue manager in esecuzione per il queue manager che si desidera rimuovere.
2. Modificare il file MQSINI per eliminare la stanza queue manager e se necessario, modificare la stanza queue manager predefinita. Notare l'ubicazione dei file queue manager prima di cancellare la stanza.
3. Cancellare tutti i file nei volumi secondari del queue manager utilizzando il comando FUP PURGE. Ad esempio, FUP PURGE \$VOL.QMSVOL*.*.

Rimozione manuale dei queue manager

Appendice F. MQSeries e Compaq NonStop Server for Java

MQSeries for Compaq NSK è compatibile con Compaq NonStop Server for Java, Version 1.5 e successive. Il prodotto supporta tutte le classi Java di MQSeries. Esse sono dettagliatamente descritte nel manuale *MQSeries Using Java*.

E' possibile utilizzare MQSeries con NonStop Server for Java in due modi:

- Come servlet in esecuzione nel contesto di Compaq iTP Webserver
- Come applicazione Java in esecuzione direttamente dalla riga comandi

Compaq NonStop Server for Java 1.5 Reference fornisce informazioni sull'utilizzo dei servlet.

Per accedere a MQSeries da Java, utilizzando i servlet o le applicazioni, è necessario creare una versione personalizzata di NonStop JVM che fornisce collegamenti con le seguenti librerie MQSeries:

- Il prodotto SRL MQSRLLIB, dal volume secondario Guardian ZMQSEXE
- Archivio Java libMQSESSION.a da /opt/mqm/lib

La documentazione Java di Compaq fornisce le informazioni sulle modalità di ricollegamento del JVM. La directory di esempio MQSeries (/opt/mqm/samp) contiene due file make di esempio:

MakeJVM.smp

E' un file make di esempio che illustra come ricreare NonStop JVM per fornire accesso a MQSeries. Per utilizzare questo file make, modificarlo in modo da riflettere l'ubicazione di installazione delle librerie MQSeries.

MakeJava.smp

Un file make di esempio che illustra come creare un'applicazione Java.

Considerazioni transazionali

Il linguaggio Java non fornisce l'accesso diretto all'interfaccia TM/MP. MQSeries supporta le transazioni in Java attraverso la *Classe corrente* JTS. La tabella riportata di seguito descrive l'interfaccia *corrente*.

Tabella 22. *Interfaccia con linguaggio Java*

Funzione	Descrizione
begin()	Avvia una nuova transazione e la associa al thread di chiamata.
commit(boolean)	Esegue il commit della transazione associata al thread di chiamata.
get_control()	Richiama un oggetto Control che rappresenta la transazione associata al thread di chiamata.
get_status()	Richiama lo stato della transazione associata al thread di chiamata.
get_transaction_name()	Richiama un nome descrittivo della transazione associata al thread di chiamata.
resume(ControlRef)	Imposta o riprende l'associazione di una transazione con il thread di chiamata.

Tabella 22. Interfaccia con linguaggio Java (Continua)

Funzione	Descrizione
rollback()	Esegue il rollback della transazione associata al thread di chiamata.
suspend()	Sospendere l'associazione dei thread di chiamata con un contesto di transazione.

Per ulteriori informazioni sui JTS, fare riferimento alla documentazione Compaq NSK NonStop Java o alla documentazione JDK.

Appendice G. MQSC supportato da MQSeries for Compaq NSK

Quest'appendice elenca i comandi MQSeries (MQSC) supportati da MQSeries for Compaq NSK. Per informazioni sulla sintassi di questi comandi, consultare: *MQSeries Guida di riferimento per i comandi*.

Tabella 23. MQSC supportato da MQSeries for Compaq NSK

Comando	Descrizione	Come descritto in MQSeries Command Reference
ALTER CHANNEL	Modifica attributi di canale	Sì, ma con le eccezioni descritte in: "CONNAME" a pagina 340.
ALTER NAMELIST	Modifica un elenco di nomi	Sì
ALTER PROCESS	Modifica gli attributi di proprietà	Sì
ALTER QALIAS	Modifica gli attributi di una coda alias	Sì
ALTER QLOCAL	Modifica gli attributi di una coda locale.	Sì, ma con le eccezioni descritte in: "HARDENBO e NOHARDENBO" a pagina 340.
ALTER QMGR	Modifica gli attributi del queue manager.	Sì, ma con le eccezioni descritte in: "MAXUMSGS e MAXHANDS" a pagina 339.
ALTER QMODEL	Modifica gli attributi di una coda modello.	Sì, ma con le eccezioni descritte in: "HARDENBO e NOHARDENBO" a pagina 340.
ALTER QREMOTE	Modifica gli attributi di una definizione locale di una coda remota, un alias queue-manager, o un alias di coda di risposta.	Sì
CLEAR QLOCAL	Elimina i messaggi da una coda locale.	Sì
DEFINE CHANNEL	Crea una definizione di canale.	Sì, ma con le eccezioni descritte in: "CONNAME" a pagina 340.
DEFINE NAMELIST	Definisce un elenco di nomi.	Sì
DEFINE PROCESS	Crea una definizione di processo.	Sì
DEFINE QALIAS	Crea una definizione di coda alias.	Sì
DEFINE QLOCAL	Crea una definizione di coda locale.	Sì, ma con le eccezioni descritte in: "HARDENBO e NOHARDENBO" a pagina 340.

Comandi MQSeries

Tabella 23. MQSC supportato da MQSeries for Compaq NSK (Continua)

Comando	Descrizione	Come descritto in MQSeries Command Reference
DEFINE QMODEL	Crea una definizione di coda modello.	Sì, ma con le eccezioni descritte in: "HARDENBO e NOHARDENBO" a pagina 340.
DEFINE QREMOTE	Crea una definizione locale di una coda remota, un alias di queue-manager o un alias di coda di risposta.	Sì
DELETE CHANNEL	Cancella una definizione di canale.	Sì
DELETE NAMELIST	Cancella un elenco di nomi.	Sì
DELETE PROCESS	Cancella una definizione di processo.	Sì
DELETE QALIAS	Cancella una definizione di coda alias.	Sì
DELETE QLOCAL	Cancella una definizione di coda locale.	Sì
DELETE QMODEL	Cancella una definizione di coda modello.	Sì
DELETE QREMOTE	Cancella una definizione locale di una coda remota.	Sì
DISPLAY CHANNEL	Visualizza una definizione di canale.	Sì, ma con le eccezioni descritte in: "CONNNAME" a pagina 340.
DISPLAY CHSTATUS	Visualizza lo stato di uno o più canali.	Sì, ma con le eccezioni descritte in "Informazioni sullo stato del canale (DISPLAY CHSTATUS)" a pagina 339 e "CONNNAME" a pagina 340.
DISPLAY CLUSQMGR	Visualizza lo stato di uno o più canali.	Sì, ma con le eccezioni descritte in: "CONNNAME" a pagina 340.
DISPLAY NAMELIST	Visualizza un elenco di nomi.	Sì
DISPLAY PROCESS	Visualizza una definizione di processo.	Sì
DISPLAY QMGR	Visualizza gli attributi queue-manager.	Sì, ma con le eccezioni descritte in: "MAXUMSGS e MAXHANDS" a pagina 339.
DISPLAY QUEUE	Visualizza gli attributi della coda.	Sì, ma con le eccezioni descritte in: "HARDENBO e NOHARDENBO" a pagina 340.
PING CHANNEL	Esegue il test di una canale.	Sì
PING QMGR	Esegue il test se un queue manager risponde ai comandi.	Sì

Tabella 23. MQSC supportato da MQSeries for Compaq NSK (Continua)

Comando	Descrizione	Come descritto in MQSeries Command Reference
REFRESH CLUSTER	Elimina tutte le informazioni cluster locali e ne esegue il rebuilt.	Sì
RESET CHANNEL	Ripristina il numero della sequenza messaggi per un canale.	Sì
RESET CLUSTER	Esegue operazioni speciali sui cluster.	Sì
RESOLVE CHANNEL	Risolve i messaggi in sospeso sul Sender o sul canale del server.	Sì
RESUME QMGR	Informa gli altri queue manager di un cluster che il queue manager locale è nuovamente disponibile per l'elaborazione e può ricevere messaggi.	Sì
START CHANNEL	Avvia un canale.	Sì
STOP CHANNEL	Arresta una canale.	Sì
SUSPEND QMGR	Informa gli altri queue manager di un cluster che il queue manager locale non è disponibile per l'elaborazione e non può ricevere messaggi.	Sì

Se si esegue il build dei comandi MQSC in uno script, ogni riga può comprendere massimo 72 caratteri.

Attributi di MQSC

Questa sezione fornisce informazioni relative agli attributi MQSC specifiche per MQSeries for Compaq NSK.

Informazioni sullo stato del canale (DISPLAY CHSTATUS)

Il comando DISPLAY CHSTATUS viene eseguito come descritto nella sezione *MQSeries Guida di riferimento per i comandi*, a eccezione dell'aggiornamento dello stato del canale che viene eseguito solo nell'ambito dell'elaborazione batch. Le informazioni sullo stato del canale non vengono aggiornate per ciascun trasferimento del messaggio a causa dell'impatto potenziale sulle prestazioni dei canali. Quindi i valori dei dati relativi agli stati comuni sono identici per entrambe le serie, correnti e salvate.

MAXUMSGS e MAXHANDS

Gli attributi oggetto del queue manager MAXUMSGS e MAXHANDS vengono ignorati. Eseguire i seguenti comandi:

```
ALTER QMGR
DISPLAY QMGR
```

HARDENBO e NOHARDENBO

HARDENBO e NOHARDENBO

In MQSeries for Compaq NSK, gli attributi della coda modello e della coda locale HARDENBO e NOHARDENBO vengono ignorati. Per i messaggi permanenti, il viene sempre eseguito l'harden del *backoutcount* e non viene mai eseguito per i messaggi non permanenti. Eseguire i seguenti comandi:

```
ALTER QLOCAL
ALTER QMODEL
DEFINE QLOCAL
DEFINE QMODEL
DISPLAY QUEUE
```

CONNAME

L'attributo CONNAME dei canali TCP può facoltativamente utilizzare un campo aggiuntivo all'inizio del valore, specificando il nome di un processo Guardian TCP/IP Server specifico da utilizzare per il canale. Eseguire i seguenti comandi:

```
ALTER CHANNEL
DEFINE CHANNEL
DISPLAY CHANNEL
DISPLAY CHSTATUS
DISPLAY CLUSQMGR
```

USERDATA per i programmi con la funzione trigger

I dati inoltrati al monitor trigger attraverso l'attributo USERDATA di MQSC DEFINE PROCESS o ALTER PROCESS devono essere compresi tra doppi apici se si tratta di una stringa contenente spazi. Ad esempio, se l'attributo USERDATA -o \$DISK.VOLUME.PROGRAM -d 1 deve essere inoltrato al monitor trigger, nell'input di MQSC è necessario specificarlo tra doppi apici, come segue:

```
'" -o $DISK.VOLUME.PROGRAM -d 1"'
```

Se la definizione del processo viene riportato attraverso MQSC, verrà visualizzata come segue:

```
"-o $DISK.VOLUME.PROGRAM -d 1"
```

Utilizzo dei nomi di uscita come attributi di oggetti

Quando i nomi di uscita sono specificati negli attributi degli oggetti, saranno in un formato specifico per MQSeries for Compaq NSK.

Appendice H. Application Programming Reference

Le seguenti sezioni sono nuove per MQSeries for Compaq NSK e vanno utilizzate insieme a: *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*.

Tipi di dati di struttura

Questa sezione descrive le modifiche apportate ai tipi di dati.

Tipo di dati di struttura	Supportata nella V2.x	Supportato nella V5.1	Funziona come descritto in: <i>MQSeries Application Programming Reference</i>
MQBO – Begin Options	No	No	
MQCH – CICS Bridge Header	No	Sì	Sì
MQCNO – Connect Options	No	Sì	Sì, na con alcune note aggiuntive. Consultare la sezione “MQCNO – Connect Options” a pagina 342 per ulteriori informazioni.
MQDH – Distribution Header	No	Sì	Sì
MQDLH – Dead Letter Header	Sì	Sì	Sì
MQGMO – Get Message Options	Sì	Sì	Sì, na con alcune note aggiuntive. Consultare la sezione “MQGMO – Get Message Options” a pagina 342 per ulteriori informazioni.
MQIH – IMS Bridge Header	Sì	Sì	Sì
MQMD – Message Descriptor	Sì	Sì	Sì, na con alcune note aggiuntive. Consultare la sezione “MQMD – Message Descriptor” a pagina 343 per ulteriori informazioni.
MQMDE – Message Descriptor Extension	No	Sì	Sì
MQOD – object Descriptor	Sì	Sì	Sì
MQOR – Object Record	No	Sì	Sì
MQPMO – Put Message Options	Sì	Sì	Sì, na con alcune note aggiuntive. Consultare la sezione “MQPMO – Put Message Options” a pagina 343 per ulteriori informazioni.
MQPMR – Put Message Record	No	Sì	Sì

Tipi di dati di struttura

Tipo di dati di struttura	Supportata nella V2.x	Supportato nella V5.1	Funziona come descritto in: <i>MQSeries Application Programming Reference</i>
MQRMH – Message Reference Header	No	Sì	Sì
MQRR – Response Record	No	Sì	Sì
MQTM – Trigger Message	Sì	Sì	Sì
MQTMC2 – Trigger Message Character Format	Sì	Sì	Sì
MQWIH – Workload Information Header	No	Sì	Sì
MQXQH – Transmission Queue Header	Sì	Sì	Sì

Questa sezione descrive i seguenti tipi di dati di struttura MQSeries:

MQCNO – Connect Options

La struttura di dati MQCNO è uguale a quella specificata in: *MQSeries for Tandem Non Stop Kernal, V5.1 Guida operativa* con le seguenti note aggiuntive:

- L'unità di esecuzione è definita come un processo
- MQCNO_FASTPATH_BINDING può essere utilizzato solo in un processo che dispone di una connessione singola ad un queue manager
- MQCNO_FASTPATH_BINDING richiede che l'applicazione venga eseguita sotto l'ID utente, appartenente al gruppo di utenti di gestione MQM, che ha creato il queue manager
- Il parametro Guardian MQCONNECTTYPE può essere utilizzato in associazione al tipo di bind specificato dal campo *Options*, per controllare il tipo di binding utilizzato. Se questo parametro viene specificato, deve avere il valore **FASTPATH** oppure **STANDARD**; se è impostato su un valore diverso, viene ignorato. Il valore del parametro è sensibile al maiuscolo/minuscolo.
- MQSeries for Compaq NSK supporta sia MCNO_VERSION_2 che MQCNO_VERSION_1 ma i campi *ClientConnOffset* e *ClientConnPtr* vengono ignorati.

MQGMO – Get Message Options

La struttura MQGMO è un parametro di immissione/emissione della chiamata MGET. Notare le seguenti informazioni relative alle opzioni MQGMO_SET_SIGNAL, MQGMO_WAIT, MQGMO_SYNCPOINT e MQGMO_NO_SYNCPOINT in *MQSeries for Compaq NSK*:

- Se si desidera che l'applicazione proceda a svolgere altre operazioni mentre attende l'arrivo di un messaggio, è possibile utilizzare l'opzione di segnalazione MQGMO_SET_SIGNAL invece di MQGMO_WAIT. Tuttavia, l'opzione di segnalazione è specifica per l'ambiente e non può essere utilizzata da applicazioni utilizzate in vari ambienti.
- Se esiste più di una chiamata MQGET in attesa per lo stesso messaggio con una varietà di opzioni di attesa e di segnalazione, ciascuna chiamata in attesa viene considerata allo stesso modo. E' un errore specificare MQGMO_SET_SIGNAL

con MQGMO_WAIT. E' anche un errore specificare questa opzione con un handle di coda per cui è in sospeso un segnale.

- Se un'applicazione specifica MQGET con MQGMO_SET_SIGNAL ed un WaitInterval di 0, l'opzione MQGMO_SET_SIGNAL verrà ignorata ed elaborata come una MQGET con MQGMO_NO_WAIT.

Questo significa che un'applicazione deve essere preparata a ricevere MQRC_NO_MSG_AVAILABLE su una MQGET con MQGMO_SET_SIGNAL se il WaitInterval è pari a zero. Le applicazioni ricevono un segnale IPC solo se:

- L'applicazione riceve MQRC_SIGNAL_REQUEST_ACCEPTED da MQGET (indica che un segnale è stato *inoltrato*)
- l'applicazione ha potuto elaborare il messaggio di sistema file_open_ ed accettare l'IPC di segnalazione entro il timeout de server di coda per il recapito della segnalazione. Per impostazione predefinita, questo valore è pari a 60s, ma può essere sostituito per un server di coda specificando il PARAM MQQSSIGTIMEOUT nell'ambiente del server di coda.

Il queue manager registra la condizione di impossibilità di recapitare un messaggio IPC ad un'applicazione se non è stato in grado di aprire il processo e di inviare l'IPC prima della scadenza del timeout. A questo punto il queue manager non tenterà nuovamente il recapito. Le applicazioni dovrebbero essere in grado di evitare situazioni di stallo simili non attendendo indefinitamente per un segnale IPC.

- MQGMO_SYNCPOINT_IF_PERSISTENT è adesso supportato.
- Se non viene impostata l'opzione MQGMO_SYNCPOINT o l'opzione MQGMO_NO_SYNCPOINT, MQSeries for Compaq NSK assume come valore predefinito MQGMO_SYNCPOINT.
- MQSeries for Compaq NSK non supporta il campo *MsgToken*.

MQMD – Message Descriptor

La struttura MQMD contiene le informazioni di controllo che descrivono un messaggio. Notare le seguenti informazioni:

- *BackoutCount* funziona come descritto in: *MQSeries for Tandem Non Stop Kernal, V5.1 Guida operativa*. Questo è un conteggio di quante volte il messaggio è stato in precedenza restituito dalla chiamata MQGET come parte di un'unità di lavoro (UOW) e ne è stato successivamente eseguito il backout. Ciò consente alle applicazioni di rilevare più facilmente eventuali errori di elaborazione basati sul contenuto del messaggio. Nella Versione 2.2.0.1, il valore di BackoutCount era stimato.
- In MQSeries for Compaq NSK, l'eliminazione di un messaggio (e la generazione di un prospetto, se necessario) non viene eseguita durante una chiamata MQGET ma è sotto il controllo del server di coda che la esegue su base periodica, sulla base delle impostazioni per il queue manager.
- Il valore del campo UserIdentifier, quando impostato dal queue manager durante un'operazione MQPUT o MQPUT1 è il nome del principal di MQSeries che si trova nel database dei principal del queue manager corrispondente all'effettivo ID utente dell'applicazione.

MQPMO – Put Message Options

La struttura MQPMO è un parametro di immissione/emissione delle chiamate MQPUT e MQPUT1. Notare le seguenti informazioni sull'opzione MQPMO_NO_SYNCPOINT in MQSeries for Compaq NSK:

Tipi di dati di struttura

- Se non vengono impostate l'opzione MQPMO_SYNCPOINT o l'opzione MQPMO_NO_SYNCPOINT, MQSeries for Compaq NSK assume automaticamente il valore predefinito di MQPMO_SYNCPOINT.

Chiamate MQI

Questa sezione descrive le modifiche apportate alle seguenti chiamate MQI:

Descrizione chiamata MQI	Supportata nella V2.x	Supportato nella V5.1	Funziona come descritto in: <i>MQSeries Application Programming Reference</i>
MQBACK – Esegui backout modifiche	Restituisce l'errore ^{1 a} pagina 345	Restituisce l'errore ^{1 a} pagina 345	Si
MQBEGIN – Avvia unità di lavoro	No	Restituisce l'errore ^{1 a} pagina 345	Si
MQCLOSE – Chiudi oggetto	Si	Si	Si, na con alcune note aggiuntive. Consultare la sezione "MQCLOSE – Chiudi oggetto" a pagina 345 per ulteriori informazioni.
MQCMIT – Esegui commit delle modifiche	No	Restituisce l'errore ^{1 a} pagina 345	Si
MQCONN – Connetti il queue manager	Si	Si	Si
MQCONNX – Connetti il Queue manager (Esteso)	No	Si	Si
MQDISC – Disconnetti il queue manager	Si	Si	Si, na con alcune note aggiuntive. Consultare la sezione "MQDISC – Disconnetti il queue manager" a pagina 345 per ulteriori informazioni.
MQGET – Estrai messaggio	Si	Si	Si
MQINQ – Interroga attributi oggetto	Si	Si	Si, na con alcune note aggiuntive. Consultare la sezione "MQINQ – Interroga attributi oggetto" a pagina 346 per ulteriori informazioni.
MQOPEN – Apri oggetto	Si	Si	Si, na con alcune note aggiuntive. Consultare la sezione "MQOPEN – Apri oggetto" a pagina 346 per ulteriori informazioni.
MQPUT – Inserisci messaggio	Si	Si	Si
MQPUT1 – Inserisci un messaggio	Si	Si	Si

Descrizione chiamata MQI	Supportata nella V2.x	Supportato nella V5.1	Funziona come descritto in: <i>MQSeries Application Programming Reference</i>
MQSET – Imposta attributi oggetto	Sì	Sì	Sì, ma con alcune note aggiuntive. Consultare la sezione “MQSET– Imposta attributi oggetto” a pagina 346 per ulteriori informazioni.
MQSYNC – Sincronizza aggiornamenti statistiche	Restituisce l’errore ²	Restituisce l’errore ²	Sì

Note:

1. La chiamata MQI può essere emessa dall’applicazione ma restituisce sempre il codice di completamento MQCC_FAILED ed il codice ragione MQRC_ENVIRONMENT_ERROR.
2. Questa chiamata restituisce un *codice completamento* di MQCC_OK ed un codice ragione di MQRC_NONE.

MQCLOSE – Chiudi oggetto

La chiamata MQCLOSE, che è l’opposto della chiamata MQOPEN, rilascia l’accesso ad un oggetto.

In MQSeries for Compaq NSK, se esiste una richiesta MQGET con l’opzione MQGMO_SET_SIGNAL attiva sull’handle di coda in chiusura, la richiesta viene annullata. Le richieste di segnalazione per la stessa coda ma indirizzate a handle diversi (*Hobj*) non sono influenzate (tranne nel caso in cui si tratti di una coda dinamica in fase di cancellazione; in questo caso, anch’esse vengono annullate).

Per un’applicazione FASTPATH che sta aprendo o chiudendo una coda dinamica, MQSeries può avviare e chiudere una transazione TM/MP al fine di aggiornare i database controllati. Se l’applicazione ha aperto il T-file TM/MP (poiché può avviare più transazioni), ENDTRANSACTION è un’operazione per cui non si attende e l’applicazione riceverà un completamento per la transazione avviata da MQSeries. Bisogna controllare la struttura e la modalità di funzionamento delle applicazioni al fine di determinare se questo è il caso e verificare che i completamenti di gestione logici siano in grado di gestire i completamenti ENDTRANSACTION causati da MQSeries.

MQDISC – Disconnetti il queue manager

La chiamata MQDISC, che è l’opposto di MQCONN, interrompe la connessione tra il Queue Manager di MQSeries ed il programma applicativo.

La nota di utilizzo 3 in *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa* non è corretta. In MQSeries for Compaq NSK un punto di sincronizzazione (syncpoint) implicito non si verifica se un’unità di lavoro coordinata dal queue manager è in corso quando viene eseguita la chiamata di MQDISC. Ciò è dovuto al fatto che il queue manager NSK non può essere un coordinatore di un’unità di lavoro (Unit Of Work). La coordinazione è fornita dal sottosistema TM/MP.

Chiamate MQI

MQINQ – Interroga attributi oggetto

La chiamata MQPUT restituisce una schiera di valori interi ed un gruppo di stringhe di caratteri che contengono gli attributi di un oggetto.

MQOPEN – Apri oggetto

La chiamata MQOPEN stabilisce l'accesso ad un oggetto. In MQSeries for Compaq NSK, l'attributo *MaxHandles* del queue manager viene ignorato.

Per un'applicazione FASTPATH che sta aprendo o chiudendo una coda dinamica, MQSeries può avviare e chiudere una transazione TM/MP al fine di aggiornare i database controllati. Se l'applicazione ha aperto il T-file TM/MP (poiché può avviare più transazioni), ENDTRANSACTION è un'operazione per cui non si attende e l'applicazione riceverà un completamento per la transazione avviata da MQSeries. Bisogna controllare la struttura e la modalità di funzionamento delle applicazioni al fine di determinare se questo è il caso e verificare che i completamenti di gestione logici siano in grado di gestire i completamenti ENDTRANSACTION causati da MQSeries.

MQSET– Imposta attributi oggetto

La chiamata MQSET modifica gli attributi di un oggetto rappresentato da un handle. L'oggetto deve essere una coda. In MQSeries for Compaq NSK, è supportato il selettore MQIA_DIST_LISTS.

Per un'applicazione FASTPATH che sta modificando gli attributi di una coda utilizzando MQSET, MQSeries avvierà e chiuderà una transazione TM/MP al fine di aggiornare i database controllati. Se l'applicazione ha aperto il T-file TM/MP (poiché può avviare più transazioni), ENDTRANSACTION è un'operazione per cui non si attende e l'applicazione riceverà un completamento per la transazione avviata da MQSeries. Bisogna controllare la struttura e la modalità di funzionamento delle applicazioni al fine di determinare se questo è il caso e verificare che i completamenti di gestione logici siano in grado di gestire i completamenti ENDTRANSACTION causati da MQSeries.

Attributi degli oggetti MQSeries

In MQSeries for Compaq NSK, gli attributi di tutti gli oggetti sono come descritti in *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*, con le seguenti eccezioni ed aggiunte.

Attributi per tutte le code

In MQSeries for Compaq NSK, gli attributi di tutte le code sono come descritti in *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*, con le seguenti eccezioni ed aggiunte.

Gli attributi AlterationDate e AlterationTime sono aggiornati solo quando vengono eseguite modifiche amministrative agli attributi di un oggetto. Gli attributi CurrentQDepth, OpenInputCount e OpenOutputCount possono essere modificati solo dinamicamente. Gli attributi QDepthHighCount, QDepthLowEvent, QDepthMaxEvent e QServiceIntervalEvent possono essere modificati sia dinamicamente che amministrativamente, ma solo le modifiche amministrative (come quelle eseguita utilizzando i comandi MQSC o via MQSET) determineranno delle modifiche negli attributi AlterationDate e AlterationTimes.

Attributi delle code locali e modello

In MQSeries for Compaq NSK,

- L'attributo *Archive* viene ignorato.
- L'attributo *HardenGetBackout* viene ignorato poiché il conteggio di backout non viene salvato su disco. Non è possibile archiviare i messaggi.
- Per i messaggi permanenti, viene sempre eseguito l'harden dell'attributo *BackoutCount*. Per i messaggi non permanenti, non viene mai eseguito l'harden dell'attributo *BackoutCount*. Se tuttavia la coda locale ha il proprio attributo *-q* server C option impostato, verrà eseguito il checkpoint di *BackoutCount* sul queue server di backup. I messaggi di cui viene eseguito il checkpoint in questo modo possono resistere a condizioni di malfunzionamento del queue server. Per conservare la compatibilità con le altre piattaforme MQSeries, la chiamata MQINQ può eseguire una query dell'attributo utilizzando il selettore MQIA_HARDEN_GET_BACKOUT.

Attributi dei queue manager

- *MaxMsgLength* è 100 MB.
- *CommandLevel* è MQCMDL_LEVEL_510.
- *SyncPoint* è MQSP_AVAILABLE.
- Il valore di *CodedCharSetId* è come specificato quando è stata creata l'istanza del queue manager.
- L'attributo *MaxHandles* viene ignorato. Non è possibile specificare un numero massimo di handle aperti per MQSeries for Compaq NSK. Il valore massimo verrà determinato dai limiti delle risorse di sistema.
- L'attributo *MaxUncommittedMsgs* viene ignorato. Non è possibile specificare un numero massimo di messaggi da consentire in una singola unità di lavoro. Il valore massimo viene determinato dai limiti delle risorse.
- E' possibile modificare il CCSID.

Conversione dei dati

Consultare: "Appendice L. User exit" a pagina 367 che descrive lo schema per il supporto di tutte le uscite in MQSeries for Compaq NSK V5.1. Il meccanismo è stato modificato rispetto alle precedenti versioni pr supportare un'implementazione delle uscite più congruente e flessibile.

Conversione dei dati

Appendice I. Creazione ed esecuzione delle applicazioni

I programmi di esempio e gli script di binding e di compilazione di esempio, forniti nel volume secondario ZMQSSMPL, illustrano le principali funzioni di MQI in MQSeries for Compaq NSK, e dimostrano come eseguire la compilazione e l'esecuzione del bind di un'applicazione.

Scrittura delle applicazioni

Questa sezione fornisce le informazioni aggiornate per alcune piccole differenze tra l'interfaccia di MQI Version 5.1 standard, come documentato in *MQSeries Application Programming Guide* e l'interfaccia MQI per MQSeries for Compaq NSK. Consultare questa sezione per aggiornare *MQSeries Application Programming Guide* per MQSeries for Compaq NSK V5.1.

Utilizzo dell'intervallo di attesa MQGET e del canale DISCINT e HBINT

Durante l'esecuzione di MQGET tramite MQGMO_NO_SYNCPOINT, una transazione TM/MP viene avviata e terminata da MQSeries solo quando un messaggio persistente è disponibile e soddisfa i criteri di richiamo. Non è necessaria alcuna considerazione al valore di WaitInterval per nessuna operazione di syncpoint.

Per un MQGET emesso con l'opzione MQGMO_SYNCPOINT o MQGMO_SYNCPOINT_IF_PERSISTENT, la transazione TM/MP è sotto il controllo dell'applicazione dell'utente che emette BEGINTRANSACTION. L'intervallo di attesa non dovrebbe superare il valore di timeout TMF Autoabort e dovrebbe essere breve per evitare il pinning di una quantità significativa della traccia di controllo TM/MP (di solito, vengono utilizzati i valori inferiori al minuto). La specifica di attesa illimitata su una coda leggermente utilizzata o una coda che è inattiva per una notte potrebbe causare il superamento del timeout di interruzione automatica e la restituzione di MQRC_UOW_CANCELLED (2297) a MQGET quando un messaggio diventa disponibile sulla coda. Un intervallo di attesa elevato o l'utilizzo illimitato può provocare il pinning delle tracce di controllo TMF, eventualmente iniziali (se non sono corrotte) al sottosistema TM/MP disabilitando le transazioni su un ampio sistema.

In modo simile, i valori DISCINT e HBINT per l'invio dei canali controllano la lunghezza di una transazione TM/MP. I canali sono in grado di eseguire il ciclo di transazioni quando lo stato inattivo contiene Disconnect Interval lunghi, senza disporre di un'influenza dannosa sulle tracce di controllo TM/MP.

Un parametro, MQTRANSACTIONLIFE, può essere utilizzato per controllare l'aggiornamento della transazione TM/MP per Disconnect Interval e Heartbeat di canale che hanno valore zero. E' utile se si desidera una durata di transazione TM/MP breve o lunga oppure per modificare la quantità di attività TM/MP prodotta dal canale inattivo. Un valore elevato produrrà un'impostazione inferiore.

Aggiungere a ciascuna classe server MQS-ECxx PATHWAY di MQSeries:

```
PARAM MQTRANSACTIONLIFE <number>
```

dove <number> indica un numero come 100.

Questo parametro ricopre l'utilizzo dell'intervallo di aggiornamento della transazione TM/MP predefinita di 10 secondi. Ad esempio, un canale, con un Disconnect Interval illimitato senza parametro, potrebbe causare un aggiornamento di 10 secondi mentre attende un messaggio in arrivo.

Gestione dell'unità di lavoro (transazione)

La gestione di transazione viene eseguita sotto il controllo del prodotto TM/MP di Compaq, piuttosto che MQSeries stesso.

Gli effetti di questa differenza sono:

- L'opzione SYNCPOINT predefinita per le chiamate MQPUT e MQGET è SYNCPOINT, piuttosto che NO_SYNCPOINT.
- Per utilizzare l'opzione predefinita (SYNCPOINT) per le operazioni MQPUT, MQGET e MQPUT1, l'applicazione deve disporre di una transazione TM/MP attiva che definisce l'unità di lavoro da assegnare. Un'applicazione inizia una transazione TM/MP chiamando la funzione BEGINTRANSACTION(). Durante l'attivazione di tale transazione, tutte le operazioni MQPUT, MQPUT1 e MQGET eseguite dall'applicazione sono interne alla stessa unità di lavoro (transazione). Qualsiasi altra operazione del database eseguita dall'applicazione è interna a questa unità di lavoro. Esistono limiti imposti dal sistema sul numero e sulla dimensione dei messaggi da poter scrivere ed annullare all'interno di una singola transazione TM/MP. Quando l'applicazione ha completato l'unità di lavoro, la transazione TM/MP viene terminata (l'unità di lavoro viene assegnata) mediante la funzione ENDTRANSACTION(). Se si verifica qualsiasi errore, l'applicazione può annullare la transazione TM/MP (backout dell'unità di lavoro) mediante la funzione ABORTTRANSACTION(). Di conseguenza, le funzioni della versione 5 standard MQCMIT(), MQBACK() e MQBEGIN() non sono supportate su tale prodotto. Se essi sono definiti, viene riportato un errore.
- Se un'applicazione utilizza l'opzione NO_SYNCPOINT per le operazioni MQPUT, MQGET e MQPUT1, MQSeries avvia una transazione TM/MP, esegue l'accodamento e assegna la transazione prima di restituirla all'applicazione. Quindi, ciascuna operazione viene eseguita nella propria unità di lavoro e, una volta completata, non può essere eseguito il back out dall'applicazione mediante TM/MP.
- E' necessario per MQSeries avviare una transazione TM/MP per un'operazione NO_SYNCPOINT solo se il messaggio persiste e viene richiesto l'aggiornamento di un file di coda protetto TM/MP.
- Non è necessario attivare una transazione TM/MP per le chiamate MQI diverse da MQGET, MQPUT e MQPUT1.
- Il codice di errore MQRC_SYNCPOINT_LIMIT_REACHED viene utilizzato da MQSeries for Compaq NSK per informare l'applicazione che è stato raggiunto il limite imposto dal sistema sul numero delle operazioni I/O all'interno di una singola transazione TM/MP. Se l'applicazione ha specificato l'opzione SYNCPOINT, essa dovrebbe eliminare la transazione (back out dell'unità di lavoro) e riprovare con un numero inferiore delle operazioni dell'unità di lavoro.
- Il codice di errore MQRC_UOW_CANCELED informa l'applicazione che l'unità di lavoro (transazione TM/MP) è stata eliminata, dal sistema stesso (TM/MP

imposta alcuni limiti per l'utilizzo delle risorse del sistema), dall'azione dell'utente o dall'iniziatore della transazione.

- Il codice di errore MQRC_BACKED_OUT informa l'applicazione che MQSeries ha eliminato l'unità di lavoro a causa di un errore o di un errore di Queue Server primario. L'applicazione dovrebbe chiamare ABORTTRANSACTION (se è stato eseguito il syncpoint sull'operazione) e riprovare.
- Il codice di errore MQRC_SYNCPOINT_NOT_AVAILABLE informa l'applicazione che MQSeries non è stato in grado di avviare o utilizzare una transazione TM/MP, richiesta per completare un'operazione. Di solito, ciò indica un problema con TM/MP e le informazioni aggiuntive potrebbero essere disponibili nel log di errore o da FFST prodotto dal queue manager.

Considerazioni sulla progettazione generale

Considerare:

- La libreria MQI (collegata al processo di applicazione) non apre \$RECEIVE e non apre \$TMP (pseudo file di transazione TM/MP), in modo tale che è possibile codificare l'applicazione per utilizzare tali funzioni.
- La libreria MQI utilizza una chiamata SERVERCLASS_SEND_() nella comunicazione iniziale con il queue manager. Durante il collegamento, esso mantiene un minimo di due aperture file di processo (con il processo LINKMON e un Local Queue Manager Agent) e un piccolo numero di aperture file disco (inferiore a 10). Le aperture di processo sono anche effettuate su qualsiasi queue server che supporta le code locali che sono aperte per l'input, output o browse come risultato di una chiamata MQOPEN.
- Prima di chiamare MQCONN, verificare che non sia indipendente nowait PATHSEND I/O. MQCONN esegue il nowait di PATHSEND I/O e non è in grado di intercettare il completamento dell'errore che ha causato l'indipendenza I/O dell'applicazione.
- Le applicazioni collegate tramite FASTPATH hanno speciali considerazioni se essi sono anche se si trattano di requester TM/MP di diversi thread (consultare le descrizioni di MQOPEN, MQCLOSE e MQSET precedenti).

Interfaccia XA

Non viene fornita alcuna interfaccia XA per il coordinamento dell'unità di lavoro. Il coordinamento dell'unità di lavoro viene eseguito da TM/MP.

MQGMO_BROWSE_* con MQGMO_LOCK

MQGMO_BROWSE_* con MQGMO_LOCK è supportato. Consultare *MQSeries for Tandem Non Stop Kernal, V5.1 Guida operativa*.

Applicazioni con la funzione trigger

In ambiente Compaq NSK, le applicazioni MQSeries con funzione trigger ricevono i dati dell'utente mediante variabili di ambiente impostate nel processo TAQL in corso. Si tratta di un limite della lunghezza dell'elenco di argomento che può essere trasferito al processo Compaq C.

Per accedere a tali informazioni, le applicazioni con funzione trigger dovrebbero contenere codici simili a quelli riportati di seguito (per ulteriori informazioni dettagliate, consultare l'esempio amqsinqa):

Scrittura delle applicazioni

```
MQTMC2 *trig;                /* struttura messaggio trigger */
MQTMC2 trigdata;            /* struttura messaggio trigger */
char    *applId;
char    *envData;
char    *usrData;
char    *qmName;

/*****
/*
/* Impostare l'argomento di programma nel messaggio trigger */
/*
/*
*****/
trig = (MQTMC2*)argv[1];    /* -> messaggio trigger */

/* richiamare le variabili di ambiente e caricare il resto del trigger */
memcpy(&trigdata, trig, sizeof(trigdata));

memset(trigdata.ApplId, ' ', sizeof(trigdata.ApplId));
memset(trigdata.EnvData, ' ', sizeof(trigdata.EnvData));
memset(trigdata.UserData, ' ', sizeof(trigdata.UserData));
memset(trigdata.QMgrName, ' ', sizeof(trigdata.QMgrName));

if( (applId = getenv("TRIGAPPLID")) != 0)
{
    strncpy( trigdata.ApplId ,applId, strlen(applId) );
}

if ( (envData = getenv("TRIGENVDATA")) != 0)
{
    strncpy( trigdata.EnvData , envData, strlen(envData) );
}

if ( (usrData = getenv("TRIGUSERDATA")) != 0)
{
    strncpy( trigdata.UserData, usrData, strlen(usrData) );
}

if ( (qmName = getenv("TRIGQMGRNAME")) != 0)
{
    strncpy( trigdata.QMgrName, qmName, strlen(qmName) );
}

trig = &trigdata;
```

Linguaggi ed ambienti supportati

MQSeries for Compaq NSK V5.1 supporta i linguaggi e gli ambienti descritti in Tabella 24. La tabella descrive inoltre se l'applicazione può utilizzare i binding FASTPATH o STANDARD.

Tabella 24. Riepilogo dei linguaggi e degli ambienti supportati

Linguaggio	Viene eseguito su Guardian? ¹	Viene eseguito su OSS?	può utilizzare il binding STANDARD?	Può utilizzare il binding FASTPATH? ¹
C nativo	Sì	Sì	Sì	Sì

Tabella 24. Riepilogo dei linguaggi e degli ambienti supportati (Continua)

Linguaggio	Viene eseguito su Guardian? ¹	Viene eseguito su OSS?	può utilizzare il binding STANDARD?	Può utilizzare il binding FASTPATH? ¹
C non nativo	Sì	No	Sì	No
COBOL nativo	Sì	Sì	Sì	Sì
COBOL non nativo	Sì	No	Sì	No
C++ nativo	Sì	Sì	Sì	Sì
TAL non nativo	Sì	No	Sì	No
NonStop Java ²	No	Sì	Sì	No

Note:

1. L'ambiente Guardian e le applicazioni OSS collegate tramite FASTPATH non possono utilizzare thread. Solo Java e OSS collegato tramite STANDARD possono utilizzare thread. Per ulteriori informazioni sull'utilizzo di thread nell'applicazione in uso, consultare "Considerazioni per la creazione di applicazioni con thread".
2. Le applicazioni NonStop Java utilizzano JTS (Java Transaction Services) per eseguire transazioni.
3. Un'applicazione nativa che utilizza il binding FASTPATH potrebbe risolvere MQI solo tramite Shared Resource Library di MQSeries for Compaq NSK. Un'applicazione nativa che utilizza il binding STANDARD è in grado di risolvere MQI utilizzando Shared Resource Library di MQSeries for Compaq NSK o una libreria MQI statica nativa. Per ulteriori informazioni sui binding FASTPATH e STANDARD, consultare "Messa a confronto dei binding FASTPATH e STANDARD" a pagina 354.

Considerazioni per la creazione di applicazioni con thread

Le applicazioni Guardian non supportano i thread. Essi possono implementare il proprio meccanismo di thread, ma le regole per utilizzare MQI in ambiente Guardian devono essere rispettate per l'elaborazione in corso.

In ambiente OSS, un pacchetto di emulazione thread basato su thread POSIX è disponibile. L'emulazione implementa un meccanismo di pianificazione in cui un thread deve fornire il controllo di esecuzione prima che il codice, presente in qualsiasi altro thread, possa essere eseguito. Le applicazioni possono utilizzare questo pacchetto con funzione di thread per organizzare l'elaborazione nei thread, ma vengono applicate le seguenti limitazioni:

- MQI non supporta la pianificazione tra i thread. Questo significa che quando MQI viene chiamato da un thread di applicazione, nessun altro thread può ottenere il controllo di esecuzione, a prescindere dall'intervallo di tempo. Ad esempio, se un thread chiama MQGET con l'opzione 'wait indefinitely', nessun altro thread può essere eseguito nel processo di applicazione fino a quando non viene restituito MQGET.
- Le applicazioni, collegate tramite FASTPATH in esecuzione in un ambiente OSS, non possono utilizzare i thread.
- Il queue manager non supporta processi LQMA o MQLQMAG (multi-threaded Local Queue Manager Agents) o MCA (Message Channel Agents).

Compilazione e binding delle applicazioni

MQI MQSeries for Compaq NSK viene implementato tramite il modello di memoria WIDE Compaq (il tipo di dati int è di 4 byte) e CRE (Common Runtime Environment). Per un corretto funzionamento, le applicazioni devono essere compatibili con questo ambiente. Consultare i file per la creazione per le appropriate opzioni di ciascun compilatore per verificare la compatibilità.

In particolare, le applicazioni TAL e COBOL devono seguire le regole richieste per la compatibilità con CRE, documentato nei manuali Compaq relativi all'ambiente CRE.

Per un corretto utilizzo della funzione MQGMO_SET_SIGNAL di MQGET, è necessario impostare l'attributo HIGHREQUESTERS su ON nel codice oggetti per le applicazioni COBOL e TAL.

Per effettuare l'installazione, tre versioni di MQI library sono distribuiti con MQSeries for Compaq NSK, contenuto in ZMQSLIB. E' necessario utilizzare la corretta library, come descritto nella sezione Tabella 25.

Tabella 25. Utilizzo della corretta versione di MQI library

Linguaggio di programmazione	Non nativo	Nativo/Statico	Nativo/Dinamico
TAL	MQMLIB	Non applicabile	Non applicabile
COBOL	MQMLIB	MQMLIBN	MQSRLLIB
C	MQMLIB	MQMLIBN	MQSRLLIB
C++	Non applicabile	MQMLIBN	MQSRLLIB

Messa a confronto dei binding FASTPATH e STANDARD

MQSeries for Compaq NSK V5.1 supporta sia i binding FASTPATH che quelli STANDARD. Tabella 25 descrive i linguaggi e gli ambienti che supportano ciascun tipo di binding.

Binding STANDARD

Quando vengono utilizzati i binding STANDARD in un'applicazione, considerare quanto segue:

- Le applicazioni non native e native possono utilizzare i binding STANDARD.
- Un'applicazione nativa che utilizza i binding STANDARD può risolvere MQI tramite:
 - Shared Resource Library di MQSeries for Compaq NSK
 - Una library MQI nativa statica. Essa fornisce il supporto per le applicazioni che già utilizzano una Shared Resource Library.

Binding FASTPATH

Quando vengono utilizzati i binding FASTPATH in un'applicazione, considerare quanto segue:

- Solo le applicazioni native possono utilizzare il binding FASTPATH.
- Un'applicazione nativa, che utilizza un binding FASTPATH, può risolvere MQI solo tramite Shared Resource Library di MQSeries for Compaq NSK.
- Le applicazioni collegate tramite FASTPATH in esecuzione su OSS non possono utilizzare thread.

- Le applicazioni collegate tramite FASTPATH devono essere eseguite sotto l'ID utente appartenente al gruppo di utenti di gestione di MQM Compaq NSK, che ha creato il queue manager.

Esecuzione delle applicazioni

Per collegarsi a un queue manager, l'ambiente di un programma applicativo deve essere definito correttamente:

- In ambiente di applicazione, il parametro PARAM MQDEFAULTPREFIX è obbligatorio.
- Se è stata selezionata un'ubicazione alternativa (ubicazione non predefinita) per il file MQSINI, un'applicazione non sarà in grado di collegarsi al queue manager se il parametro PARAM MQMACHINIFILE non è impostato correttamente.
- Le applicazioni TAL e COBOL devono disporre del parametro PARAM SAVE-ENVIRONMENT ON definito nel proprio ambiente o esse non saranno in grado di collegarsi al queue manager.

Un'applicazione può essere eseguita come low-pin o high-pin. Gli eseguibili MQSeries stessi sono configurati come high-pin.

Le applicazioni MQSeries sono supportate in ambienti Guardian e OSS.

Un'applicazione MQSeries può essere eseguita sotto PATHWAY, da TAACL o come processo secondario di un altro processo. Le applicazioni possono anche essere aggiunte alla configurazione stessa di PATHWAY del queue manager, che ha eseguito la chiusura corretta sul queue manager.

Esecuzione delle applicazioni

Appendice J. MQSeries Administration Interface (MQAI)

MQSeries for Compaq NSK V5.1 supporta l'interfaccia MQAI.

MQAI è un'interfaccia di programmazione di MQSeries, utilizzando il linguaggio C. Esegue le attività di gestione su un queue manager di MQSeries mediante le cartelle di dati. La cartella dati (Data bag) consente di gestire proprietà o parametri di oggetti in modo più semplice rispetto a PCF (Program Command Format).

MQAI offre una semplice manipolazione di PCF mediante le chiamate MQGET e MQPUT. E' possibile utilizzare MQAI per:

- Implementare le applicazioni di autogestione e gli strumenti di gestione.
- Semplificare l'utilizzo dei messaggi PCF. MQAI consente una facile gestione di MQSeries; non è necessario scrivere i messaggi PCF e questo evita i problemi che nascono dalla complessità delle strutture dei dati.
- Gestire in modo più semplice le condizioni di errore. È difficile risalire al codice dai comandi MQSC, tuttavia MQAI rende più facile al programma la gestione quando si verifica un errore.

Nota: MQSeries for Compaq NSK V5.1 fornisce solo file di intestazione C. Non fornisce i file di intestazione Visual Basic.

Per ulteriori informazioni su MQAI, consultare *MQSeries Administration Interface Programming Guide and Reference*

Appendice K. Programmi di esempio di MQSeries for Compaq NSK

I seguenti programmi di esempio C e COBOL sono forniti con MQSeries for Compaq NSK V5.1:

Descrizione	Sorgente C	Eseguibile C	Sorgente COBOL85	Eseguibile COBOL85
Descrittore di messaggio output di lettura e contesto per ciascun messaggio di una coda	amqsbcg0	amqsbcg	Nessun esempio	Nessun esempio
Restituire un messaggio da una coda di messaggi alla coda di risposta	amqsecha	amqsech	amq0ech0	amq0ech
Scrivere messaggi da una cosa a stdout, lasciare i messaggi in coda (Browse)	amqsgbr0	amqsgbr	amq0gbr0	amq0gbr
Eliminare i messaggi dalla coda denominata e scrivere a stdout	amqsget0	amqsget	amq0get0	amq0get
Leggere la coda di trigger, rispondere con le informazioni di coda	amqsinqa	amqsinq	Nessun esempio	Nessun esempio
Utilizzare una coda di input predefinita	Nessun esempio	Nessun esempio	amq0inq0	amq0inq
Copiare stdin in un messaggio ed inserire il messaggio nella coda specificata	amqsput0	amqsput	amq0put0	amq0put
Inserire un messaggio di richiesta in una coda specifica e visualizzare le risposte	amqsreq0	amqsreq	amq0req0	amq0req
(Funzione trigger) Inibire i put su una coda nominata e risponde con una istruzione del risultato.	amqsseta	amqsset	amq0set0	amq0set
Monitor trigger	amqstrg0	amqstrg	Nessun esempio	Nessun esempio
Schema di esempio per uscita di conversione dati	amqsvfcn	Nessun esempio	Nessun esempio	Nessun esempio
Schema di esempio per la channel exit	amqsvchn	Nessun esempio	Nessun esempio	Nessun esempio
Schema di esempio per la cluster workload exit	amqswlm0	Nessun esempio	Nessun esempio	Nessun esempio
Schema di esempio per MQLOADEXIT	amqslxp0	Nessun esempio	Nessun esempio	Nessun esempio

I seguenti programmi di esempio TAL sono forniti con MQSeries for Compaq NSK V5.1:

Descrizione	fonte TAL	eseguibile TAL
Leggere <i>n</i> messaggi da una coda	zmqreadt	zmqread
Scrivere <i>n</i> messaggi di <i>n</i> lunghezza in una coda	zmqwritt	zmqwrit

Creazione di programmi di esempio in C

Non nativo (con l'utilizzo della libreria statica non nativa MQMLIB)

Il volume secondario ZMQSSMPL contiene i seguenti file macro TACL da utilizzare per la creazione delle applicazioni C di esempio non native:

CSAMP	Utilizzo: CSAMP <i>nome-file-codice-sorgente</i> Macro di base per la compilazione di un file di origine C che utilizza i file contenuti nel volume secondario ZMQSLIB. Ad esempio, per compilare AMQSBCG0, utilizzare CSAMP AMQSBCG0. Se la compilazione ha esito positivo, la macro produrrà un file oggetto nel cui nome l'ultimo carattere è stato sostituito dalla lettera O; ad esempio, AMQSBCGO.
BSAMP	Utilizzo: BSAMP <i>nome-file-exe</i> Macro di base utilizzata per eseguire il bind di un file oggetto alla libreria utente MQMLIB in ZMQSLIB. Ad esempio, per eseguire il bind dell'esempio compilato AMQSBCG0, utilizzare BSAMP AMQSBCG. La macro produrrà il file eseguibile <i>nome-file-exeE</i> ; ad esempio, AMQSBCGE.
COMPALL	Utilizzo: COMPALL Macro TACL per la compilazione di tutti i file del codice sorgente di esempio mediante l'utilizzo della macro CSAMP.
BINDALL	Utilizzo: BINDALL Macro TACL per l'esecuzione del bind di ciascuno dei file di oggetto di esempio con file eseguibili con l'utilizzo della macro BSAMP.
BUILDC	Utilizzo: BUILDC Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio C con l'utilizzo delle macro COMPALL e BINDALL.

Nativo (con l'utilizzo della libreria statica nativa MQMLIBN)

Se si desidera eseguire un'installazione nativa, utilizzare i seguenti file macro TACL per creare le applicazioni MQI di esempio:

NMCALL	Utilizzo: NMCALL Macro per la compilazione di tutti gli esempi nativi con l'utilizzo di NMCSAMP.
NMCSAMP	Utilizzo: NMCSAMP <i>nome-file-codice-sorgente</i> Macro di base per la compilazione di un file di origine C che utilizza i file contenuti nel volume secondario ZMQSLIB. Ad esempio, per compilare

AMQSBCG0, utilizzare NMCSAMP AMQSBCG0. Se la compilazione ha esito positivo, la macro produrrà un file oggetto nel cui nome l'ultimo carattere è stato sostituito dalla lettera O; ad esempio, AMQSBCGO.

NMLDSAMP

Utilizzo: NMLDSAMP *nome-file-exe*

Macro di base per il collegamento di un file di oggetto alla libreria MQI nativa MQMLIBN in ZMQSLIB.

NMLDALL

Utilizzo: NMLDALL

Macro TACL per l'esecuzione del bind di ciascuno dei file di oggetto di esempio con file eseguibili con l'utilizzo della macro NMLDSAMP.

NMBUILD C

Utilizzo: NMBUILD C

Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio C nativi con l'utilizzo delle macro NMCALL e NMLDALL.

Nativo (con l'utilizzo di SRL MQSRLLIB)

NMLDSSMP

Utilizzo: NMLDSSMP *nome-file-exe*

Macro di base per il collegamento di un file di oggetto a SRL MQ SRL nativo MQSRLLIB in ZMQSLIB.

NMLDSALL

Utilizzo: NMLDSALL

Macro TACL per l'esecuzione del bind di ciascuno dei file di oggetto di esempio con file eseguibili con l'utilizzo della macro NMLDSSMP.

NMBULDSC

Utilizzo: NMBULDSC

Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio C nativi con l'utilizzo delle macro NMCALL e NMLDSALL.

NMLDPSRL

Utilizzo: NMLDPSRL *nome-file-exe*

Macro di base per il collegamento di un file di oggetto con SRL privato MQSeries in ZMQSLIB

NMCPSRL

Utilizzo: NMCPSRL *nome-file-codice-sorgente*

Macro per la compilazione del codice utente affinché venga incluso in MQSeries PSRL.

NMLDUSRL

Utilizzo: NMLDUSRL *file-input-oggetto*, dove *file-input-oggetto* è un file che contiene un elenco di oggetti da collegare.

Macro di base per il collegamento del codice utente in una libreria ricollegabile.

Nota: Le applicazioni non native possono collegarsi ai queue manager, e viceversa. Tutte le combinazioni tra operazioni native e non native sono supportate.

Creazione di programmi di esempio in C++

Nativo (con l'utilizzo della libreria statica nativa MQMLIBN)

NMCPALL	Utilizzo: NMCPALL Macro per la compilazione di tutti gli esempi nativi con l'utilizzo di NMCCPP.
NMCCPP	Utilizzo: NMCCPP <i>nome-file-codice-sorgente</i> Macro di base per la compilazione di un file di origine C ++ che utilizza i file contenuti nel volume secondario ZMQSLIB. Ad esempio, per compilare IMQSGETP, utilizzare NMCCPP IMQSGETP. Se la compilazione ha esito positivo, la macro produrrà un file oggetto nel cui nome l'ultimo carattere è stato sostituito dalla lettera O; ad esempio, IMQSGETO.
NMLDCPP	Utilizzo: NMLDCPP <i>nome-file-exe</i> Macro di base per il collegamento di un file di oggetto alla libreria MQI statica nativa MQMLIBN in ZMQSLIB.
NMLDCPPA	Utilizzo: NMLDCPPA Macro TACL per l'esecuzione del bind di ciascuno dei file di oggetto di esempio con file eseguibili con l'utilizzo della macro NMLDSAMP.
NMBLDCPP	Utilizzo: NMBUILD Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio C++ nativi con l'utilizzo delle macro NMCPALL e NMLDCPPA.

Nativo (con l'utilizzo di SRL MQSRLLIB)

NMLDCPPS	Utilizzo: NMLDCPPS <i>nome-file-exe</i> Macro di base per il collegamento di un file di oggetto a SRL MQ SRL nativo MQSRLLIB in ZMQSLIB.
NMLDCPSA	Utilizzo: NMLDCPSA Macro TACL per l'esecuzione del bind di ciascuno dei file di oggetto di esempio con file eseguibili con l'utilizzo della macro NMLDCPPS.
NMBLDSCP	Utilizzo: NMBLDSCP Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio C nativi con l'utilizzo delle macro NMCPALL e NMLDCPSA.

Creazione di programmi di esempio in COBOL

Non nativo (con l'utilizzo della libreria statica non nativa MQMLIB)

Il volume secondario ZMQSSMPL contiene i seguenti file da utilizzare per la creazione delle applicazioni COBOL di esempio.

COBSAMP	Utilizzo: <i>COBSAMP nome-file-codice-sorgente</i> Macro di base per la compilazione di un file di origine COBOL che utilizza i file di definizione contenuti nel volume secondario ZMQSLIB. Ad esempio, per compilare il programma AMQ0GBR0, utilizzare <i>COBSAMP AMQ0GBR0</i> . Se la compilazione ha esito positivo, la macro produrrà un file oggetto nel cui nome l'ultimo carattere è stato sostituito dalla lettera O; ad esempio, AMQ0GBR0.
BCOBSAMP	Utilizzo: <i>BCOBSAMP nome-file-exe</i> Macro di base utilizzata per eseguire il bind di un oggetto alle librerie utente in ZMQSLIB. Ad esempio, per eseguire il bind dell'esempio compilato AMQ0GBR0, utilizzare <i>BCOBSAMP AMQ0GBR</i> . La macro produce il file eseguibile <i>nome-file-exe AMQ0GBR</i> .
CCBSMPLS	Utilizzo: <i>CCBSMPLS</i> Macro TACL per la compilazione di tutti i file del codice sorgente COBOL.
BCBSMPLS	Utilizzo: <i>BIND /IN BCBSMPLS/</i> File input di bind per l'esecuzione di tutti i file oggetto di esempio COBOL, con file eseguibili.
BUILDCOB	Utilizzo: <i>BUILDCOB</i> Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio COBOL con l'utilizzo delle macro <i>CCBSMPLS</i> e <i>BCBSMPLS</i> .

Nativo (con l'utilizzo della libreria statica nativa MQMLIBN)

NMCOBSMP	Utilizzo: <i>NMCOBSMP nome-file-codice-sorgente</i> Macro per la compilazione del modo nativo COBOL 'NMCOBOL' con l'utilizzo della libreria nativa MQSeries MQMLIBN in ZMQSLIB. Ad esempio, per compilare il programma AMQ0GBR0, utilizzare <i>NMCOBSMP AMQ0GBR0</i> . Se la compilazione ha esito positivo, la macro produrrà un file oggetto nel cui nome l'ultimo carattere è stato sostituito dalla lettera O; ad esempio, AMQ0GBR0.
NMLDCOB	Utilizzo: <i>NMLDCOB nome-file-exe</i> Macro per l'esecuzione del bind dell'oggetto con la libreria nativa MQSeries MQMLIBN in ZMQSLIB. Ad esempio, per eseguire il bind dell'esempio

Creazione di esempi in COBOL

compilato AMQ0GBRO, utilizzare NMLDCOB AMQ0GBR. La macro produce il file eseguibile *nome-file-exe* AMQ0GBR.

NMCOBALL

Utilizzo: NMCOBALL

Macro TACL per la compilazione di tutti i file del codice sorgente COBOL con l'utilizzo di NMCOSMP.

NMLDACOB

Utilizzo: NMLDACOB

File input di bind per l'esecuzione di tutti i file oggetto di esempio NMLDCOB, con file eseguibili.

NMBLDCOB

Utilizzo: NMBLDCOB

Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio COBOL con l'utilizzo delle macro NMCOBALL e NMLDACOB.

Nativo (con l'utilizzo di SRL MQSRLLIB)

NMCBSSMP

Utilizzo: NMCBSSMP *nome-file-codice-sorgente*

Macro per la compilazione del modo nativo COBOL 'NMCOBOL' con l'utilizzo di MQSeries SRL MQSRLLIB in ZMQSLIB. Ad esempio, per compilare il programma AMQ0GBR0, utilizzare NMCBSSMP AMQ0GBR0. Se la compilazione ha esito positivo, la macro produrrà un file oggetto nel cui nome l'ultimo carattere è stato sostituito dalla lettera O; ad esempio, AMQ0GBRO.

NMLDSCOB

Utilizzo: NMLDSCOB *nome-file-exe*

Macro per l'esecuzione del bind degli oggetti con MQSeries SRL MQSRLLIB in ZMQSLIB. Ad esempio, per eseguire il bind dell'esempio compilato AMQ0GBRO, utilizzare NMLDSCOB AMQ0GBR. La macro produce il file eseguibile *nome-file-exe* AMQ0GBR.

NMCBSALL

Utilizzo: NMCBSALL

Macro TACL per la compilazione di tutti i file del codice sorgente COBOL con l'utilizzo di NMCBSSMP.

NMLDSCOB

Utilizzo: NMLDSCOB

File input di bind per l'esecuzione di tutti i file oggetto di esempio NMLDSCOB, con file eseguibili.

NMBLDSCB

Utilizzo: NMBLDSCB

Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio COBOL con l'utilizzo delle macro NMCBSALL e NMLDSCOB.

Creazione di programmi di esempio in TAL

Non nativo (con l'utilizzo della libreria statica non nativa MQMLIB)

Il volume secondario ZMQSSMPL contiene i seguenti file da utilizzare per la creazione delle applicazioni TAL di esempio.

TALSAMP	Utilizzo: TALSAMP <i>nome-file-codice-sorgente</i> Macro di base per la compilazione di un file di origine TAL con l'utilizzo dei file di definizione contenuti nel volume secondario Appendix I. MQSeries for Compaq NSK sample programs 335.ZMQSLIB. Ad esempio, per compilare il programma ZMQWRITT, utilizzare TALSAMP ZMQWRITT. Se la compilazione ha esito positivo, la macro produrrà un file oggetto nel cui nome l'ultimo carattere è stato sostituito dalla lettera O; ad esempio, ZMQWRITO.
BTALSAMP	Utilizzo: BTALSAMP <i>nome-file-exe</i> Macro di base utilizzata per eseguire il bind di un oggetto alle librerie utente in ZMQSLIB. Ad esempio, per eseguire il bind dell'esempio compilato ZMQWRITO, utilizzare BTALSAMP ZMQWRIT.
CTLSMPLS	Utilizzo: CTLSMPLS Macro TACL per la compilazione di tutti i file del codice sorgente TAL.
BTLSMPLS	Utilizzo: BIND /IN BTLSMPLS/ File input di bind per l'esecuzione di tutti i file oggetto di esempio TAL, con file eseguibili.
BUILDTAL	Utilizzo: BUILDTAL Macro TACL per la compilazione e l'esecuzione del bind dei file di esempio TAL con l'utilizzo delle macro CTLSMPLS e BTLSMPLS.

Creazione dei programmi di esempio in OSS (solo modo nativo)

La directory - /opt/mqm/samp contiene i programmi di esempio per MQSeries il file MQMAKE.SMP. Questo MakeFile contiene le informazioni necessarie per creare tutti gli esempi C, C++ e NMCOBOL

La directory - /opt/mqm/inc contiene tutti i copylibs e i file di intestazione necessari per la creazione dei programmi in OSS.

Nota: Il file MQSeries SRL e la libreria MQI nativa—MQMLIBN è presente solo nel file system Guardian. È necessario editare gli script di creazione e far sì che i file siano indirizzati agli stessi script, ad esempio, /G/system/zmqslib/mqsrllib.

Creazione di esempi in OSS

Appendice L. User exit

MQSeries for Compaq NSK V5.1 supporta i programmi di uscita relativi ai canali, alla conversione dei dati e a CLWL (Cluster Workload Management). Inoltre è supportato un programma di uscita dal caricamento specifico Compaq NSK. Per informazioni relative alle uscite di canale, consultare il manuale *MQSeries Intercommunication*. Per informazioni sulle uscite di conversioni dei dati, fare riferimento a *MQSeries Application Programming Guide* e *MQSeries for Tandem Non Stop Kernal, V5.1 Guida operativa*. Per informazioni sui programmi di uscita Cluster Workload Management, fare riferimento al manuale *MQSeries Queue Manager Clusters*.

Questa appendice fornisce informazioni specifiche per l'utilizzo dei programmi di uscita in MQSeries for Compaq NSK.

User exit supportati

Tabella 26 elenca le caratteristiche di ciascun tipo di uscita utente supportata per MQSeries for Compaq NSK.

Tabella 26. User exit supportati per MQSeries for Compaq NSK

Tipo di User Exit	Lunghezza massima nome uscita	Lunghezza massima dati di uscita	Dove abilitato	In catena?
Uscita MSG canale	32	32	DEFINE CHANNEL	Sì
Uscita SEND canale	32	32	DEFINE CHANNEL	Sì
Uscita RECEIVE canale	32	32	DEFINE CHANNEL	Sì
Uscita SECURITY canale	32	32	DEFINE CHANNEL	
Uscita MSGRETRY canale	32	32	DEFINE CHANNEL	
Uscita auto-definizione canale	32	32	ALTER QMGR	
Uscita Cluster Workload Management	32	32	ALTER QMGR	
Uscita conversione dati	8	Non applicabile	Nome FORMAT sconosciuto	
MQ_LOAD_ENTRY_POINT_EXIT	Nome fisso	Non applicabile	Richiamato quando una delle uscite precedenti è richiesta o abilitata.	

MQ_LOAD_ENTRY_POINT_EXIT è l'unica user exit specifica per Compaq NSK.

Formato nome uscita

Nomi di uscita (diversi da MQ_LOAD_ENTRY_POINT_EXIT) possono essere costituiti da stringhe alfanumeriche per un massimo di 32 caratteri. Per le uscite che supportano un campo di dati associati, i dati possono contenere massimo 32 caratteri.

MQ_LOAD_ENTRY_POINT_EXIT - Caricamento User Exit

Tutti i programmi user exit devono essere collegati a MQSeries Private SRL o alla libreria statica. I programmi user exit devono contenere almeno una funzione esterna (simbolo) che può essere richiamata da MQSeries quando richiesto.

Prima di abilitare qualsiasi altra user exit MQSeries, è necessario installare un programma MQ_LOAD_ENTRY_POINT_EXIT per mappare i nomi di uscita sugli indirizzi di entry-point. Il programma MQ_LOAD_ENTRY_POINT_EXIT deve essere collegato a MQSeries SRL o alla libreria statica e viene richiamato da MQSeries quando una delle altre user exit è abilitata.

Il nome del programma MQ_LOAD_ENTRY_POINT_EXIT è fisso, quindi il suo nome di funzione esterno deve rimanere MQ_LOAD_ENTRY_POINT.

MQ_LOAD_ENTRY_POINT_EXIT è libero di creare una mappatura con un nome uscita su qualsiasi indirizzo di entry point oppure una mappatura di molti nomi di uscita sullo stesso indirizzo di entry-point.

MQSeries fornisce una funzione stub MQ_LOAD_ENTRY_POINT_EXIT che restituisce sempre MQXCC_FAILED (Nome uscita non rilevato) quando è richiamata. E' necessario sostituire questa uscita stub con la propria prima di abilitare qualsiasi altra user exit.

MQ_LOAD_ENTRY_POINT_EXIT (*ExitParms*)

Parametri:

Exitparms (PMQLXP) – input/output
Blocco parametri LoadExit

Note sull'utilizzo: La funzione eseguita dal programma MQ_LOAD_ENTRY_POINT_EXIT viene definita dal fornitore di tale uscita.

Figura 46 a pagina 371 contiene un esempio MQ_LOAD_ENTRY_POINT_EXIT che mappa le tre uscite sugli indirizzi di entry point.

MQLXP - Struttura parametri MQ_LOAD_ENTRY_POINT_EXIT

La struttura MQLXP descrive le informazioni che vengono inoltrate all'uscita di caricamento.

Questa struttura è supportata solo per Compaq NSK.

Campi

StrucId (MQCHAR4)

Identificativo struttura.

Il valore è MQLXP_STRUC_ID.

Programmi di uscita supportati

Identificativo per la struttura del parametro di uscita del caricamento.

Per il linguaggio di programmazione C, anche la costante `MQLXP_STRUC_ID_ARRAY` viene definita. Il valore della costante è lo stesso di `MQLXP_STRUC_ID`, ma è costituito da una serie di caratteri e non da una stringa.

E' un campo di immissione all'uscita.

Version (MQLONG)

Il numero di versione della struttura.

Il valore è: `MQLXP_VERSION_1`

Struttura del parametro di uscita del caricamento della versione-1.

La costante di seguito riportata specifica il numero della versione corrente:
`MQLXP_CURRENT_VERSION`

La versione corrente della struttura del parametro di uscita del caricamento.

E' un campo di immissione all'uscita.

QMgrName (MQCHAR48)

Nome del queue manager locale.

E' il nome del queue manager che ha richiamato l'uscita di caricamento. Il nome contiene spazi vuoti per la lunghezza del campo.

E' un campo di immissione all'uscita.

EntryPointName (MQCHAR32)

Nome dell'Entry Point richiesto.

E' il nome dell'Entry Point che occorre all'uscita di caricamento per risolvere un indirizzo richiamabile. Il nome contiene spazi vuoti per la lunghezza del campo.

E' un campo di immissione all'uscita.

EntryAddress (PMQFUNC)

Indirizzo richiamabile restituito.

E' l'indirizzo dell'EntryPoint richiesto.

E' un campo di emissione dall'uscita.

ExitResponse (MQLONG)

Risposta dall'uscita.

Il valore è impostato dall'uscita per indicare se la risoluzione del nome di immissione su un indirizzo richiamabile è riuscito correttamente. Il valore deve essere uno dei seguenti:

`MQXCC_OK` Riuscito.

Indica che l'elaborazione dell'uscita ha risolto correttamente l'EntryPointName fornito nell'ExitParms su un indirizzo richiamabile. L'indirizzo richiamabile viene restituito nel campo EntryAddress della struttura MQLXP.

`MQXCC_FAILED` Non riuscito.

Indica che l'uscita non è stata in grado di risolvere l'EntryPointName fornito nell'ExitParms su un indirizzo richiamabile.

Programmi di uscita supportati

Qualsiasi altro valore restituito nel campo ExitResponse ha lo stesso significato di MQXCC_FAILED.

E' un campo di emissione dall'uscita.

Esempio MQ_LOAD_ENTRY_POINT_EXIT

Figura 46 a pagina 371 è un esempio di un programma MQ_LOAD_ENTRY_POINT_EXIT che mappa tre nomi di uscita (due uscite canali e una uscita di conversione dati) sugli indirizzi di entry point. Il codice di origine per il programma di esempio MQ_LOAD_ENTRY_POINT_EXIT viene fornito nel volume secondario di esempio (AMQSLXP0).

```

/*****/
/*
/* Program name: AMQSLXP0          (Compaq NSK only)
/*
/* Description: Sample C skeleton of a Load Exit function
/*
/* Statement:      Licensed Materials - Property of IBM
/*
/*                  (C) Copyright IBM Corp. 1993, 2001
/*
/*****/
/*
/* Function:
/*
/* AMQSLXP0 is a sample C skeleton of a Load Exit function
/*
/* The function resolves EntryNames to callable addresses
/*
/*
/* Once complete the code should be compiled into a loadable
/* object, the name of the object should be the name of the
/* format to be converted. Instructions on how to do this are
/* contained in the README file in this directory.
/*
/*****/
/*
/* AMQSLXP0 takes the parameters defined for a Load Exit
/* routine in the CMQXC.H header file.
/*
/*****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>

#include <cmqc.h>
#include <cmqxc.h>

/*****/
/* Load Exit
/*
/*
/*****/
void
MQENTRY MQ_LOAD_ENTRY_POINT(
    PMQLXP pExitParms          /* exit Parameter */
)
{
    /* No loadable entry points are defined */
    pExitParms->ExitResponse = MQXCC_FAILED;

    return;
}
/*****/
/* End of AMQSLXP0
/*****/

```

Figura 46. Esempio MQLOADEXIT

Programmi di uscita supportati

Con il programma MQ_LOAD_ENTRY_POINT_EXIT e il canale e le uscite di conversione dati installati, è possibile abilitare le uscite di ricezione e invio dei canali utilizzando il seguente comando MQSC:

```
ALTER CHANNEL(CHAN) CHLTYPE(SDR) SENDEXIT(MY_CHANNEL_SEND_EXIT)
ALTER CHANNEL(CHAN) CHLTYPE(SDR) RCVEXIT(MY_CHANNEL_RCV_EXIT)
```

L'uscita di conversione dei dati verrà richiamata da MQSeries quando un MQGET viene eseguito con la conversione abilitata (MQGMO_CONVERT) e il nome del formato dei messaggi è MY_FORMAT.

Installazione user exit

Tutte le user exit create devono essere installate nell'SRL privato di MQSeries. Inoltre, alcuni tipi di uscita devono essere installati in qualsiasi libreria statica di MQSeries utilizzata dai programmi di applicazione.

Tipo di User Exit	Dove installato
Uscita MSG canale	SRL privato
Uscita SEND canale	SRL privato
Uscita RECEIVE canale	SRL privato
Uscita SECURITY canale	SRL privato
Uscita MSGRETRY canale	SRL privato
Uscita auto-definizione canale	SRL privato
Uscita Cluster Workload Management	SRL privato
Uscita conversione dati	SRL privato (anche libreria statica nativa e/o libreria statica non-nativa se utilizzato da applicazioni)
MQ_LOAD_ENTRY_POINT_EXIT	SRL privato (anche libreria statica nativa e/o libreria statica non-nativa se utilizzato da applicazioni)

Installazione di un'uscita nell'SRL privato di MQSeries

Per installare un programma di uscita nell'SRL privato di MQSeries creare una nuova versione dell'SRL privato di MQSeries che contenga la nuova uscita:

1. Compilare la funzione di uscita. Ad esempio:

```
NMCP SRL AMQSVFCN
```

2. Compilare la funzione di entry point MQLOAD. Ad esempio:

```
NMCP SRL AMQSLXP0
```

3. Collegare l'uscita e gli oggetti di entry point su una libreria a cui è possibile ricollegarsi successivamente nell'SRL. Ad esempio:

```
NMLDUSRL OBJECTS EXITS
```

4. Creare una nuova versione dell'SRL privato di MQSeries chiamata NEWMQSRL collegando questo oggetto di conversione dei dati con la versione ricollegabile dell'SRL privato di MQSeries. Ad esempio:

```
MAKEPSRL EXITS $VOL.ZMQSLIB NEWMQSRL
```

5. Interrompere tutti i queue manager e le applicazioni che accedono all'SRL privato di MQSeries corrente.
6. Ricollegare tutte le applicazioni MQSeries al nuovo PSRL. Ad esempio:

```
NMLDEXES $VOL.ZMQSLIB.NEWMQSRL $VOL.ZMQSEXE
```

7. Compilare l'applicazione get. Ad esempio:

```
NMCSAMP AMQSGET0
```

8. Ricollegare tutte le applicazioni utente al nuovo PSRL. Ad esempio:

```
AMQSGET NMLDPSRL AMQSGET
```

9. Riavviare MQSeries e tutte le applicazioni MQI.

I passi da 6 a 8 sono abbastanza rapidi, ma possono essere omessi se il nuovo PSRL MQSeries è ubicato in ZMQSLIB e chiamato MQSRLLIB. I passi possono essere ripetuti per collegarsi ad un PSRL di MQSeries diverso.

Installazione di un'uscita nella libreria statica nativa di MQSeries

Un'uscita può essere collegata con l'applicazione scelta e con la libreria MQI utilizzando la macro TACL NMLDEXIT. Ad esempio:

```
NMLDEXIT Object-File Exit-Object-File
```

Installazione di un'uscita nella libreria statica non-nativa MQSeries

Un'uscita può essere collegata all'eseguibile scelto (o libreria) utilizzando la macro TACL BEXITE.

Nota: Tale procedura consente di modificare l'eseguibile di destinazione; si consiglia di creare una copia di riserva dell'eseguibile di destinazione prima di utilizzare la macro.

Programmi di uscita supportati

Le funzioni di uscita, dopo la compilazione, devono essere collegate direttamente all'eseguibile di destinazione o alla libreria a cui accedere da MQSeries. La macro TACL, BEXITE, viene utilizzata a tale scopo. Ad esempio:

```
BEXITE Target-Executable-Or-Library Source-Exit-File-Or-Library
```

Ad esempio, per collegare l'uscita della conversione di dati di esempio nell'esempio MQSGETA, procedere nel modo seguente:

1. Compilare la funzione di uscita. Ad esempio:

```
CSAMP AMQSVFCN
```

2. Compilare la funzione di entry point MQLOAD. Ad esempio:

```
CSAMP AMQSLXP0
```

3. Compilare l'applicazione get. Ad esempio:

```
CSAMP AMQSGET0
```

4. Collegare l'applicazione get. Ad esempio:

```
BSAMP AMQSGET
```

5. Collegare la funzione di uscita nell'applicazione get. Ad esempio:

```
BEXITE AMQSGET AMQSVFC0
```

6. Collegare la funzione di entry point nell'applicazione get. Ad esempio:

```
BEXITE AMQSGET AMQSLXP0
```

In alternativa, se tutte le applicazioni devono disporre dell'uscita di conversione dei dati, seguire le procedure descritte per creare sia una libreria utente che un'applicazione con l'uscita collegata:

1. Compilare la funzione di uscita. Ad esempio:

```
CSAMP AMQSVFCN
```

2. Compilare la funzione di entry point MQLOAD. Ad esempio:

Programmi di uscita supportati

```
CSAMP AMQSLXP0
```

3. Compilare l'applicazione get. Ad esempio:

```
CSAMP AMQSGET0
```

4. Collegare la funzione di uscita nella libreria utente. Ad esempio:

```
BEXITE ZMQSLIB.MQMLIB AMQSVFC0
```

5. Collegare la funzione di uscita nella libreria utente. Ad esempio:

```
BEXITE ZMQSLIB.MQMLIB AMQSLXP0
```

6. Collegare l'applicazione get con la libreria modificata. Ad esempio:

```
BSAMP AMQSGET
```

Programmi di uscita supportati

Appendice M. Impostazione delle comunicazioni

Questa appendice descrive come impostare le comunicazioni per MQSeries for Compaq NSK mediante l'utilizzo dei protocolli di comunicazione SNA e TCP/IP. Sono forniti i seguenti esempi:

- "Esempi di comunicazione SNAX" a pagina 384
- "Esempi di comunicazione ICE" a pagina 389
- "Esempi di comunicazione TCP/IP" a pagina 393

Canali SNA

Gli attributi di canale riportati di seguito sono necessari per i canali SNA in MQSeries for Compaq NSK V5.1:

CONNNAME

Il valore di CONNNAME dipende dal protocollo di comunicazione utilizzato, SNAX o ICE:

Se viene utilizzato il protocollo SNAX:

CONNNAME('\$PPPP.LOCALLU.REMOTELU')

Viene applicato al mittente, al richiedente e ai canali completi del server, dove:

\$PPPP	Corrisponde al nome del processo SNAX/APC.
LOCALLU	Corrisponde al nome del LU locale.
REMOTELU	Corrisponde al nome del partner LU sulla macchina remota.

Ad esempio:

```
CONNNAME('$BP01.IYAHT080.IYCNVM03')
```

Se viene utilizzato il protocollo ICE:

CONNNAME('\$PPPP.#OPEN.LOCALLU.REMOTELU')

Viene applicato al mittente, al richiedente e ai canali completi del server, dove:

\$PPPP	Corrisponde al nome del processo ICE.
#OPEN	Corrisponde al nome open ICE.
LOCALLU	Corrisponde al nome del LU locale.
REMOTELU	Corrisponde al nome del partner LU sulla macchina remota.

Ad esempio:

```
CONNNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+)
```

MODENAME

Corrisponde al nome del modo SNA. Ad esempio, MODENAME(LU62PS).

TPNAME('LOCALTP[.REMOTETP]')

Corrisponde al nome del TP (Transaction Process).

Impostazione delle comunicazioni

LOCALTP	Corrisponde al nome della classe di server (generalmente MQSeries) nel PATHWAY utilizzato per le comunicazioni. Se viene utilizzato il protocollo of SNAX, sarà necessario che la classe di server sia ubicata nello stesso PATHWAY del del dispatcher SNAX e della classe di server di APC Process. Se viene utilizzato il protocollo ICE, dovrà essere ubicata nel PATHWAY dichiarato nel comando NOF - ADD TP.
REMOTETP	Corrisponde al nome del TP sulla macchina remota. Questo valore è facoltativo. Se non diversamente specificato e se il canale avvia una conversazione (cioè un mittente, un richiedente o un canale completo del server), viene utilizzato il nome del LOCALTP.

La lunghezza di LOCALTP e REMOTETP può raggiungere un massimo di 16 caratteri.

Processi responder LU 6.2

In MQSeries for Compaq NSK V5.1, è necessario un processo SNA Listener pr eseguire il listen delle richieste in entrata allegate dal canale del queue manager.

Quando un allegato in arrivo viene ricevuto da un canale remoto, i listener MQSeries SNA listener dovranno essere inviati dal prodotto SNA.

Utilizzando il dispatcher SNAX APC, SNAX è in grado accettare le richieste di allegato in entrata da un programma di transazione partner. Affinché il Dispatcher SNAX esegua questa operazione, è necessario che:

- il Dispatcher SNAX venga eseguito in un PATHWAY diverso da quello utilizzato dai queue manager
- il processo APC venga eseguito come una classe di server dallo stesso PATHWAY

Per utilizzare ICE di Insession, è necessario che:

- un programma di transazione (classe di server) venga definito in un PATHWAY
- un programma di transazione venga aggiunto a NOF in modo che venga indirizzato a questa classe di server.

Non sono presenti Dispatcher ICE, il processo ICE gestisce da solo le richieste di allegato in entrata.

Invio di SNAX TP

Il prodotto Compaq SNAX SNA supporta l'avvio dei programmi di transazione (TP) APPC al ricevimento di un allegato APPC da un programma di transazione partner. Un Dispatcher SNAX invia tali richieste ai rispettivi processi SNAX \$APC mediante la classe di server di un programma di transazione locale.

Per utilizzare il Dispatcher SNAX per MQSeries, è necessario che:

- Il Dispatcher SNAX venga eseguito nello stesso PATHWAY del \$APC ad esso associato.
- Ciascun TPNAME in entrata sia definito come una classe di server (generalmente denominata MQSERIES) nello stesso PATHWAY utilizzato dal Dispatcher e dal processo \$APC.
- Il nome del programma della classe di server sia il programma **runmqtsr** presente nel volume secondario degli eseguibili MQSeries (in genere ZMQSEXE).

Impostazione delle comunicazioni

- Il processo del Dispatcher avvia la classe di server e trasmette tutte le informazioni importanti (\$Processo APC, Nome LU, Nome TP) alla classe di server (TP) in una richiesta DISPATCH-TP IPC.

Invio di ICE TP

ICE Versione 3.2 implementa i rispettivi SNA Attach Manager in modo simile a SNAX in quanto il TP utilizzato è una classe di server del PATHWAY. Il processo ICE accetta una richiesta di allegato e agisce come dispatcher. Non è necessario che questo processo ICE venga eseguito nello stesso PATHWAY del TP (ServerClass). In questo caso, è necessario che:

- Un processo attivo Ice sia in esecuzione.
- Un Dispatch TP venga aggiunto a NOF. Ad esempio:

```
ADD TP <tpname>, PROCESS <$process>, SERVERCLASS <Serverclass name>
```

Il *nome del Serverclass* è generalmente MQSeries.

- Ciascun TPNAME in entrata venga definito come una classe di server (generalmente definita MQSERIES) nel PATHWAY<\$processo>.
- Il nome del programma della classe di server sia il programma RUNMQLSR presente nel volume secondario degli eseguibili MQSeries (in genere ZMQSEXE)
- Il processo ICE avvia la classe di server e trasmetta tutte le informazioni importanti (\$Processo ICE, Nome Appl, Nome TP) a questa classe di server (TP) in una richiesta DISPATCH-TP IPC.

Impostazione ambiente SNA di esempio

Di seguito vengono riportati degli esempi di configurazione degli ambienti SNA.

Utilizzo di SNAX APC: Se viene utilizzato SNAX APC:

- È necessario creare un PATHWAY da utilizzare esclusivamente per questo listener
- Da questo PATHWAY, eseguire una classe di server del processo APC.

Da una richiesta comandi TAQL, immettere quanto segue:

```
TAQL> Pathmon /name $PMAP, nowait, out $vhs, cpu 3/4  
TAQL> Pathcom $PMAP  
= 0 LU62SCFG
```

dove LU62SCFG corrisponde a un file di editazione che contiene quanto segue:

Impostazione delle comunicazioni

```
[ SET PATHMON BACKUPCPU 6
SET PATHWAY MAXTCPS 10
SET PATHWAY MAXTERMS 10
SET PATHWAY MAXPROGRAMS 10
SET PATHWAY MAXSERVERCLASSES 10
SET PATHWAY MAXSERVERPROCESSES 10
SET PATHWAY MAXSTARTUPS 10
SET PATHWAY MAXPATHCOMS 40
SET PATHWAY MAXASSIGNS 32
SET PATHWAY MAXPARAMS 32
START PATHWAY COLD!

SET TCP PROGRAM $ SYSTEM.SYSTEM.PATHTCP2
SET TCP CPUS 3:4
SET TCP MAXTERMS 5
SET TCP MAXSERVERCLASSES 010
SET TCP MAXSERVERPROCESSES 010
SET TCP MAXTERMDATA 08960
SET TCP MAXREPLY 20000 SET TCP NONSTOP 0
SET TCP TCLPROG $system.system.APCP
ADD TCP SNAXAPC-TCP
```

Figura 47. File di impostazione MQSeries SNAX di esempio (Numero 1 di 3)

```
[Configurare il SNAX/APC SERVER]
RESET SERVER
SET SERVER PARAM LOGFILE APCLOG
SET SERVER PARAM TRACEFILE APCTRC
SET SERVER PARAM BACKUPCPU -1
SET SERVER PARAM MAXINRUSIZE 4096
SET SERVER PARAM MAXOUTRUSIZE 4096
SET SERVER PARAM MAXAPPLIOSIZE 4096
SET SERVER PARAM DATAPAGES 100
SET SERVER PARAM TRACEPAGES 300
SET SERVER PARAM RMTATTACHDISP QUEUE
SET SERVER PARAM RMTATTACHTIMER -1
SET SERVER PARAM CONFIG APCCFG
SET SERVER PROGRAM $system.system.APCOBJ
SET SERVER OUT $VHS
SET SERVER HOMETERM $VHS
SET SERVER PROCESS $AP02
SET SERVER NUMSTATIC 1
SET SERVER MAXSERVERS 1
SET SERVER CREATEDELAY 0 SECS
SET SERVER DELETEDELAY 1 MINS
SET SERVER CPUS 3:4
ADD SERVER SNAXAPCSVR
```

Figura 47. File di impostazione MQSeries SNAX di esempio (Numero 2 di 3)

```
[Aggiungere Listener MQSeries SNAX]
RESET SERVER
SET SERVER PROGRAM $DATA00.ZMQSEXE.RUNMQLSR
SET SERVER PROCESS $1rcv
SET SERVER NUMSTATIC 1
SET SERVER MAXSERVERS 1
SET SERVER CREATEDELAY 0 SECS
SET SERVER DELETEDELAY 1 MINS
SET SERVER STARTUP "-t LU62"
SET SERVER PARAM MQQUEMGRNAME "QMGR"
SET SERVER PARAM MQMACHINIFILE "$DATA03.QMGRD.UMQSINI"
SET SERVER PARAM MQDEFAULTPREFIX "$DATA00"
SET SERVER OUT $VHS
SET SERVER HOMETERM $VHS
SET SERVER CPUS 3:4
[ADD SERVER MQSERIES]
ADD SERVER MQSERIES
START TCP *
```

```
[Configurare il DISPATCHER]
SET TERM FILE $s.#displg
SET TERM INITIAL SNAXAPC-DISPATCHER
SET TERM TYPE CONVERSATIONAL
SET TERM TCP SNAXAPC-TCP
ADD TERM SNAXAPCSVR01 [I primi 10 caratteri costituiscono il nome del server SNAX/APC]
```

```
start server MQSERIES
start server SNAXAPCSVR
start term SNAXAPCSVR01
```

Figura 47. File di impostazione MQSeries SNAX di esempio (Numero 3 di 3)

Nota: La classe di server del Listener è identica alla classe di server MQS-TCPLIS00 nei queue manager del PATHWAY, ad eccezione del parametro di avvio in più: SET SERVER STARTUP "-t LU62"

Utilizzo di Insession ICE

Se si utilizza Insession ICE, è necessario creare un PATHWAY da utilizzare esclusivamente per questo listener. Il processo ICE non viene eseguito da questo PATHWAY.

1. Aggiungere il TP a NOF come segue:

```
ADD TP <tpname>, PROCESS <processo>, SERVERCLASS <server> [, <opzione> ...]
```

in cui:

processo

è il nome del processo PATHMON che gestisce il TP

server

è il nome del PATHMON SERVERCLASS a cui appartiene il TP può essere

opzione

- [ATTACHTIMER n] -- l'intervallo di tempo (in centesimi di secondo) in ICE attenderà un ATTACH dopo l'invio di un nuovo thread TP. Il valore predefinito è 6000 (60 secondi).
- [MAXDISPATCHTHREADS n] -- il numero massimo di TP inviati simultaneamente DEFAULT: 0 (nessun limite per il numero di DISPATCHED TP)
- [TIMEOUT n] -- stabilisce la modalità di risposta di ICE ad un ATTACH se MAXDISPATCHTHREADS è stato raggiunto su un TP TIMEOUT -1 = ATTACH in coda per TIMEOUT >0 = ATTACH in coda per n/100 secondi DEFAULT: 0 (ATTACH è rifiutato immediatamente)

Impostazione delle comunicazioni

2. È tuttavia necessario aggiungere la classe di server al PATHWAY. Alla richiesta comandi TACL, immettere:

```
TACL> Pathmon /name $PMAP, nowait, out $vhs, cpu 3/4
TACL> Pathcom $PMAP
= 0 LU62ICFG
```

dove LU62ICFG corrisponde a un file di editazione che contiene quanto segue:

```
[
SET PATHMON BACKUPCPU 6
SET PATHWAY MAXTCPS 10
SET PATHWAY MAXTERMS 10
SET PATHWAY MAXPROGRAMS 10
SET PATHWAY MAXSERVERCLASSES 10
SET PATHWAY MAXSERVERPROCESSES 10
SET PATHWAY MAXSTARTUPS 10
SET PATHWAY MAXPATHCOMS 40
SET PATHWAY MAXASSIGNS 32
SET PATHWAY MAXPARAMS 32
START PATHWAY COLD!
SET TCP PROGRAM $SYSTEM.SYSTEM.PATHTCP2
SET TCP CPUS 3:4
SET TCP MAXTERMS 5
SET TCP MAXSERVERCLASSES 010
SET TCP MAXSERVERPROCESSES 010
SET TCP MAXTERMDATA 08960SET TCP MAXREPLY 20000
SET TCP NONSTOP 0
SET TCP TCLPROG $system.system.APCP
ADD TCP SNAXAPC-TCP
```

Figura 48. File di impostazione MQSeries SNA di esempio per ICE (Numero 1 di 2)

```
[Aggiungere Listener MQSeries ICE]
RESET SERVER
SET SERVER PROGRAM $DATA00.ZMQSEXE.RUNMQLSR
SET SERVER PROCESS $lrcv
SET SERVER NUMSTATIC 1
SET SERVER MAXSERVERS 1
SET SERVER CREATEDELAY 0 SECS
SET SERVER DELETEDELAY 1 MINS
SET SERVER STARTUP "-t LU62"
SET SERVER PARAM MQQUEMGRNAME "QMGR"
SET SERVER PARAM MQMACHINIFILE "$DATA03.QMGRD.UMQSINI"
SET SERVER PARAM MQDEFAULTPREFIX "$DATA00"
SET SERVER OUT $VHS
SET SERVER HOMETERM $VHS
SET SERVER CPUS 3:4
[ADD SERVER MQSERIES]
ADD SERVER MQSERIES

START TCP *
start server MQSERIES
```

Figura 48. File di impostazione MQSeries SNA di esempio per ICE (Numero 2 di 2)

Nota: La classe di server del Listener è identica alla classe di server MQS-TCPLIS00 nei queue manager del PATHWAY, ad eccezione del parametro di avvio in più: SET SERVER STARTUP "-t LU62"

Canali TCP/IP

MQSeries for Compaq NSK consente di utilizzare più processi TCP/IP con un singolo ambiente MQSeries queue manager. Quindi è possibile selezionare i processi TCP/IP utilizzati con un queue manager associando il processo TCP/IP richiesto a un canale specifico. I canali in uscita (mittente, server, richiedente) possono indicare il nome del processo TCP/IP nel campo CONNAME della definizione del canale.

Utilizzo di **runmqsc**:

```
alter channel ... conname ('$ZTC1.123.456.789.012(1415)')
alter channel ... conname ('$ZTC1.dnshostname(1415)')
```

Utilizzo dei pannelli MQMC:

```
Processo TCPIP/SNA: $ZTC1
```

Utilizzo dei comandi PCF:

```
strncpy( pPCFString->String, ('$ZTC1.123.456.789.012(1415)', len );
```

Per riconfigurare la risoluzione DNS per un resolver non predefinito, aggiungere a tutte le classi di server PATHWAY ECnn quanto segue:

```
DEFINE =TCPIP^RESOLVER^NAME, FILE nome file
```

dove nome file è l'ubicazione del file resolver.

Se si utilizza un file host, aggiungere a tutte le classi di server PATHWAY ECnn quanto segue:

```
DEFINE =TCPIP^HOST^FILE, FILE nome file
```

dove nome file è l'ubicazione del file host.

I canali in entrata utilizzano variabili di ambiente per determinare i processi TCP/IP da utilizzare. I Listener TCP/IP trasmettono questo valore del processo ai rispettivi EC e Responder TCP/IP tramite i propri agenti. Per impostare il processo TCP/IP del canale in entrata, procedere come segue:

Utilizzando TACL:

```
ADD DEFINE =TCPIP^PROCESS^NAME, FILE nome processo
```

dove nome processo è il nome del processo TCP/IP.

Impostazione delle comunicazioni

Utilizzando PATHWAY, per le classi di server MQS-TCPLIS nn dove nn è il numero della classe di server del Listener:

```
DEFINE =TCPIP^PROCESS^NAME, FILE \HAWK.$ZTC1
PARAM MQLISTENPORTNUM "1415"
```

Per informazioni sull'utilizzo di un processo TCP/IP non predefinito per le comunicazioni via TCP/IP, fare riferimento a "Riconfigurazione della classe server MQS-TCPLIS nn per una porta ed un processo TCP/IP non predefiniti" a pagina 55. Per informazioni sulle porte TCP/IP su cui un queue manager esegue il listen, fare riferimento a "Porte TCP/IP su cui è in ascolto il queue manager" a pagina 54.

Esempi di comunicazione

Questa sezione contiene degli esempi di impostazione delle comunicazioni per SNA (SNAX e ICE) e TCP/IP.

Esempi di comunicazione SNAX

Questa sezione contiene:

- Un file di configurazione SCF di esempio per la linea SNA
- Alcuni parametri SYSGEN di esempio per il supporto della linea
- Un file di configurazione SCF di esempio per la definizione del processo SNA
- Alcune definizioni di esempio del canale MQSC

File di configurazione linea SCF SNA

Esempio di file di configurazione SCF:

```
==
== SCF configuration file for defining SNA LINE, PUs and LUs to VTAM®
== Line is called $SNA02 and SYSGEN'd into the Compaq system
==

ALLOW ALL
ASSUME LINE $SNA02

ABORT, SUB LU
ABORT, SUB PU
ABORT

DELETE, SUB LU
DELETE, SUB PU
DELETE

==
== ADD $SNA02 LINE DEFINITION
==

ADD LINE $SNA02, STATION SECONDARY, MAXPUS 5, MAXLUS 1024, RECSIZE 2048, &
CHARACTERSET ASCII, MAXLOCALLUS 256, &
PUIDBLK %H05D, PUIDNUM %H312FB

==
== ADD REMOTE PU OBJECT, LOCAL IS IMPLICITLY DEFINED AS #ZNT21
==

ADD PU #PU2, ADDRESS 1, MAXLUS 16, RECSIZE 2046, TYPE (13,21), &
TRRMTADDR 04400045121088, DYNAMIC ON, &
ASSOCIATESUBDEV $CHAMB.#p2, &
TRSSAP %H04, &
CPNAME IYAQCDRM, SNANETID GBIBMIYA
```

Impostazione delle comunicazioni

```
==
== ADD LOCAL LU OBJECT
==

ADD LU #ZNTLU1, TYPE (14,21), RECSIZE 1024, &
      CHARACTERSET ASCII, PUNAME #ZNT21, SNANAME IYAHT080

==
== ADD PARTNER LU OBJECTS
==

== spinach (HP)

ADD LU #PU2LU1, TYPE(14,21), PUNAME #PU2, SNANAME IYABT0F0

== stingray (AIX)

ADD LU #PU2LU2, TYPE(14,21), PUNAME #PU2, SNANAME IYA3T995

== coop007 (OS/2)

ADD LU #PU2LU3, TYPE(14,21), PUNAME #PU2, SNANAME IYAFT170

== MVS CICS

ADD LU #PU2LU4, TYPE(14,21), PUNAME #PU2, SNANAME IYCMVM03

== MVS Non-CICS

ADD LU #PU2LU5, TYPE(14,21), PUNAME #PU2, SNANAME IYCNVM03

== finnrl00 (NT)

ADD LU #PU2LU6, TYPE(14,21), PUNAME #PU2, SNANAME IYAFT080

== winas18 (AS400)

ADD LU #PU2LU7, TYPE(14,21), PUNAME #PU2, SNANAME IYAFT110

== MQ-Portugese (OS/2)

ADD LU #PU2LU8, TYPE(14,21), PUNAME #PU2, SNANAME IYAHT090

== VSE

ADD LU #PU2LU10, TYPE(14,21), PUNAME #PU2, SNANAME IYZMZSI2

== START UP TOKEN RING ASSOCIATE SUB DEVICE $CHAMB.#P2
== then start the line, pu's and lu's

START LINE $CHAMB, SUB ALL

START
START, SUB PU

STATUS
STATUS, SUB PU
STATUS, SUB LU
```

Parametri SYSGEN

Di seguito è riportato il contenuto del file CONFTEXT relativo a SYSGEN affinché supporti le linee SNA e token ring:

Impostazione delle comunicazioni

```
!*****
!                               LAN MACRO
!*****
! This macro is used for all 361x LAN controllers
! REQUIRES T9375 SOFTWARE PACKAGE

C3613^MLAM          = MLAM
                    TYPE 56,          SUBTYPE 0,
                    PROGRAM            C9376P00,
                    INTERRUPT          IOP^INTERRUPT^HANDLER,
                    MAXREQUESTSIZE    32000,
                    RSIZE              32000,
                    BURSTSIZE          16,
                    LINEBUFFERSIZE    32,
                    STARTDOWN #;

!*****
!                               SNAX macro for Token ring lines
!*****
TOKEN^RING^SNAX^MACRO = SNATS
                    TYPE 58,
                    SUBTYPE 4,
                    RSIZE 1024,
                    SUBTYPE 4,
                    FRAMESIZE 1036 # ;

!*****
!                               SNAX MANAGER
!*****
SSCP^MACRO          = SNASVM
                    TYPE 13,          SUBTYPE 5,
                    RSIZE              256 #;

!*****
!                               LAN CONTROLLER
!*****
LAN1      3616    0,1    %130    ;

!***** Service manager
SNAX      6999    0,1    %370    ;

!***** SNAX/Token Ring Pseudocontroller
RING      6997    0,1    %360    ;

!***** Token Ring Line
$CHAMB    LAN1.0, LAN1.1    C3613^MLAM, NAME #LAN1;

!***** Configure the SSCP
$SSCP     SNAX.0, SNAX.1 SSCP^MACRO;

!***** Sna lines for Dummy Controller over Token Ring
$SNA01    RING.0, RING.1 TOKEN^RING^SNAX^MACRO;
$SNA02    RING.2, RING.3 TOKEN^RING^SNAX^MACRO;
```

Configurazione del processo SNAX/APC

Le seguenti definizioni configurano il processo APC di esempio (nome processo \$BP01) via SCF per la linea SNA.

Nota: Il processo APC \$BP01 è definito come processo della classe di server in esecuzione nello stesso PATHWAY del Dispatcher SNAX APC.

```
==
== SCF Configuration file for SNAX/APC Lus
==
```

ALLOW ERRORS

Impostazione delle comunicazioni

```
ASSUME PROCESS $BP01

ABORT SESSION *
ABORT TPN *
ABORT PTNR-MODE *
ABORT PTNR-LU *
ABORT LU *

DELETE TPN *
DELETE PTNR-MODE *
DELETE PTNR-LU *
DELETE LU *

==
== ADD LOCAL LU
==
ADD LU IYAHT080, SNANAME GBIBMIYA.IYAHT080, SNAXFILENAME $SNA02.#ZNTLU1, &
    MAXSESSION 256, AUTOSTART YES

== TPnames for MQSeries

ADD TPN IYAHT080.MQSeries

=== Spinach (HP) Partner LU

ADD PTNR-LU IYAHT080.IYABT0F0, SNANAME GBIBMIYA.IYABT0F0, &
    PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYABT0F0.LU62PS, MODENAME LU62PS, &
    DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
    DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
    DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
    SENDWINDOW 4

==
== Winas18 (AS400) Partner LU
==

ADD PTNR-LU IYAHT080.IYAFT110, SNANAME GBIBMIYA.IYAFT110, &
    PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYAFT110.LU62PS, MODENAME LU62PS, &
    DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
    DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
    DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
    SENDWINDOW 4

==
== Stingray (AIX) Partner LU
==

ADD PTNR-LU IYAHT080.IYA3T995, SNANAME GBIBMIYA.IYA3T995, &
    PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYA3T995.LU62PS, MODENAME LU62PS, &
    DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
    DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
    DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
    SENDWINDOW 4

==
== coop007 (OS/2) Partner LU
==

ADD PTNR-LU IYAHT080.IYAFT170, SNANAME GBIBMIYA.IYAFT170, &
    PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYAFT170.LU62PS, MODENAME LU62PS, &
    DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
    DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
```

Impostazione delle comunicazioni

```
                DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
                SENDWINDOW 4
==
== MQ-Portugese (OS/2) Partner LU
==

ADD PTNR-LU    IYAHT080.IYAHT090, SNANAME GBIBMIYA.IYAHT090, &
                PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYAHT090.LU62PS, MODENAME LU62PS, &
                DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
                DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
                DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
                SENDWINDOW 4

==
== finnr100 (NT) Partner LU
==

ADD PTNR-LU    IYAHT080.IYAFT080, SNANAME GBIBMIYA.IYAFT080, &
                PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYAFT080.LU62PS, MODENAME LU62PS, &
                DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
                DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
                DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
                SENDWINDOW 4

==
== MVS CICS      Partner LU
==

ADD PTNR-LU    IYAHT080.IYCMVM03, SNANAME GBIBMIYA.IYCMVM03, &
                PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYCMVM03.LU62PS, MODENAME LU62PS, &
                DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
                DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
                DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
                SENDWINDOW 4

==
== MVS Non CICS Partner LU
==

ADD PTNR-LU    IYAHT080.IYCNVM03, SNANAME GBIBMIYA.IYCNVM03, &
                PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYCNVM03.LU62PS, MODENAME LU62PS, &
                DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
                DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
                DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
                SENDWINDOW 4

==
== VSE Partner LU
==

ADD PTNR-LU    IYAHT080.IYZMZSI2, SNANAME GBIBMIYA.IYZMZSI2, &
                PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYZMZSI2.LU62PS, MODENAME LU62PS, &
                DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
                DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
                DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
                SENDWINDOW 4

==
== Start the LUs
==
```

```
START LU IYAHT080, SUB ALL
START TPN *
```

Le applicazioni MQSeries richiedono il valore Maxapplies; tale valore controlla che la dimensione massima del trasferimento dati interprocesso tra MQSeries e il processo del server di comunicazione sia impostata su 32000, valore maggiore di quello predefinito.

Definizioni di canale

Di seguito sono riportati alcuni esempi di definizione di canale MQSeries che supportano la configurazione SNAX:

- Un canale mittente per MQSeries su MVS/ESA (non-CICS mover):

```
DEFINE CHANNEL(MT01.VM03.SDR0.0002) CHLTYPE(SDR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.SDR0.0001') +
  CONNAME('$BP01.IYAHT080.IYCNVM03') +
  MODENAME('LU62PS') TPNAME(MQSERIES)
```

- Un canale ricevente da MQSeries su MVS/ESA:

```
DEFINE CHANNEL(VM03.MT01.SDR0.0002) CHLTYPE(RCVR) +
  TRPTYPE(LU62) REPLACE DESCR('Receiver channel from VM03NCM') +
  SEQWRAP(9999999) +
  MAXMSGL(2048)
```

- Un canale server per MQSeries su MVS/ESA in grado di avviare una conversazione o ricevere una conversazione da un canale richiedente remoto:

```
DEFINE CHANNEL(MT01.VM03.RQSV.0002) CHLTYPE(SVR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.RQSV.0001') +
  CONNAME('$BP01.IYAHT080.IYCNVM03') +
  MODENAME('LU62PS') TPNAME(MQSERIES)
```

dove MQSeries è il TPNAME per il quale il queue manager MVS™ esegue il listen.

Esempi di comunicazione ICE

La configurazione di ICE per MQSeries precede due fasi:

1. È necessario configurare il processo ICE.
2. Le informazioni sulla linea (\$ICE01, nell'esempio seguente) e su SNA dovranno essere inseriti nel processo ICE.

Configurazione del processo ICE

Di seguito è riportato un esempio della configurazione del processo ICE. Tale configurazione viene ubicata per impostazione predefinita nel file GOICE:

Impostazione delle comunicazioni

```
?tacl macro
clear all
param backupcpu 1
param cinittimer 120
param collector $0
param config icectl
param idblk 05d
param idnum 312FF
param cpname IYAHR00C
param datapages 64
param dynamicrlu yes
param genesis $gen
param maxrcv 32000
param loglevel info
param netname GBIBMIYA
param password xxxxxxxxxxxxxxxxxxxxxx
param retrysl 5
param secuserid super.super
param startup %1%
param timer1 20
param timer2 300
param usstable default
run $system.ice.ice/name $ICE,nowait,cpu 0,pri 180,highpin off/
```

Note:

1. La password PARAM è stata sostituita da xxxxxxxxxxxxxxxxxxxxxx.
2. Le applicazioni MQSeries richiedono il PARAM maxrcv; tale valore controlla che la dimensione massima del trasferimento dati interprocesso tra MQSeries e il processo del server di comunicazione sia impostata su 32000, valore maggiore di quello predefinito.

Definizione delle informazioni sulla linea e su APC

Dopo aver avviato il processo ICE con questa configurazione, le seguenti informazioni vengono immesse nel processo ICE mediante il NOF** (Node Operator Facility). Questo esempio definisce la linea \$ICE01 in esecuzione sulla porta token ring \$CHAMB.#ICE:

```
==
== ICE definitions for PU IYAHR00C.
== Local LU for this PU is IYAHT0C0.
==

ALLOW ERRORS

OPEN $ICE

ABORT LINE $ICE01, SUB ALL

DELETE LINE $ICE01, SUB ALL

==
== ADD TOKEN RING LINE
==

ADD LINE $ICE01, TNDM $CHAMB.#ICE, &
      IDBLK %H05D, &
      PROTOCOL TOKENRING, WRITEBUFFERSIZE 8192

==
== ADD PU OBJECT
==
```

Impostazione delle comunicazioni

```
ADD PU IYAHRO0C, LINE $ICE01, MULTIRoute YES, &  
    DMAC 400045121088, DSAP %H04, &  
    NETNAME GBIBMIYA, IDNUM %H312FF, IDBLK %H05D, &  
    RCPNAME GBIBMIYA.IYAQCDRM, SSAP %H08
```

```
==  
== Add Local APPL Object  
==
```

```
DELETE APPL IYAHT0C0  
ADD APPL IYAHT0C0, ALIAS IYAHT0C0, PROTOCOL CPIC, &  
    OPENNAME #IYAHT0C
```

```
==  
== Add Mode LU62PS  
==
```

```
DELETE MODE LU62PS  
ADD MODE LU62PS, MAXSESS 8, MINCONWIN 4, MINCONLOS 3
```

```
==  
== Add Partner LU Objects  
==
```

```
== spinach (HP)
```

```
ABORT RLU IYABT0F0  
DELETE RLU IYABT0F0  
ADD RLU IYABT0F0, MODE LU62PS, PARSESS YES
```

```
== stingray (AIX)
```

```
ABORT RLU IYA3T995  
DELETE RLU IYA3T995  
ADD RLU IYA3T995, MODE LU62PS, PARSESS YES
```

```
== coop007 (OS/2)
```

```
ABORT RLU IYAFT170  
DELETE RLU IYAFT170  
ADD RLU IYAFT170, MODE LU62PS, PARSESS YES
```

```
== MVS CICS
```

```
ABORT RLU IYCMVM03  
DELETE RLU IYCMVM03  
ADD RLU IYCMVM03, MODE LU62PS, PARSESS YES
```

```
== MVS Non-CICS
```

```
ABORT RLU IYCNVM03  
DELETE RLU IYCNVM03  
ADD RLU IYCNVM03, MODE LU62PS, PARSESS YES
```

```
== finnr100 (NT)
```

```
ABORT RLU IYAFT080  
DELETE RLU IYAFT080  
ADD RLU IYAFT080, MODE LU62PS, PARSESS YES
```

```
== winas18 (AS400)
```

```
ABORT RLU IYAFT110  
DELETE RLU IYAFT110  
ADD RLU IYAFT110, MODE LU62PS, PARSESS YES
```

```
ABORT RLU IYAHT080
```

Impostazione delle comunicazioni

```
DELETE RLU IYAHT080
ADD RLU IYAHT080, MODE LU62PS, PARSESS YES

==
== START UP ICE LINE $ICE01 AND SUB DEVICE
==

START LINE $ICE01, SUB ALL
```

Nota: Affinché questa configurazione funzioni, è necessario che la porta #ICE sia stata definita sulla linea token ring.

Ad esempio, è possibile immettere i seguenti comandi in SCF:

```
add port $chamb.#ice, type tr8025, address %H08
start port $chamb.#ice
```

dove \$chamb è un controller token-ring e il SAP della porta è %08.

Definizioni di canale per ICE

Alcune definizioni di canale MQSeries supportano questa configurazione ICE:

- Un canale mittente per MQSeries su MVS/ESA (non-CICS mover):

```
DEFINE CHANNEL(MT01.VM03.SDRC.ICE) CHLTYPE(SDR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.SDRC.ICE') +
  CONNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+
  MODENAME('LU62PS') TPNAME(MQSERIES)
```

- Un canale ricevente da MQSeries su MVS/ESA:

```
DEFINE CHANNEL(VM03.MT01.SDRC.ICE) CHLTYPE(RCVR) +
  TRPTYPE(LU62) REPLACE DESCR('Receiver channel from VM03NCM') +
  SEQWRAP(9999999) +
  MAXMSGL(2048) +
  TPNAME(VM03NCMSDRCRCVR)
```

- Un canale server per MQSeries su MVS/ESA in grado di avviare una conversazione o ricevere una conversazione da un canale richiedente remoto:

```
DEFINE CHANNEL(MT01.VM03.RQSV.ICE) CHLTYPE(SVR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.RQSV.ICE') +
  CONNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+
  MODENAME('LU62PS') TPNAME(MQSERIES) +
```

dove MQSeries è il TPNAME per il quale il queue manager MVS esegue il listen.

Esempi di comunicazione TCP/IP

Questo esempio mostra come stabilire le comunicazioni con un sistema remoto MQSeries attraverso TCP/IP.

Stanza TCPConfig in QMINI

È necessario che il file QMINI contenga una stanza TCPConfig. Ad esempio:

```
TCPConfig:
  TCPPort=1414
  TCPNumListenerPorts=1
  TCPListenerPort=1996
  TCPKeepAlive=1
```

Il valore TCPPort è la porta in uscita predefinita per i canali per i quali non è stato specificato un valore di porta nel campo OCONNAME. TCPListenerPort identifica la porta predefinita utilizzata quando l'opzione -p non viene indicata se si utilizza `runmqtsr` da una riga comandi.

Definizione di un canale mittente TCP/IP

È necessario definire un canale mittente TCP/IP. In questo esempio, il queue manager è MH01 sull'host SPINACH:

```
DEFINE CHANNEL(MT01_MH01_SDR0001) CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  SEQWRAP(9999999) MAXMSGL(4194304) +
  XMITQ('MH01_TQ_SDR0001') +
  CONNAME('SPINACH.HURSLEY.IBM.COM(2000)')
```

Il canale tenta di collegarsi alla porta TCP/IP numero 2000 sull'host SPINACH.

Il seguente esempio contiene la definizione di canale mittente TCP/IP per un queue manager MH01 sull'host SPINACH mediante la porta TCP/IP in uscita *predefinita*:

```
DEFINE CHANNEL(MT01_MH01_SDR0001) CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  SEQWRAP(9999999) MAXMSGL(4194304) +
  XMITQ('MH01_TQ_SDR0001') +
  CONNAME('SPINACH.HURSLEY.IBM.COM')
```

In CONNAME non è stato specificato alcun numero di porta. In questo caso, viene utilizzato il valore specificato alla voce TCPPort nel file QMINI (1414).

Definizione di un canale ricevente TCP/IP

Esempio di canale ricevente TCP/IP:

```
DEFINE CHANNEL(MH01_MT01_SDR0001) CHLTYPE(RCVR) +
  TRPTYPE(TCP)
```

Un canale ricevente TCP/IP non richiede alcun valore CONNAME, ma è necessario che un listener TCP/IP sia in esecuzione. Esistono due modalità per avviare un listener TCP/IP. Uno dei seguenti:

Impostazione delle comunicazioni

1. Passare al PATHWAY del queue manager utilizzando PATHCOM ed immettere:

```
start server mqs-tcplis00
```

oppure

Alla richiesta comandi TAQL, immettere:

```
runmq1sr -m QMgrName
```

Viene avviato un listener TCP/IP, che eseguirà il listen sulla porta definita nel file QMINI (in questo esempio, 1996).

Nota: È possibile sostituire questo numero di porta dall'indicatore *-p Port* in **runmq1sr**.

Definizione di un canale mittente TCP/IP sul sistema remoto

La definizione del canale mittente sul sistema remoto per la connessione a questo canale ricevente, sarà:

```
DEFINE CHANNEL(MH01_MT01_SDR_0001) CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  XMITQ('MT01_TQ_SDR_0001') +  
  CONNAME('Compaq.ISC.UK.IBM.COM(1996)')
```

Configurazione di QMINI per il supporto di più listener TCP/IP

Per abilitare un queue manager al supporto di più listener TCP/IP, è necessario creare una nuova classe di server PATHWAY per ciascun listener aggiuntivo, in base a MQS-TCPLIS00.

Inoltre, è necessario che ciascun listener TCP/IP abbia una voce per la porta nella stanza TCPConfig del file QMINI.

Ad esempio:

```
TCPConfig:  
  TCPPort=1414  
  TCPNumListenerPorts=3  
  TCPLListenerPort=1996  
  TCPLListenerPort=1997  
  TCPLListenerPort=1998  
  TCPKeepAlive=1
```

TCPNumListenerPorts dovrà corrispondere al numero di voci TCPLListenerPort (in questo esempio tre). Il file QMINI è in grado di supportare tre listener TCP/IP attivi sulle porte 1996, 1997 e 1998. In genere le classi del server destinate al supporto delle tre poste vengono denominate MQS-TCPLIS00, MQS-TCPLIS01 e MQS-TCPLIS02.

Per ulteriori informazioni sull'aggiunta delle classi del server, fare riferimento a "Gestione di TS/MP (PATHWAY)" a pagina 32.

Appendice N. Client MQSeries

Un client MQSeries è un sistema MQSeries che non comprende un queue manager. Il codice client MQSeries indirizza le chiamate MQI provenienti dalle applicazioni in esecuzione sul sistema client di un queue manager del sistema server MQSeries con cui è collegato.

Questa appendice fornisce le informazioni relative ai client MQSeries specifiche per MQSeries for Compaq NSK V5.1. Utilizzarla insieme al manuale *MQSeries Client*.

Supporto client

MQSeries for Compaq NSK può essere utilizzato come sistema server MQSeries di tutti i client MQSeries che possono collegarsi al server utilizza i protocollo TCP/IP o SNA LU 6.2. Tuttavia, non esiste alcun client MQSeries for Compaq NSK.

Quando un client MQSeries si collega a un queue manager su MQSeries for Compaq NSK:

- Le funzioni MQGET, MQPUT o MQPUT1 con un'opzione MQ*_SYNCPOINT inizializzano una transazione Compaq, se una transazione non è già associata all'id interno della connessione.
- Le funzioni MQGET, MQPUT o MQPUT1 non associate all'opzione MQ*_SYNCPOINT o MQ*_NO_SYNCPOINT inizializzano una transazione Compaq, se una transazione non è già associata all'id interno della connessione.
- La chiamata MQCMIT esegue il commit di una transazione Compaq, se una transazione è associata all'id interno della connessione. La chiamata MQBACK annulla la transazione Compaq, se una transazione è associata all'id interno della connessione.

Nei vasi casi, se Compaq BEGINTRANSACTION non viene eseguito regolarmente, viene restituito al caller un *CompCode* di MQCC_FAILED e un *Reason* di MQRC_SYNCPOINT_NOT_AVAILABLE.

Considerazioni sulla sicurezza

MQSeries for Compaq NSK supporta l'utilizzo delle uscite di protezione del canale per la convalida dei client, come segue:

- Una volta stabilita una connessione tra il client MQSeries e il server, il client richiama l'uscita di protezione sul server prima di ritornare dalla chiamata MQCONN.
- L'uscita di protezione del server può restituire le informazioni all'uscita di protezione del client.

Questa finestra di dialogo consente, ad esempio, la trasmissione dei dati confidenziali tra il server e il client. Se il client non dispone di alcuna uscita di protezione definita i valori delle variabili di ambiente locale MQ_USER_ID e MQ_PASSWORD vengono inoltrati al server attraverso gli attributi di canale. Questi attributi sono disponibili sull'uscita di protezione server per la convalida.

Appendice O. Programmable System Management

MQSeries for Compaq NSK supporta queste funzioni di gestione del sistema di MQSeries:

- Eventi di strumentazione
- Programmable Command Format (PCF)
- Servizi installabili

La presente appendice fornisce un riepilogo di tali funzioni di MQSeries for Compaq NSK. Per una descrizione dettagliata, consultare il manuale *MQSeries Programmable System Management*.

Eventi di strumentazione

MQSeries for Compaq NSK supporta gli eventi di strumentazione MQSeries standard, che generano un messaggio di evento in una coda di evento.

È possibile abilitare e disabilitare gli eventi specificando i valori appropriati per gli attributi di coda e del Queue Manager mediante:

- MQSC, come descritto nel manuale *MQSeries Guida di riferimento per i comandi*
- Comandi PCF, come descritto nel manuale *MQSeries Programmable System Management*
- MQM (Message Queue Management), come descritto nella sezione “Capitolo 4. Gestione dei Queue Manager” a pagina 45

Tipi di evento supportati da MQSeries for Compaq NSK

MQSeries for Compaq NSK supporta i seguenti tipi di evento:

Tabella 27. I tipi di evento supportati da MQSeries for Compaq NSK

Tipo di evento	Nome evento
Eventi di autorizzazione	Not Authorized (type 1)
Eventi Channel	Channel Activated Channel Conversion Error Channel Not Activated Channel Started Channel Stopped
Eventi Inhibit	Get Inhibited Put Inhibited
Eventi locali	Alias Base Queue Type Error Queue Type Error Unknown Alias Base Queue Unknown Object Name
Eventi prestazioni	Queue Depth High Queue Depth Low Queue Full Queue Service Interval High Queue Service Interval OK

Tabella 27. I tipi di evento supportati da MQSeries for Compaq NSK (Continua)

Tipo di evento	Nome evento
Eventi remoti	Default Transmission Queue Type Error Default Transmission Queue Usage Error Queue Type Error Remote Queue Name Error Transmission Queue Usage Error Unknown Default Transmission Queue Unknown Remote Queue Manager Unknown Transmission Queue
Eventi Start e Stop	Queue Manager Active Queue Manager Not Active

Formato dei messaggi di evento

MQSeries for Compaq NSK supporta il formato di messaggio di evento standard di MQSeries. In altre parole, il messaggio di evento consta di due parti, *il descrittore messaggio* (MQMD) e *i dati di messaggio*. I dati di messaggio comprendono un'intestazione evento e alcuni dati che sono specifici del tipo di evento.

La struttura MQMD di un messaggio di evento è riepilogata nella sezione “MQMD – Message Descriptor” a pagina 343. La struttura dell'intestazione di evento (MQCFH) è riepilogata nella sezione Tabella 29 a pagina 399.

Tabella 28. Struttura MQMD di un messaggio di evento

Parametro	Tipo	Valori
<i>StrucId</i>	MQCHAR4	MQMD_STRUC_ID
<i>Version</i>	MQLONG	MQMD_VERSION_1
<i>Report</i>	MQLONG	MQRO_NONE
<i>MsgType</i>	MQLONG	MQMT_DATAGRAM
<i>Expiry</i>	MQLONG	MQEI_UNLIMITED
<i>Feedback</i>	MQLONG	MQFB_NONE
<i>Encoding</i>	MQLONG	Codifica del Queue Manager che genera l'evento.
<i>CodedCharSetId</i>	MQLONG	CCSID (Coded Character Set ID) del Queue Manager che genera l'evento.
<i>Format</i>	MQCHAR8	MQFMT_EVENT
<i>Priority</i>	MQLONG	Priorità predefinita della coda di evento, se si tratta di una coda locale oppure della relativa definizione locale al Queue Manager che genera l'evento.
<i>Persistence</i>	MQLONG	Permanenza predefinita della coda di evento, se si tratta di una coda locale oppure della relativa definizione locale al Queue Manager che genera l'evento.
<i>MsgId</i>	MQBYTE24	Il valore è generato univocamente dal Queue Manager.
<i>CorrelId</i>	MQBYTE24	MQCI_NONE
<i>BackoutCount</i>	MQLONG	Il valore è sempre 0.
<i>ReplyToQ</i>	MQCHAR48	Sempre vuoto.

Tabella 28. Struttura MQMD di un messaggio di evento (Continua)

Parametro	Tipo	Valori
<i>ReplyToQMgr</i>	MQCHAR48	Il nome del Queue Manager al sistema d'origine.
<i>UserIdentifier</i>	MQCHAR12	Sempre vuoto.
<i>AccountingToken</i>	MQBYTE32	MQACT_NONE
<i>ApplIdentityData</i>	MQCHAR32	Sempre vuoto.
<i>PutApplType</i>	MQLONG	Tipo di applicazione che ha inserito il messaggio.
<i>PutApplName</i>	MQCHAR28	Nome dell'applicazione che ha inserito il messaggio.
<i>PutDate</i>	MQCHAR8	Data in cui il messaggio è stato inserito, generata dal Queue manager.
<i>PutTime</i>	MQCHAR8	Ora in cui il messaggio è stato inserito, generata dal Queue manager.
<i>ApplOriginData</i>	MQCHAR4	Sempre vuoto.

Tabella 29. Struttura intestazione evento (MQCFH)

Parametro	Tipo	Valori
<i>Type</i>	MQLONG	MQCFT_EVENT
<i>StrucLength</i>	MQLONG	MQCFH_STRUC_LENGTH
<i>Version</i>	MQLONG	MQCFH_VERSION_1
<i>Command</i>	MQLONG	MQCMD_Q_MGR_EVENT MQCMD_PERFM_EVENT MQCMD_CHANNEL_EVENT
<i>MsqSeqNumber</i>	MQLONG	Sempre 1.
<i>Control</i>	MQLONG	MQCFC_LAST
<i>CompCode</i>	MQLONG	MQCC_OK MQCC_WARNING
<i>Reason</i>	MQLONG	Codice ragione che identifica l'evento.
<i>ParameterCount</i>	MQLONG	Il numero di strutture di parametro che seguono la struttura MQCFH.

Programmable Command Format (PCF)

MQSeries for Compaq NSK supporta le funzioni PCF (Programmable Command Format) standard, come descritto nel manuale *MQSeries Programmable System Management*. I messaggi PCF sono composti da due parti, il *descrittore messaggio* (MQMD) e i *dati di messaggio*. I dati messaggio comprendono un'intestazione PCF (MQCFH) e alcuni parametri PCF definiti dalle strutture MQCFIN, MQCFIL, MQCFST e MQCFSL.

Il descrittore messaggio PCF (MQMD) è riepilogato nella sezione Tabella 30 a pagina 400. La struttura dell'intestazione PCF (MQCFH) è riepilogata nella sezione Tabella 31 a pagina 400. Le strutture del parametro PCF sono riepilogate nella sezione Tabella 32 a pagina 401 attraverso Tabella 35 a pagina 402.

Descrittore messaggio PCF

Per MQSeries for Compaq NSK, si applica il descrittore messaggio PCF standard, cioè il descrittore messaggio contiene i seguenti campi:

Tabella 30. Descrittore messaggio PCF

Campo	Valori
<i>Report</i>	Un valore valido
<i>MsgType</i>	MQMT_REQUEST
<i>Expiry</i>	Un valore valido
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	Codifica utilizzata per i dati di messaggio; se necessario, viene eseguita una conversione.
<i>CodedCharSetId</i>	CCSID utilizzato per i dati di messaggio; se necessario, viene eseguita una conversione.
<i>Format</i>	MQFMT_ADMIN MQFMT_PCF (per i dati utente)
<i>Priority</i>	Un valore valido
<i>Persistence</i>	Un valore valido
<i>MsgId</i>	Un valore valido, compreso MQMI_NONE
<i>CorrelId</i>	Un valore valido, compreso MQMI_NONE
<i>ReplyToQ</i>	Nome coda
<i>ReplyToQMgr</i>	Il nome del Queue Manager
Campi di contesto del messaggio	Un valore valido, compreso MQPMO_DEFAULT_CONTEXT

Intestazione PCF (MQCFH)

Per MQSeries for Compaq NSK, si applica l'intestazione PCF standard, cioè la struttura di intestazione PCF contiene i seguenti campi:

Tabella 31. Intestazione PCF

Campo	Tipo	Valori
<i>Type</i>	MLONG	MQCFT_COMMAND MQCFT_RESPONSE MQCFT_EVENT
<i>StrucLength</i>	MLONG	MQCFH_STRUC_LENGTH
<i>Version</i>	MLONG	MQCFH_VERSION_1
<i>Command</i>	MLONG	ID comando valido.
<i>MsgSeqNumber</i>	MLONG	Numero sequenza del messaggio.
<i>Control</i>	MLONG	MQCFC_LAST MQCFC_NOT_LAST
<i>CompCode</i>	MLONG	MQCC_OK MQCC_WARNING MQCC_FAILED MQCC_UNKNOWN
<i>Reason</i>	MLONG	Codice ragione che qualifica il codice di completamento.
<i>ParameterCount</i>	MLONG	Conteggio delle strutture di parametro.

Parametro di stringa PCF (MQCFST)

Per MQSeries for Compaq NSK, si applica la struttura standard del parametro di stringa PCF (MQCFST), cioè la struttura del parametro di stringa PCF contiene i seguenti campi:

Tabella 32. Parametro di stringa PCF

Campo	Tipo	Valore
<i>Type</i>	MQLONG	MQCFT_STRING
<i>StrucLength</i>	MQLONG	Lunghezza in byte della struttura MQCFST.
<i>Parameter</i>	MQLONG	ID parametro.
<i>CodedCharSetId</i>	MQLONG	Coded character set identifier (CCSID).
<i>StringLength</i>	MQLONG	Lunghezza in byte dei dati nel campo <i>String</i> .
<i>String</i>	MQCHAR × <i>StringLength</i>	Valore stringa.

Parametro integer list PCF (MQCFIL)

Per MQSeries for Compaq NSK, si applica la struttura standard del parametro integer list PCF (MQCFIL), cioè la struttura del parametro integer list PCF contiene i seguenti campi:

Tabella 33. Integer list PCF

Campo	Tipo	Valore
<i>Type</i>	MQLONG	MQCFT_INTEGER_LIST
<i>StrucLength</i>	MQLONG	Lunghezza in byte della struttura MQCFIL.
<i>Parameter</i>	MQLONG	ID parametro.
<i>Count</i>	MQLONG	Numero di elementi nella schiera <i>Values</i> .
<i>Values</i>	MQLONG × <i>Count</i>	Valori di parametro.

Integer PCF (MQCFIN)

Per MQSeries for Compaq NSK, si applica la struttura standard integer PCF (MQCFIN), cioè la struttura integer PCF contiene i seguenti campi:

Tabella 34. Integer PCF

Campo	Tipo	Valore
<i>Type</i>	MQLONG	MQCFT_INTEGER
<i>StrucLength</i>	MQLONG	MQCFIN_STRUC_LENGTH
<i>Parameter</i>	MQLONG	ID di parametro
<i>Value</i>	MQLONG	Valore di parametro

String list PCF (MQCFSL)

Per MQSeries for Compaq NSK, si applica la struttura string list PCF (MQCFSL), cioè la struttura di string list PCF contiene i seguenti campi:

Tabella 35. String list PCF

Campo	Tipo	Valore
Type	MLONG	MQCFT_STRING_LIST
StrucLength	MLONG	Lunghezza in byte della struttura MQCFSL
Parameter	MLONG	ID di parametro
CodedCharSetId	MLONG	CCSID dei dati nel campo <i>Strings</i> .
Count	MLONG	Numero di stringhe nel campo <i>Strings</i> .
StringLength	MLONG	Lunghezza in byte nel campo <i>Strings</i> .
Strings	MQCHAR \times StringLength \times Count	Gruppo di valori di stringa per il parametro identificato dal campo <i>Parameter</i> .

Comandi PCF supportati da MQSeries for Compaq NSK

I seguenti comandi PCF di MQSeries sono supportati da MQSeries for Compaq NSK. Per una descrizione completa di tali comandi, consultare il manuale *MQSeries Programmable System Management*.

Tabella 36. Comandi PCF supportati da MQSeries for Compaq NSK

Comando	ID comando
Change Channel	MQCMD_CHANGE_CHANNEL
Change Namelist	MQCMD_CHANGE_NAMELIST
Change Process	MQCMD_CHANGE_PROCESS
Change Queue	MQCMD_CHANGE_Q
Modifica Queue Manager (Change Queue Manager)	MQCMD_CHANGE_Q_MGR
Clear Queue	MQCMD_CLEAR_Q
Copy Channel	MQCMD_COPY_CHANNEL
Copy Namelist	MQCMD_COPY_NAMELIST
Copy Process	MQCQ_COPY_PROCESS
Copy Queue	MQCMD_COPY_Q
Create Channel	MQCMD_CREATE_CHANNEL
Create Namelist	MQCMD_CREATE_NAMELIST
Create Process	MQCMD_CREATE_PROCESS
Create Queue	MQCMD_CREATE_Q
Delete Channel	MQCMD_DELETE_CHANNEL
Delete Namelist	MQCMD_DELETE_NAMELIST
Delete Process	MQCMD_DELETE_PROCESS
Delete Queue	MQCMD_DELETE_Q
Escape	MQCMD_ESCAPE
Inquire Channel	MQCMD_INQUIRE_CHANNEL
Inquire Channel Names	MQCMD_INQUIRE_CHANNEL_NAMES
Inquire Channel Status	MQCMD_INQUIRE_CHANNEL_STATUS

Tabella 36. Comandi PCF supportati da MQSeries for Compaq NSK (Continua)

Comando	ID comando
Inquire Cluster Queue Manager	MQCMD_INQUIRE_CLUSTER_Q_MGR
Inquire Namelist	MQCMD_INQUIRE_NAMELIST
Inquire Process	MQCMD_INQUIRE_PROCESS
Inquire Process Names	MQCMD_INQUIRE_PROCESS_NAMES
Inquire Queue	MQCMD_INQUIRE_Q
Interroga Queue Manager (Inquire Queue Manager)	MQCMD_INQUIRE_Q_MGR
Inquire Queue Names	MQCMD_INQUIRE_Q_NAMES
Ping Channel	MQCMD_PING_CHANNEL
Ping Queue Manager	MQCMD_PING_Q_MGR
Refresh Cluster	MQCMD_REFRESH_CLUSTER
Reset Channel	MQCMD_RESET_CHANNEL
Reset Queue Statistics	MQCMD_RESET_Q_STATS
Resolve Channel	MQCMD_RESOLVE_CHANNEL
Resume Queue Manager Cluster	MQCMD_RESUME_Q_MGR_CLUSTER
Start Channel	MQCMD_START_CHANNEL
Stop Channel	MQCMD_STOP_CHANNEL
Suspend Queue Manager Cluster	MQCMD_SUSPEND_Q_MGR_CLUSTER

Nota: MQSeries for Compaq NSK non supporta i comandi Start Channel Initiator e Start Channel Listener.

Risposte dei comandi PCF

In MQSeries for Compaq NSK, il server di comandi genera messaggi di risposta standard a ciascun comando PCF. Esistono tre tipi di risposta:

- Risposta OK
- Risposta di errore
- Risposta di dati

Per ulteriori informazioni, consultare il manuale *MQSeries Programmable System Management*.

Servizi installabili

MQSeries for Compaq NSK supporta i servizi Authorization Service e Name Service.

Interfaccia di Authorization Service

Authorization Service consente ai Queue Manager di richiamare funzioni di autorizzazione. Ad esempio, un Queue Manager può verificare che un particolare ID utente sia autorizzato ad aprire una coda mediante Authorization Service.

Servizi installabili

Un componente Authorization Service è fornito con MQSeries for Compaq NSK. Tale componente è denominato Object Authority Manager (OAM). Per impostazione predefinita, OAM è attivo e funziona con i comandi di controllo **dspmqaout** (display authority) e **setmqaut** (set authority).

È possibile incrementare o sostituire OAM con il proprio componente Authorization Service, come descritto nel manuale *MQSeries Programmable System Management*.

Interfaccia di Name Service

Name Service fornisce supporto al Queue Manager per la risoluzione del nome del Queue Manager proprietario di una coda.

L'interfaccia standard di Name Service, come descritto nel manuale *MQSeries Programmable System Management*, è supportata da MQSeries for Compaq NSK.

Appendice P. Modello di evento EMS utilizzato da MQSeries for Compaq NSK

Il file di modello EMS (SMQSTMPL) contiene il codice di origine per le definizioni degli eventi MQSeries EMS. Tali definizioni controllano come vengono visualizzate le informazioni nei messaggi di eventi EMS e mostrano anche il tipo e il significato dei dati contenuti in ciascun messaggio di eventi EMS.

I seguenti tipi di evento sono creati:

ZMQS-VAL-EVT-ERROR

Un FFST (un problema risorse di sistema, un problema relativo al software oppure all'hardware).

ZMQS-VAL-EVT-ERR

Un errore con MQSeries, che fa riferimento ad un evento FFST e ai dati registrati in un disco.

ZMQS-VAL-EVT-MSG

Un messaggio MQSeries, come l'avvio di queue manager o di un canale. Tutti questi eventi corrispondono ad un messaggio di log MQSeries AMQxxxxx e contengono le stesse informazioni e lo stesso testo. I dati della variabile in ciascun messaggio sono contenuti in token individuali nell'ambito del messaggio di eventi. Per ulteriori informazioni sui messaggi AMQxxxxx, consultare il manuale *MQSeries Messages*.

ZMQS-VAL-EVT-QMGR

Un evento queue manager per eventi authority, inhibit, local, remote, start e stop. Tali eventi EMS dispongono effettivamente dello stesso contenuto di informazioni come i messaggi di eventi corrispondenti PCF, descritti nel manuale *MQSeries Programmable System Management*. I token individuali nel messaggio di eventi contiene i dati della variabile in ciascun messaggio di evento.

ZMQS-VAL-EVT-PERF

Un evento prestazione, che corrisponde agli eventi prestazione MQSeries standard. Tali eventi presentano i dati statistici sulle code nell'ambito di un queue manager. I dati della variabile negli eventi prestazione sono contenuti in token individuali nell'ambito del messaggio di eventi.

ZMQS-VAL-EVT-CHNL

Un evento canale, che corrisponde agli eventi canale MQSeries standard. Gli eventi canale riportano le modifiche relative allo stato dei canali o i problemi nella comunicazione tra i queue manager. Come per gli altri tipi di messaggi di eventi, i dati della variabile negli eventi canale sono contenuti in token individuali nell'ambito del messaggio di eventi.

Questo è un estratto dalle definizioni del modello EMS:

```
VERSION: "IBM.MQS - 10JAN97"  
SSID: ZMQS-VAL-SSID  
SSNAME: "MQSeries", "MQS"
```

```
==
```

```
== Questo è un messaggio EMS FFST
```

```
==
```

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-ERROR  
OVERRIDE ZEMS-TKN-EMPHASIS ZSPI-VAL-TRUE
```

modello di evento EMS

```

MQSeries FFST dal componente COMP_<1>
"<*CR> Codice errore : <2>
"<*CR> Gravità : <3>
"<*CR> Nome modulo : <4>
"<*CR> ID analisi : <5>
"<*CR> Testo errore :
"<*CR> <6>"
1: ZMQS-TKN-COMPONENT
2: ZMQS-TKN-ERROR-CODE

3: ZMQS-TKN-SEVERITY
4: ZMQS-TKN-MODULE-NAME
5: ZMQS-TKN-PROBE-ID
6: ZMQS-TKN-ERROR-TEXT

==
== Questo è un evento di visualizzazione messaggio EMS
==
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-MSG
"Messaggio MQSeries: <1>
"
"<*CR> EXPLANATION :
" <*CR> <2>
"
"<*CR> ACTION :
" <*CR> <3>"
1: ZMQS-TKN-ERROR-TEXT
2: ZMQS-TKN-ERROR-TEXT-2
3: ZMQS-TKN-ERROR-TEXT-3

==
== Questo è un evento di notifica errore EMS
==
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-ERR
OVERRIDE ZEMS-TKN-EMPHASIS ZSPI-VAL-TRUE
"Errore MQSeries
"<*CR> Codice errore : <1>
"<*CR> Funzione : <2>
"<*CR> ID analisi : <3>
"<*CR> File FFST : <4>
1: ZMQS-TKN-ERROR-CODE
2: ZMQS-TKN-MODULE-NAME
3: ZMQS-TKN-PROBE-ID
4: ZMQS-TKN-FILE-NAME

==
== Questa è una copia EMS di messaggi di evento Queue Manager PCF
== per eventi di tipo authority, inhibit, local, remote, start_and_stop
==
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-QMGR
"Evento QMgr MQSeries da <1>
"<*CR> Ragione : <2>
"<*IF 3><*CR> Qualificatore ragione : <4>
"<*IF 5><*CR> ID utente : <6>
"<*IF 13><*CR> QMgr oggetto : <14>
"<*IF 9><*CR> Opzioni : <10>
"<*IF 11><*CR> Comando : <12>
"<*IF 15><*CR> Nome coda : <16>
"<*IF 17><*CR> Tipo coda : <18>
"<*IF 19><*CR> Nome coda di base : <20>
"<*IF 21><*CR> Nome coda XMit : <22>
"<*IF 30><*CR> Tipo applicazione : <31>
"<*IF 32><*CR> Nome applicazione : <33>
1: ZMQS-TKN-QMGR

```

- 2: ZMQS-TKN-REASON
- 3: TOKENPRESENT(ZMQS-TKN-REASON-QUALIFIER)
- 4: ZMQS-TKN-REASON-QUALIFIER
- 5: TOKENPRESENT(ZMQS-TKN-USER-ID)
- 6: ZMQS-TKN-USER-ID
- 9: TOKENPRESENT(ZMQS-TKN-OPTIONS)
- 10: ZMQS-TKN-OPTIONS
- 11: TOKENPRESENT(ZMQS-TKN-COMMAND)
- 12: ZMQS-TKN-COMMAND
- 13: TOKENPRESENT(ZMQS-TKN-OBJ-QMGR)
- 14: ZMQS-TKN-OBJ-QMGR
- 15: TOKENPRESENT(ZMQS-TKN-Q-NAME)
- 16: ZMQS-TKN-Q-NAME
- 17: TOKENPRESENT(ZMQS-TKN-Q-TYPE)
- 18: ZMQS-TKN-Q-TYPE
- 19: TOKENPRESENT(ZMQS-TKN-BASE-Q-NAME)
- 20: ZMQS-TKN-BASE-Q-NAME
- 21: TOKENPRESENT(ZMQS-TKN-XMIT-Q-NAME)
- 22: ZMQS-TKN-XMIT-Q-NAME
- 30: TOKENPRESENT(ZMQS-TKN-APPL-TYPE)
- 31: ZMQS-TKN-APPL-TYPE
- 32: TOKENPRESENT(ZMQS-TKN-APPL-NAME)
- 33: ZMQS-TKN-APPL-NAME

==
 == Questa è una copia EMS di messaggi di eventi prestazione PCF
 ==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-PERF
"evento prestazione MQSeries da <1>          "
"<*CR> Ragione : <2>                          "
"<*CR> Nome coda : <3>                          "
"<*CR> Ora dall'ultimo ripristino : <4>         "
"<*CR> Capacità massima della coda : <5>
" "<*CR> # dei messaggi in coda : <6>          "
"<*CR> # dei messaggi non in coda : <7>        "
1: ZMQS-TKN-QMGR
2: ZMQS-TKN-REASON
3: ZMQS-TKN-Q-NAME
4: ZMQS-TKN-TIME-SINCE-RESET
5: ZMQS-TKN-HIGH-Q-DEPTH
6: ZMQS-TKN-MSG-ENQ-COUNT
7: ZMQS-TKN-MSG-DEQ-COUNT
```

==
 == Questa è una copia EMS di messaggi di eventi canale PCF
 ==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-CHNL
"Eventi canale MQSeries da <1>          "
"<*CR> Ragione : <2>                          "
"<*CR> Nome canale : <3>                       "
"<*CR> XMit Nome coda : <5>                    "
"<*CR> Nome connessione : <7>                 "
"<*CR> Qualificatore ragione : <9>            "
"<*CR> Formato : <11>                          "
"<*CR> Codice di ritorno : <13>                 "
"<*CR> rc ausiliario 1 : <15>                   "
"<*CR> rc ausiliario 2 : <17>                   "
"<*CR> CCSID 1 : <19>                           "
"<*CR> Stringa ausiliario 1 : <21>              "
"<*CR> CCSID 2 : <23>                           "
"<*CR> Stringa ausiliaria 2 : <25>              "
"<*CR> CCSID 3 : <27>                           "
"<*CR> Stringa ausiliaria 3 : <29>              "
1: ZMQS-TKN-QMGR
2: ZMQS-TKN-REASON
```

modello di evento EMS

3: ZMQS-TKN-CHANNEL-NAME
5: ZMQS-TKN-XMIT-Q-NAME
7: ZMQS-TKN-CONN-NAME
9: ZMQS-TKN-REASON-QUALIFIER
11: ZMQS-TKN-FORMAT
13: ZMQS-TKN-RETURN-CODE
15: ZMQS-TKN-RETURN-CODE-2
17: ZMQS-TKN-RETURN-CODE-3
19: ZMQS-TKN-CCSID
21: ZMQS-TKN-ERROR-TEXT
23: ZMQS-TKN-CCSID-2
25: ZMQS-TKN-ERROR-TEXT-2
27: ZMQS-TKN-CCSID-3
29: ZMQS-TKN-ERROR-TEXT-3

Appendice Q. Informazioni particolari

Queste informazioni sono state sviluppate per i prodotti e servizi offerti negli Stati Uniti. E' possibile che negli altri paesi l'IBM non offra i prodotti, le funzioni o i servizi illustrati in questo documento. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti dall'IBM, possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. E' responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri programmi e/o prodotti, fatta eccezione per quelli espressamente indicati dall'IBM.

L'IBM può avere brevetti o domande di brevetto in corso relativi a quanto trattato nella presente pubblicazione. La fornitura di questa pubblicazione non implica la concessione di alcuna licenza su di essi. Chi desiderasse ricevere informazioni relative alle licenze può rivolgersi per iscritto a:

Director of Commercial Relations
IBM Europe
Schoenaicher str. 220
D-7030 Boeblingen
Deutschland

Il seguente paragrafo non è valido per il Regno Unito o per tutti i paesi le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute:

L'INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA", SENZA ALCUNA GARANZIA, ESPLICITA O IMPLICITA, IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZABILITÀ ED IDONEITÀ AD UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe non essere a voi applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche verranno incorporate nelle nuove edizioni della pubblicazione. L'IBM si riserva il diritto di apportare miglioramenti e/o modifiche al prodotto o al programma descritto in questa pubblicazione in qualsiasi momento e senza preavviso.

Tutti i riferimenti a siti Web non dell'IBM contenuti in questo documento sono forniti unicamente a scopo di consultazione. I materiali contenuti in tali siti Web non fanno parte di questo prodotto e l'utente si assume ogni rischio relativo al loro utilizzo.

L'IBM può utilizzare o divulgare le informazioni ricevute dagli utenti secondo le modalità ritenute appropriate, senza alcun obbligo nei loro confronti.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

Informazioni particolari

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

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

Il programma su licenza descritto in queste informazioni e tutto il materiale su licenza ad esso relativo sono forniti dall'IBM nel rispetto delle condizioni previste dalla licenza d'uso.

Le informazioni relative a prodotti non IBM sono state ottenute dai fornitori di tali prodotti. L'IBM non ha verificato tali prodotti e, pertanto, non può garantirne l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

LICENZA RELATIVA AI DIRITTI D'AUTORE:

Queste informazioni contengono programmi applicativi di esempio in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. Potete copiare, modificare o distribuire questi esempi di programmi sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (Application Programming Interface) a seconda della piattaforma operativa per cui tali esempi di programmi sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. La IBM, quindi, non può garantire o assicurare l'affidabilità, la praticità o il funzionamento di questi programmi. Potete copiare, modificare e distribuire questi esempi di programmi sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in modo conforme alle API (Application Programming Interface) della IBM.

Marchi

I seguenti termini sono marchi della IBM Corporation negli Stati Uniti e/o in altri paesi:

AIX	IBM
MQSeries	AS/400
MVS/ESA	FFST
CICS	OS/2
First Failure Support Technology	VSE/ESA
OS/390	BookManager
IBMLink	MVS
SupportPac	VTAM

Compaq e NonStop sono marchi della Compaq Computer Corporation.

Intel è un marchio registrato di Intel Corporation negli Stati Uniti e/o in altri paesi.

Informazioni particolari

Microsoft, Windows e Windows NT sono marchi della Microsoft Corporation.

Java e tutti i logo basati su Java sono marchi della Sun Microsystems, Inc. negli Stati Uniti e/o in altri paesi.

UNIX è un marchio registrato negli Stati Uniti e in altri paesi concesso in licenza esclusivamente tramite la X/Open Company Limited.

Nomi di altri servizi, prodotti o società possono essere marchi di altre società.

Bibliografia

Questa sezione descrive la documentazione disponibile per tutti i prodotti correnti MQSeries.

MQSeries pubblicazioni multiplatforma

La maggior parte di queste pubblicazioni, a volte indicate come manuali della “famiglia” MQSeries, sono validi per tutti i prodotti MQSeries. I prodotti MQSeries più recenti sono:

- MQSeries per AIX, V5.2
- MQSeries per AS/400, V5.2
- MQSeries per AT&T GIS UNIX, V2.2
- MQSeries for Compaq OpenVMS Alpha, V5.1
- MQSeries per Compaq Tru64 UNIX, V5.1
- MQSeries per HP-UX, V5.2
- MQSeries per Linux, V5.2
- MQSeries per OS/2 Warp, V5.1
- MQSeries per OS/390, V5.2
- MQSeries per SINIX e DC/OSx, V2.2
- MQSeries per Sun Solaris, V5.2
- MQSeries per Sun Solaris, Intel Platform Edition, V5.1
- MQSeries for Compaq NSK, V5.1
- MQSeries per VSE/ESA, V2.1.1
- MQSeries per Windows, V2.0
- MQSeries per Windows, V2.1
- MQSeries per Windows NT e Windows 2000, V5.2

Le pubblicazioni multiplatforma MQSeries sono:

- *MQSeries Brochure*, G511-1908
- *An Introduction to Messaging and Queuing*, GC33-0805
- *MQSeries Intercommunication*, SC33-1872
- *MQSeries Queue Manager Clusters*, SC34-5349
- *MQSeries Client*, GC13-2676
- *MQSeries System Administration*, SC33-1873
- *MQSeries Guida di riferimento per i comandi*, SC13-2823
- *MQSeries Event Monitoring*, SC34-5760
- *MQSeries Programmable System Management*, SC33-1482
- *MQSeries Administration Interface Programming Guide and Reference*, SC34-5390
- *MQSeries Messages*, GC33-1876
- *MQSeries Application Programming Guide*, SC33-0807

- *MQSeries for Tandem Non Stop Kernel, V5.1 Guida operativa*, SC33-1673
- *MQSeries Programming Interfaces Reference Summary*, SX33-6095
- *MQSeries Using C++*, SC33-1877
- *MQSeries Using Java*, SC34-5456
- *MQSeries Application Messaging Interface*, SC34-5604

Pubblicazioni MQSeries per piattaforme specifiche

Ogni prodotto MQSeries è riportato in almeno una delle seguenti pubblicazioni, oltre che nei manuali della famiglia MQSeries.

MQSeries per AIX, V5.2

MQSeries per AIX Guida rapida,
GC13-2677

MQSeries per AS/400, V5.2

MQSeries per AS/400 Guida operativa,
GC13-2839

*MQSeries per AS/400 System
Administration*, SC34-5558

*MQSeries per AS/400 Application
Programming Reference (ILE RPG)*,
SC34-5559

MQSeries per AT&T GIS UNIX, V2.2

*MQSeries per AT&T GIS UNIX System
Management Guide*, SC33-1642

MQSeries for Compaq OpenVMS Alpha, V5.1

*MQSeries per Compaq OpenVMS Alpha
Guida operativa*, GC13-2967

*MQSeries per Compaq OpenVMS Alpha
Guida alla gestione del sistema*,
SC13-2966

MQSeries for Compaq NSK, V5.1

*MQSeries for Compaq NSK . Guida
operativa*, GC13-2969-00

*MQSeries for Compaq NSK - Guida alla
gestione del sistema*, SC13-2968-00

MQSeries per Compaq Tru64 UNIX, V5.1

*MQSeries per Compaq Tru64 UNIX
Guida rapida*, GC13-2882

MQSeries per HP-UX, V5.2

Bibliografia

MQSeries per HP-UX Guida rapida,
GC13-2679

MQSeries per Linux, V5.2

MQSeries per Linux Guida operativa,
GC13-2862

MQSeries per OS/2 Warp, V5.1

MQSeries per OS/2 Warp Guida rapida,
GC13-2678

MQSeries per OS/390, V5.2

*MQSeries per OS/390 Concepts and
Planning Guide*, GC34-5650

*MQSeries per OS/390 System Setup
Guide*, SC34-5651

*MQSeries per OS/390 System
Administration Guide*, SC34-5652

*MQSeries for OS/390 System
Administration Guide*, GC34-5892

*MQSeries for OS/390 Messages and
Codes*, GC34-5891

*MQSeries for OS/390 Licensed Program
Specifications*, GC34-5893

MQSeries for OS/390 Program Directory

MQSeries link for R/3, Version 1.2

*MQSeries collegamento per R/3 User's
Guide*, GC13-2743

MQSeries per SINIX e DC/OSx, V2.2

*MQSeries per SINIX e DC/OSx System
Management Guide*, GC33-1768

MQSeries per Sun Solaris, V5.2

MQSeries per Sun Solaris Guida rapida,
GC13-2680

MQSeries per Sun Solaris, Intel Platform Edition, V5.1

*MQSeries for Sun Solaris, Intel Platform
Edition Quick Beginnings*, GC34-5851

MQSeries per VSE/ESA, V2.1.1

*MQSeries for VSE/ESA Licensed Program
Specifications*, GC34-5365

*MQSeries per VSE/ESA System
Management Guide*, GC34-5364

MQSeries per Windows, V2.0

MQSeries per Windows User's Guide,
GC33-1822

MQSeries per Windows, V2.1

MQSeries per Windows User's Guide,
GC33-1965

MQSeries per Windows NT e Windows 2000, V5.2

*MQSeries per Windows NT e Windows
2000 Guida rapida*, GC13-2818

*MQSeries per Windows NT Using the
Component Object Model Interface*,
SC34-5387

MQSeries LotusScript Extension,
SC34-5404

Manuali in formato elettronico

Molti manuali MQSeries sono disponibili sia in formato cartaceo che in formato elettronico.

Formato HTML

Documentazione MQSeries è fornita in formato HTML con questi prodotti MQSeries:

- MQSeries per AIX, V5.2
- MQSeries per AS/400, V5.2
- MQSeries for Compaq OpenVMS Alpha, V5.1
- MQSeries per Compaq Tru64 UNIX, V5.1
- MQSeries per HP-UX, V5.2
- MQSeries per Linux, V5.2
- MQSeries per OS/2 Warp, V5.1
- MQSeries per OS/390, V5.2
- MQSeries per Sun Solaris, V5.2
- MQSeries per Sun Solaris, Intel Platform Edition, V5.1
- MQSeries per Windows NT e Windows 2000, V5.2 (compiled HTML)
- MQSeries link for R/3, V1.2

I manuali MQSeries in formato HTML sono disponibili anche nel sito Web dei prodotti della famiglia MQSeries all'indirizzo:

<http://www.ibm.com/software/mqseries/>

Portable Document Format (PDF)

I file PDF possono essere letti e stampati utilizzando Adobe Acrobat Reader.

Se non si possiede Adobe Acrobat Reader o si desidera ottenere informazioni aggiornate sulle piattaforme che supportano Acrobat Reader, visitare il sito Web Adobe Systems Inc. all'indirizzo:

<http://www.adobe.com/>

Versioni PDF relative ai manuali MQSeries sono fornite con i seguenti prodotti MQSeries:

- MQSeries per AIX, V5.2
- MQSeries per AS/400, V5.2
- MQSeries for Compaq OpenVMS Alpha, V5.1

- MQSeries per Compaq Tru64 UNIX, V5.1
- MQSeries per HP-UX, V5.2
- MQSeries per Linux, V5.2
- MQSeries per OS/2 Warp, V5.1
- MQSeries per OS/390, V5.2
- MQSeries per Sun Solaris, V5.2
- MQSeries per Sun Solaris, Intel Platform Edition, V5.1
- MQSeries per Windows NT e Windows 2000, V5.2
- MQSeries link for R/3, V1.2

Versioni PDF di tutti gli attuali manuali MQSeries sono inoltre disponibili nel sito Web dei prodotti della famiglia MQSeries all'indirizzo:

<http://www.ibm.com/software/mqseries/>

Formato BookManager

La libreria MQSeries è disponibile in formato IBM BookManager in una varietà di librerie in linea, tra cui *Transaction Processing and Data collection kit*, SK2T-0730. Per visualizzare questi manuali in formato IBM BookManager, utilizzare i seguenti programmi su licenza IBM:

BookManager READ/2
 BookManager READ/6000
 BookManager READ/DOS
 BookManager READ/MVS
 BookManager READ/VM
 BookManager READ for Windows

Formato PostScript

La libreria MQSeries viene fornita in formato PostScript (.PS) con molti prodotti MQSeries Versione 2. I manuali in formato PostScript possono essere stampati utilizzando una stampante PostScript e possono essere letti utilizzando un apposito visualizzatore.

Formato Windows Help

Il testo *MQSeries for Windows User's Guide* viene fornito in formato Windows Help con MQSeries per Windows, Versione 2.0 e MQSeries per Windows, Versione 2.1.

Informazioni su MQSeries disponibili su Internet

Il sito Web della famiglia dei prodotti MQSeries è all'indirizzo:

<http://www.ibm.com/software/mqseries/>

Seguendo i link contenuti in questo sito Web è possibile:

- Ottenere informazioni aggiornate relative alla famiglia di prodotti MQSeries.
- Accedere ai manuali MQSeries in formato HTML e PDF.
- Scaricare MQSeries SupportPac.

Pubblcazioni correlate

- *SNAX/APC Planning and Configuration Manual*, (Numero parte Compaq 098289)
 SNAX/APC fornisce il supporto LU 6.2 per l'implementazione Compaq di SNA. In questa guida viene spiegato come installare e configurare SNAX/APC.
- *SCF Reference Manual for SNAX/APC*, (Numero parte Compaq 064525)
 SNAX/APC fornisce il supporto LU 6.2 per l'implementazione Compaq di SNA. Questo manuale descrive l'interfaccia interattiva SCF (Subsystem Control Facility) che consente agli operatori ed agli amministratori di rete di configurare e controllare SNAX/APC.
- *Pathway System Management Guide*, (Numero parte Compaq)
 Questo manuale contiene le informazioni di base necessarie per configurare e controllare i sistemi di elaborazione delle transazioni Pathway.
- *Introduction to NonStop Transaction Manager/MP (TM/MP)*, (Numero parte Compaq 085812)
 Questo manuale descrive come utilizzare il sottosistema TMF per proteggere le proprie transazioni aziendali e l'integrità dei database utilizzati.
- *Introduzione ai Servizi di rete e alle Comunicazioni di dati di Compaq*, (Numero di codice Compaq 093148)
 Questa guida fornisce una panoramica dei concetti, delle attività, dei prodotti e dei manuali relativi ai servizi di rete e alle comunicazioni di dati Compaq.
- *Intersystem Communications Environment (ICE) Installation Guide*, (Versione 3 Rilascio 2 o edizioni successive)
 Questo manuale descrive come installare ICE e come configurare i parametri di avvio di ICE. (ICE fornisce il supporto LU 6.2 per l'implementazione Insessions di SNA).
- *Intersystem Communications Environment (ICE) Administrator's Guide*, (Versione 3 Rilascio 2 o successive edizioni)

Pubblicazioni correlate

Questo manuale descrive come configurare ed utilizzare ICE, la relativa interfaccia ed i relativi programmi di utilità.

Glossario dei termini e delle abbreviazioni

Questo glossario definisce MQSeries i termini e le abbreviazioni utilizzate nel presente manuale. Se non si riesce a trovare il termine desiderato, consultare l'indice oppure la pubblicazione *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

Questo glossario include i termini e le definizioni tratte dalla pubblicazione *American National Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 del American National Standards Institute (ANSI). È possibile ordinare delle copie presso American National Standards Institute, 11 West 42 Street, New York, New York 10036. Le definizioni sono identificate dal simbolo (A) dopo la definizione.

A

accodamento. Vedere *accodamento messaggi*.

accodamento di messaggi. Una tecnica di programmazione per cui ciascun programma nell'ambito di un'applicazione comunica con gli altri programmi inserendo messaggi nelle code.

accodamento remoto. Nell'accodamento di messaggi, la fornitura di servizi che consentono di inserire messaggi nelle code appartenenti ad altri Queue Manager.

administration bag. Nella MQAI, un tipo di data bag creato per la gestione di MQSeries (modificare gli elementi dei dati, creare elenchi e controllare i selettori in un messaggio).

Advanced Program-to-Program Communication (APPC). La funzione generale che caratterizza l'architettura LU 6.2 e le sue varie implementazioni nei prodotti.

APAR. Authorized Program Analysis Report.

APPC. Advanced Program to Program Communication.

applicazione del client. Un'applicazione eseguita su una stazione di lavoro e collegata a un client, che consente all'applicazione di accedere ai servizi di accodamento su un server.

applicazione distribuita. Nell'accodamento messaggi, un insieme di programmi applicativi che è possibile

collegare individualmente a un diverso Queue Manager, ma costituiscono collettivamente un'unica applicazione.

archivio di memoria. Una raccolta di informazioni sui queue manager membri di un cluster. Queste informazioni comprendono i nomi di queue manager, la loro posizione, i loro canali, le code che ospitano e così via.

arresto. Vedere *arresto immediato*, *arresto preventivo* e *arresto a riposo*.

arresto a riposo. In MQSeries, un arresto di un Queue Manager che consente la disconnessione di tutte le applicazioni connesse. Confrontare con *arresto immediato* e *arresto preventivo*.

arresto controllato. Vedere *arresto a riposo*.

arresto immediato. In MQSeries, un arresto di un Queue Manager che non attende la disconnessione delle applicazioni. È consentito il completamento delle chiamate MQI, ma le nuove chiamate MQI non vengono portate a termine dopo che è stato richiesto un arresto immediato. Confrontare con *arresto a riposo* e *arresto preventivo*.

arresto preventivo. In MQSeries, un arresto di un Queue Manager che non attende la disconnessione delle applicazioni, né il completamento delle chiamate MQI. Confrontare con *arresto immediato* e *arresto a riposo*.

attributo. Una delle proprietà di un insieme di proprietà che definisce le caratteristiche di un oggetto MQSeries.

avviso. Un messaggio inviato a un punto cruciale dei servizi di gestione in una rete allo scopo di identificare un problema o un problema imminente.

B

back out. Un'operazione che inverte tutte le modifiche apportate durante la corrente unità di ripristino o unità di lavoro. Al termine dell'operazione, inizia una nuova unità di ripristino o un'unità di lavoro.

backout a fase singola. Un metodo secondo il quale non è consentito il completamento di un'azione in corso e tutte le modifiche che fanno parte di tale azione devono essere annullate.

bag. Vedere *data bag*.

Basic Mapping Support (BMS). Un'interfaccia tra CICS e i programmi applicativi che formatta i dati di

visualizzazione in entrata e in uscita e indirizza i messaggi in uscita di pagina multipla indipendentemente dai caratteri di controllo utilizzati dai vari terminali.

BMS. Basic Mapping Support.

C

canale. Vedere *canale messaggi*.

canale messaggi. Nell'accodamento messaggi distribuiti, un meccanismo per spostare messaggi da un Queue Manager a un altro. Un canale messaggi comprende due agenti del canale messaggi (un mittente a un'estremità e un destinatario dall'altra) e una connessione. Confrontare con *canale MQI*.

canale mittente. Nell'accodamento di messaggi, un canale che inizia il trasferimento, rimuove messaggi da una coda di trasmissione e li trasferisce tramite una connessione a un canale ricevente o richiedente.

Canale MQI. Connette un client MQI a un Queue Manager su un sistema di server e trasferisce solo le chiamate MQI e risponde in maniera bidirezionale. Confrontare con *canale messaggi*.

canale ricevente. Nell'accodamento di messaggi, un canale che può essere avviato in maniera remota da un canale mittente. Il canale richiedente accetta messaggi dal canale mittente tramite una connessione e inserisce i messaggi sulla coda locale indicata nel messaggio. Vedere anche *canale del server*.

canale ricevente. Nell'accodamento messaggi, un canale che risponde a un canale mittente, prende i messaggi da una connessione e li inserisce in una coda locale.

canale server. Nell'accodamento di messaggi, un canale che risponde a un canale richiedente, rimuove messaggi da una coda di trasmissione e li trasferisce tramite una connessione al canale richiedente.

carattere nullo. Il carattere rappresentato da X'00'.

client MQI. Parte di un prodotto di MQSeries che può essere installato su un sistema senza installare il Queue Manager completo. Il client MQI accetta chiamate MQI dalle applicazioni e comunica con un Queue Manager su un sistema di server.

CCF. Channel Control Function (funzione di controllo canale).

CCSID. Coded Character Set Identifier (identificatore insieme di caratteri codificati).

CDF. Channel Definition File (file di definizione canale).

Channel Control Function (CCF). Un programma che sposta messaggi da una coda di trasmissione a una connessione e da quest'ultima a una coda locale, assieme a un'interfaccia del pannello operatore per consentire l'impostazione e il controllo dei canali.

Channel Definition File (CDF). In MQSeries, un file contenente definizioni del canale di comunicazione che associano code di trasmissione a connessioni.

checkpoint. Un intervallo durante il quale vengono scritte informazioni significative sul log. Confrontare con *syncpoint*. In MQSeries su sistemi UNIX, il punto in cui un record di dati descritto nel log è identico al record di dati nella coda. I checkpoint vengono generati automaticamente e sono utilizzati durante il processo di riavvio del sistema.

CICS. Customer Information Control System.

client. Un componente di runtime che fornisce accesso ai servizi di accodamento su un server per applicazioni di utenti locali. Le code utilizzate dalle applicazioni risiedono sul server. Vedere anche *client MQI*.

cluster. Una rete di Queue Manager logicamente associati in qualche modo.

codà. Un oggetto MQSeries. Le applicazioni di accodamento messaggi possono inserire e richiamare i messaggi da una coda. Una coda è posseduta e gestita da un Queue Manager. Le code locali possono contenere un elenco di messaggi in attesa di elaborazione. Le code di altri tipi non possono contenere messaggi—puntano ad altre code oppure possono essere utilizzate come modelli per code dinamiche.

codà di applicazione. Una coda utilizzata da un'applicazione.

codà di iniziazione. Una coda locale sulla quale il Queue Manager inserisce messaggi di trigger.

codà di messaggi. Sinonimo di *codà*.

codà dinamica. Una coda locale creata quando un programma apre un oggetto coda modello. Vedere anche *codà dinamica permanente* e *codà dinamica temporanea*.

codà dinamica permanente. Una coda dinamica che viene eliminata quando è chiusa solo se l'eliminazione è richiesta espressamente. Le code dinamiche permanenti vengono ripristinate se il Queue Manager subisce un malfunzionamento, in modo che possano contenere messaggi permanenti. Confrontare con *codà dinamica temporanea*.

codà dinamica temporanea. Una coda dinamica che viene eliminata quando è chiusa. Le code dinamiche temporanee non sono ripristinate se il Queue Manager subisce un malfunzionamento, in modo che possano

contenere solo messaggi non permanenti. Confrontare con *coda dinamica permanente*.

coda di risposta RTQ (reply-to queue). Il nome di una coda alla quale il programma che ha emesso la chiamata MQPUT intende inviare un messaggio di risposta o di notifica.

coda di trasmissione. Una coda locale in cui i messaggi preparati destinati a un Queue Manager remoto sono temporaneamente memorizzati.

coda evento. La coda nella quale il Queue Manager inserisce un messaggio di evento dopo che ha rilevato un evento. Ciascuna categoria di evento (Queue Manager, prestazioni o evento canale) dispone della propria coda di eventi.

coda locale. Una coda appartenente al Queue Manager locale. Una coda locale può contenere un elenco di messaggi in attesa di elaborazione. Confrontare con *coda remota*.

coda messaggi non recapitati. Vedere *Dead-letter Queue*.

coda remota. Una coda appartenente ad un Queue Manager remoto. I programmi possono inserire i messaggi su code remote, ma non possono ricevere messaggi da code remote. Confrontare con *coda locale*.

coda system.command.input. Una coda locale in cui i programmi applicativi possono inserire comandi di MQSeries. I comandi vengono recuperati nella coda dal server dei comandi, che li convalida e li trasferisce al processore dei comandi affinché li esegua.

Coded Character Set Identifier (CCSID). Il nome di un insieme codificato di caratteri e le relative assegnazioni di punti di codice.

codice di errore. Un codice restituito che descrive il motivo del malfunzionamento o la parziale riuscita di una chiamata MQI.

codice di completamento. Un codice restituito che indica in che modo è terminata una chiamata MQI.

codici di errore. Il nome collettivo per i codici di completamento e codici di errore.

comandi di controllo del sistema. I comandi utilizzati per manipolare entità specifiche della piattaforma, quali pool buffer, classi di memoria e insiemi di pagina.

comandi di gestione. Comandi di MQSeries utilizzati per gestire gli oggetti MQSeries, quali code, processi e namelist.

Comandi MQSeries (MQSC). Comandi leggibili dagli utenti, uniformi in tutte le piattaforme, che sono utilizzati per manipolare oggetti MQSeries. Confrontare con *Programmable Command Format (PCF)*.

comando. In MQSeries, un'istruzione che può essere eseguita dal Queue Manager.

comando PCF. Vedere *formato comando programmabile*.

command bag. Nella MQAI, un tipo di bag creato per gestire oggetti MQSeries che non può però modificare l'ordine degli elementi dei dati né creare elenchi in un messaggio.

commit. L'operazione di completamento di una transazione al fine di registrare e rendere permanenti le modifiche apportate al database. Le risorse protette vengono rilasciate dopo che è stato eseguito il commit di una transazione.

commit a due fasi. Un protocollo per la coordinazione delle modifiche a risorse recuperabili quando viene utilizzato più di un gestore risorse da una singola transazione. Confrontare con *commit a singola fase*.

commit a fase singola. Un metodo secondo il quale un programma può impegnare aggiornamenti in una coda senza coordinarli con gli aggiornamenti effettuati dal programma sulle risorse controllate da un altro gestore di risorse. Confrontare con il *commit a due fasi*.

Common Run-Time Environment (CRE). Un gruppo di servizi che consente ai programmatori di applicazioni e di sistemi di scrivere dei programmi in linguaggi misti. Questi servizi di run-time condivisi possono essere utilizzati da programmi in C, COBOL85, FORTRAN, Pascal e TAL.

connessione. Per fornire un handle di connessione del Queue Manager, che un'applicazione utilizza su chiamate MQI successive. La connessione viene eseguita sia dalla chiamata MQCONN o automaticamente dalla chiamata MQOPEN.

consegna sequenziale. In MQSeries, un metodo di trasmissione di messaggi con un numero di sequenza in modo che il canale ricevente possa ristabilire la sequenza di messaggi memorizzando i messaggi. Ciò è necessario quando occorre consegnare solo una volta i messaggi e nell'ordine corretto.

contesto. Informazioni sull'origine di un messaggio.

CRE. Common Run-Time Environment.

cursore browse. Nell'accodamento di messaggi, un indicatore utilizzato nello sfogliare una coda allo scopo di identificare il successivo messaggio in sequenza.

Customer Information Control System (CICS). Un sistema di gestione delle transazioni IBM che fornisce l'accesso online simultaneo ai file di dati tramite programmi applicativi scritti dagli utenti. CICS comprende anche funzioni per creare, utilizzare e gestire i database.

D

data bag. Nella MQAI, un bag che consente di gestire le proprietà (o i parametri) degli oggetti.

Data Conversion Interface (DCI). L'interfaccia di MQSeries a cui devono conformarsi i programmi scritti dal cliente (o dal rivenditore) che convertono i dati applicativi tra le codifiche di macchine diverse e i CCSID. Fa parte di MQSeries Framework.

datagramma. Il messaggio più semplice supportato da MQSeries. Questo tipo di messaggio non richiede una risposta.

dati evento. In un messaggio di evento, la parte di dati del messaggio che contiene informazioni sull'evento (come il nome del Queue Manager e l'applicazione che ha dato origine all'evento). Vedere anche *intestazione evento*.

DCE. Distributed Computing Environment.

DCI. Interfaccia di conversione di dati.

Dead-Letter Queue (DLQ). Una coda alla quale il Queue Manager o l'applicazione invia messaggi che non può consegnare alla rispettiva destinazione corretta.

definizione locale. Un oggetto MQSeries che appartiene al queue manager locale.

definizione locale di una coda remota. Un oggetto MQSeries che appartiene al queue manager locale. Quest'oggetto definisce gli attributi di una coda che appartiene a un altro Queue Manager. Inoltre, viene utilizzato per l'aliasing del manager di code e per l'aliasing RTQ.

descrittore messaggio. Informazioni di controllo che descrivono il formato del messaggio e la presentazione che è eseguita come parte di un messaggio di MQSeries. Il formato del descrittore messaggio è definito dalla struttura di MQMD.

descrittore oggetto. Una struttura di dati che identifica un particolare oggetto di MQSeries (MQOD). Nel descrittore sono inclusi il nome dell'oggetto e il tipo di oggetto.

Distributed Computing Environment (DCE). Middleware che fornisce servizi di base, semplificando lo sviluppo di applicazioni distribuite. DCE è definito da Open Software Foundation (OSF).

distributed queue management. Nell'accodamento di messaggi, l'impostazione e il controllo dei canali di messaggi a Queue Manager su altri sistemi.

DLQ (dead-letter queue). Una coda alla quale il Queue Manager o l'applicazione invia messaggi che non può consegnare alla rispettiva destinazione corretta.

E

EC. EC è un processo di controllo secondario nel queue manager, responsabile per un gruppo di agenti.

EC Boss. L'Execution Controller Boss è il processo di controllo principale nel queue manager.

elemento di dati. Nella MQAI, un elemento contenuto in un data bag. Può trattarsi di un elemento numero intero o stringa di caratteri, oppure un elemento utente o sistema.

EMS. Event Monitoring System.

evento. Vedere *evento canale*, *evento strumentazione*, *evento prestazioni* e *evento Queue Manager*.

evento canale. Un evento che indica che un'istanza del canale è divenuta disponibile o non disponibile. Gli eventi di canale sono generati sui Queue Manager su entrambe le estremità del canale.

evento gestore coda. Un evento che indica:

- Una condizione di errore che si è verificata in relazione alle risorse utilizzate da un Queue Manager. Ad esempio, una condizione di errore determinata dal fatto che un Queue Manager non è disponibile.
- Una modifica significativa che si è verificata nel Queue Manager. Ad esempio, un Queue Manager è stato arrestato o avviato.

evento intervallo di servizio. Un evento correlato all'intervallo di servizio.

evento prestazione. Una categoria di evento che indica che si è verificata una condizione di limite.

evento sospeso. Un evento non pianificato che si verifica quale risultato di una richiesta di connessione da un adattatore CICS.

evento strumentazione. Una funzione che può essere utilizzata per monitorare il funzionamento dei Queue Manager in una rete di sistemi MQSeries. MQSeries fornisce eventi di strumentazione per il monitoraggio di definizioni di risorse del Queue Manager, condizioni delle prestazioni e condizioni del canale. Gli eventi di strumentazione possono essere utilizzati da un meccanismo di notifica scritta dall'utente in un'applicazione amministrativa che visualizza gli eventi a un operatore di sistema. Essi inoltre consentono alle applicazioni di comportarsi come agenti per altre reti amministrative per monitorare prospetti e creare gli avvisi appropriati.

evento trigger. Un evento (quale un messaggio in arrivo in una coda) che consente al Queue Manager di creare un messaggio trigger su una coda di iniziazione.

F

FFST. First Failure Support Technology (tecnologia di assistenza per guasti).

FFST (First Failure Support Technology). Utilizzata da MQSeries su sistemi UNIX, MQSeries per OS/2 Warp, MQSeries per Windows NT e Windows 2000 e MQSeries per AS/400 per rilevare e riportare problemi di software.

FIFO (first-in-first-out). Una tecnica di accodamento in cui l'elemento successivo da recuperare è quello che è rimasto in coda più a lungo.

file di autorizzazione. Un file che fornisce le definizioni di sicurezza per un oggetto, una classe di oggetti o tutte le classi di oggetto.

file di configurazione (noto anche come file ini). Un file che contiene le informazioni di configurazione correlate ai log, alle comunicazioni o ai servizi installabili. Vedere anche *stanza*.

file ini. Vedere *file di configurazione*.

first-in-first-out (FIFO). Una tecnica di accodamento in cui l'elemento successivo da recuperare è quello che è rimasto in coda più a lungo.

Framework. In MQSeries, una raccolta di interfacce di programmazione che consentono ai clienti o ai rivenditori di scrivere programmi che estendono o sostituiscono alcune funzioni fornite in MQSeries. Le interfacce sono:

- Data Conversion Interface (DCI, interfaccia di conversione dati) di MQSeries
- Message Channel Interface (MCI, interfaccia del canale messaggio) di MQSeries
- Name Service Interface (NSI, interfaccia del servizio nome) di MQSeries
- Security Enabling Interface (SEI, interfaccia di abilitazione protezione) di MQSeries
- Trigger Monitor Interface (TMI, interfaccia di monitoraggio trigger) di MQSeries

G

get. Nell'accodamento di messaggi, per utilizzare la chiamata MQGET allo scopo di rimuovere un messaggio da una coda. Vedere anche *browse*.

H

handle. Vedere *handle di connessione* e *handle dell'oggetto*.

handle dell'oggetto. L'identificativo, o token, con il quale un programma accede all'oggetto MQSeries con il quale sta lavorando.

handle di connessione. L'identificativo, o token, tramite il quale un programma accede al Queue Manager a cui è connesso.

handler DLQ. Un'utilità fornita da MQSeries che controlla una coda di messaggi non recapitati (DLQ) ed elabora messaggi in coda in conformità alla tabella delle regole scritte dall'utente.

I

ICE. Intersystem Communications Environment è una famiglia di prodotti software basati su Compaq che abilita all'accesso ad una gamma di applicazioni sui computer Compaq.

Internet Protocol (IP). Un protocollo utilizzato per instradare i dati dalla loro fonte alla loro destinazione in un ambiente Internet. Questo è il livello di base, su cui si basano altri livelli di protocollo, quali TCP e UDP.

intervallo di servizio. Un intervallo di tempo, rispetto al quale il tempo trascorso tra put e get e un successivo get è confrontato dal Queue Manager nello stabilire se le condizioni per un evento di intervallo di servizio sono state soddisfatte. L'intervallo di servizio per una coda è specificato da un attributo di coda.

intestazione evento. In un messaggio di evento, la parte dei dati di messaggio che identifica il tipo di evento del codice di errore relativo a tale evento.

IP. Internet Protocol

L

listener. Nell'accodamento distribuito di MQSeries, un programma che controlla le informazioni relative alle connessioni di rete in arrivo.

log. In MQSeries, registra il lavoro svolto dai Queue Manager mentre ricevono, trasmettono e consegnano messaggi, per consentirne il ripristino in caso di malfunzionamento.

Logical Unit of Work (LUW). Vedere *unità di lavoro*.

M

MCA (Message Channel Agent). Un programma che trasmette messaggi da una coda di trasmissione a una connessione oppure da una connessione a una coda di destinazione. Vedere anche *interfaccia di accodamento messaggi*.

MCA (message channel agent). Un programma che trasmette messaggi da una coda di trasmissione a una connessione oppure da una connessione a una coda di destinazione.

MCI (message channel interface). L'interfaccia di MQSeries a cui devono conformarsi i programmi scritti dal cliente (o dal rivenditore) che trasmettono messaggi tra un Queue Manager di MQSeries e un altro sistema di messaggistica. Fa parte di MQSeries Framework.

memorizzazione e invio. La memorizzazione temporanea di pacchetti, messaggi o frame in una rete di dati prima che siano ritrasmessi verso la loro destinazione.

Message Channel Interface (MCI). L'interfaccia di MQSeries a cui devono conformarsi i programmi scritti dal cliente (o dal rivenditore) che trasmettono messaggi tra un Queue Manager di MQSeries e un altro sistema di messaggistica. Fa parte di MQSeries Framework.

message queue management. La funzione MQM (Message Queue Management) in MQSeries for Compaq NSK V2.2 utilizza i formati di comando PCF ed i comandi di controllo. MQM viene eseguito come un requester PATHWAY SCOBOL sotto il TCP (Terminal Control Process) ed utilizza un server MQM SERVERCLASS, che richiama l'API in linguaggio C per eseguire i comandi PCF. Esiste un'istanza di MQM separata per ciascun queue manager configurato su un sistema, poiché ogni queue manager è controllato sotto la propria configurazione PATHWAY. Di conseguenza, un MQM è limitato alla gestione del queue manager cui appartiene.

messaggio. Nelle applicazioni di accodamento messaggi, una comunicazione inviata tra programmi. Vedere anche *messaggio permanente* e *messaggio non permanente*. Nella programmazione del sistema, le informazioni dirette all'operatore del terminale o all'amministratore di sistema.

messaggio di evento. Contiene informazioni (quali la categoria dell'evento, il nome dell'applicazione che ha provocato l'evento e le statistiche del Queue Manager) in relazione all'origine di un evento di strumentazione in una rete di sistemi MQSeries.

messaggio di notifica. Un tipo di messaggio che fornisce informazioni circa un altro messaggio. Un messaggio di notifica può indicare che un messaggio è stato consegnato, è giunto a destinazione, è scaduto o non è stato elaborato per qualche motivo.

messaggio di richiesta. Un tipo di messaggio utilizzato per richiedere una risposta da un altro programma.

messaggio di risposta. Un tipo di messaggio utilizzato per le risposte allo scopo di richiedere messaggi.

messaggio non permanente. Un messaggio che non sopravvive al riavvio del Queue Manager. Confrontare con *messaggio permanente*.

messaggio permanente. Un messaggio che sopravvive al riavvio del Queue Manager. Confrontare con *messaggio non permanente*.

messaggio trigger. Un messaggio che contiene le informazioni sul programma che un monitor trigger deve avviare.

messaggistica. Vedere *messaggistica sincrona* e *messaggistica asincrona*.

messaggistica asincrona. Un metodo di comunicazione tra programmi in cui questi inseriscono messaggi in code di messaggi. Con la messaggistica asincrona, il programma di invio procede con la propria elaborazione senza attendere una risposta al messaggio. Contrario di *messaggistica asincrona*.

messaggistica indipendente dal tempo. Vedere *messaggistica asincrona*.

messaggistica sincrona. Un metodo di comunicazione tra programmi in cui questi inseriscono messaggi in code di messaggi. Con la messaggistica sincrona, il programma mittente attende una risposta al proprio messaggio prima di riprendere l'elaborazione. Confrontare con *messaggistica asincrona*.

monitor trigger. Un'applicazione continuamente in esecuzione dedicata ad una o più code di iniziazione. Quando un messaggio trigger giunge su una coda di iniziazione, il monitor trigger recupera il messaggio. Esso utilizza le informazioni nel messaggio trigger per avviare un processo che serve la coda in cui si è verificato un evento trigger.

MQAI. MQSeries Administration Interface (interfaccia di amministrazione di MQSeries).

MQI (Message Queue Interface). L'interfaccia di programmazione fornita dai Queue Manager di MQSeries. Questa interfaccia di programmazione consente ai programmi applicativi di accedere ai servizi di accodamento dei messaggi.

MQI (message queuing interface). L'interfaccia di programmazione fornita dai Queue Manager di MQSeries. Questa interfaccia di programmazione consente ai programmi applicativi di accedere ai servizi di accodamento dei messaggi.

MQM. Message Queue Management.

MQSC. Comandi di MQSeries.

MQSeries. Una famiglia di programmi concessi in licenza da IBM che offre servizi di accodamento messaggi.

MQSeries Administration Interface (MQAI).
Un'interfaccia di programmazione di MQSeries.

N

namelist. Un oggetto di MQSeries che contiene un elenco di nomi, ad esempio, nomi di code.

Name Service Interface (NSI). L'interfaccia di MQSeries a cui devono conformarsi i programmi scritti dal cliente (o dal rivenditore) che risolvono la proprietà del nome della coda. Fa parte di MQSeries Framework.

NetBIOS. Network Basic Input/Output System. Un'interfaccia di sistema operativo per i programmi applicativi utilizzati sui personal computer IBM connessi ad una rete Token-Ring della IBM.

NSI. Name Service Interface.

numerazione sequenza messaggi. Una tecnica di programmazione che assegna numeri univoci ai messaggi durante la trasmissione attraverso una connessione. Ciò consente al processo di ricezione di verificare se tutti i messaggi sono ricevuti, inserirli in una coda secondo l'ordine originale ed eliminare i messaggi duplicati.

O

Object authority manager (OAM). In MQSeries sui sistemi UNIX, MQSeries for Compaq e MQSeries per Windows NT e Windows 2000, il servizio di autorizzazione predefinito per la gestione di comandi e oggetti. OAM può essere sostituito da, o eseguito in combinazione con un servizio di protezione fornito dall'utente.

oggetto. In MQSeries, un oggetto è un queue manager, una coda, una definizione di processo oppure un canale.

oggetto coda dell'alias. Un oggetto MQSeries, il cui nome è un alias per una coda di base definita al gestore di coda locale. Quando un'applicazione o un gestore di coda utilizzano un alias di coda, il nome alias viene risolto e l'operazione richiesta viene eseguita sulla coda di base associata.

oggetto coda modello. Un insieme di attributi di coda che agisce da modello quando un programma crea una coda dinamica.

oggetto coda remota. Vedere *definizione locale di una coda remota*.

oggetto di definizione del processo. Un oggetto MQSeries che contiene la definizione di un'applicazione MQSeries. Ad esempio, un Queue Manager utilizza la definizione quando opera con messaggi trigger.

oggetto predefinito. Una definizione di un oggetto (ad esempio, una coda) con tutti gli attributi definiti. Se un utente definisce un oggetto, ma non specifica tutti i possibili attributi per tale oggetto, il Queue Manager utilizza attributi predefiniti in luogo di quelli che non sono stati specificati.

P

parametro di input. Un parametro di una chiamata MQI in cui si forniscono informazioni durante la chiamata.

parametro di output. Un parametro di una chiamata MQI in cui il Queue Manager restituisce le informazioni quando la chiamata è completata o non riuscita.

parametro input/output. Un parametro di una chiamata MQI in cui si forniscono informazioni durante la chiamata e in cui il Queue Manager modifica le informazioni quando la chiamata è completata o non riuscita.

PCF. Programmable Command Format (formato comando programmabile).

percorso di risoluzione. L'insieme di code che sono aperte quando un'applicazione specifica una coda alias o una coda remota su immissione nella chiamata MQOPEN.

piattaforma. In MQSeries, il sistema operativo sotto il quale il Queue Manager è in esecuzione.

ping. Nell'accodamento distribuito, un ausilio diagnostico che utilizza lo scambio di un messaggio di prova per confermare che un canale di messaggi sta funzionando.

priorità del messaggio. In MQSeries, un attributo di un messaggio che può influenzare l'ordine di recupero dei messaggi di una coda e l'eventuale generazione di un evento trigger.

processore comandi. Il componente di MQSeries che elabora i comandi.

Programmable Command Format (PCF). Un tipo di messaggio di MQSeries utilizzato da:

- Applicazioni amministrative dell'utente che inseriscono comandi PCF nella coda di immissione comandi del sistema di un Queue Manager specifico.
- Applicazioni amministrative dell'utente, per ottenere i risultati di un comando PCF da un gestore di code specifico.
- Un Queue Manager, come notifica del verificarsi di un evento.

Confrontare con *MQSC*.

programma di trasmissione. Vedere *agente di canale messaggi*.

Program Temporary Fix (PTF). Una soluzione o by-pass di un problema diagnosticato dagli ingegneri IBM come risultato di un difetto in un rilascio corrente e non modificato di un programma.

prospetto analitico del programma autorizzato (APAR). Un prospetto di un problema provocato da un presunto difetto in una versione corrente e non modificata di un programma.

protezione contestuale. Un metodo per consentire la gestione della protezione in modo che i messaggi siano obbligati a recare dettagli delle relative origini nella descrizione del messaggio.

protezione utente alternativo. Una funzione di protezione grazie alla quale l'autorizzazione di un ID utente può essere utilizzata da un altro ID utente, ad esempio, per aprire un oggetto MQSeries.

PTF. Program Temporary Fix (riparazione temporanea del programma).

Q

Queue Manager. Un programma di sistema che fornisce i servizi di accodamento alle applicazioni. Fornisce un'API (la MQI) che consente ai programmi di accedere ai messaggi nelle code appartenenti al Queue Manager. Vedere anche *gestore coda locale* e *gestore coda remota*. Un oggetto MQSeries che definisce gli attributi di un particolare Queue Manager.

queue manager . Per un programma, un queue manager è remoto se non è il queue manager cui è connesso il programma.

Queue Manager locale. Il Queue Manager a cui è connesso un programma, che fornisce servizi di accodamento dei messaggi al programma. I Queue Manager a cui non è connesso un programma sono denominati *Queue Manager remoto*, anche se sono in esecuzioni sullo stesso sistema del programma.

queue server. NonStop Process Pair che supporta tutte le operazioni di messaggistica per le code locali.

R

RBA. Relative Byte Address (indirizzo byte relativo).

registrazione nei log lineare. In MQSeries su sistemi UNIX, MQSeries per OS/2 Warp e MQSeries per Windows NT e Windows 2000, il processo che conserva i dati di riavvio in un sequenza di file. I nuovi file vengono aggiunti alla sequenza secondo le necessità. Lo spazio in cui i dati vengono scritti non viene riutilizzato finché il Queue Manager non è riavviato. Confrontare con *registrazione nei log circolare*.

responder. Nell'accodamento distribuito, un programma che risponde alle richieste di connessione alla rete da un altro sistema.

resynch. In MQSeries, un'opzione che consente di indirizzare il canale ad avviare e risolvere qualsiasi messaggio di stato in sospeso, ma senza riavviare il trasferimento di messaggi.

richiamata. Un canale di messaggi di richiesta inizia un trasferimento da un canale di invio chiamando prima il mittente, quindi chiudendo e restando in attesa di una richiamata.

riposo. In MQSeries, lo stato di un Queue Manager precedente al suo arresto. In tale stato, i programmi i programmi possono terminare l'elaborazione, ma non è possibile avviare nuovi programmi.

rollback. Sinonimo di *back out*.

S

Security Enabling Interface (SEI). L'interfaccia di MQSeries a cui devono conformarsi i programmi scritti dal cliente (o dal rivenditore) che verificano le autorizzazioni, forniscono ID o eseguono l'autenticazione. Fa parte di MQSeries Framework.

SEI. Security Enabling Interface (interfaccia di attivazione protezione).

server. (1) In MQSeries, un Queue Manager che fornisce servizi di code alle applicazioni client in esecuzione su una stazione di lavoro remota. (2) Il programma che risponde alle richieste di informazioni nel particolare modello di client/server di flusso di informazioni a due programmi. Vedere anche *client*.

server di comandi. Il componente di MQSeries che legge i comandi dalla coda d'immissione dei comandi di sistema, li verifica e passa i comandi validi al processore comandi.

server MQI. Un server MQI è un Queue Manager che fornisce servizi di accodamento ad uno o più client. Tutti gli oggetti MQSeries, ad esempio le code, esistono sul sistema queue manager, cioè sulla macchina server MQI. Un server può supportare anche normali applicazioni locali MQI.

servizi installabili. In MQSeries sui sistemi UNIX, MQSeries for Compaq, MQSeries per OS/2 Warp e MQSeries per Windows NT e Windows 2000, funzioni aggiuntive fornite come componenti indipendenti. L'installazione di ciascun componente è facoltativa; è possibile utilizzare invece componenti IBM o di terzi. Vedere anche *servizio di autorizzazione*, *servizio nome e servizio identificatore utente*.

sfoglia. Nell'accodamento di messaggi, per utilizzare la chiamata MQGET al fine di copiare un messaggio senza rimuoverlo dalla coda. Vedere anche *get*.

SIT. System Initialization Table (tabella di inizializzazione del sistema).

SNA. Systems Network Architecture.

stanza. Un gruppo di righe in un file di configurazione che assegna un valore a un parametro che modifica il comportamento di un Queue Manager, di un client o di un canale. In MQSeries sui sistemi, un file di configurazione (ini) può contenere un certo numero di stanze.

Status Server. Supporta tutte le informazioni sugli stati per tutti gli oggetti diversi dalle code locali. Lo Status Server predefinito gestisce inoltre le informazioni sugli stati dei canali per il queue manager.

stringa sintomo. Informazioni diagnostiche visualizzate in un formato strutturato progettato per la ricerca del database di supporto del software IBM.

syncpoint. Un punto intermedio o finale durante l'elaborazione di una transazione al quale le risorse protette sono congruenti. A un syncpoint, le modifiche delle risorse possono essere impegnate con sicurezza, oppure è possibile eseguire il backout al precedente syncpoint.

system bag. Un tipo di data bag creato dalla MQAI.

T

tabella regole. Un file di controllo contenente una o più regole che l'handler della coda messaggi non recapitati applica alla DLQ.

TACL. Tandem Advanced Command Language.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

thread. In MQSeries, il livello inferiore dell'esecuzione parallela disponibile su una piattaforma di sistema operativo.

tipo canale di connessione del server. Il tipo di definizione canale MQI associata al server che esegue un Queue Manager. Vedere anche *tipo di canale di connessione del client*.

tipo di canale di connessione del client. Il tipo di definizione del canale MQI associato con un client MQI. Vedere anche la sezione *tipo di canale di connessione del server*.

TMF. Transaction Management Facility.

TMI. Trigger Monitor Interface (interfaccia monitor del trigger).

TM/MP. NonStop Transaction Manager/MP.

traccia di prestazione. Un'opzione di traccia di MQSeries in cui i dati di traccia devono essere utilizzati per l'analisi e l'ottimizzazione delle prestazioni.

tranid. Vedere *identificatore transazione*.

Transmission Control Protocol/Internet Protocol (TCP/IP). Una suite di protocolli per le comunicazioni che supporta le funzioni di connettività peer-to-peer sia per le LAN che per le WAN.

Transmission Control Protocol (TCP). Parte della suite di protocolli TCP/IP. Un protocollo host-to-host tra host in reti di comunicazioni a commutazione di pacchetti. TCP fornisce un recapito dei flussi di dati a connessioni. Il recapito è affidabile ed ordinato.

trasmissione ciclica del numero sequenziale. In MQSeries, un metodo che garantisce che entrambe le estremità di una connessione reimpostino contemporaneamente i numeri di sequenza dei messaggi correnti. La trasmissione di messaggi con un numero di sequenza garantisce che il canale ricevente possa ristabilire la sequenza di messaggio quando memorizza i messaggi.

triggering. In MQSeries, una funzione che consente a un Queue Manager di avviare un'applicazione automaticamente quando sono soddisfatte condizioni predeterminate su una coda.

Trigger Monitor Interface (TMI). L'interfaccia di MQSeries a cui devono conformarsi i programmi di monitor trigger scritti dal cliente (o dal rivenditore). Fa parte di MQSeries Framework.

U

UDP. User Datagram Protocol.

unità di lavoro. Una sequenza recuperabile di operazioni eseguite da un'applicazione tra due punti di consistenza. Un'unità di lavoro inizia all'avvio di una transazione o al termine di un syncpoint richiesto dall'utente. Essa termina sia al syncpoint richiesto dall'utente o al termine di una transazione. Confrontare con *unità di ripristino*.

unità di ripristino. Una sequenza recuperabile di operazioni nell'ambito di un singolo gestore di risorse. Confrontare con *unità di lavoro*.

user bag. Nella MQAI, un tipo di data bag creato dall'utente.

User Datagram Protocol (UDP). Parte della suite di protocolli TCP/IP. Un protocollo a pacchetti creato direttamente sul livello Internet Protocol. UDP è

un'alternativa senza connessioni e meno affidabile a TCP. Viene utilizzato per i programmi application-to-application tra sistemi host TCP/IP.

utilità. In MQSeries, un insieme di programmi incluso che fornisce all'operatore di sistema o all'amministratore di sistema funzioni aggiuntive oltre a quelle fornite dai comandi di MQSeries. Alcune utilità invocano più di una funzione.

V

verifiche di autorizzazione. Controlli di sicurezza eseguiti quando un utente tenta di aprire un oggetto MQSeries.

verifiche di autorizzazione. Controlli di sicurezza che vengono eseguiti quando un utente prova a emettere comandi di gestione su un oggetto, ad esempio per aprire una coda o per connettersi a un Queue Manager.

Indice analitico

Caratteri speciali

- , requisiti
- hardware 319
- software 319

A

- abilitazione
 - eventi di strumentazione 172
 - sicurezza 143
- accesso non autorizzato, protezione da 139
- accodamento distribuito
 - code messaggi non recapitati 14
 - code undelivered-message 14
 - output non corretto 210
- ADD SERVER, comando PATHCOM 32
- aggiunta di queue server non predefinito 60
- aggiunta di status server non predefinito 57
- algoritmi di richiamo per i messaggi 10
- alias
 - code di risposta RTQ 136
 - queue manager 136
- alias di coda di risposta RTQ 136
- AllQueueManagers - stanza, file MQSINI 188
- ALTER SERVER
 - comando PATHCOM 33
- ambiente operativo specificato 319
- amministrazione
 - remota 127
 - canali 128
 - code di trasmissione 128
 - oggetti 125
- annullamento di una coda locale 111
- applicazione
 - attendibile 228
 - Binding FASTPATH 354
 - Binding STANDARD 354
 - collegata a FASTPATH
 - ripristino errore 239
 - collegata a STANDARD
 - ripristino errore 239
 - compilazione e binding 354
 - considerazioni sulla progettazione 207
 - dati 8
 - errori di programmazione, esempi di 203
 - esecuzione 355
 - funzione trigger 351
 - gestione di transazioni simultanee 238
 - guardian 353
 - indipendente dal tempo 7
 - limitazioni per l'utilizzo di FASTPATH 229
 - ottimizzazione 219

- applicazione (*Continua*)
 - scrittura 349
 - sfogliare messaggi permanenti nella coda 225
 - supporto gestione MQI 97
 - utilizzo di OpenTMF 237
- applicazioni con la funzione trigger 351
 - inoltrato di USERDATA 340
- applicazioni indipendenti dal tempo 7
- arresto
 - queue manager 288, 333
- arresto traccia MQSeries 291
- assegnazione di spazio di swap 56
- attributi degli oggetti MQSeries
 - attributo ALL 108
 - code 12
 - confronto tra MQSC e PCF 31
 - gestore
 - modifica 101
 - visualizzazione 100
 - modifica della coda locale 109
 - predefiniti 108
- attributi LIKE, DEFINE QLOCAL 109
- attributo di canale NPMSPEED 221
- attributo REPLACE, comandi DEFINE 103
- Authority - stanza, file QMINI 190
- autorizzazione
 - comandi 146
 - comando dspmqaut 146
 - comando set/reset 302
 - comando setmqaut 146
 - contestuale 148
 - elenchi 144
 - eventi 397
 - file 156
 - gestione 154
 - gruppi utenti 142
 - MQI 151
 - servizi installabili 146
 - utente alternativo 148
- autorizzazione contestuale 148
- autorizzazione per utente alternativo 148
- autorizzazioni MQOPEN 151
- autorizzazioni MQPUT 151
- avvio
 - canali 130
- avvio di
 - canali 91
 - traccia 74
 - un Queue Manager 66

B

- bilanciamento del carico di lavoro 219
 - utilizzo di cluster 10
- binding 23
- binding dinamico 23
- Binding FASTPATH 228, 354
 - abilitazione 228

- Binding FASTPATH 228, 354 (*Continua*)
 - restrizioni 229
 - riduzione del carico 228
- binding nativo 23
- binding non nativo 23
- Binding STANDARD 354
- BookManager 415
- buffer
 - messaggi quando si sfogliano 225
- buffer di memoria 222

C

- C++ nativo
 - linguaggio supportato 352
- C nativo
 - linguaggio supportato 352
- C non nativo
 - linguaggio supportato 352
- canale
 - accodamento remoto 125
 - autorizzazioni di comando
 - escape 154
 - avvio 130
 - canali messaggi 221
 - comandi 150
 - comando run 294
 - comando run initiator 293
 - definizione 129
 - definizione automatica di 131
 - definizione di un canale mittente TCP/IP 393
 - definizione di un canale ricevente TCP/IP 393
 - definizione tra Queue Manager 12
 - descrizione 15
 - descrizione di 125
 - eventi 172, 397
 - gestione remota 128
 - ICE 392
 - limitazione del numero di 53
 - requisiti di sicurezza 150
 - sicurezza 150
 - uscite 367
- canale di connessione al server, definizione automatica di 131
- canale messaggi 221
- canale ricevente, definizione automatica di 131
- capacità attuale della coda 109
- capacità di coda 109
- catalogo oggetti 238
- CCSID
 - metodo di conversione 51
 - supportato da MQSeries for Compaq NSK 321
- channel
 - configurazione 190
- channel initiator
 - configurazione mediante comandi PATHCOM 32

- channel initiator (*Continua*)
 - ripristino errore 239
- ChannelInitiator - stanza, file QMINI 190
- Channels - stanza, file QMINI 190
- Channels -menu, MQM 82
- chiusura, queue manager 67
- chiusura ad attività completate 67
- chiusura controllata 67
- chiusura di un Queue Manager 67
- chiusura preventiva del Queue Manager 67
- classe di server
 - definita dall'utente 43
- classe di server MQS-STATUS00 34
- classe server
 - convenzioni di denominazione server di coda 61
 - status server 58
 - MQS-TCPLISnn 55
- classi di server definite dall'utente 43
- classi di server TS/MP
 - MQS-CHANINIT00 32
 - MQS-CMDSERV00 32
 - MQS-EC00 34
 - MQS-TCPLIS00 32
 - MQS-TRIGMON00 33
- client 17, 395
- cluster
 - accodamento remoto 125
 - descrizione di 126
 - di Queue Manager 10
 - miglioramento della disponibilità di rete 245
 - Queue Manager
 - workload exit 18
 - uscite di gestione carico del lavoro 367
- COBOL nativo
 - linguaggio supportato 352
- COBOL non nativo
 - linguaggio supportato 352
- coda
 - coda di trasmissione cluster 223
- coda comandi 14
- coda di evento 15
- coda di iniziazione
 - channel initiator, modifica 33
 - definizione 119
 - descrizione 13
- coda di trasmissione
 - creazione 135
 - definizione 129
 - definizione tra Queue Manager 12
 - descrizione 14
 - gestione remota 128
 - predefinite 135
 - predefinito 46
- coda di trasmissione cluster 223
- codici di errore
 - comando crtmqcvx 265
 - comando dltnmqm 271
 - comando dspmqaut 275
 - comando dspmqcsv 277
 - comando endmqcsv 285
 - comando endmqm 289
 - comando runmqchi 293
- codici di errore (*Continua*)
 - comando runmqchl 294
 - comando runmqtrm 301
 - comando setmqaut 306
 - comando strmqcsv 309
 - comando strmqm 310
 - comando strmqtrc 312
- code
 - alias 13
 - alias, utilizzo di 115
 - applicazione, definizione per triggering 118
 - attributi 12, 346
 - attributi, modifica 109
 - autorizzazioni per 148
 - comandi 14
 - definizione 12
 - descrizione 9
 - di risposta 15, 136
 - dimensione di 107
 - dimensione fisica 107
 - dinamiche 9
 - distribuito, output non corretto da 210
 - esame 112
 - evento 15, 172
 - funzionante con 106
 - iniziazione
 - definizione 119
 - messaggi trigger 13
 - locale
 - annullamento 111
 - copia 109
 - definizione 106
 - eliminazione 112
 - locali 11, 12
 - lunghezza 226
 - messaggi non recapitati 14, 46
 - modello 13
 - definizione 117
 - funzionante con 117
 - oggetti
 - alias 13
 - locali 12
 - modello 13
 - remote 12
 - per le applicazioni MQI 97
 - predefinite 9
 - remota
 - alias Queue Manager 136
 - creazione 133
 - funzionante con 136
 - remote 11, 12
 - temporanee 9
 - trasmissione 14
 - creazione 135
 - definizione 129
 - gestione remota 128
 - predefinite 135
 - predefinito 46
- code di alias
 - autorizzazioni per 148
 - descrizione 13
- code di risposta 15
- code dinamiche 9
 - autorizzazioni per 148
 - descrizione 9
- code locali
 - annullamento 111
 - attributi 347
 - comandi 14
 - copia di definizioni 109
 - definizione 106
 - descrizione 11, 12
 - eliminazione 112
 - iniziazione 13
 - messaggi non recapitati 14
 - trasmissione 14
- code messaggi non recapitati
 - descrizione 14
 - handler 295
 - specifica 46
- code modello
 - definizione 117
 - descrizione 13
 - funzionante con 117
- code permanenti 9
- code predefinite 9
- code remote 125
 - descrizione 11, 12
- code temporanee 9
- codice di errore 11
- codici di errore 200
 - comando altnmqfls 257
 - comando altnmqusr 259
 - comando cleanrdf 261
 - comando crtmqcm 269
 - comando dspmqfls 278
 - comando dspmqusr 283
 - comando endmqtrc 291
 - comando runmqflsr 297
 - comando runmqsc 299
- comandi
 - aggiorna il gestore code V2.2.0.1 (upgmqm) 314
 - avvia server di comandi (strmqcsv) 309
 - avvia trigger monitor (runmqtrm) 301
 - avvio Queue Manager (strmqm) 310
 - avvio traccia di MQSeries (strmqtrc) 311
 - comandi di sicurezza
 - dspmqaut 147
 - setmqaut 144
 - confronto tra gruppi 329
 - controllo 29
 - altnmqfls 254
 - altnmqusr 259
 - cleanrdf 261
 - cnvlchl 263
 - crtmqcvx 265
 - crtmqm 267
 - dltnmqm 271
 - dspmqaut 273
 - dspmqcsv 277
 - dspmqfls 278
 - dspmqtrc 282
 - dspmqusr 283
 - endmqcsv 285
 - endmqm 288
 - endmqtrc 291
 - instmqm 292
 - runmqchi 293

- comandi (*Continua*)
 - runmqchl 294
 - runmqdlq 295
 - runmqlsr 296
 - runmqsc 298
 - runmqtrm 301
 - setmqaut 302
 - strmqcsv 309
 - strmqm 310
 - strmqtrc 311
 - upgmqm 314
- converti le definizioni di canale client (cnvclchl) 263
- crea Queue Manager (crtmqm) 267
- definire un principal corrispondente ad un Id utente Compaq NSK (altmqusr) 259
- delete Queue Manager (dltmqm) 271
- display authority (dspmqaut) 273
- display command server (dspmqcsv) 277
- end command server (endmqcsv) 285
- end MQSeries trace (endmqtrc) 291
- end queue manager (endmqm) 288
- eseguire la pulizia RDF (cleanrdf) 261
- impostare/reimpostare
 - l'autorizzazione (setmqaut) 145, 302
- installa MQSeries for Compaq NSK (instmqm) 292
- modifica attributi del file di coda (altmqfls) 254
- MQSC
 - ALTER QLOCAL 109
 - ALTER QREMOTE 135
 - DEFINE CHANNEL 129
 - DEFINE QALIAS 115
 - DEFINE QLOCAL 109
 - DEFINE QLOCAL LIKE 109
 - DEFINE QLOCAL REPLACE 109
 - DEFINE QMODEL 117
 - DEFINE QREMOTE 133
 - DELETE QLOCAL 112
 - DISPLAY QREMOTE 135
 - file di comandi 103
 - utilizzo 31
 - verifica 104
- PCF (Programmable Command Format) 31
- run channel (runmqchl) 294
- run channel initiator (runmqchi) 293
- run dead-letter queue handler 295
- run DLQ handler (runmqdlq) 159
- run listener (runmqlsr) 296
- runmqsc 99
- visualizza informazioni un principal specificato (dspmqusr) 283
- visualizza traccia formattata di MQSeries (dspmqtrc) 282
- comandi dell'operatore, nessuna risposta da 204
- comandi di controllo 29
 - altmqfls 254
 - altmqusr 259
 - cleanrdf 261
- comandi di controllo 29 (*Continua*)
 - cnvclchl 263
 - crtmqcvx 265
 - crtmqm 267
 - dltmqm 271
 - dspmqaut 273
 - dspmqcsv 277
 - dspmqfls 278
 - dspmqtrc 282
 - dspmqusr 283
 - endmqcsv 285
 - endmqm 288
 - endmqtrc 291
 - instmqm 292
 - runmqchi 293
 - runmqchl 294
 - runmqdlq 295
 - runmqlsr 296
 - runmqsc 99, 298
 - runmqtrm 301
 - sensibile al maiuscolo/minuscolo 29
 - setmqaut 302
 - strmqcsv 309
 - strmqm 310
 - strmqtrc 311
 - upgmqm 314
- comandi MQSC
 - ALTER QLOCAL 109
 - ALTER QREMOTE 135
 - DEFINE CHANNEL 129
 - DEFINE QALIAS 115
 - DEFINE QLOCAL 109
 - DEFINE QLOCAL LIKE 109
 - DEFINE QLOCAL REPLACE 109
 - DEFINE QMODEL 117
 - DEFINE QREMOTE 133
 - DELETE QLOCAL 112
 - DISPLAY QREMOTE 135
 - lunghezza massima della linea 103
 - sensibilità al
 - maiuscolo/minuscolo 99
 - utilizzo 31
- comandi PATHCOM
 - ADD SERVER 32
 - ALTER SERVER 33
 - FREEZE SERVER 32
 - START SERVER 32
 - STATUS SERVER 32
 - STOP SERVER 32
 - THAW SERVER 32
- comandi PCF
 - attributi in MQSC e PCF 122
 - automazione delle attività di gestione utilizzando PCF 121
 - escape PCF 122
 - MQAI, utilizzo per semplificare l'impiego di 122
- comando altmqfls 110, 254
 - comandi correlati 258
 - conservazione del server di code in memoria 226
 - controllo dei messaggi non permanenti 226
 - creazione della struttura di dati all'avvio del server di code 225
 - impostazione della soglia di browse 225
- comando altmqfls 110, 254 (*Continua*)
 - lunghezza della coda 226
 - riposizionamento dei file di coda 223
- comando altmqusr 259
 - comandi correlati 260
- comando cleanrdf 261
- comando cnvclchl 263
- comando crtmqcvx 265
- comando crtmqm 267
- comando di avvio Queue Manager 310
- comando di avvio traccia di MQSeries 311
- Comando DISPLAY CHSTATUS 339
- comando dltmqm 271
- comando dspmqaut 273
 - utilizzo 144, 146
- comando dspmqcsv 277
- comando dspmqfls 278
- comando dspmqtrc 282
- comando dspmqusr 283
- comando endmqcsv 285
- comando endmqm 67, 288
- comando endmqtrc 291
- comando fix 30
- comando instmqm 292
- comando runmqchi 293
- comando runmqchl 294
- comando runmqdlq 159, 295
- comando runmqlsr 296
- comando runmqtrm 301
- comando setmqaut 302
 - comandi correlati 308
 - servizi installabili 146
 - utilizzo 144, 145
- comando strmqcsv 309
 - comandi correlati 309
- comando strmqm 310
 - comandi correlati 310
- comando strmqtrc 311
 - comandi correlati 313
- comando upgmqm 314
- command server
 - arresto del server di comando 124
 - avvio di un server di comando 123
 - comando start 309
 - end comando 285
 - gestione remota 123
 - ripristino errore 239
 - visualizza comando 277
 - visualizzazione dello stato 123
- commit a due fasi 179
- commit a fase singola 179
- commit e bailout 179
- Compaq NonStop Server for Java 335
- Compaq NSKID utente connesso 147
- compilazione e binding delle applicazioni 354
- componente servizio 18
- Configuration - stanza, file QMINI 189
- configurazione per l'integrità dei dati 246
- configurazione per la disponibilità 246
- CONNNAME
 - attributo process 340
 - conservazione nella cache 226
 - contenuto di MQSINI 187

- contenuto di (*Continua*)
 - QMINI 189
- controllo
 - messaggi non permanenti dal server di code 226
 - Stato TMF 181
- Controllo dei Queue Manager 171
- controllo di un canale 89
- controllo di una coda 81
- convenzioni di denominazione
 - classi queue server 61
 - classi status server 58
 - supporto lingua nazionale 251
- conversione dei dati 137, 347
 - comando crtmqcvx 265
 - conversione di dati predefinita 137
 - conversione di formati di messaggio definiti dall'utente 138
 - EBCDIC 51
- conversione di dati predefinita 137
- copia di un canale 94
- copia di una coda 79
- CorrelId
 - considerazioni sulle prestazioni durante l'utilizzo 207
- Costanti MQZAO e autorizzazioni 151
- CPU
 - bilanciamento del carico di lavoro 34
 - configurazione per la disponibilità 246
 - elevato utilizzo 227
 - ottimizzazione 219
 - server di coda 222
- creazione
 - coda 75
 - coda di trasmissione 135
 - definizione di canale 85
 - definizioni di processo 119
 - principali 65
 - queue manager 45, 64
- creazione di file
 - Java 335
- CURDEPTH 226

D

- database
 - congruenza 236
 - congruenza, esterni 236
 - file dei database controllati 238
 - protetti da TM/MP 236
- database dei principal 238
- database OAM 238
- database XA compatibili 20
- dati non permanenti 233
- dati permanenti 233
- dead-letter header, MQDLH 159
- DefaultProcess - stanza, file QMINI 189
- DefaultQueueManager - stanza, file MQSINI 188
- definizione APC PATHWAY, esempio 386
- definizione automatica dei canali 131
- definizione delle code 75
- definizione dinamica dei canali 131
- definizioni di canali client 238
- definizioni di canali non client 238

- definizioni di processo
 - creazione 119
 - descrizione 15
 - visualizzazione 120
- definizioni elenco nomi 238
- determinazione dei problemi
 - controlli principali 199
 - controlli ulteriori 204, 207
 - errori di programmazione 203
 - file di configurazione 215
 - nessuna risposta dai comandi 204
 - output non corretto 210
- diagrammi railroad, come leggere 252
- diagrammi syntax, come leggere 252
- dimensione del messaggio, diminuzione 110
- dimensione traccia di controllo, TM/MP 181
- directory, queue manager 147
- disabilitazione dell'OAM (object authority manager) 143
- disponibilità
 - configurazione per 246
 - definizione 232
- DLQ handler
 - richiamo 159
 - tabella regole 160

E

- EBCDIC
 - conversione dei dati 51
- EC
 - file di controllo 238
 - funzione 22
 - ripristino errore 239
- EC - stanza, file QMINI 189
- EC Boss
 - ripristino errore 239
- EC Boss, ruolo di 49
- ECBoss
 - funzione 22
- ECBoss - stanza, file QMINI 189
- elaborazione, evento guidato 8
- elaborazione di evento guidato 8
- elaborazione di messaggi guidati 7
- eliminazione
 - canale 90
 - coda locale 81, 112
 - queue manager 68, 271
- EMSCollectorName 52
- errore di sintassi nei comandi MQSC 100
- errori di programmazione, esempi di 203
- errori relativi ai comandi 203
- esame delle code 112
- escape PCF 31, 122
- esecuzione del debug
 - controlli preliminari 199
 - controlli secondari 204, 207
 - errori di programmazione frequenti 203
- esempi
 - comando altmqfls 257
 - comando altmqusr 260
 - comando cleanrdf 262

- esempi (*Continua*)
 - comando cnvclchl 263
 - comando crtmqcvx 265
 - comando crtmqm 269
 - comando dltnqm 271
 - comando dspmqaut 275
 - comando dspmqcsv 277
 - comando dspmqfls 279
 - comando dspmqusr 283
 - comando endmqcsv 285
 - comando endmqm 289
 - comando endmqtrc 291
 - comando runmqsc 299
 - comando setmqaut 307
 - comando strmqcsv 309
 - comando strmqm 310
 - comando strmqtrc 313
 - comando upgmqm 315
 - errori di programmazione 203
 - impostazione comunicazioni 384
 - log degli errori 213
- esempi di comunicazione
 - ICE 389
 - SNAX 384
 - TCP/IP 393
- esempi di comunicazione ICE 389
- esempi di comunicazione SNAX 384
- esempi di comunicazione TCP/IP 393
- esempio di dati traccia 216
- eventi
 - canale 172
 - code 172
 - EMS 405
 - strumentazione
 - abilitazione 172
 - cosa sono 171
 - descrizione 171
 - messaggio 173
 - perché usarli 171
 - tipi di 171
 - supporto in MQSeries for Compaq NSK 397
 - tipi di 171
 - trigger 172
- eventi di strumentazione
 - abilitazione 172
 - descrizione 171
 - messaggi 173
 - scopo 171
 - supportato da MQSeries for Compaq NSK 397
 - tipi di 171
- eventi EMS 173, 214
 - collector alternativo, specifica 177
 - collector predefinito 177
 - impostazione di MQEMSEVENTS PARAM 176
 - scrittura dei programmi da elaborare 177
- Eventi EMS (Event Management Service) 173
- eventi inhibit 397
- eventi locali 397
- eventi prestazioni 171, 397
- eventi start e stop 397
- exit
 - channel exit 18

exit (*Continua*)
cluster workload exit 18
user exit 18
ExpectedNumECs 48

F

feedback da comandi MQSC 100
FFST
analisi 216
volume secondario 61
file
autorizzazione 156
catalogo oggetti 238
configurazione
MQSeries (MQSINI) 187
nella determinazione dei problemi 215
queue manager (QMINI) 189
database OAM 238
database principal 238
definizioni di canali non client 238
definizioni elenco nomi 238
ENSCRIBE 27
file dei database controllati 238
file di coda 222, 238
posizione 223
file di controllo EC 238
file di overflow coda 222, 238
posizione 223
inizializzazione 25
namelist 28
overflow messaggio 222
modifica della soglia 224
posizione 223
touch
per alias e remote 28
touch file 222, 238
posizione 223
touch file oggetti 238
file CCSID 137
file di coda 222, 238
riposizionamento 223
suddivisione 224
file di comandi 103
file di configurazione
modifica 197
MQSeries (MQSINI) 187
backup 46
contenuto 187
panoramica 187
percorso 105
panoramica 187
queue manager (QMINI)
backup 46
contenuto 189
disabilitazione di OAM (object authority manager) 143
voci 189
file di configurazione SCF, esempio 384
file di overflow coda 222, 238
riposizionamento 223
file di overflow messaggi
modifica della soglia 224
file di overflow messaggio 222
configurazione per l'integrità dei dati 246

file di overflow messaggio 222
(*Continua*)
riposizionamento 223
file ENSCRIBE 20, 27
file modelli di evento EMS,
ZMQSTMPL 173
formati messaggio definiti
dall'utente 138
formato messaggi di evento 398
Formato PostScript 415
forzatura dell'arresto di un canale
(arresto immediato) 91
FREEZE SERVER, comando
PATHCOM 32

G

gestione
autorizzazioni 154
gruppi di comandi 29
comandi di controllo 29
comandi MQSeries (MQSC) 30
Comandi PCF (Programmable Command Format) 31
locale 97
gestione degli oggetti per 118
gestione dell'unità di lavoro 350
gestione delle licenze 321
Gestione di TS/MP 32
gestione locale 97
gestione remota
command server 123
problemi iniziali 132
glossario 417
gruppo comandi
confronto 329
gestione 29
gruppo utenti
MQM 139
per l'autorizzazione 142

H

HARDENBO
attributo process 340
HTML (Hypertext Markup Language) 414
Hypertext Markup Language (HTML) 414

I

ICE TP 379
ID di segmento 54
ID di segmento guardian 54
ID utente
autorizzazione 139, 147
Compaq NSK utente connesso 147
ID utente Guardian 238
immissione di comandi MQSeries 98
impostazione comunicazioni 377
integrità dei dati
configurazione per 246
definizione 232
Inter-Process Communication Component (IPCC) 26

interfaccia Name Service 404
interfaccia XA 351
interruzione
canali 91
traccia 74

J

Java
compatibile 335
linguaggio supportato 352
JTS (Java Transaction Services) 353

L

library
MQI 354
linguaggio
supportato 352
listener TCP/IP
configurazione 32
ripristino errore 239
start 32
stop 32
Local Queue Manager Agent (LQMA)
funzione 22
log, errori 212, 213
log degli errori 212
errori verificatisi prima
dell'installazione 213
esempio 213
volume secondario 62
LQMA (local queue manager agent)
ripristino errore 239
LQMA (Local Queue Manager Agent) 228
LQMAgent - stanza, file QMINI 190
LQMAgentPriority 53
LU6.2 listener 25
lunghezza massima della linea di comandi MQSC 103

M

manuali in formato elettronico 414
massima
dimensione di una coda 9
massimo
numero di messaggi 9
MaxActiveChannels 53
Maxapplio, valore di 389
MaxChannels 53
MAXHANDS
attributo process 339
MaxIdleAgentReuse 52
MaxIdleAgents 52
maxrcv PARAM 390
MAXUMSGS
attributo process 339
MCA (Message Channel Agent)
funzione 23
ripristino errore 239
MCAAgentPriority 53
MCACaller - stanza, file QMINI 190
MCALU62Responder - stanza, file QMINI 190

MCATCPResponder - stanza, file
 QMINI 190
 measure counter 226
 menu Queues, MQM 74
 Message Queue Management (MQM) 69
 messaggi
 non permanenti 220
 permanente 220
 messaggi di errore 100
 messaggi non permanenti
 configurazione per l'integrità dei
 dati 246
 controllo 226
 differenza dai permanenti 8
 disponibilità 235
 memorizzazione 222
 ottimizzazione 220
 registrazione nei log
 sincronizzata 221
 messaggi per l'operatore 213
 messaggi permanenti
 differenza dai non permanenti 8
 disponibilità 234
 memorizzazione 222
 ottimizzazione 220
 sostituzione del numero massimo
 predefinito 327
 messaggio
 accodamento 7
 algoritmi di richiamo 10
 assenti nelle code 208
 che contiene informazioni
 inattese 210
 considerazioni sulle prestazioni 207
 descrittore 8
 descrizione 8
 dimensione 222
 dimensione di 8
 lunghezza di 207
 lunghezza variabile 208
 non recapitato 215
 operatore 213
 per eventi di strumentazione 173
 raggruppamento 9
 ricerca di un particolare 207
 segmento 9
 MinIdleLQMAgents 52
 MinIdleMCACallers 52
 MinIdleMCALU62Responders 52
 MinIdleMCATCPResponders 52
 modalità accodamento, di runmqsc 131
 modelli, evento EMS 173
 modello code
 attributi 347
 modello di evento EMS, NQSeries 405
 modifica
 canale 84
 CCSID 51
 coda 80
 modifica degli attributi di coda 109
 modifica degli attributi Queue
 Manager 101
 MQ_LOAD_ENTRY_POINT_EXIT 368
 MQAI
 descrizione di 122, 357
 MQCFH, intestazione PCF 400
 MQCFIL, parametro integer list PCF 401
 MQCFIN, integer PCF 401
 MQCFSL, string list PCF 401
 MQCFST, parametro di stringa PCF 401
 MQCNO 342
 MQCNO_FASTPATH_BINDING 228
 MQCONN 228
 MQCONNECTTYPE 228
 MQCONNX 228
 MQDATACONVEXIT 347
 MQDEFAULTPREFIX 325
 MQDISC 345
 MQDLH, dead-letter header 159
 MQEMSEVENTS 325
 MQEMSEVENTS, variabile di
 ambiente 176
 MQGMO 342
 MQGMO_BROWSE_* 351
 MQI
 autorizzazioni 151
 chiamate 344
 chiamate Queue Manager 12
 descrizione 7
 library 354
 Linguaggio di programmazione TAL
 MQCLOSE 345
 MQDISC 345
 MQINQ 346
 MQSET 346
 supporto gestione locale 97
 MQI (Message Queue Interface) 7
 MQINQ 346
 MQIServer - stanza, file QMINI 190
 MQLISTENPORTNUM 325
 MQM
 gruppo utenti 139
 ID utente 139, 147
 MQM (Message Queue Management) -
 Interfaccia 69
 MQMACHINIFILE 325
 MQMACHINIFILE, variabile di
 ambiente 187
 MQMC - funzione 69
 MQMD 343
 MQMREFRESHINT
 parametro PATHWAY 35
 MQPMO 343
 MQPUT e MQPUT1, considerazioni sulle
 prestazioni 208
 MQQSHKEEPINT 327
 MQQSMAXBATCHEXPURE 327
 MQQSMAXMSGSEXPURE 327
 MQQSSIGTIMEOUT 327
 MQRC_Q_SPACE_NOT_AVAILABLE
 return code 107
 MQRDF 325
 MQRDFUPPROCESSNAME 325
 MQRDFUPPROGNAME 325
 MQS-CHANINIT00
 classe di server TS/MP 32
 MQS-CMDSERV00
 classe di server TS/MP 32
 MQS-EC00
 classe di server TS/MP 34
 MQS-TCPLIS00
 classe di server TS/MP 32
 MQS-TCPLISnn
 riconfigurazione 55
 MQS-TRIGMON00
 classe di server TS/MP 33
 MQSC 31
 come immettere i comandi 98
 emissione remota 131
 file di comandi
 esecuzione 104
 prospetti di output 103
 immissione interattiva di comandi 99
 lunghezza massima della linea 103
 problemi
 locali 105
 remoto 132
 reindirizzamento di input e
 output 102
 requisiti di sicurezza sui canali 150
 risposte dai comandi scadute 131
 supportato da MQSeries for Compaq
 NSK 337
 termine di input interattivo 100
 utilizzo dei comandi 102
 verifica dei comandi 104
 MQSC interattivi
 feedback dei 100
 termine 100
 utilizzo 99
 MQSeries for Compaq NSK
 architettura 21
 componenti 25
 creazione delle applicazioni 349
 esecuzione delle applicazioni 349
 file 27
 modello di evento EMS utilizzato
 da 405
 MQSC supportato da 337
 prestazioni 219
 riepilogo 319
 scalabilità 219
 struttura dei volumi 61
 super user, MQM 139
 supporto client 395
 Supporto TM/MP (TMF) 179
 uscite utente 367
 volumi secondari 27
 MQSET 346
 MQSINI, file di configurazione
 esempio 188
 modifica 197
 panoramica 187
 percorso per 105
 MQSNOAUT 325
 MQSNOAUT, variabile di ambiente 143
 MQSS server 24
 MQTRANSACTIONLIFE
 aggiornare la transazione
 TM/MP 349
 MsgId
 considerazioni sulle prestazioni
 durante l'utilizzo 207

N

Name Service 19
 NOHARDENBO
 attributo process 340
 nome dello queue server predefinito 59
 nome dello status server predefinito 56

- nomi
 - oggetti 10
 - valido per gli oggetti 251
- NonStop Java
 - linguaggio supportato 352
- NonStop Process Pair 24
- NonStop TM/MP (Transaction Manager) 179
- NonStop Tuxedo
 - ambiente di transazione con TM/MP 238
- notificazione di eventi 172
- NSKSegidRange 54

O

- OAM (object authority manager)
 - comando dspmqaut 147
 - comando setmqaut 144, 145
 - come funziona 141
 - disabilitazione 143
 - gruppi 141
 - operazioni riservate 147
- OAM (Object Authority Manager)
 - introduzione a 19
 - sostituzione 403
- oggetti
 - accesso a 139
 - coda 12
 - convenzioni di denominazione 251
 - definizione di processo 15
 - gestione remota 125
 - nomi 98
 - per triggering 118
 - predefiniti
 - attributi 108
 - predefinito
 - ripristino 66
 - Queue Manager
 - Chiamate MQI 12
 - sistema
 - predefiniti 16, 323
 - ripristino 66
 - tipi di 10
- oggetti del sistema
 - ripristino 66
- oggetti predefiniti di sistema 16
- OpenTMF 237
- opzione di segnalazione 342
- opzione MQGMO_SET_SIGNAL 342
- ottimizzazione di MQSeries for Compaq NSK 219
- output non corretto 208

P

- pannelli
 - MQM 69
- panoramica di MQSeries for Compaq NSK 319
- PARAM (variabili di ambiente) 325
- parameters
 - comando crtmqcvx 265
 - comando dltnmqm 271
 - comando dspmqaut 273
 - comando dspmqcsv 277

- parameters (*Continua*)
 - comando dspmqtrc 282
 - comando endmqcsv 285
 - comando endmqm 288
 - comando instmqm 292
 - comando runmqchi 293
 - comando runmqchl 294
 - comando runmqtrm 301
 - comando setmqaut 304
 - comando strmqcsv 309
 - comando strmqm 310
 - comando strmqtrc 311
 - comando upgmqm 315
- parametri
 - comando almqfls 255
 - comando almqusr 259
 - comando cleanrdf 261
 - comando cnvclchl 263
 - comando crtmqm 267
 - comando dspmqfls 278
 - comando dspmqusr 283
 - comando endmqtrc 291
 - comando runmqdlq 295
 - comando runmqslr 296
 - comando runmqsc 299
- parametro CCSID 51
- Parametro DefaultPrefix, comando crtmqm 48
- parametro HomeTerminalName 49
- Parametro NumECs, comando crtmqm 48
- parola chiave APPLIDAT, tabella regole 162
- parola chiave APPLNAME, tabella regole 162
- parola chiave APPLTYPE, tabella regole 162
- parola chiave DESTQ, tabella regole 162
- parola chiave DESTQM, tabella regole 162
- parola chiave di azione 163
- parola chiave FEEDBACK, tabella regole 162
- parola chiave FORMAT, tabella regole 163
- parola chiave FWDQ, tabella regole 164
- parola chiave FWDQM, tabella regole 164
- parola chiave HEADER, tabella regole 164
- parola chiave INPUTQ, tabella regole 160
- parola chiave INPUTQM, tabella regole 161
- parola chiave MSGTYPE, tabella regole 163
- parola chiave PERSIST, tabella regole 163
- parola chiave PUTAUT, tabella regole 164
- parola chiave REASON, tabella regole 163
- parola chiave REPLYQM, tabella regole 163
- parola chiave RETRY, tabella regole 164
- parola chiave RETRYINT, tabella regole 161

- parola chiave USERID, tabella regole 163
- parola chiave WAIT, tabella regole 161
- parole chiave dello schema di confronto, tabella regole 162
- parole chiave REPLYQ, tabella regole 163
- PathmonProcName 49
- PATHWAY
 - classe di server 42
 - classe server per il server di code 222
 - configurazione per la disponibilità 246
 - errori 202
 - esempio di configurazione 35
- PCF (programmable command format) 399
 - descrittore messaggio 400
 - integer (MQCFIN) 401
 - intestazione (MQCFH) 400
 - introduzione a 31
 - MQCFH (intestazione) 400
 - MQCFIL (parametro integer list) 401
 - MQCFIN (integer) 401
 - MQCFSL (string list) 401
 - MQCFST (parametro di stringa) 401
 - parametro di stringa (MQCFST) 401
 - parametro integer list (MQCFIL) 401
 - string list (MQCFSL) 401
 - supportato da MQSeries for Compaq NSK 402
- PDF (Portable Document Format) 414
- PMSEARCH 325
- Portable Document Format (PDF) 414
- predefinite
 - coda di trasmissione 135
- predefiniti
 - oggetti 16
 - oggetti del sistema 323
 - queue manager
 - comandi elaborati 99
 - eliminazione involontaria 268
 - modifica 101
- predefinito
 - coda di trasmissione 46
 - oggetti 66
 - porta TCP/IP 54
 - queue manager 46
 - modifica 68
 - modifica accidentale 68
 - queue server, nome dello 59
 - status server, nome dello 56
- prestazioni 219
 - considerazioni durante l'utilizzo di traccia 215
 - considerazioni sulla progettazione delle applicazioni 207
- PRIDB, database dei principal 156
- principal
 - scopo di 19
- principal OAM 238
- principali
 - creazione 65, 144
 - database, PRIDB 156
 - denominazione di 141
- priorità dei processi queue manager 53

- processi agente 26, 52
- processi agente inattivi 52
- processi EC, numero di 48
- Processo responder LU6.2 378
- processo TCP/IP 55
- processo TCP/IP non predefinito 55
- programmabile command format (PCF) 31, 399
- programmi, esempi forniti 359
- programmi di esempio
 - creazione versioni C 360
 - creazione versioni in C++ 362
 - creazione versioni in COBOL 363
 - creazione versioni in OSS 365
 - creazione versioni in TAL 365
 - forniti con MQSeries for Compaq NSK 359
- protocollo SNA 377
- protocollo TCP/IP 377
- pubblicazioni correlate 415
- Pubblicazioni MQSeries 413
- punti molteplici di errore 231
- punto singolo di errore 231

Q

- QMDefaultVolume 48
- QMINI, file di configurazione
 - configurazione per il supporto dei listener TCP/IP 394
 - esempio 191
 - modifica 197
 - panoramica 189
- queue manager
 - alias, coda remota 136
 - arresto
 - manuale 333
 - attributo 347
 - autorizzazioni 147
 - avvio di 66
 - chiusura
 - ad attività controllata 67
 - controllata 67
 - chiusura immediata 67
 - chiusura prioritaria 67
 - comando endmqm 288
 - command server 123
 - componenti 25
 - controllo 171
 - creazione
 - comando crtmqm 267
 - modalità 45
 - predefinito 64
 - directory 147
 - disponibilità 232
 - eliminazione 68, 271
 - eventi 171
 - file di configurazione
 - backup 46
 - contenuto di 189
 - gestione locale 97
 - gestione remota 125
 - interruzione 67
 - nome univoco 46
 - numeri di 46
 - OAM (object authority manager)
 - descrizione 140

- queue manager (*Continua*)
 - OAM (object authority manager) (*Continua*)
 - disabilitazione 143
 - predefiniti
 - eliminazione involontaria 268
 - predefinito 46
 - modifica 68
 - modifica accidentale 68
 - processi 21
 - proprietà configurabili di 48
 - riavvio 68
 - rimozione, manuale 333
 - ripristino errore 239
 - specifiche di runmqsc 101
 - su MVS/ESA 132
 - volume iniziale 48
 - volume secondario coda messaggi 63
 - volume secondario FFST 61
 - volume secondario file di dati 61
 - volume secondario log errori 62
 - volume secondario sincronizzazione canali 63
- Queue Manager
 - descrizione 11
 - oggetti
 - Chiamate MQI 12
- Queue Manager Menu, MQM 70
- Queue Manager MVS/ESA 132
- queue manager server
 - ripristino errore 239
- queue server
 - ripristino e riavvio 185
 - ripristino errore 239
- queue server non predefinito, aggiunta e rimozione 60
- QueueManager - stanza, file MQSINI 188

R

- RDF (Remote Database Duplication Facility)
 - comando cleanrdf 66, 261
 - ripristino mediante 185
- reimpostazione di un MSN (message sequence number) 92
- reindirizzamento di input e output, con comandi MQSC 102
- remota
 - accodamento
 - consigli 132
 - code
 - come alias di coda di risposta RTQ 136
 - come alias di Queue Manager 136
 - definizione di coda, creazione 133
 - emissione di comandi MQSC 131
 - gestione oggetti 125
 - oggetto coda, funzionante con 136
- remote
 - code
 - autorizzazioni per 148
 - considerazioni sulla sicurezza 149
- remoti
 - eventi 397

- repository cache manager
 - ripristino errore 239
- repository manager
 - processo 23
 - ripristino errore 239
- restrizioni
 - accesso agli oggetti MQM 139
 - nomi degli oggetti 251
- riavvio di un Queue Manager 68
- rimozione del queue server 61
- rimozione di queue server non predefinito 60
- rimozione di status server non predefinito 57
- rimozione di uno status server 58
- rimozione manuale dei queue manager 333
- ripartizione del carico di lavoro 49
- ripristino
 - oggetti del sistema 66
 - oggetti predefiniti 66
- ripristino errore 239
- risoluzione dei problemi 182
- risoluzione di un canale 93
- risorse, protezione 139
- risorse protette 142
- risposte dai comandi PCF 403
- risposte scadute dai comandi MQSC 131
- rollback 179
- run listener (comando runmqslr) 296
- runmqsc
 - comandi 298
 - feedback 100
 - immissione di comandi MQSC 98
 - modalità accodamento 131
 - problemi 105
 - reindirizzamento di input e output 102
 - specifiche di un Queue Manager 101
 - termine 100
 - utilizzo 102
 - utilizzo interattivo 99
 - verifica 104

S

- SAFEGUARD 19, 142, 320
- SAVE-ENVIRONMENT ON 325
- scalabilità 219
- segmenti di memoria condivisi 24
- sensibilità al maiuscolo/minuscolo
 - comandi di controllo 29
 - comandi MQSC 99
- server di coda
 - aggiunta e rimozione di non definiti 60
 - descrizione 24
 - gestione dei dati non permanenti 235
 - memorizzazione dei messaggi 225
 - nome del predefinito 59
 - PARAM 327
 - processi
 - aggiunta 35
 - rimozione 61
 - server di code
 - conservazione della struttura di dati e delle catene in memoria 226

- server di code (*Continua*)
 - controllo dei messaggi non permanenti 226
 - creazione della struttura di dati all'avvio del server di code 225
 - distribuzione della CPU 222
 - distribuzione delle code tra 223
 - measure counter per la coda 226
 - ottimizzazione 222
- Server di code NonStop 221
- server di comando
 - configurazione mediante comandi PATHCOM 32
- server di stato
 - aggiunta e rimozione di non definiti 57
 - nome del predefinito 56
 - rimozione 58
 - ripristino errore 239
- Service - stanza, file QMINI 190
- ServiceComponent - stanza, file QMINI 190
- servizi installabili
 - Name Service 404
 - object authority manager (OAM) 140
 - disabilitazione 143
 - servizio di autorizzazione 403
- servizio di autorizzazione 19
 - interfaccia 403
- Shared Resource Library 23
- sicurezza 139
 - abilitazione 143
 - OAM (Object Authority Manager) 19
 - principal, creazione 65
 - remote 149
 - utilizzo dei comandi 144, 147
- SNA
 - consumo della CPU 227
 - Listener SNA 211
- stanza TCPConfig
 - configurazione TCP/IP 393
- Start Channel Initiator 403
- Start Channel Listener 403
- START SERVER, comando PATHCOM 32
- status server
 - classe di server MQS-STATUS00 34
 - processi
 - aggiunta 35
 - ripristino e riavvio 185
- STATUS SERVER, comando PATHCOM 32
- status server non predefinito, aggiunta e rimozione 57
- stdin, su runmqsc 102
- stdout, su runmqsc 102
- STOP SERVER, comando PATHCOM 32
- suddivisione
 - file di coda 224
- super user (MQSeries) MQM 139
- supporto euro 321
- supporto TM/MP (TMF) 350
- Supporto TM/MP (TMF) 179
- supporto transazionale 179
- SupportPac 415
- sync file canali 238

- synchpoint 236
- syncpoint
 - limiti 180
- syncpoint, considerazioni sulle prestazioni 208
- system.cluster.transmit.queue 223

T

- tabella regole, DLQ handler 160
 - elaborazione di 166
 - esempio 168
 - schemi e azioni, (regole)
 - parola chiave ACTION 163
 - parola chiave APPLIDAT 162
 - parola chiave APPLNAME 162
 - parola chiave APPLTYPE 162
 - parola chiave DESTQ 162
 - parola chiave DESTQM 162
 - parola chiave FEEDBACK 162
 - parola chiave FORMAT 163
 - parola chiave FWDQ 164
 - parola chiave FWDQM 164
 - parola chiave HEADER 164
 - parola chiave MSGTYPE 163
 - parola chiave PERSIST 163
 - parola chiave PUTAUT 164
 - parola chiave REASON 163
 - parola chiave REPLYQ 163
 - parola chiave REPLYQM 163
 - parola chiave USERID 163
 - schemi e azioni (regole)
 - parola chiave RETRY 164
 - sintassi 165
 - voce dati di controllo
 - parola chiave INPUTQ 160
 - parola chiave INPUTQM 161
 - parola chiave RETRYINT 161
 - parola chiave WAIT 161
- TAL non nativo
 - linguaggio supportato 352
- TCP/IP
 - canali 383
 - gestione remota 17
- TCPConfig - stanza, file QMINI 191
- TCPLListener - stanza, file QMINI 190
- TCPLListenerPort 54
- TCPNumListenerPorts 54
- TCPPort 54
- termine dei comandi interattivi MQSC 100
- terminologia utilizzata in questo manuale 417
- THAW SERVER, comando PATHCOM 32
- thread
 - applicazioni 353
- tipi di dati
 - tipi di dati di struttura 341
- tipi di dati di struttura 341
- tipi di eventi 171
- tipi di oggetto 10
- TM/MP 20
 - controllo dei file di database 236
 - coordinazione con i database esterni 236
 - file di controllo 220

- touch file 222, 238
 - riposizionamento 223
- touch file oggetti 238
- traccia
 - considerazioni sulle prestazioni 215
 - esempio di dati 216
- traccia degli oggetti di MQSeries 72
- Transaction Manager (NonStop TM/MP) 179
- transazione
 - gestione multipla 238
- trasformazione dei nomi di oggetto 63
- trasformazione nomi, oggetto 63
- trigger
 - eventi 172
 - messaggi su code di iniziazione 13
 - monitor
 - comando start 301
 - configurazione mediante comandi PATHCOM 33
 - descrizione 14
- triggering
 - definizione 8
 - definizione della coda applicazioni per 118
 - gestione degli oggetti per 118
- TuningParameters - stanza, file QMINI 190
- Tuxedo, NonStop
 - ambiente di transazione con TM/MP 238

U

- unità di elaborazione globale
 - descrizione 19
- unità di elaborazione locale
 - descrizione 19
- UOW (unit of work)
 - descrizione 19
- uscita
 - formato nome 368
 - installazione 372
 - MQ_LOAD_ENTRY_POINT_EXIT 368
 - uscite utente 367
- uscite di conversione dati 367
- uscite utente 367
- user exit
 - channel exit 18
 - cluster workload 18
 - data conversion exit 18
 - descrizione 18
- USERDATA
 - attributo process 340
- utenti
 - appartenenza a più di un gruppo utenti 141
 - gruppi 141

V

- valori predefiniti di sistema 323
- variabili di ambiente 325
- variabili di ambiente TAQL 325
- verifica dei comandi MQSC 104

- visualizzazione
 - attributi Queue Manager 100
 - comando autorizzazione 273
 - comando output di traccia formattata di MQSeries 282
 - comando server di comando 277
 - definizioni di processo 120
 - file MQSeries comando 278
 - stato del server di comando 123
 - stato di un canale 90
- voce ExtPoolSize
 - file QMINI 56
- voci
 - MQSINI 187
 - QMINI 189
- volume
 - struttura 61
- volume, modifica 110
- volume disco
 - suddivisione dei file di coda 224
- volume iniziale del queue manager 48
- volume secondario 27
- volume secondario coda messaggi 63
- volume secondario file di dati, queue manager 61
- volume secondario sincronizzazione canali 63

W

- Windows Help 415

Riservato ai commenti del lettore

MQSeries® for Compaq NonStop™ Kernel
Guida alla gestione del sistema
Versione 5 Rilascio 1

Pubblicazione N. SC13-2968-00

Commenti relativi alla pubblicazione in oggetto potranno contribuire a migliorarla. Sono graditi commenti pertinenti alle informazioni contenute in questo manuale ed al modo in cui esse sono presentate. Si invita il lettore ad usare lo spazio sottostante citando, ove possibile, i riferimenti alla pagina ed al paragrafo.

Si prega di non utilizzare questo foglio per richiedere informazioni tecniche su sistemi, programmi o pubblicazioni e/o per richiedere informazioni di carattere generale.

Per tali esigenze si consiglia di rivolgersi al punto di vendita autorizzato o alla filiale IBM della propria zona oppure di chiamare il "Supporto Clienti" IBM al numero verde 167-017001.

I suggerimenti ed i commenti inviati potranno essere usati liberamente dall'IBM e dalla Selfin e diventeranno proprietà esclusiva delle stesse.

Commenti:

Si ringrazia per la collaborazione.

Per inviare i commenti è possibile utilizzare uno dei seguenti modi.

- Spedire questo modulo all'indirizzo indicato sul retro.
- Inviare un fax al numero: +39-081-660236
- Spedire una nota via email a: translationassurance@selfin.it

Se è gradita una risposta dalla Selfin, si prega di fornire le informazioni che seguono:

Nome

Indirizzo

Società

Numero di telefono

Indirizzo e-mail

Indicandoci i Suoi dati, Lei avrà l'opportunità di ottenere dal responsabile del Servizio di Translation Assurance della Selfin S.p.A. le risposte ai quesiti o alle richieste di informazioni che vorrà sottoporci. I Suoi dati saranno trattati nel rispetto di quanto stabilito dalla legge 31 dicembre 1996, n.675 sulla "Tutela delle persone e di altri soggetti rispetto al trattamento di dati personali". I Suoi dati non saranno oggetto di comunicazione o di diffusione a terzi; essi saranno utilizzati "una tantum" e saranno conservati per il tempo strettamente necessario al loro utilizzo.

Selfin S.p.A.
Translation Assurance

Via F. Giordani, 7

80122 NAPOLI



Numero parte: CT8Z0IT

Printed in Denmark by IBM Danmark A/S

SC13-2968-00



(1P) P/N: CT8Z0IT

