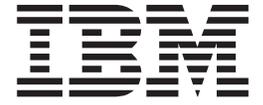


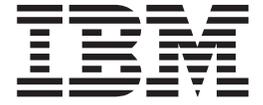
MQSeries for Compaq NonStop Kernel



Systemverwaltung

Version 5 Release 1

MQSeries for Compaq NonStop Kernel



Systemverwaltung

Version 5 Release 1

Anmerkung

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen in „Anhang Q: Bemerkungen“ auf Seite 417 gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: **ibm.com**
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business Symbol ist eine Marke der International Business Machines Corporation
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle Java-basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Änderungen in der IBM Terminologie

Die ständige Weiterentwicklung der deutschen Sprache nimmt auch Einfluß auf die IBM Terminologie. Durch die daraus resultierende Umstellung der IBM Terminologie, kann es u. U. vorkommen, dass in diesem Handbuch sowohl alte als auch neue Termini gleichbedeutend verwendet werden. Dies ist der Fall, wenn auf ältere existierende Handbuchausschnitte und/oder Programmteile zurückgegriffen wird.

Erste Ausgabe (Juni 2001)

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
MQSeries for Compaq NonStop Kernel System Administration Version 5 Release 1,
IBM Form SC34-5886-00,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1993, 2001
© Copyright IBM Deutschland GmbH 2001

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
Juni 2001

Inhaltsverzeichnis

Abbildungsverzeichnis xi

Tabellen xiii

Zu diesem Handbuch xv

Zielgruppe xv

Voraussetzungen zur Benutzung dieses Handbuchs xv

Benutzung des Handbuchs xv

Informationen zu MQSeries im Internet. xvi

**Neuerungen in MQSeries for Compaq
NSK V5.1 xvii**

Leistungsverbesserungen xvii

Aufrüstungen an MQSeries-Funktionen xviii

Kommunikation xix

Compaq NSK-spezifische Verbesserungen der
Benutzerfreundlichkeit. xx

Teil 1. Anleitung 1

Kapitel 1. Einführung 7

MQSeries und Message-Queuing 7

 Zeitunabhängige Anwendungen 7

 Nachrichtengesteuerte Verarbeitung 8

Nachrichten und Warteschlangen 8

 Nachrichten 8

 Warteschlangen 9

Objekte 10

 Objektnamen 11

 Objekte verwalten 11

 MQSeries-WS-Manager 12

 MQSeries-Warteschlangen 12

 Prozessdefinitionen 16

 Kanäle 16

 Cluster 17

 Namenslisten 17

Systemstandardobjekte 17

Verwaltung 18

 Lokale und ferne Verwaltung 18

Clients und Server 19

 MQI-Anwendungen in einer Client/Server-
 Umgebung 19

WS-Managerfunktionen erweitern 19

 Benutzer-Exits 19

 Installierbare Services 20

Sicherheit 21

 Objektberechtigungsmanager (OAM, Object
 Authority Manager 21

Transaktionsunterstützung 21

Durchsatzverbesserung, Zuverlässigkeit, Skalierbar-
keit und Größenanpassung 22

Kapitel 2. MQSeries for Compaq NSK

V5.1 - Architektur 23

WS-Managerprozesse - Übersicht 23

Produktbindungen 25

Ausführbare Module 26

Vergleich der Architekturen von Version 5.1 und
Version 2.2.0.1 26

WS-Manager — Funktionsübersicht 27

WS-Managerprozessmodell 28

MQSeries-Dateien und -Teildatenträger 29

**Kapitel 3. MQSeries-Befehlssätze ver-
wenden 31**

Verwaltungs-Tasks über Steuerbefehle ausführen 31

 Steuerbefehle verwenden 31

Verwaltungs-Tasks über MQSC-Befehle ausführen 32

 MQSC-Befehle ausführen 32

Verwaltungs-Tasks über PCF-Befehle ausführen 33

 Attribute in MQSC- und PCF-Befehlen 33

 PCF-Escape-Befehle. 33

TS/MP-Verwaltung (PATHWAY-Verwaltung) 34

 TCP/IP-Empfangsprogramme konfigurieren und
 steuern 34

 Steuerung des Befehlsservers 34

 Kanalinitiatoren konfigurieren und steuern 35

 Auslösemonitore konfigurieren und steuern 35

 Prozesse über mehrere CPUs verteilen 36

 Neue Statusserverprozesse hinzufügen 37

 Neue Warteschlangenserverprozesse hinzufügen 37

 Aktualisierungsintervall für MQM-
 Überwachungsanzeigen festlegen 38

 PATHWAY-Konfiguration für einen WS-Manager 38

 Parameter von PATHWAY-Serverklassen ändern 45

 Benutzerdefinierte Serverklassen einer
 PATHWAY-Konfiguration in MQSeries hinzufü-
 gen 45

Kapitel 4. WS-Manager verwalten 47

Einführung 47

Vorgehensweise beim Erstellen von WS-Managern 47

 Sicherungskopien von Konfigurationsdateien
 nach Erstellung eines WS-Managers erstellen 50

WS-Managermerkmale ändern 50

 Ausgangsdatenträger des WS-Managers 50

 Anzahl der Ausführungs-Controller pro
 WS-Manager 50

 Ausgangsterminal des WS-Managers 51

 PATHMON-Prozessname für den WS-Manager 52

 CCSID des WS-Managers. 53

 EBCDIC-Datenkonvertierung steuern. 53

 EMS-Ereignissammler für WS-Manager 54

 Agenten-Pool der Ausführungs-Controller im
 WS-Manager 54

 Maximale Anzahl inaktiver Agenten und Prozess-
 wiederverwendung. 55

Priorität der WS-Managerprozesse	55
Maximale Anzahl an Kanälen für den WS-Manager	56
Maximale Anzahl an aktiven Kanälen für den WS-Manager	56
Guardian-Segment-IDs für MQSeries	56
TCP/IP-Standard-Port	57
Anzahl an TCP/IP-Ports, die vom WS-Manager überwacht werden	57
Der vom WS-Manager verwendete TCP/IP- Prozess	57
Serverklasse MQS-TCPLISnn für die Verwendung eines anderen TCP/IP-Prozesses und Ports rekonfigurieren	58
Zuordnung des Auslagerungsspeichers	59
Name des Standardstatusservers	59
Zusätzliche Statusserver hinzufügen und entfer- nen	60
Name des Standardwarteschlangenservers	62
Zusätzliche Warteschlangenserver hinzufügen und entfernen	63
Datenträgerstruktur	64
Teildatenträger mit den FFST-Dateien des WS-Managers.	65
Teildatenträger mit den Datendateien des WS-Managers.	65
Datenträger mit dem Fehlerprotokoll des WS-Managers.	66
Teildatenträger für Nachrichtenwarteschlangen des WS-Managers	66
Teildatenträger für Kanalsynchronisation des WS-Managers.	67
Umwandlung von Objektnamen	67
Mit WS-Managern arbeiten	67
Standard-WS-Manager erstellen	68
MQSeries-Principals erstellen	69
'cleanrdf' für einen RDF-aktivierten WS-Manager ausführen	70
WS-Manager starten	70
Standard- und Systemobjekte wiederherstellen. Objektdateien.	70
WS-Manager stoppen	71
Maßnahmen bei eventuellen Problemen	71
Sofortiger und erzwungener Abschluss eines WS-Managers.	71
WS-Manager erneut starten	72
Vorhandenen WS-Manager als Standard-WS- Manager definieren.	72
WS-Manager löschen	73
MQM (Message Queue Management) verwenden	73
'Queue Manager Menu' (WS-Managermenü)	74
Anzeige 'Queues Menu'	77
Anzeige 'Channels Menu'	83

Kapitel 5. Lokale MQSeries-Objekte ver- walten 95

Unterstützung von Anwendungsprogrammen, die MQI verwenden.	95
Lokale Verwaltungs-Tasks mit Hilfe von MQSC- Befehlen ausführen	96
Vor dem Start	97

Die MQSC-Funktion interaktiv verwenden	97
Rückmeldung auf MQSC-Befehle	98
Interaktive Eingabe von MQSC-Befehlen beenden	99
WS-Managerattribute anzeigen	99
Einen WS-Manager verwenden, der nicht der Standard-WS-Manager ist	100
WS-Managerattribute ändern	100
MQSC-Befehle aus Textdateien ausführen	100
MQSC-Befehlsdateien verwenden	101
MQSC-Berichtsdateien verwenden	102
Im Lieferumfang enthaltene MQSC-Befehlsdatei ausführen	103
Befehle mit Hilfe von runmqsc prüfen	103
MQSC-Probleme lösen	104
Mit lokalen Warteschlangen arbeiten	105
Eine lokale Warteschlange definieren	105
Größe der physischen Datei für Warteschlangen ändern	106
Eine Warteschlange für nicht zustellbare Nach- richten definieren	106
Standardobjektattribute anzeigen	107
Eine lokale Warteschlangendefinition kopieren	108
Attribute einer lokalen Warteschlange ändern	108
Datenträger einer lokalen Warteschlange ändern	109
Optionen einer lokalen Warteschlange ändern	109
Objekte Statusservern und Warteschlangen- servern erneut zuordnen.	110
Inhalt einer lokalen Warteschlange löschen	110
Eine lokale Warteschlange löschen	111
Warteschlangen durchsuchen	111
Mit Aliaswarteschlangen arbeiten.	114
Eine Aliaswarteschlange definieren	114
Andere Befehle für Aliaswarteschlangen ver- wenden	115
Mit Modellwarteschlangen arbeiten	116
Eine Modellwarteschlange definieren	116
Andere Befehle für Modellwarteschlangen ver- wenden	117
Objekte für Auslösefunktion verwalten	117
Anwendungswarteschlange für die Auslöse- funktion definieren	117
Eine Initialisierungswarteschlange definieren	119
Eine Prozessdefinition erstellen	119
Prozessdefinition anzeigen	120

Kapitel 6. Verwaltungs-Tasks automa- tisieren 121

PCF-Befehle	121
Attribute in MQSC- und PCF-Befehlen	122
PCF-Escape-Befehle	122
Verwendung von PCF-Befehlen mit Hilfe von MQAI vereinfachen	122
Befehlsserver für Fernverwaltung verwalten	123
Befehlsserver starten	123
Status des Befehlsservers anzeigen	124
Befehlsserver stoppen	124

Kapitel 7. Ferne MQSeries-Objekte verwalten 125

Kanäle, Cluster und fernes Queuing.	125
---	-----

Fernverwaltung mit Hilfe von Clustern.	126
Einen fernen WS-Manager verwalten	127
WS-Manager für Fernverwaltung vorbereiten	127
Kanäle und Übertragungswarteschlangen für	
Fernverwaltung vorbereiten	128
Kanäle und Übertragungswarteschlangen defi-	
nieren	129
Kanäle starten	130
Ferne Ausführung von MQSC-Befehlen.	131
Empfehlungen für fernes Queuing	132
Probleme bei der Verwendung von MQSC-Befehlen	
für die Fernverwaltung	133
Eine lokale Definition für eine ferne Warteschlange	
erstellen	133
Funktionsweise von lokalen Definitionen ferner	
Warteschlangen.	133
Ein alternative Möglichkeit, Nachrichten in eine	
ferne Warteschlange einzureihen	135
Weitere Befehle für ferne Warteschlangen verwen-	
den.	135
Eine Übertragungswarteschlange erstellen.	136
Standardübertragungswarteschlangen	136
Definitionen ferner Warteschlangen als Aliasnamen	
verwenden	136
Aliasname eines WS-Managers	136
Aliasnamen für Warteschlangen für zu beant-	
wortende Nachrichten	137
Datenkonvertierung	137
Ein WS-Manager kann Nachrichten nicht in	
integrierte Formate konvertieren	138
Datei CCSID	138
Konvertierung von Nachrichten in benutzer-	
definierte Formate.	139
CCSID des WS-Managers ändern.	139
Kapitel 8. MQSeries-Objekte schützen 141	
Gründe für den Schutz von MQSeries-Ressourcen	141
Konzept der Benutzer-IDs in der Benutzergruppe	
MQM	141
Weitere Informationen	142
Objektberechtigungsmanager (OAM, Object Autho-	
rity Manager)	142
Funktionsweise des OAM	143
Zugriffsverwaltung über Benutzergruppen	143
Ressourcen über den OAM schützen	144
Berechtigungserteilung über Gruppen	144
OAM inaktivieren	146
OAM-Befehle verwenden	146
Angaben in OAM-Befehlen.	146
Der Befehl 'altmqusr'.	147
Der Befehl 'dspmqusr'	148
Der Befehl 'setmqaut'.	148
Zugriffsberechtigungen	149
Befehl zum Anzeigen von Berechtigungen.	149
Hinweise zur Verwendung des OAM	150
Benutzer-IDs	150
Datenträger des WS-Managers.	150
Warteschlangen.	150
Alternative Benutzerberechtigung	151
Kontextberechtigung	151

Sicherheitsüberlegungen bei fernen	
WS-Managern	152
Sicherheit für Kanalbefehle	152
Erläuterungen zu den Tabellen mit den	
Berechtigungspezifikationen	153
MQI-Berechtigungen	154
Verwaltungsberechtigungen	157
Berechtigungen für MQSC-Befehle in PCF-	
Escape-Befehlen	157
Erläuterung der Berechtigungsdateien	159
Die Principal-Datenbank.	159
Die OAM-Datenbank.	159
Datensätze für Klassenberechtigungen	161
Datensatz mit Berechtigungen für alle Klassen	161

Kapitel 9. MQSeries-Steuerroutine der Warteschlange für nicht zustellbare Nachrichten 163

DLQ-Steuerroutine aufrufen	163
Regeltabelle für die DLQ-Steuerroutine.	164
Steuerdaten	164
Regeln (Muster und Aktionen)	166
Schlüsselwörter für die Mustererkennung	166
Aktionsschlüsselwörter	168
Konventionen für die Regeltabelle	170
Verarbeitung der Regeltabelle	171
Verarbeitung aller DLQ-Nachrichten sicherstel-	
len	172
Regeltabelle für DLQ-Steuerroutine - Beispiel.	173

Kapitel 10. Instrumentierungs- und EMS-Ereignisse 175

MQSeries-Instrumentierungsereignisse	175
Ereignisarten	176
Ereignisaufzeichnung durch Ereigniswarte-	
schlangen	176
Ereigniswarteschlangen mit Auslösern verwen-	
den.	177
Instrumentierungsereignisse aktivieren	177
Ereignisnachrichten	177
EMS-Ereignisse.	178
EMS-Schablonendateien in MQSeries for Com-	
paq NSK	178
MQSeries-EMS-Ereignisschablonen integrieren	179
Umgebungsvariable MQEMSEVENTS definieren	181
Alternativen Ereignissammler (Event Collector)	
verwenden	182
Programme für die Verarbeitung von MQSeries-	
EMS-Ereignissen erstellen	182

Kapitel 11. Transaktionale Unterstützung und transaktionale Nachrichtenübertragung 183

NonStop-TM/MP (Transaktionsmanager) verwen-	
den.	183
Begrenzung innerhalb von Transaktionen	184
Operationen mit Option NOSYNCPPOINT für	
permanente Nachrichten ausführen	185

Operationen mit SYNCPOINT für nicht permanente Nachrichten	185
Konfigurationsvoraussetzungen für TM/MP und MQSeries for Compaq NSK	185
Überwachung	186
Größe für Prüfprotokolle	186
Konfiguration des Ressourcenmanagers	186
Fehlerbehebung	187

Kapitel 12. Wiederherstellung und Neustart 189

Fehlertoleranz und Fehlerbehebung	189
MQSeries-Daten sichern und zurückschreiben	190
MQSeries-Daten sichern	190
MQSeries zurückschreiben	190
Status- und Warteschlangenserver wiederherstellen und erneut starten.	191
Fehlerbehebung mit RDF	192

Kapitel 13. Konfigurationsdateien. . . 193

Konfigurationsdateien - Beschreibung	193
MQSeries-Konfigurationsdatei (MQSINI)	193
Inhalt der MQSeries-Konfigurationsdatei	194
Konfigurationsdatei der WS-Manager (QMINI)	195
Inhalt der Konfigurationsdatei eines WS-Managers	195
Konfigurationsdatei für WS-Manager - Beispiel	198
Konfigurationsdateien editieren	203
Änderungen in Konfigurationsdateien implementieren	203
Empfehlungen für Konfigurationsdateien	203

Kapitel 14. Fehlerbestimmung 205

Erste Überprüfungen	205
Lief MQSeries bis jetzt fehlerfrei?	205
Werden Fehlernachrichten angezeigt?	205
Werden Rückkehrcodes mit Fehlererläuterungen angezeigt?	206
Können Sie den Fehler reproduzieren?	206
Trat der Fehler erst auf, nachdem Änderungen vorgenommen wurden?	206
Lief die Anwendung bisher fehlerfrei?	207
Die Anwendung lief bisher nicht fehlerfrei	207
Betrifft der Fehler bestimmte Teile des Netzes?	208
Tritt der Fehler zu einer bestimmten Tageszeit auf?	208
Tritt der Fehler unregelmäßig auf?	209
Haben Sie Funktionsaktualisierungen installiert?	209
Typische Programmierfehler	209
Fehler in Zusammenhang mit Befehlen	210
Die nächsten Schritte	210
Haben Sie falsche Ausgaben erhalten?	210
Haben Sie von einem PCF-Befehl keine Antwort erhalten?	211
Sind einige Ihrer Warteschlangen fehlerhaft?	212
Betrifft der Fehler nur ferne Warteschlangen?	212
Läuft Ihre Anwendung oder MQSeries for Compaq NSK langsam?	213
Überlegungen zum Anwendungsdesign	214
Auswirkung der Nachrichtenlänge	214

Nach einer bestimmten Nachricht suchen	214
Warteschlangen mit Nachrichten in unterschiedlicher Länge	214
Häufigkeit von Synchronisationspunkten	215
Den Aufruf MQPUT1 verwenden.	215
Falsche Ausgabe	215
Nachrichten erscheinen nicht in der Warteschlange	215
Nachrichten mit unerwarteten oder beschädigten Informationen	216
Probleme mit falschen Ausgaben bei der Verwendung verteilter Warteschlangen	217
Fehlerprotokolle	219
Protokolldateien	219
Fehler in der Startphase	220
Bedienernachrichten	220
Beispiel für ein Fehlerprotokoll	220
EMS-Ereignisse	221
Warteschlangen für nicht zustellbare Nachrichten	222
Konfigurationsdateien und Fehlerbestimmung	222
MQSeries-Trace verwenden.	222
Namen von Trace-Dateien	222
Beispiel für Trace-Daten	223
First Failure Support Technology™ (FFST)	223
FFST-Daten überprüfen	224

Kapitel 15. Skalierbarkeit und Systemleistung 225

Einführung	225
Neue Anwendungen im Hinblick auf Leistung und Skalierbarkeit erstellen.	225
Verwendung gemeinsam benutzter Ressourcen minimieren oder eliminieren	225
Leistungsoptimierung (Durchsatzverbesserung) als iterativer Prozess	226
Permanente Nachrichten	226
Nicht permanente Nachrichten	227
Nicht permanente Nachrichten und Kanäle	227
Warteschlangenserver und Warteschlangendateien	228
Speicherverfahren für permanente Nachrichten	228
Speicherverfahren für nicht permanente Nachrichten	229
Verteilung der Warteschlangenserver auf die CPUs	229
Warteschlangen mit dem Befehl 'altmqfls' anderen Warteschlangenservern zuordnen	229
Cluster-Übertragungswarteschlange: SYSTEM.CLUSTER.TRANSMIT.QUEUE	229
Warteschlangen mit dem Befehl 'altmqfls' verschieben	230
Warteschlangendateien partitionieren	230
Nachrichtenüberlaufdateien	231
Nachrichten zum Durchsuchen im Hauptspeicher ablegen.	231
Weitere Warteschlangenserveroptionen	232
CPU-Zuordnung	233
FASTPATH-Bindungen für Anwendungsprogramme	234
Hintergrund.	235
Systemaufwand bei MQI-Aufrufen reduzieren	235
FASTPATH-Bindungen aktivieren.	236

Einschränkungen bei der Verwendung von FASTPATH-Bindungen	236
---	-----

Kapitel 16. Datenintegrität und Verfügbarkeit 237

Datenintegrität	237
Verfügbarkeit	238
Permanente und nicht permanente Daten	239
Permanente Nachrichten	240
Nicht permanente Nachrichten	241
Datenbankkonsistenz	242
Interne Datenbankkonsistenz	242
Externe Datenbankkonsistenz	243
OpenTMF	244
NonStop Tuxedo	244
Gleichzeitige Ausführung von Transaktionen in Anwendungen	245
Kritische MQSeries-Datenbankdateien	245
Kritische Prozesse	246
Cluster	253
Konfigurationsoptionen für die Verfügbarkeit	253
Konfigurationsoptionen für die Datenintegrität	254

Teil 2. Referenz 255

Kapitel 17. MQSeries-Steuerbefehle 257

Steuerbefehle - Übersicht	257
Namenskonventionen	258
Syntaxdiagramme lesen	258
altmqfls (Attribute der Warteschlangendateien ändern)	260
altmqusr (MQSeries-Benutzerinformationen ändern)	265
cleanrdf (RDF-Verwaltungsaufgaben ausführen)	267
cnvclchl (Client-Kanaldefinitionen konvertieren)	269
crtmqcvx (Datenkonvertierung)	271
crtmqm (WS-Manager erstellen)	273
dltmqm (WS-Manager löschen)	277
dspmqaout (Berechtigung anzeigen)	279
dspmqcsv (Befehlsserver anzeigen)	283
dspmqlfs (MQSeries-Dateiattribute anzeigen)	284
dspmqrtrc (Formatierte MQSeries-Trace-Ausgabe anzeigen)	287
dspmqusr (MQSeries-Benutzerinformationen anzeigen)	288
endmqcsv (Befehlsserver beenden)	290
endmqm (WS-Manager beenden)	292
endmqtrc (MQSeries-Trace beenden)	294
instmqm (MQSeries for Compaq NSK installieren)	295
runmqchi (Kanalinitiator ausführen)	296
runmqchl (Kanal ausführen)	297
runmqdlq (Steueroutine der DLQ ausführen)	298
runmqlsr (Empfangsprogramm ausführen)	300
runmqsc (MQSeries-Befehle ausführen)	302
runmqtrm (Auslösemonitor starten)	305
setmqaut (Berechtigung setzen/zurücksetzen)	306
strmqcsv (Befehlsserver starten)	312
strmqm (WS-Manager starten)	313
strmqtrc (MQSeries-Trace starten)	315

upgmqm (Upgrade eines WS-Managers der Version 2.2.0.1 durchführen)	318
--	-----

Teil 3. Anhänge und Schlussteil 321

Anhang A. MQSeries for Compaq NSK - Übersicht 323

Programm und Teilenummer	323
Hardwarevoraussetzungen	323
Softwarevoraussetzungen	323
Sicherheit	324
Verwaltungsfunktionen	324
Kompatibilität	324
Unterstützte Compiler	324
Lizenzverwaltung	325
Sprachenauswahl	325
CCSID-Auswahl	325

Anhang B. Systemstandardwerte 327

Anhang C. TAACL-Umgebungsvariablen für MQSeries for Compaq NSK setzen. 329

Optimierungsparameter für Warteschlangenserver	331
--	-----

Anhang D. Befehlssätze - Übersicht 333

Anhang E. WS-Manager manuell stoppen und entfernen 337

WS-Manager manuell stoppen	337
WS-Manager manuell entfernen	337

Anhang F. MQSeries und Compaq NonStop Server for Java 339

Transaktionale Überlegungen	340
---------------------------------------	-----

Anhang G. In MQSeries for Compaq NSK unterstützte MQSC-Befehle 341

MQSC-Attribute	344
Kanalstatusinformationen (DISPLAY CHSTATUS)	344
MAXUMSGS und MAXHANDS	344
HARDENBO und NOHARDENBO	344
CONNAME	344
USERDATA für ausgelöste Programme	345
Namen von Benutzer-Exits als Objektattribute verwenden	345

Anhang H. Anwendungsprogrammierung - Referenz 347

Strukturdatentypen	347
MQCNO (Connect Options)	348
MQGMO (Get Message Options)	349
MQMD (Message Descriptor)	350
MQPMO (Put Message Options)	350
MQI-Aufrufe	350
MQCLOSE (Close Object)	352
MQDISC (Disconnect Queue Manager)	352

MQINQ (Inquire About Object Attributes)	352
MQOPEN (Open Object)	352
MQSET (Set Object Attributes)	353
Attribute von MQSeries-Objekten.	353
Attribute für alle Warteschlangen.	353
Attribute der lokalen Warteschlangen und der Modellwarteschlangen	353
Attribute der WS-Manager	354
Datenkonvertierung	354

Anhang I. Anwendungen erstellen und ausführen 355

Anwendungen schreiben	355
Warteintervall in MQGET und Kanalattribute	
DISCONT und HBINT verwenden	355
Arbeitseinheiten (UOW) verwalten	356
Allgemeine Überlegungen bei der Erstellung von Anwendungen	358
XA-Schnittstelle	358
MQGMO_BROWSE_* mit MQGMO_LOCK	358
Ausgelöste Anwendungen	358
Unterstützte Programmiersprachen und Umgebun- gen.	359
Überlegungen bei der Erstellung von Anwen- dungen mit Threads	360
Anwendungen kompilieren und binden	361
FASTPATH- und STANDARD-Bindungen	361
Anwendungen ausführen	362

Anhang J. Die MQSeries-Verwaltungsschnittstelle 363

Anhang K. MQSeries for Compaq NSK - Beispielprogramme 365

C-Beispielprogramm erstellen	366
Nicht native Anwendungen (unter Verwendung der nicht nativen statischen Bibliothek MQM- LIB) verwenden	366
Native Anwendungen (unter Verwendung der nativen statischen Bibliothek MQMLIBN)	367
Native Anwendungen (unter Verwendung der SRL MQSRLLIB)	367
C++-Beispielprogramme erstellen.	368
Native Anwendungen (unter Verwendung der nativen statischen Bibliothek MQMLIBN)	368
Native Anwendungen (unter Verwendung der SRL MQSRLLIB)	369
COBOL-Beispielprogramme erstellen	369
Nicht native Anwendungen (unter Verwendung der nicht nativen statischen Bibliothek MQM- LIB)	369
Native Anwendungen (unter Verwendung der nativen statischen Bibliothek MQMLIBN)	370
Native Anwendungen (unter Verwendung der SRL MQSRLLIB)	371
TAL-Beispielprogramm erstellen	372
Nicht native Anwendungen (unter Verwendung der nicht nativen statischen Bibliothek MQM- LIB)	372

Beispielprogramme unter OSS erstellen (nur nati- ver Modus)	372
--	-----

Anhang L. Benutzer-Exits 373

Unterstützte Benutzer-Exits.	373
Format für Exit-Namen	374
MQ_LOAD_ENTRY_POINT_EXIT (Benutzer-Exits laden)	374
MQLXP (Parameterstruktur von MQ_LOA- D_ENTRY_POINT_EXIT)	375
Benutzer-Exits installieren	378
Exit in der privaten SRL von MQSeries installie- ren	378
Exit in der nativen statischen MQSeries- Bibliothek installieren	379
Exit in der nicht nativen statischen MQSeries- Bibliothek installieren	380

Anhang M. Kommunikation einrichten 383

SNA-Kanäle	383
LU6.2-Responder-Prozesse	384
TCP/IP-Kanäle	389
Kommunikationskonfiguration - Beispiele	390
SNAX-Kommunikationskonfiguration - Beispiel	390
ICE-Kommunikationskonfiguration - Beispiel	396
TCP/IP-Kommunikationskonfiguration - Bei- spiel	399

Anhang N. MQSeries-Clients 403

Client-Unterstützung	403
Sicherheitsüberlegungen.	403

Anhang O. Programmierbare System- verwaltung 405

Instrumentierungsereignisse	405
In MQSeries for Compaq NSK unterstützte Ereignisarten	405
Format der Ereignisnachrichten	406
PCF-Befehle (Programmable Command Format)	407
PCF-Nachrichtendeskriptor.	408
PCF-Header (MQCFH)	408
PCF-Zeichenfolgeparameter (MQCFST).	409
PCF-Parameter für Integerliste (MQCFIL)	409
PCF-Integer (MQCFIN)	409
PCF-Zeichenfolgeliste (MQCFSL).	410
In MQSeries for Compaq NSK unterstützte PCF- Befehle	410
Antworten auf PCF-Befehle.	411
Installierbare Services	412
Schnittstelle für Berechtigungsservice	412
Schnittstelle für Namensservice.	412

Anhang P. In MQSeries for Compaq NSK verwendete EMS- Ereignisschablone 413

Anhang Q. Bemerkungen 417

Marken	418
------------------	-----

Literaturverzeichnis	419
Plattformübergreifende MQSeries-Veröffentlichungen	419
Plattformspezifische MQSeries-Veröffentlichungen	419
Softcopy-Bücher	420
HTML-Format	420
Bücher im PDF-Format (Portable Document Format)	420
BookManager-Format	421
Bücher im PostScript-Format	421
Format der Online-Hilfe von Windows	421

Im Internet verfügbare Informationen zu MQSeries	421
Referenzliteratur	421

Glossar der Begriffe und Abkürzungen.	423
--	------------

Index	435
------------------------	------------

Kommentare an IBM senden	445
---	------------

Abbildungsverzeichnis

1. MQSeries for Compaq NSK - Prozesse	23	27. Die Anzeige 'Monitor Channels'	90
2. Komponenten von MQSeries for Compaq NSK V5.1	28	28. Die Anzeige 'Channel Status'	91
3. PATHWAY-Beispielkonfiguration	38	29. Die Anzeige 'Start/Stop Channel'	91
4. MQSeries for Compaq NSK - MQM- Hauptmenü	74	30. Die Anzeige 'Reset Channel'	92
5. Die Anzeige 'Queue Manager Menu'	75	31. Die Anzeige 'Resolve Channel'	93
6. Die Anzeige 'Queue Manager Menu', Teil 2	75	32. Die Anzeige 'Copy Channel'	94
7. Die Anzeige 'QUEUE MANAGER TRACE MENU'	76	33. Warteschlangen, Nachrichten und Anwendun- gen	95
8. Die Anzeige 'Search Criteria' (Warteschlangen)	77	34. Ausgabe für QMGR ALL - Beispiel.	99
9. Die Anzeige 'Queue Menu'	78	35. Auszug aus der MQSC-Befehlsdatei mymqsc- cin	102
10. Die Anzeige 'Create Queue'	79	36. Auszug aus der MQSC-Berichtsdatei mymqsc- cou	103
11. Die Anzeige 'Create Local Queue'	80	37. Fernverwaltung.	128
12. Die Anzeige 'Create Local Queue', Teil 2	80	38. Kanäle und Warteschlangen für Fern- verwaltung konfigurieren	129
13. Die Anzeige 'Create Remote Queue'	81	39. Verwendung von Gruppen mit SAFEGUARD zur Erteilung von Berechtigungen.	145
14. Die Anzeige 'Copy Queue'	81	40. Steuerdaten - Beispiel	166
15. Die Anzeige 'Display/Modify Local Queue'	82	41. Beispielregel	166
16. Die Anzeige 'Monitor Local Queues'	83	42. MQSeries-Konfigurationsdatei MQSINI - Bei- spiel	194
17. Die Anzeige 'Search Criteria' (für Kanäle)	84	43. Konfigurationsdatei für einen WS-Manager (QMINI) - Beispiel	198
18. Die Anzeige 'Channel Menu'	84	44. Trace - Beispiel	223
19. Die Anzeige 'Display/Modify Sender Channel' (Teil 1)	85	45. First Failure Symptom Report - Beispiel	224
20. Die Anzeige 'Display/Modify Sender Channel' (Teil 2)	86	46. MQLOADEXIT - Beispiel	377
21. Die Anzeige 'Create Channel'	86	47. MQSeries-SNAX-Konfigurationsdatei - Bei- spiel	386
22. Die Anzeige 'Create Sender Channel'	87	48. MQSeries-SNA-Konfigurationsdatei für ICE - Beispiel	388
23. Die Anzeige 'Create Receiver Channel'	88		
24. Die Anzeige 'Create Server Connection Chan- nel'	88		
25. Die Anzeige 'Create Cluster Sender Channel'	89		
26. Die Anzeige 'Create Cluster Receiver Channel'	89		

Tabellen

1. Ausführbare MQSeries-Module	26	17. Befehle für die Verwaltung des Befehlsservers	333
2. Liste möglicher ISO-CCSIDs	54	18. Befehle für die Warteschlangenverwaltung	334
3. Erforderliche Sicherheitsberechtigungen für MQI-Aufrufe.	154	19. Befehle für die Prozessverwaltung	334
4. MQSC-Befehle und erforderliche Sicherheitsberechtigungen	157	20. Befehle für die Kanalverwaltung	335
5. PCF-Befehle und erforderliche Sicherheitsberechtigungen	158	21. Weitere Steuerbefehle	336
6. Warteschlangenüberlaufdatei und Nachrichtenüberlaufdatei im Vergleich	240	22. Java-Schnittstelle	340
7. Kritische protokollierte Datenbankdateien	245	23. In MQSeries for Compaq NSK unterstützte MQSC-Befehle	341
8. Schutz kritischer Prozesse	246	24. Unterstützte Programmiersprachen und Umgebungen - Überblick	359
9. Sicherheitsberechtigungen mit dem Befehl dspmqaut.	280	25. Richtige Version der MQI-Bibliothek	361
10. Berechtigungen für verschiedene Objektarten erteilen	309	26. In MQSeries for Compaq NSK unterstützte Benutzer-Exits	373
11. Dateien, die nach einem Upgrade gelöscht werden können - Beispiele	318	27. In MQSeries for Compaq NSK unterstützte Ereignisarten.	405
12. System- und Standardobjekte für Warteschlangen	327	28. MQMD-Struktur einer Ereignisnachricht	406
13. System- und Standardobjekte für Kanäle	328	29. Ereignis-Header-Struktur (MQCFH)	407
14. System- und Standardobjekte für Namenslisten	328	30. PCF-Nachrichtendeskriptor	408
15. System- und Standardobjekte für Prozesse	328	31. PCF-Header	408
16. Befehle für die Verwaltung des WS-Managers	333	32. PCF-Zeichenfolgeparameter.	409
		33. PCF-Integerliste.	409
		34. PCF-Integer	409
		35. PCF-Zeichenfolgeliste	410
		36. In MQSeries for Compaq NSK unterstützte PCF-Befehle	410

Zu diesem Handbuch

MQSeries for Compaq NonStop Kernel Version 5 Release 1, das in diesem Handbuch als MQSeries for Compaq NSK oder, sofern es der Kontext zulässt, einfach als MQSeries bezeichnet wird, ist Teil der MQSeries-Produktfamilie. Diese Produkte stellen Services für die Anwendungsprogrammierung zur Verfügung, mit deren Hilfe Anwendungsprogramme über *Nachrichtenwarteschlangen* miteinander kommunizieren können. Diese Art der Kommunikation wird als *Commercial Messaging* (kommerzielle Nachrichtenübertragung) bezeichnet. Die beteiligten Anwendungen können sich auf verschiedenen Knoten auf einer Vielzahl von Maschinen- und Betriebssystemtypen befinden. Sie verwenden eine gemeinsame Anwendungsprogrammierschnittstelle, die so genannte Schnittstelle für Nachrichtenwarteschlangen (MQI, Message Queue Interface), so dass Programme, die auf der einen Plattform entwickelt wurden, ohne weiteres auf eine andere Plattform übertragen werden können.

Im vorliegenden Handbuch werden die Aspekte der Systemverwaltung von MQSeries for Compaq NSK Version 5 Release 1 und die Services, die dieses Produkt für die Unterstützung der kommerziellen Nachrichtenübertragung in einer Compaq NSK-Umgebung zur Verfügung stellt. Dazu gehört das Verwalten der Warteschlangen, die Anwendungen zum Empfangen ihrer Nachrichten verwenden, sowie das Sicherstellen des Zugriffs von Anwendungen auf die von ihnen benötigten Warteschlangen.

Zielgruppe

Dieses Buch richtet sich in erste Linie an Systemadministratoren und Systemprogrammierer, die die Konfigurations- und Verwaltungs-Tasks für MQSeries ausführen. Es ist aber auch für Anwendungsprogrammierer hilfreich, die Kenntnisse über Verwaltungs-Tasks von MQSeries benötigen.

Voraussetzungen zur Benutzung dieses Handbuchs

Für das Verständnis dieses Buches sind gute Kenntnisse über das Betriebssystem Compaq NSK und seine Dienstprogramme erforderlich. Es wird nicht vorausgesetzt, dass Sie bereits mit Message-Queuing-Produkten gearbeitet haben, Kenntnisse über die grundlegenden Konzepte des Message-Queuing sollten jedoch vorhanden sein.

Benutzung des Handbuchs

Der Hauptteil dieses Buchs enthält:

- eine Einführung in MQSeries,
- eine Beschreibung der täglich auf einem System mit MQSeries for Compaq Non-Stop Kernel anfallenden Verwaltungsaufgaben (z. B. die Verwaltung lokaler und ferner MQSeries-Objekte, Sicherheitsfunktionen, Transaktionsunterstützung und Fehlerbestimmung).

Informationen zu MQSeries im Internet

MQSeries-URL

Die Internet-Adresse der Homepage der MQSeries-Produktfamilie lautet:

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

Neuerungen in MQSeries for Compaq NSK V5.1

In dieser Ausgabe des Handbuchs *MQSeries for Compaq NSK V5.1 Systemverwaltung* werden die folgenden neuen Funktionen beschrieben:

Leistungsverbesserungen

Neuer Warteschlangenserverprozess

Im WS-Manager steht jetzt ein neuer WS-Serverprozess zur Verfügung, der Nachrichtenspeicherplatz für eine oder mehrere lokale Warteschlangen zur Verfügung stellt und die Verwaltung aller GET- und PUT-Zugriffe auf diese Warteschlangen übernimmt. Er ermöglicht die Implementierung der nicht permanenter Nachrichtenübertragung und unterstützt die neuen Messaging-Funktionen in Version 5.1. Weitere Informationen finden Sie unter „Kapitel 2. MQSeries for Compaq NSK V5.1 - Architektur“ auf Seite 23.

Änderungen an der Statusserveroperation

Der Statusserver ersetzt die Plattendatei, die zuvor zum Speichern des Kanalstatus verwendet wurde. Der Statusserver unterstützt die Statusinformationen von Objekten, bei denen es sich nicht um lokale Warteschlangen handelt, und ermöglicht einen effizienten Zugriff auf Kanalstatusinformationen. Weitere Informationen finden Sie unter „Kapitel 2. MQSeries for Compaq NSK V5.1 - Architektur“ auf Seite 23.

Nicht permanente Nachrichten

Sie können jetzt die Leistungsverbesserungen nutzen, die sich durch nicht permanente Nachrichten ergeben.

FASTPATH-Bindungen für sichere Anwendungen

Falls Sie über geeignete Anwendungen verfügen, können Sie eine Verbindung zu einem WS-Manager über FASTPATH-Bindungen herstellen, wodurch sich deutliche Leistungsverbesserungen ergeben. FASTPATH-Anwendungen sind bestimmten Einschränkungen unterworfen und müssen *well behaved* sein, d. h. bestimmten Standards entsprechen, da diese Bindungsart weniger Schutz für kritische interne Daten des WS-Managers bietet. Weitere Informationen finden Sie unter „Anhang I. Anwendungen erstellen und ausführen“ auf Seite 355.

Verbesserungen am Plattenspeicher für permanente Nachrichten

Als Teil der neuen WS-Serverarchitektur wurden Änderungen an der Nachrichtenspeicherung auf Platte vorgenommen, um eine verbesserte Leistung für alle Nachrichtenlängen zu ermöglichen. Für Warteschlangendateien sind keine Alternativschlüsseldateien erforderlich, und für besonders große Nachrichten wurde eine neue Plattenspeicherart eingeführt, die eine effiziente Speicherung von Nachrichten mit einer Größe von bis zu 100 MB ermöglicht.

Aufrüstungen an MQSeries-Funktionen

MQSeries-WS-Manager-Cluster

MQSeries-Warteschlangenmanager können zu einem WS-Manager-Cluster verbunden werden. In einem Cluster können WS-Manager die von ihnen verwalteten Warteschlangen allen anderen WS-Managern zur Verfügung stellen. Jeder WS-Manager kann eine Nachricht an einen beliebigen WS-Manager innerhalb desselben Clusters senden, ohne dass explizite Kanaldefinitionen, Definitionen ferner Warteschlangen oder Übertragungswarteschlangen für die einzelnen Zieladressen erforderlich sind. Die wichtigsten Vorteile von MQSeries-Clustern sind:

- weniger Systemverwaltungs-Tasks
- erhöhte Verfügbarkeit
- gleichmäßige Auslastung

Eine ausführliche Beschreibung dieser Funktion finden Sie im Handbuch *Cluster-Unterstützung in MQSeries*.

MQSeries-Verwaltungsschnittstelle (MQSeries Administration Interface, MQAI)

MQSeries for Compaq NSK V5.1 unterstützt jetzt die MQSeries-Verwaltungsschnittstelle (MQSeries Administration Interface, MQAI), eine Programmierschnittstelle, die die Verwendung von PCF-Nachrichten zum Konfigurieren von MQSeries vereinfacht. Weitere Informationen zu MQAI, einschließlich einer ausführlichen Beschreibung der Befehle, finden Sie im Handbuch *MQSeries Administration Interface Programming Guide and Reference*.

Erweiterte MQI-Unterstützung

MQSeries for Compaq NSK V5.1 unterstützt jetzt erweiterte Nachrichtenfunktionen, die in den Releases von MQSeries Version 5.0 und 5.1 auf anderen Plattformen zur Verfügung stehen. Dazu gehört die Verarbeitung von Verteilerlisten, Verweinsnachrichten, Nachrichtensegmentierung und eine ganze Reihe weiterer Optionen. Weitere Informationen hierzu finden Sie in den Handbüchern *MQSeries Application Programming Guide* und *MQSeries Application Programming Reference*.

Längere Nachrichten und Warteschlangen

Für Nachrichten ist jetzt eine Größe von bis zu 100 MB, für Nachrichtenwarteschlangen eine Größe von bis zu 4 GB möglich.

Automatische Erstellung von Standardobjekten

Bei der Erstellung eines WS-Managers mit dem Befehl `crtmqm` werden automatisch Systemstandardobjekte erstellt.

Gesteuerter, synchroner Abschluss von WS-Managern

Der Befehl `endmqm` wurde um eine neue Option erweitert, die einen gesteuerten, synchronen Abschluss eines WS-Managers ermöglicht.

Java-Unterstützung

MQSeries for Compaq NSK V5.1 arbeitet jetzt mit Java-Compilern, d. h., es können Java-Anwendungen verwendet werden. Weitere Informationen finden Sie unter „Anhang F. MQSeries und Compaq NonStop Server for Java“ auf Seite 339.

Unterstützung von OSS-Anwendungen

MQSeries for Compaq NSK V5.1 unterstützt jetzt NSK OSS-Anwendungen in C, C++ und Java.

Web-Verwaltung

Mit MQSeries for Compaq NSK V5.1 können Sie in einem HTML-Browser (z. B. Netscape Navigator oder Microsoft Internet Explorer) unter Microsoft® Windows NT®:

- sich als MQSeries-Administrator anmelden
- einen WS-Manager auswählen und MQSC-Befehle dafür ausgeben
- MQSC-Scripts erstellen, editieren und löschen

Bessere Einbindung und Konfiguration von Benutzer-Exits

Die Einbindung und Konfiguration von Benutzer-Exit-Code in MQSeries wurde erheblich verbessert, um eine Schnittstelle zur Verfügung zu stellen, die mehr oder weniger dem gängigen Standard entspricht, und eine für alle Exits geltende Vorgehensweise zu ermöglichen. Ausführliche Informationen finden Sie unter „Anhang L. Benutzer-Exits“ auf Seite 373.

Kommunikation

TCP/IP

MQSeries for Compaq NSK V5.1 ermöglicht WS-Managern jetzt die Verwendung mehrerer Guardian-TCP/IP-Serverprozesse. Dies erlaubt eine bessere Konfiguration für eine gleichmäßige Auslastung der gesamten Netzhardware und ermöglicht für einen WS-Manager und für Anwendungen bei Ausfällen von Netzverbindungen das Umschalten auf einen anderen Kanal (Redundanz). Weitere Informationen finden Sie unter „Anhang M. Kommunikation einrichten“ auf Seite 383.

SNA

MQSeries for Compaq NSK V5.1 ermöglicht jetzt eine bessere Verwaltung und Steuerung beim Starten von Kanälen für das SNA-Übertragungsprotokoll. Das neue Verfahren verwendet einen Empfangsprogrammprozess, der unter PATHWAY läuft, und wird für SNAX- und Insession ICE-Produkte verwendet. Das Kanalattribut AUTOSTART, bei dem es sich nicht um ein MQSeries-Standardattribut handelt, wird nicht mehr verwendet. Weitere Informationen finden Sie unter „Anhang M. Kommunikation einrichten“ auf Seite 383.

Kanäle

Kanäle unterstützen jetzt *Überwachungssignale* und die Möglichkeit, nicht permanente Nachrichten außerhalb einer Arbeitseinheit zu übertragen; hierdurch wird eine bessere Leistung erzielt.

MQSeries for Compaq NSK V5.1 unterstützt jetzt die optionale automatische Kanaldefinition für Kanäle, die von anderen WS-Managern oder Clients (also fern) gestartet wurden.

Compaq NSK-spezifische Verbesserungen der Benutzerfreundlichkeit

Compaq NSK Fix Command in `runmqsc`

`runmqsc` umfasst jetzt die Funktion Compaq NSK Fix Command, mit der Sie ein Protokoll der eingegebenen MQSC-Befehle abrufen und MQSC-Befehle bearbeiten können. Weitere Informationen finden Sie in „Compaq NSK Fix Command verwenden“ auf Seite 32.

Erweiterungen für das Dienstprogramm `altmqfls`

Am Dienstprogramm `altmqfls` wurden grundlegende Änderungen vorgenommen, so dass eine gezielte Verwaltung der Speicheroptionen für Nachrichten möglich ist. Weitere Informationen hierzu finden Sie unter „Kapitel 15. Skalierbarkeit und Systemleistung“ auf Seite 225 und „`dspmqfls` (MQSeries-Dateiattribute anzeigen)“ auf Seite 284.

MQMC-Anzeigen

Die MQMC-Verwaltungsanzeigen, die Teil der PATHWAY-Umgebung für WS-Manager sind, wurden aufgerüstet und unterstützen die MQSeries-Funktionserweiterungen dieses Releases.

Teil 1. Anleitung

Kapitel 1. Einführung	7
MQSeries und Message-Queuing	7
Zeitunabhängige Anwendungen	7
Nachrichtengesteuerte Verarbeitung	8
Nachrichten und Warteschlangen	8
Nachrichten	8
Nachrichtenlänge	9
Warteschlangen	9
Senden und Empfangen von Nachrichten durch Anwendungen.	9
Vordefinierte und dynamische Warteschlangen Nachrichten aus Warteschlangen abrufen	10
Objekte	10
Objektnamen	11
Objekte verwalten	11
Objektattribute	11
MQSeries-WS-Manager	12
MQI-Aufrufe	12
MQSeries-Warteschlangen	12
Warteschlangenobjekte verwenden.	13
Lokale Warteschlangen in MQSeries	14
Prozessdefinitionen.	16
Kanäle	16
Cluster	17
Namenslisten.	17
Systemstandardobjekte	17
Verwaltung	18
Lokale und ferne Verwaltung	18
Clients und Server	19
MQI-Anwendungen in einer Client/Server- Umgebung	19
WS-Managerfunktionen erweitern	19
Benutzer-Exits	19
Installierbare Services	20
Sicherheit	21
Objektberechtigungsmanager (OAM, Object Authority Manager	21
Transaktionsunterstützung	21
Durchsatzverbesserung, Zuverlässigkeit, Skalierbar- keit und Größenanpassung	22
Kapitel 2. MQSeries for Compaq NSK V5.1 - Architektur	23
WS-Managerprozesse - Übersicht	23
Produktbindungen	25
Ausführbare Module	26
Vergleich der Architekturen von Version 5.1 und Version 2.2.0.1	26
WS-Manager — Funktionsübersicht	27
WS-Managerprozessmodell	28
MQSeries-Dateien und -Teildatenträger	29
Kapitel 3. MQSeries-Befehlsätze verwenden	31
Verwaltungs-Tasks über Steuerbefehle ausführen	31
Steuerbefehle verwenden	31
Verwaltungs-Tasks über MQSC-Befehle ausführen	32
MQSC-Befehle ausführen	32
Compaq NSK Fix Command verwenden	32
Verwaltungs-Tasks über PCF-Befehle ausführen	33
Attribute in MQSC- und PCF-Befehlen	33
PCF-Escape-Befehle.	33
TS/MP-Verwaltung (PATHWAY-Verwaltung)	34
TCP/IP-Empfangsprogramme konfigurieren und steuern	34
Steuerung des Befehlsservers	34
Kanalinitiatoren konfigurieren und steuern	35
Standardinitialisierungswarteschlange für Kanalinitiator ändern	35
Auslösemonitore konfigurieren und steuern	35
Prozesse über mehrere CPUs verteilen	36
Neue Statusserverprozesse hinzufügen	37
Neue Warteschlangenserverprozesse hinzufügen	37
Aktualisierungsintervall für MQM- Überwachungsanzeigen festlegen	38
PATHWAY-Konfiguration für einen WS-Manager	38
Parameter von PATHWAY-Serverklassen ändern	45
Benutzerdefinierte Serverklassen einer PATHWAY-Konfiguration in MQSeries hinzufü- gen	45
Kapitel 4. WS-Manager verwalten	47
Einführung	47
Vorgehensweise beim Erstellen von WS-Managern	47
Sicherungskopien von Konfigurationsdateien nach Erstellung eines WS-Managers erstellen	50
WS-Managermerkmale ändern	50
Ausgangsdatenträger des WS-Managers	50
Anzahl der Ausführungs-Controller pro WS-Manager	50
Gleichmäßige Systemauslastung	51
Ausgangsterminal des WS-Managers	51
PATHMON-Prozessname für den WS-Manager	52
CCSID des WS-Managers.	53
EBCDIC-Datenkonvertierung steuern.	53
EMS-Ereignissammler für WS-Manager	54
Agenten-Pool der Ausführungs-Controller im WS-Manager	54
Maximale Anzahl inaktiver Agenten und Prozess- wiederverwendung.	55
Priorität der WS-Managerprozesse.	55
Maximale Anzahl an Kanälen für den WS-Manager	56
Maximale Anzahl an aktiven Kanälen für den WS-Manager	56
Guardian-Segment-IDs für MQSeries	56
TCP/IP-Standard-Port.	57
Anzahl an TCP/IP-Ports, die vom WS-Manager überwacht werden	57
Der vom WS-Manager verwendete TCP/IP- Prozess	57

Serverklasse MQS-TCPLISnn für die Verwendung eines anderen TCP/IP-Prozesses und Ports rekonfigurieren	58	Unterstützung von Anwendungsprogrammen, die MQI verwenden	95
Zuordnung des Auslagerungsspeichers	59	Lokale Verwaltungs-Tasks mit Hilfe von MQSC-Befehlen ausführen	96
Name des Standardstatusservers	59	Vor dem Start	97
Zusätzliche Statusserver hinzufügen und entfernen	60	MQSeries-Objektnamen	97
Name des Standardwarteschlangenservers	62	Groß-/Kleinschreibung bei MQSC-Befehlen.	97
Zusätzliche Warteschlangenserver hinzufügen und entfernen	63	Die MQSC-Funktion interaktiv verwenden	97
Datenträgerstruktur	64	Rückmeldung auf MQSC-Befehle	98
Teildatenträger mit den FFST-Dateien des WS-Managers.	65	Interaktive Eingabe von MQSC-Befehlen beenden	99
Teildatenträger mit den Datendateien des WS-Managers.	65	WS-Managerattribute anzeigen	99
Datenträger mit dem Fehlerprotokoll des WS-Managers.	66	Einen WS-Manager verwenden, der nicht der Standard-WS-Manager ist	100
Teildatenträger für Nachrichtenwarteschlangen des WS-Managers	66	WS-Managerattribute ändern	100
Teildatenträger für Kanalsynchronisation des WS-Managers.	67	MQSC-Befehle aus Textdateien ausführen	100
Umwandlung von Objektnamen	67	MQSC-Befehlsdateien verwenden	101
Mit WS-Managern arbeiten	67	MQSC-Berichtsdateien verwenden	102
Standard-WS-Manager erstellen	68	Im Lieferumfang enthaltene MQSC-Befehlsdatei ausführen	103
MQSeries-Principals erstellen	69	Befehle mit Hilfe von runmqsc prüfen	103
'cleanrdf' für einen RDF-aktivierten WS-Manager ausführen	70	MQSC-Probleme lösen	104
WS-Manager starten	70	Mit lokalen Warteschlangen arbeiten	105
Standard- und Systemobjekte wiederherstellen. Objektdateien.	70	Eine lokale Warteschlange definieren	105
WS-Manager stoppen	71	Größe der physischen Datei für Warteschlangen ändern	106
Maßnahmen bei eventuellen Problemen	71	Eine Warteschlange für nicht zustellbare Nachrichten definieren	106
Sofortiger und erzwungener Abschluss eines WS-Managers.	71	Standardobjektattribute anzeigen	107
WS-Manager erneut starten	72	Eine lokale Warteschlangendefinition kopieren	108
Vorhandenen WS-Manager als Standard-WS-Manager definieren.	72	Attribute einer lokalen Warteschlange ändern	108
WS-Manager löschen	73	Datenträger einer lokalen Warteschlange ändern	109
MQM (Message Queue Management) verwenden	73	Optionen einer lokalen Warteschlange ändern	109
'Queue Manager Menu' (WS-Managermenü)	74	Objekte Statusservern und Warteschlangenservern erneut zuordnen.	110
WS-Managerattribute ändern	75	Inhalt einer lokalen Warteschlange löschen	110
Trace für MQSeries-Objekte durchführen	76	Eine lokale Warteschlange löschen	111
Anzeige 'Queues Menu'	77	Warteschlangen durchsuchen	111
Warteschlange erstellen	79	Mit Aliaswarteschlangen arbeiten.	114
Warteschlange kopieren	81	Eine Aliaswarteschlange definieren	114
Warteschlange ändern	82	Andere Befehle für Aliaswarteschlangen verwenden	115
Warteschlange löschen.	82	Mit Modellwarteschlangen arbeiten	116
Warteschlange überwachen	83	Eine Modellwarteschlange definieren	116
Anzeige 'Channels Menu'	83	Andere Befehle für Modellwarteschlangen verwenden	117
Kanal ändern.	85	Objekte für Auslösefunktion verwalten	117
Kanaldefinition erstellen	86	Anwendungswarteschlange für die Auslösefunktion definieren	117
Kanal überwachen	90	Eine Initialisierungswarteschlange definieren	119
Kanal löschen	90	Eine Prozessdefinition erstellen	119
Kanalstatus anzeigen	90	Prozessdefinition anzeigen	120
Kanäle starten und stoppen	91	Kapitel 6. Verwaltungs-Tasks automatisieren	121
Nachrichtenfolgennummer zurücksetzen	92	PCF-Befehle	121
Kanal auflösen	93	Attribute in MQSC- und PCF-Befehlen	122
Kanal kopieren	94	PCF-Escape-Befehle	122
Kapitel 5. Lokale MQSeries-Objekte verwalten	95	Verwendung von PCF-Befehlen mit Hilfe von MQAI vereinfachen	122
		Befehlsserver für Fernverwaltung verwalten	123
		Befehlsserver starten	123
		Status des Befehlsservers anzeigen	124

Befehlsserver stoppen	124	Berechtigungslisten	147
Kapitel 7. Ferne MQSeries-Objekte verwalten	125	Der Befehl 'altmqusr'.	147
Kanäle, Cluster und fernes Queuing	125	Der Befehl 'dspmqusr'	148
Fernverwaltung mit Hilfe von Clustern.	126	Der Befehl 'setmqaut'.	148
Einen fernen WS-Manager verwalten	127	Berechtigungsbefehle und installierbare Ser-	
WS-Manager für Fernverwaltung vorbereiten	127	vices	149
Kanäle und Übertragungswarteschlangen für		Zugriffsberechtigungen	149
Fernverwaltung vorbereiten	128	Befehl zum Anzeigen von Berechtigungen.	149
Kanäle und Übertragungswarteschlangen defi-		Hinweise zur Verwendung des OAM	150
nieren	129	Benutzer-IDs	150
Kanäle starten	130	Datenträger des WS-Managers.	150
Automatische Definition von Kanälen	131	Warteschlangen.	150
Ferne Ausführung von MQSC-Befehlen.	131	Alternative Benutzerberechtigung	151
Mit WS-Managern unter MVS/ESA arbeiten	132	Kontextberechtigung	151
Empfehlungen für fernes Queuing	132	Sicherheitsüberlegungen bei fernen	
Probleme bei der Verwendung von MQSC-Befehlen		WS-Managern	152
für die Fernverwaltung	133	Sicherheit für Kanalbefehle	152
Eine lokale Definition für eine ferne Warteschlange		PCF-Befehle	152
erstellen	133	MQSC-Kanalbefehle	153
Funktionsweise von lokalen Definitionen ferner		Erläuterungen zu den Tabellen mit den	
Warteschlangen.	133	Berechtigungspezifikationen	153
Beispiel	133	MQI-Berechtigungen	154
Funktionsweise.	134	Verwaltungsberechtigungen	157
Ein alternative Möglichkeit, Nachrichten in eine		Berechtigungen für MQSC-Befehle in PCF-	
ferne Warteschlange einzureihen	135	Escape-Befehlen	157
Weitere Befehle für ferne Warteschlangen verwen-		Berechtigungen für PCF-Befehle	158
den.	135	Erläuterung der Berechtigungsdateien	159
Eine Übertragungswarteschlange erstellen.	136	Die Principal-Datenbank.	159
Standardübertragungswarteschlangen	136	Die OAM-Datenbank.	159
Definitionen ferner Warteschlangen als Aliasnamen		Datensätze für Klassenberechtigungen	161
verwenden	136	Datensatz mit Berechtigungen für alle Klassen	161
Aliasname eines WS-Managers	136	Kapitel 9. MQSeries-Steuerroutine der Warte-	
Aliasnamen für Warteschlangen für zu beant-		schlange für nicht zustellbare Nachrichten	163
wortende Nachrichten	137	DLQ-Steuerroutine aufrufen	163
Datenkonvertierung	137	Regeltabelle für die DLQ-Steuerroutine.	164
Ein WS-Manager kann Nachrichten nicht in		Steuerdaten	164
integrierte Formate konvertieren	138	Regeln (Muster und Aktionen)	166
Datei CCSID	138	Schlüsselwörter für die Mustererkennung	166
Standarddatenkonvertierung	138	Aktionsschlüsselwörter	168
Konvertierung von Nachrichten in benutzer-		Konventionen für die Regeltabelle	170
definierte Formate.	139	Verarbeitung der Regeltabelle	171
CCSID des WS-Managers ändern.	139	Verarbeitung aller DLQ-Nachrichten sicherstel-	
Kapitel 8. MQSeries-Objekte schützen	141	len	172
Gründe für den Schutz von MQSeries-Ressourcen	141	Regeltabelle für DLQ-Steuerroutine - Beispiel.	173
Konzept der Benutzer-IDs in der Benutzergruppe		Kapitel 10. Instrumentierungs- und EMS-	
MQM	141	Ereignisse	175
Weitere Informationen	142	MQSeries-Instrumentierungsereignisse	175
Objektberechtigungsmanager (OAM, Object Autho-		Ereignisarten	176
rity Manager)	142	Ereignisaufzeichnung durch Ereigniswarte-	
Funktionsweise des OAM	143	schlangen	176
Zugriffsverwaltung über Benutzergruppen	143	Ereigniswarteschlangen mit Auslösern verwen-	
Benutzer, die zu mehreren Benutzergruppen		den.	177
gehören	143	Instrumentierungsereignisse aktivieren	177
Gruppen und Primärgruppe	144	Ereignisnachrichten	177
Ressourcen über den OAM schützen	144	EMS-Ereignisse.	178
Berechtigungserteilung über Gruppen	144	EMS-Schablonendateien in MQSeries for Com-	
OAM inaktivieren	146	paq NSK	178
OAM-Befehle verwenden	146	MQSeries-EMS-Ereignisschablonen integrieren	179
Angaben in OAM-Befehlen.	146	Umgebungsvariable MQEMSEVENTS definieren	181

Alternativen Ereignissammler (Event Collector) verwenden	182	Die nächsten Schritte	210
Programme für die Verarbeitung von MQSeries- EMS-Ereignissen erstellen	182	Haben Sie falsche Ausgaben erhalten?	210
Kapitel 11. Transaktionale Unterstützung und transaktionale Nachrichtenübertragung.	183	Haben Sie von einem PCF-Befehl keine Antwort erhalten?	211
NonStop-TM/MP (Transaktionsmanager) verwenden.	183	Sind einige Ihrer Warteschlangen fehlerhaft?	212
Begrenzung innerhalb von Transaktionen	184	Betrifft der Fehler nur ferne Warteschlangen?	212
Operationen mit Option NOSYNCPPOINT für permanente Nachrichten ausführen	185	Läuft Ihre Anwendung oder MQSeries for Com- paq NSK langsam?	213
Operationen mit SYNCPPOINT für nicht perma- nente Nachrichten	185	Überlegungen zum Anwendungsdesign	214
Konfigurationsvoraussetzungen für TM/MP und MQSeries for Compaq NSK	185	Auswirkung der Nachrichtenlänge	214
Überwachung	186	Nach einer bestimmten Nachricht suchen	214
Größe für Prüfprotokolle	186	Warteschlangen mit Nachrichten in unterschied- licher Länge	214
Konfiguration des Ressourcenmanagers	186	Häufigkeit von Synchronisationspunkten	215
Fehlerbehebung	187	Den Aufruf MQPUT1 verwenden.	215
Kapitel 12. Wiederherstellung und Neustart	189	Falsche Ausgabe	215
Fehlertoleranz und Fehlerbehebung	189	Nachrichten erscheinen nicht in der Warte- schlange	215
MQSeries-Daten sichern und zurückschreiben	190	Nachrichten mit unerwarteten oder beschädig- ten Informationen	216
MQSeries-Daten sichern	190	Probleme mit falschen Ausgaben bei der Ver- wendung verteilter Warteschlangen	217
MQSeries zurückschreiben	190	Fehlerprotokolle	219
Status- und Warteschlangenserver wiederherstellen und erneut starten.	191	Protokolldateien	219
Fehlerbehebung mit RDF	192	Fehler in der Startphase	220
Kapitel 13. Konfigurationsdateien	193	Bedienernachrichten	220
Konfigurationsdateien - Beschreibung	193	Nummern der Ausführungs-Controller in der Datei MQERRLG ermitteln	220
MQSeries-Konfigurationsdatei (MQSINI)	193	Beispiel für ein Fehlerprotokoll	220
Inhalt der MQSeries-Konfigurationsdatei	194	EMS-Ereignisse	221
Konfigurationsdatei der WS-Manager (QMINI)	195	Warteschlangen für nicht zustellbare Nachrichten	222
Inhalt der Konfigurationsdatei eines WS-Managers	195	Konfigurationsdateien und Fehlerbestimmung	222
Konfigurationsdatei für WS-Manager - Beispiel	198	MQSeries-Trace verwenden.	222
Konfigurationsdateien editieren	203	Namen von Trace-Dateien	222
Änderungen in Konfigurationsdateien imple- mentieren	203	Beispiel für Trace-Daten	223
Empfehlungen für Konfigurationsdateien	203	First Failure Support Technology™ (FFST)	223
Kapitel 14. Fehlerbestimmung	205	FFST-Daten überprüfen	224
Erste Überprüfungen	205	Kapitel 15. Skalierbarkeit und Systemleistung	225
Lief MQSeries bis jetzt fehlerfrei?	205	Einführung	225
Werden Fehlernachrichten angezeigt?	205	Neue Anwendungen im Hinblick auf Leistung und Skalierbarkeit erstellen.	225
Werden Rückkehrcodes mit Fehlererläuterungen angezeigt?	206	Verwendung gemeinsam benutzter Ressourcen minimieren oder eliminieren	225
Können Sie den Fehler reproduzieren?	206	Leistungsoptimierung (Durchsatzverbesserung) als iterativer Prozess	226
Trat der Fehler erst auf, nachdem Änderungen vorgenommen wurden?	206	Permanente Nachrichten	226
Lief die Anwendung bisher fehlerfrei?	207	Nicht permanente Nachrichten	227
Die Anwendung lief bisher nicht fehlerfrei	207	Nicht permanente Nachrichten und Kanäle	227
Betrifft der Fehler bestimmte Teile des Netzes?	208	Warteschlangenserver und Warteschlangendateien	228
Tritt der Fehler zu einer bestimmten Tageszeit auf?	208	Speicherverfahren für permanente Nachrichten	228
Tritt der Fehler unregelmäßig auf?	209	Speicherverfahren für nicht permanente Nach- richten	229
Haben Sie Funktionsaktualisierungen installiert?	209	Verteilung der Warteschlangenserver auf die CPUs	229
Typische Programmierfehler	209	Warteschlangen mit dem Befehl 'altmqfls' ande- ren Warteschlangenservern zuordnen	229
Fehler in Zusammenhang mit Befehlen	210	Cluster-Übertragungswarteschlange: SYSTEM.CLUSTER.TRANSMIT.QUEUE	229
		Warteschlangen mit dem Befehl 'altmqfls' ver- schieben	230

Warteschlangendateien partitionieren	230
Nachrichtenüberlaufdateien	231
Nachrichten zum Durchsuchen im Hauptspeicher ablegen.	231
Weitere Warteschlangenserveroptionen	232
Laden beim Start	232
Sperrung im Cache	232
Prüfpunktinformationen für nicht permanente Nachrichten.	232
Längenzähler	233
CPU-Zuordnung	233
FASTPATH-Bindungen für Anwendungsprogramme	234
Hintergrund.	235
Systemaufwand bei MQI-Aufrufen reduzieren	235
FASTPATH-Bindungen aktivieren.	236
Einschränkungen bei der Verwendung von FASTPATH-Bindungen	236
Kapitel 16. Datenintegrität und Verfügbarkeit	237
Datenintegrität	237
Verfügbarkeit	238
Permanente und nicht permanente Daten	239
Permanente Nachrichten	240
Nicht permanente Nachrichten	241
Datenbankkonsistenz	242
Interne Datenbankkonsistenz	242
Externe Datenbankkonsistenz	243
OpenTMF	244
NonStop Tuxedo	244
Gleichzeitige Ausführung von Transaktionen in Anwendungen	245
Kritische MQSeries-Datenbankdateien	245
Kritische Prozesse	246
Cluster	253
Konfigurationsoptionen für die Verfügbarkeit	253
Konfigurationsoptionen für die Datenintegrität	254

Kapitel 1. Einführung

Dieses Kapitel gibt eine Einführung in MQSeries for Compaq NonStop Kernel Version 5.1 (MQSeries for Compaq NSK V5.1) aus der Sicht des Administrators und beschreibt die grundlegenden Konzepte von MQSeries und der Nachrichtenübertragung. Es enthält die folgenden Abschnitte:

- „MQSeries und Message-Queuing“
- „Nachrichten und Warteschlangen“ auf Seite 8
- „Objekte“ auf Seite 10
- „MQSeries-Warteschlangen“ auf Seite 12
- „Prozessdefinitionen“ auf Seite 16
- „Kanäle“ auf Seite 16
- „Systemstandardobjekte“ auf Seite 17
- „Verwaltung“ auf Seite 18
- „Clients und Server“ auf Seite 19
- „WS-Managerfunktionen erweitern“ auf Seite 19
- „Sicherheit“ auf Seite 21
- „Transaktionsunterstützung“ auf Seite 21
- „Durchsatzverbesserung, Zuverlässigkeit, Skalierbarkeit und Größenanpassung“ auf Seite 22

MQSeries und Message-Queuing

MQSeries ermöglicht Anwendungsprogrammen mit Hilfe von Message-Queuing die Teilnahme an einer nachrichtengesteuerten Verarbeitung. Anwendungsprogramme können über verschiedene Plattformen miteinander kommunizieren, indem sie die entsprechenden Softwareprodukte für Message-Queuing verwenden. Beispielsweise können Compaq NSK- und MVS/ESA™-Anwendungen über MQSeries for Compaq NSK bzw. MQSeries for OS/390® kommunizieren. Die Anwendungen sind von den technischen Spezifikationen der zu Grunde liegenden Kommunikationshardware unabhängig.

MQSeries-Produkte implementieren eine einheitliche Anwendungsprogrammierschnittstelle, die so genannte Schnittstelle für Nachrichtenwarteschlangen (MQI, Message Queue Interface) auf jeder Plattform, auf der die Anwendungen ausgeführt werden. Das ermöglicht eine einfache Übertragung von Anwendungen von einer Plattform auf eine andere.

MQI wird im Handbuch *MQSeries Application Programming Reference* ausführlich beschrieben.

Zeitunabhängige Anwendungen

Durch Message-Queuing kann der Austausch von Nachrichten zwischen den sendenden und empfangenden Programmen zeitunabhängig erfolgen. Das heißt, dass die sendenden und empfangenden Anwendungen voneinander entkoppelt werden, der Sender also die Verarbeitung fortsetzen kann, ohne darauf warten zu müssen, dass der Empfänger den Eingang der Nachricht bestätigt. Tatsächlich muss die Zielanwendung nicht einmal aktiv sein, wenn die Nachricht gesendet wird. Sie kann die Nachricht abrufen, nachdem sie gestartet wurde.

Nachrichtengesteuerte Verarbeitung

Bei der Ankunft in einer Warteschlange können Nachrichten eine Anwendung automatisch über einen Mechanismus, die so genannte *Auslösefunktion*, starten. Bei Bedarf können die Anwendungen gestoppt werden, nachdem die Nachricht(en) verarbeitet wurde(n).

Nachrichten und Warteschlangen

Nachrichten und Warteschlangen sind die Basiskomponenten eines Message-Queuing-Systems.

Nachrichten

Eine *Nachricht* ist eine Zeichenfolge von Bytes, die für die Anwendungen, die diese Zeichenfolge verwenden, eine Bedeutung hat. Nachrichten werden zur Übertragung von Informationen von einer Anwendung an eine andere (oder an verschiedene Komponenten derselben Anwendung) verwendet. Die Anwendungen können auf derselben Plattform oder auf verschiedenen Plattformen aktiv sein.

MQSeries-Nachrichten bestehen aus zwei Teilen:

- *Anwendungsdaten*

Der Inhalt und die Struktur von Anwendungsdaten werden durch das Anwendungsprogramm, das diese Daten verwendet, definiert.

- *Nachrichtendeskriptor*

Der Nachrichtendeskriptor identifiziert die Nachricht und enthält weitere Steuerinformationen, z. B. den Nachrichtentyp und die Priorität, die der Nachricht von der sendenden Anwendung zugewiesen wurden.

Das Format des Nachrichtendeskriptors wird von MQSeries definiert. Eine vollständige Beschreibung des Nachrichtendeskriptors finden Sie im Handbuch *MQSeries Application Programming Reference*.

Das Format des Nachrichtendeskriptors wird von MQSeries definiert. Eine vollständige Beschreibung des Nachrichtendeskriptors finden Sie im Handbuch *MQSeries Application Programming Reference*.

Bei Nachrichten unterscheidet man zwischen permanenten und nicht permanenten Nachrichten. Sie unterscheiden sich wie folgt:

- Permanente Nachrichten sind auch nach dem Neustart eines WS-Managers noch vorhanden, nicht permanente Nachrichten hingegen nicht.
- Nicht permanente Nachrichten werden in der Regel nicht auf Platte geschrieben; sie können daher rascher in Warteschlangen eingereiht bzw. aus Warteschlangen entfernt werden und benötigen weniger Ressourcen als permanente Nachrichten.
- Bei bestimmten Störfällen sind nicht permanente Nachrichten weniger zuverlässig als permanente Nachrichten.
- Permanente Nachrichten können nicht in temporäre dynamische Warteschlangen eingereiht werden.

Nachrichtenlänge

In MQSeries beträgt die maximale Nachrichtenlänge 100 MB (1 MB entsprechen 1.048.576 Byte). Die Nachrichtenlänge kann durch folgende Vorgaben begrenzt sein:

- die maximale Nachrichtenlänge, die für die Empfangswarteschlange definiert wurde,
- die maximale Nachrichtenlänge, die für den Warteschlangenmanager definiert wurde,
- die maximale Nachrichtenlänge, die entweder von der sendenden oder von der empfangenden Anwendung definiert wurde,
- durch den für die Nachricht verfügbaren Speicherbereich.

Unter Umständen sind für die Übertragung aller für eine Anwendung erforderlichen Informationen mehrere Nachrichten erforderlich.

Eine Erhöhung der maximalen Nachrichtenlänge könnte unter Umständen negative Auswirkungen haben. Außerdem könnte sie eventuell zu groß für die Warteschlange oder den WS-Manager sein. In diesen Fällen ist es möglich, die Nachricht in einzelne Segmente zu teilen, die anschließend wieder zu einer logischen Nachricht zusammengefasst werden. Die logische Gruppierung von Nachrichten ermöglicht es Anwendungen, ähnliche Nachrichten zusammenzufassen und ihre Reihenfolge sicherzustellen. Weitere Informationen zur Nachrichtensegmentierung und -zusammenfassung finden Sie im Handbuch *MQSeries Application Programming Guide*.

Warteschlangen

Eine *Warteschlange* ist eine Datenstruktur, die zum Speichern von Nachrichten verwendet wird. Die Nachrichten können von Anwendungsprogrammen oder einem WS-Manager (als Teil seines normalen Betriebsablaufs) in die Warteschlange eingeordnet werden.

Jede Warteschlange ist einem *WS-Manager* zugeordnet. Der WS-Manager ist für die Verwaltung der ihm zugeordneten Warteschlangen und für die Speicherung aller von ihm empfangenen Nachrichten in den entsprechenden Warteschlangen zuständig.

Die maximale Größe einer Warteschlange beträgt 4 GB. Weitere Informationen zur Planung des für Warteschlangen erforderlichen Speicherbereichs finden Sie im Handbuch *MQSeries Planning Guide*, oder besuchen Sie die folgende Website für plattformspezifische Leistungsberichte:

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

Senden und Empfangen von Nachrichten durch Anwendungen

Anwendungen senden und empfangen Nachrichten über *MQI-Aufrufe*. Um zum Beispiel eine Nachricht in eine Warteschlange einzureihen, geht eine Anwendung folgendermaßen vor:

1. Sie öffnet die betreffende Warteschlange mit dem MQI-Aufruf MQOPEN.
2. Sie reiht die Nachricht mit dem MQI-Aufruf MQPUT in die Warteschlange ein.
3. Eine andere Anwendung kann die Nachricht mit dem MQI-Aufruf MQGET aus der betreffenden Warteschlange abrufen.

Weitere Informationen zu MQI-Aufrufen finden Sie im Handbuch *MQSeries Application Programming Reference*.

Nachrichten und Warteschlangen

Vordefinierte und dynamische Warteschlangen

Warteschlangen können nach der Art und Weise ihrer Erstellung unterschieden werden:

- *Vordefinierte Warteschlangen* werden von einem Administrator mit Hilfe des entsprechenden Befehlssatzes erstellt. Beispiel: Mit dem MQSC-Befehl DEFINE QLOCAL wird eine vordefinierte lokale Warteschlange erstellt. Vordefinierte Warteschlangen sind permanent, d. h., sie existieren unabhängig von den Anwendungen, von denen sie verwendet werden, und bleiben auch bei einem Neustart von MQSeries erhalten.
- *Dynamische Warteschlangen* werden erstellt, wenn eine Anwendung eine OPEN-Anforderung absetzt und dabei den Namen einer *Modellwarteschlange* angibt. Die erstellte Warteschlange basiert auf einer Vorlage mit einer Warteschlangendefinition, der so genannten *Modellwarteschlange*. Eine *Modellwarteschlange* können Sie mit dem MQSC-Befehl DEFINE QMODEL erstellen. Jede dynamische Warteschlange, die aus einer *Modellwarteschlange* erstellt wird, übernimmt die Attribute der betreffenden *Modellwarteschlange*, z. B. die maximale Anzahl der Nachrichten, die gespeichert werden können.

Modellwarteschlangen verfügen über ein Attribut, das angibt, ob die dynamische Warteschlange permanent oder temporär ist. Permanente Warteschlangen bleiben beim Neustart einer Anwendung oder eines WS-Managers erhalten, temporäre Warteschlangen dagegen gehen bei einem Neustart verloren.

Nachrichten aus Warteschlangen abrufen

In MQSeries können Anwendungen, die über eine entsprechende Berechtigung verfügen, Nachrichten nach den folgenden Abrufalgorithmen aus einer Warteschlange abrufen:

- First In/First Out (FIFO)
- Nachrichtenpriorität, so wie im Nachrichtendeskriptor definiert. Nachrichten mit derselben Priorität werden nach der FIFO-Methode abgerufen.
- ein Programmaufruf für eine bestimmte Nachricht

Die jeweils verwendete Methode wird durch den MQGET-Aufruf aus der Anwendung bestimmt.

Objekte

Viele der in diesem Buch beschriebenen Tasks beinhalten eine Bearbeitung von MQSeries-*Objekten*. In MQSeries Version 5.1 gibt es folgende Objektarten: WS-Manager, Warteschlangen, Prozessdefinitionen, Kanäle, Cluster und Namenslisten.

Die Bearbeitung oder *Verwaltung* von Objekten umfasst Folgendes:

- Starten und Stoppen von WS-Managern
- Erstellen von Objekten, insbesondere Warteschlangen, für Anwendungen
- Arbeiten mit Kanälen zum Erstellen von Kommunikationspfaden zu WS-Managern auf anderen (fernen) Systemen. Dies wird im Handbuch *MQSeries Intercommunication* ausführlich beschrieben.
- Erstellen von *Clustern* aus WS-Managern, um den Gesamtverwaltungsprozess zu vereinfachen oder um eine gleichmäßige Auslastung zu erreichen.

Die folgenden Kapitel dieses Buches enthalten detaillierte Informationen zur Verwaltung:

- „Kapitel 3. MQSeries-Befehlssätze verwenden“ auf Seite 31
- „Kapitel 4. WS-Manager verwalten“ auf Seite 47
- „Kapitel 5. Lokale MQSeries-Objekte verwalten“ auf Seite 95
- „Kapitel 7. Ferne MQSeries-Objekte verwalten“ auf Seite 125

Objektnamen

Jedes Exemplar eines WS-Managers wird durch einen Objektnamen gekennzeichnet. Dieser Objektname muss innerhalb des WS-Managernetzes eindeutig sein, damit der Zielwarteschlangenmanager ermittelt werden kann, an den eine Nachricht gesendet wird.

Der Objektname muss innerhalb eines WS-Managers und einer Objektart eindeutig sein. Eine Warteschlange und ein Prozess können beispielsweise denselben Namen haben, nicht jedoch zwei Warteschlangen.

Objektnamen können aus maximal 48 Zeichen bestehen; ausgenommen hiervon sind *Kanäle*. Kanalobjektnamen können aus maximal 20 Zeichen bestehen. Weitere Informationen zu Namen finden Sie unter „Namenskonventionen“ auf Seite 258.

Objekte verwalten

MQSeries stellt Möglichkeiten zum Erstellen, Ändern, Anzeigen und Löschen von Objekten zur Funktionen. Dazu gehören folgende:

- MQSC-Befehle, die über die Tastatur eingegeben oder aus einer Datei übernommen werden können.
- MQM (eine Schnittstelle in Form von Bildschirmanzeigen).
- PCF-Befehle (Programmable Command Format), die von Programmen verwendet werden können.
- Steuerbefehle, die Sie über eine Befehlszeile des Betriebssystems eingeben können.

Weitere Informationen finden Sie in „Kapitel 3. MQSeries-Befehlssätze verwenden“ auf Seite 31.

Objektattribute

Die Eigenschaften eines Objekts werden über seine Attribute definiert. Einige dieser Attribute können angegeben bzw. geändert, andere wiederum nur angezeigt werden. Beispielsweise wird die in einer Warteschlange maximal zulässige Nachrichtenlänge über das Attribut *MaxMsgLength* festgelegt. Sie können den Wert dieses Objektattributs bei der Erstellung einer Warteschlange angeben. Das Attribut *DefinitionType* gibt an, wie eine Warteschlange erstellt wurde. Dieses Attribut kann nur angezeigt werden.

In MQSeries gibt es zwei Möglichkeiten, auf ein Objektattribut zu verweisen:

- über seinen PCF-Namen, z. B. *MaxMsgLength*. Bei dem PCF-Namen handelt es sich um den formalen Namen eines Attributs.
- über seinen MQSC-Namen, z. B. *MAXMSGL*.

Der formale Name eines Attributs ist sein PCF-Name. Da die MQSC-Funktion in diesem Buch eine wichtige Rolle spielt, wird in Beispielen in der Regel der MQSC-Name eines Attributs verwendet, und nicht sein PCF-Name.

MQSeries-WS-Manager

Ein WS-Manager stellt Message-Queuing-Services für Anwendungen zur Verfügung. Er stellt Folgendes sicher:

- Objektattribute werden den empfangenen Befehlen entsprechend geändert.
- Es werden besondere Ereignisse wie Auslöse- oder Instrumentierungseignisse generiert, wenn die entsprechenden Bedingungen eintreten.
- Nachrichten werden so in die richtige Warteschlange eingereiht, wie von der Anwendung durch den MQPUT-Aufruf gefordert. Sollte dies nicht möglich sein, wird dies der Anwendung gemeldet und ein entsprechender Ursachencode ausgegeben.

Jede Warteschlange gehört zu einem bestimmten WS-Manager und wird als *lokale Warteschlange* des betreffenden WS-Managers bezeichnet. Der WS-Manager, mit dem eine Anwendung verbunden ist, wird als der lokale WS-Manager der betreffenden Anwendung bezeichnet. Für die Anwendung sind die Warteschlangen ihres lokalen WS-Managers lokale Warteschlangen.

Eine *ferne Warteschlange* ist nichts anderes als eine Warteschlange, die einem anderen WS-Manager zugeordnet ist.

Ein *ferner WS-Manager* ist jeder WS-Manager, der nicht der lokale WS-Manager ist. Ein ferner WS-Manager kann sich auf einem fernen System im Netz oder auf demselben System wie der lokale WS-Manager befinden.

MQSeries unterstützt mehrere WS-Manager auf demselben System.

MQI-Aufrufe

WS-Managerobjekte können für verschiedene MQI-Aufrufe verwendet werden. Mit dem MQI-Aufruf MQINFO beispielsweise können Sie Objektattribute abfragen.

Anmerkung: Nachrichten werden immer in Warteschlangenobjekte, nicht in WS-Managerobjekte eingereiht. Letzteres ist nicht möglich.

MQSeries-Warteschlangen

Warteschlangen werden in MQSeries for Compaq NSK folgendermaßen definiert:

- mit DEFINE-Befehlen (ein MQSC-Befehl),
- mit dem Tool MQM (Message Queue Management), das in MQSeries for Compaq NSK zur Verfügung steht,
- mit dem PCF-Befehl **Create Queue**,
- mit MQAI-Befehlen.

In diesen Befehlen werden der Typ der Warteschlange und ihre Objektattribute angegeben. Eine lokale Warteschlange verfügt beispielsweise über Objektattribute, die festlegen, welche Aktion ausgeführt wird, wenn Anwendungen in MQI-Aufrufen diese Warteschlange angeben. Über Objektattribute wird beispielsweise Folgendes festgelegt:

- ob Anwendungen Nachrichten aus der Warteschlange abrufen können (Warteschlange ist für GET-Vorgänge aktiviert),
- ob Anwendungen Nachrichten in die Warteschlange einreihen können (Warteschlange ist für PUT-Vorgänge aktiviert),

- ob der Zugriff auf die Warteschlange ausschließlich einer Anwendung vorbehalten ist oder ob mehrere Anwendung auf die Warteschlange zugreifen können,
- die maximale Anzahl an Nachrichten, die gleichzeitig in der Warteschlange gespeichert werden können (maximale Warteschlangenlänge),
- die maximal zulässige Länge von Nachrichten, die in die Warteschlange eingereiht werden können.

Weitere Informationen finden Sie:

- im Handbuch *MQSeries MQSC - Befehle* (zu MQSC-Befehlen),
- unter „MQM (Message Queue Management) verwenden“ auf Seite 73 (zu MQM),
- im Handbuch *MQSeries Programmable System Management* (zu PCF-Befehlen).

Warteschlangenobjekte verwenden

In MQSeries gibt es verschiedene Arten von Warteschlangenobjekten. Jede Objektart kann mit Hilfe der Produktbefehle bearbeitet werden, und jeder werden auf unterschiedliche Art und Weise tatsächliche Warteschlangen zugeordnet:

- **Lokales Warteschlangenobjekt**

Ein lokales Warteschlangenobjekt identifiziert eine lokale Warteschlange, die dem WS-Manager zugeordnet ist, mit dem die Anwendung verbunden ist. Alle Warteschlangen sind lokale Warteschlangen in dem Sinne, dass jede Warteschlange einem WS-Manager zugeordnet ist, und die Warteschlange für diesen WS-Manager eine lokale Warteschlange darstellt.

- **Fernes Warteschlangenobjekt**

Ein fernes Warteschlangenobjekt identifiziert eine Warteschlange, die einem anderen WS-Manager zugeordnet ist. Diese Warteschlange muss als lokale Warteschlange des betreffenden WS-Managers definiert werden. Anhand der Informationen, die Sie beim Definieren eines fernen Warteschlangenobjekts angeben, kann der lokale WS-Manager den fernen WS-Manager ausfindig machen, so dass alle Nachrichten für die ferne Warteschlange an den richtigen WS-Manager gesendet werden.

Bevor Anwendungen Nachrichten an eine Warteschlange eines fernen WS-Managers senden können, müssen Sie eine Übertragungswarteschlange und Kanäle zwischen den WS-Managern definieren, es sei denn, Sie haben mehrere WS-Manager in einem Cluster zusammengefasst. Weitere Informationen zu Clustern finden Sie unter „Fernverwaltung mit Hilfe von Clustern“ auf Seite 126.

- **Aliaswarteschlangenobjekt**

Mit Hilfe eines Aliaswarteschlangenobjekts können Anwendungen auf eine Warteschlange zugreifen, indem sie in MQI-Aufrufen indirekt darauf verweisen. Wenn ein Aliaswarteschlangenname in einem MQI-Aufruf verwendet wird, wird der Name entweder in den Namen einer lokalen oder den einer fernen Warteschlange aufgelöst. Dies gibt Ihnen die Möglichkeit, die von einer Anwendung verwendeten Warteschlangen zu ändern, ohne die Anwendung selbst ändern zu müssen. Sie ändern einfach die Definition der Aliaswarteschlange und geben dort den Namen der neuen Warteschlange ein, in den der Aliasname aufgelöst werden soll.

Eine Aliaswarteschlange ist keine Warteschlange, sondern ein Objekt, über das Sie auf eine andere Warteschlange zugreifen können.

MQSeries-Warteschlangen

- **Modellwarteschlangenobjekt**

Ein Modellwarteschlangenobjekt definiert eine Gruppe von Warteschlangenattributen, die als Vorlage zum Erstellen einer dynamischen Warteschlange verwendet wird. Dynamische Warteschlangen werden vom WS-Manager erstellt, wenn eine Anwendung einen MQOPEN-Aufruf absetzt und es sich bei dem darin angegebenen Warteschlangennamen um den Namen einer Modellwarteschlange handelt. Die auf diese Weise erstellte dynamische Warteschlange ist eine lokale Warteschlange, deren Attribute aus der Modellwarteschlangendefinition übernommen werden. Der Name der dynamischen Warteschlange kann von der Anwendung angegeben oder vom WS-Manager generiert und an die Anwendung zurückgegeben werden.

Auf diese Weise definierte dynamische Warteschlangen können temporäre Warteschlangen sein, die beim Neustart des Produkts verloren gehen, oder permanente Warteschlangen, die bei einem Neustart erhalten bleiben.

Lokale Warteschlangen in MQSeries

MQSeries verwendet verschiedene lokale Warteschlangen für jeweils bestimmte Operationen. Diese Warteschlangen *müssen* Sie definieren, bevor MQSeries sie verwenden kann.

Anwendungswarteschlangen: Eine Warteschlange, die über MQI-Aufrufe von Anwendungen verwendet wird, wird als Anwendungswarteschlangen bezeichnet. Dabei kann es sich um eine lokale Warteschlange im WS-Manager, mit dem die Anwendung verbunden ist, oder um eine ferne Warteschlange, die einem anderen WS-Manager zugeordnet ist, handeln.

Anwendungen können Nachrichten in lokale und ferne Warteschlangen einreihen. Sie können Nachrichten jedoch nur aus einer lokalen Warteschlange abrufen.

Initialisierungswarteschlangen: Initialisierungswarteschlangen werden für Auslösefunktionen verwendet. Ein WS-Manager reiht eine Auslösenachricht in eine Initialisierungswarteschlange ein, sobald ein Auslöseereignis eintritt. Ein Auslöseereignis ist eine logische Kombination aus Bedingungen, die von einem WS-Manager erkannt wird. Ein Auslöseereignis kann beispielsweise generiert werden, wenn die Anzahl der Nachrichten in einer Warteschlange die vorgegebene Größe erreicht. Dieses Ereignis veranlasst den WS-Manager, eine Auslösenachricht in eine bestimmte Initialisierungswarteschlange einzureihen. Diese Auslösenachricht wird von einem *Auslösemonitor*, einer speziellen Anwendung für die Überwachung von Initialisierungswarteschlangen, abgerufen. Der Auslösemonitor startet dann das Anwendungsprogramm, das in der Auslösenachricht angegeben ist.

Wenn ein WS-Manager die Auslösefunktion verwenden soll, muss mindestens eine Initialisierungswarteschlange für den betreffenden WS-Manager definiert werden.

Siehe hierzu „Objekte für Auslösefunktion verwalten“ auf Seite 117. Weitere Informationen zur Auslösefunktion finden Sie im Handbuch *MQSeries Application Programming Guide*.

Übertragungswarteschlangen: In einer Übertragungswarteschlange werden temporär Nachrichten gespeichert, die an einen fernen WS-Manager gerichtet sind. Sie müssen mindestens eine Übertragungswarteschlange für jeden fernen WS-Manager definieren, an den der lokale WS-Manager Nachrichten direkt senden soll. Diese Warteschlangen werden auch für die Fernverwaltung verwendet. Siehe hierzu „Einen fernen WS-Manager verwalten“ auf Seite 127. Weitere Informationen zur Verwendung von Übertragungswarteschlangen bei einem verteilten Message-Queuing finden Sie im Handbuch *MQSeries Intercommunication*.

Cluster-Übertragungswarteschlangen: Jeder WS-Manager in einem Cluster verfügt über eine Cluster-Übertragungswarteschlange mit dem Namen SYSTEM.CLUSTER.TRANSMIT.QUEUE. Eine Definition dieser Warteschlange wird standardmäßig in jedem WS-Manager erstellt.

Ein WS-Manager in einem Cluster kann über die Cluster-Übertragungswarteschlange Nachrichten an die anderen WS-Manager innerhalb desselben Clusters senden.

WS-Manager in einem Cluster können darüber hinaus auch mit WS-Managern kommunizieren, die nicht zu dem Cluster gehören. Dazu muss der WS-Manager wie in traditionellen Umgebungen mit verteilter Steuerung von Warteschlangen Kanäle und eine Übertragungswarteschlange zu dem anderen WS-Manager definieren.

Bei der Namensauflösung hat die Cluster-Übertragungswarteschlange Vorrang vor der Standardübertragungswarteschlange. Wenn ein WS-Manager, der dem Cluster nicht angehört, eine Nachricht in eine ferne Warteschlange einreicht, wird immer die Standardübertragungswarteschlange verwendet, falls keine Übertragungswarteschlange mit demselben Namen wie dem des Ziel-WS-Managers vorhanden ist.

Gehört ein WS-Manager zu einem Cluster, wird standardmäßig SYSTEM.CLUSTER.TRANSMIT.QUEUE verwendet, es sei denn, der Ziel-WS-Manager gehört nicht zum Cluster.

Warteschlangen für nicht zustellbare Nachrichten: In einer Warteschlange für nicht zustellbare Nachrichten werden Nachrichten gespeichert, die nicht an das vorgegebene Ziel weitergeleitet werden können. Dieser Fall tritt beispielsweise ein, wenn die Zielwarteschlange voll ist. Der Namen der standardmäßigen Warteschlange für nicht zustellbare Nachrichten lautet SYSTEM.DEAD.LETTER.QUEUE. Diese Warteschlangen werden auf anderen Plattformen auch als Warteschlangen für nicht übermittelte Nachrichten bezeichnet.

Für verteiltes Message-Queuing sollten Sie in jedem aktiven WS-Manager eine Warteschlange für nicht zustellbare Nachrichten definieren.

Befehlswarteschlangen: Die Befehlswarteschlange SYSTEM.ADMIN.COMMAND.QUEUE ist eine lokale Warteschlange, an die entsprechend berechnete Anwendungen MQSeries for Compaq NSK-Befehle zur Verarbeitung senden können. Diese Befehle werden dann von einer MQSeries-Komponente, dem so genannten Befehlsserver, abgerufen. Der Befehlsserver überprüft die Befehle, übergibt die gültigen Befehle zur Verarbeitung an den WS-Manager, und gibt eventuell vorhandene Antworten an die Warteschlange für zu beantwortende Nachrichten zurück.

MQSeries-Warteschlangen

Warteschlangen für zu beantwortende Nachrichten: Sendet eine Anwendung eine Anforderungsnachricht, kann die Anwendung, die diese Nachricht empfängt, eine Antwort an die sendende Anwendung zurückschicken. Diese Nachricht wird in eine Warteschlange, die so genannte Warteschlange für zu beantwortende Nachrichten, eingereiht. Dabei handelt es sich normalerweise um eine lokale Warteschlange der sendenden Anwendung. Der Name der Warteschlange für zu beantwortende Nachrichten wird von der sendenden Anwendung als Teil des Nachrichtendesktors angegeben.

Ereigniswarteschlangen: MQSeries for Compaq NSK unterstützt Instrumentierungsereignisse, die zur Überwachung von WS-Managern unabhängig von MQI-Anwendungen verwendet werden können. Instrumentierungsereignisse können auf verschiedene Weise generiert werden, z. B. wie folgt:

- Eine Anwendung versucht eine Nachricht in eine Warteschlangen einzureihen, die nicht verfügbar oder nicht vorhanden ist.
- Eine Warteschlange ist voll.
- Ein Kanal wird gestartet.

Wenn ein Instrumentierungsereignis eintritt, reiht der WS-Manager eine Ereignisnachricht in eine Ereigniswarteschlange ein. Diese Nachricht kann anschließend von einer Überwachungsanwendung gelesen werden, die wiederum einen Administrator informieren oder eine Fehlerbehebung einleiten kann, falls das Ereignis auf einen Fehler hinweist.

Anmerkung: Auslöseereignisse unterscheiden sich von Instrumentierungsereignissen insofern, als Erstere nicht durch dieselben Bedingungen verursacht werden und keine Ereignisnachrichten generieren.

Weitere Informationen zu Instrumentierungsereignissen finden Sie im Handbuch *MQSeries Programmable System Management*.

Prozessdefinitionen

Ein *Prozessdefinitionsobjekt* definiert eine Anwendung, die auf ein Auslöseereignis in einem WS-Manager von MQSeries for Compaq NSK hin gestartet wird. Weitere Informationen hierzu finden Sie unter „Initialisierungwarteschlangen“ auf Seite 14.

Die Prozessdefinitionsattribute umfassen die Anwendungs-ID, den Anwendungstyp und anwendungsspezifische Daten.

Verwenden Sie zum Erstellen einer Prozessdefinition den MQSC-Befehl DEFINE PROCESS oder den PCF-Befehl 'Create Process'.

Kanäle

Kanäle sind Objekte, die einen Kommunikationspfad zwischen zwei WS-Managern bereitstellen. Kanäle werden beim verteilten Message-Queuing zum Verschieben von Nachrichten von einem WS-Manager zu einem anderen verwendet. Durch sie werden Anwendungen von den zu Grunde liegenden Kommunikationsprotokollen unabhängig. Die WS-Manager können sich auf derselben oder auf verschiedenen Plattformen befinden. Damit WS-Manager miteinander kommunizieren können, müssen Sie ein Kanalobjekt für den WS-Manager, der Nachrichten senden wird, und ein Kanalobjekt für den WS-Manager, der diese Nachrichten empfangen wird, definieren.

MQSeries for Compaq NSK unterstützt die Übertragungsprotokolle TCP/IP und SNA LU6.2 (SNAX oder ICE).

Weitere Informationen zu Kanälen und zu deren Verwendung finden Sie im Handbuch *MQSeries Intercommunication* sowie unter „Kanäle und Übertragungswarteschlangen für Fernverwaltung vorbereiten“ auf Seite 128.

Cluster

In einem herkömmlichen MQSeries-Netz arbeitet bei einem verteilten Message-Queuing jeder WS-Manager unabhängig. Damit ein WS-Manager Nachrichten an einen anderen WS-Manager senden kann, muss für ihn eine Übertragungswarteschlange, ein Kanal zum fernen WS-Manager und eine Definition einer fernen Warteschlange definiert werden, und das für jede Warteschlange, an die der WS-Manager Nachrichten senden soll.

Bei einem Cluster handelt es sich um eine Gruppe von WS-Managern, die so konfiguriert sind, dass die WS-Manager über ein einziges Netz direkt miteinander kommunizieren können, ohne dass dazu komplexe Übertragungswarteschlangen, Kanäle und Warteschlangendefinitionen benötigt werden.

Weitere Informationen zu Clustern finden Sie in „Kapitel 7. Ferne MQSeries-Objekte verwalten“ auf Seite 125 und im Handbuch *Cluster-Unterstützung in MQSeries*.

Namenslisten

Eine Namensliste ist ein MQSeries-Objekt, das eine Liste anderer MQSeries-Objekte enthält. Namenslisten werden üblicherweise von Anwendungen wie Auslösemonitoren verwendet, um eine Warteschlangengruppe zu identifizieren. Der Vorteil einer Namensliste besteht darin, dass sie unabhängig von Anwendungen verwaltet wird; d. h., sie kann aktualisiert werden, ohne dass Anwendungen, die darauf zugreifen, gestoppt werden müssen. Außerdem ist die Namensliste nicht betroffen, wenn in einer Anwendung ein Fehler auftritt, so dass andere Anwendungen sie weiter verwenden können.

Namenslisten werden auch mit WS-Manager-Clustern verwendet, so dass Sie eine Liste von Clustern verwalten können, auf die von mehreren MQSeries-Objekten verwiesen wird.

Systemstandardobjekte

Bei *Systemstandardobjekten* handelt es sich um eine Gruppe von Objektdefinitionen, die automatisch zusammen mit einem WS-Manager erstellt werden.

Standardobjektnamen beginnen immer mit SYSTEM.DEFAULT; die standardmäßige lokale Warteschlange heißt zum Beispiel SYSTEM.DEFAULT.LOCAL.QUEUE und die standardmäßige Empfängerwarteschlange SYSTEM.DEFAULT.RECEIVER. Sie können diese Objekte nicht umbenennen, da Standardobjekte mit diesen Namen immer vorhanden sein müssen.

Systemstandardobjekte

Wenn Sie ein Objekt definieren, werden alle Attribute, die Sie nicht explizit angeben, aus dem zugehörigen Standardobjekt kopiert. Wenn Sie zum Beispiel eine lokale Warteschlange definieren, werden die Attribute, die Sie nicht angeben, von der Standardwarteschlange SYSTEM.DEFAULT.LOCAL.QUEUE übernommen.

Nach der Erstellung eines WS-Managers können Sie Standardeinstellungen mit dem Befehl **runmqsc** durch andere Werte ersetzen.

Weitere Informationen zu Systemstandardwerten finden Sie in „Anhang B. Systemstandardwerte“ auf Seite 327.

Verwaltung

In MQSeries werden Verwaltungs-Tasks durch die Eingabe von *Befehlen* ausgeführt. Hierfür stehen vier Befehlsgruppen zur Verfügung. Welche Gruppe verwendet wird hängt davon ab, welche Tasks ausgeführt werden sollen und wie dies geschehen soll. Eine Beschreibung dieser Befehlsgruppen finden Sie in „Kapitel 3. MQSeries-Befehlssätze verwenden“ auf Seite 31. Bei den Verwaltungs-Tasks handelt es sich unter anderem um folgende:

- Starten und Stoppen von WS-Managern
- Erstellen von Objekten, insbesondere Warteschlangen, für Anwendungen
- Arbeiten mit Kanälen zum Erstellen von Kommunikationspfaden zu WS-Managern auf anderen (fernen) Systemen. Eine ausführliche Beschreibung finden Sie im Handbuch *MQSeries Intercommunication*.

Lokale und ferne Verwaltung

Lokale Verwaltung bezeichnet die Ausführung von Verwaltungs-Tasks für WS-Manager, die Sie auf Ihrem lokalen System definiert haben. Sie können aber auch auf andere Systeme zugreifen, z. B. über das TCP/IP-Terminalemulationsprogramm **telnet**, und dort Verwaltungs-Tasks ausführen. In MQSeries kann dies ebenfalls als lokale Verwaltung angesehen werden, weil keine Kanäle benötigt werden, d. h., die Kommunikation wird vom Betriebssystem gesteuert.

MQSeries unterstützt die Verwaltung von einem Einzelpunkt aus. Dies wird als *Fernverwaltung* bezeichnet. Dadurch ist es Ihnen möglich, Befehle an Ihrem lokalen System auszugeben, die auf einem anderen System verarbeitet werden. Dazu müssen Sie sich nicht an dem anderen System anmelden, Sie müssen jedoch vorher die erforderlichen Kanäle definieren. Der WS-Manager und der Befehlsserver auf dem Zielsystem müssen aktiv sein. Sie können zum Beispiel einen Fernbefehl zum Ändern einer Warteschlangendefinition in einem fernen WS-Manager ausgeben.

Nicht alle Befehle können auf diese Weise ausgegeben werden. Dies gilt insbesondere für Befehle zum Erstellen und Starten von WS-Managern und zum Starten von Befehlsservern. Um diese Art von Tasks auszuführen, müssen Sie sich entweder an dem fernen System anmelden und die Befehle dort ausgeben, oder Sie müssen einen Prozess erstellen, der die Befehle an Ihrer Stelle ausgeben kann.

Clients und Server

MQSeries for Compaq NSK unterstützt Client/Server-Konfigurationen für MQI-Anwendungen. Es gibt keine MQSeries for Compaq NSK-Clients, nur einen MQSeries for Compaq NSK-Server; es besteht jedoch die Möglichkeit, Clients auf anderen Plattformen mit einem MQSeries for Compaq NSK-Server zu verbinden.

Ein *MQI-Client* ist ein Teil des MQSeries-Produkts, das auf einem System installiert wird, um MQI-Aufrufe von Anwendungen zu empfangen und an ein MQI-Serversystem weiterzuleiten. Dort werden sie dann von einem WS-Manager verarbeitet. In der Regel befinden sich Client und Server auf verschiedenen Systemen, sie können sich jedoch auch auf demselben System befinden.

Ein *MQI-Server* ist ein WS-Manager, der Queuing-Services für einen oder mehrere Clients bereitstellt. Alle MQSeries-Objekte (z. B. Warteschlangen) sind nur im WS-Managersystem, d. h. auf dem MQI-Serversystem, vorhanden. Ein Server kann auch normale lokale MQI-Anwendungen unterstützen.

Weitere Informationen hierzu finden Sie in den Handbüchern *MQSeries Intercommunication* und *MQSeries Clients*.

MQI-Anwendungen in einer Client/Server-Umgebung

MQI-Client-Anwendungen, die mit einem Server verbunden sind, können MQI-Aufrufe auf dieselbe Weise wie lokale Anwendungen ausgeben. Die Client-Anwendung gibt den Aufruf MQCONN aus, um eine Verbindung zu einem angegebenen WS-Manager herzustellen. Jeder weitere MQI-Aufruf, in dem die Verbindungskennung angegeben wird, die von der Verbindungsanforderung zurückgegeben wurde, wird dann von diesem WS-Manager verarbeitet.

Sie müssen Ihre Anwendungen mit den zugehörigen Client-Bibliotheken verbinden. Weitere Informationen hierzu finden Sie im Handbuch *MQSeries Application Programming Guide*. Für Compaq NSK steht derzeit noch kein MQI-Client zur Verfügung; da es sich bei Compaq NSK jedoch um einen MQI-Server handelt, akzeptiert er Verbindungsanforderungen von MQSeries-MQI-Clients auf anderen Plattformen.

WS-Managerfunktionen erweitern

Die von einem WS-Manager zur Verfügung gestellten Funktionen können wie folgt erweitert werden:

- durch Benutzer-Exits
- durch installierbare Services

Benutzer-Exits

Über Benutzer-Exits können Sie Ihren eigenen Programmiercode in eine WS-Managerfunktion integrieren. Zwei Arten von Benutzer-Exits werden unterstützt:

- *Kanal-Exits*

Diese Exits ändern die Funktionsweise von Kanälen. Kanal-Exits werden im Handbuch *MQSeries Intercommunication* beschrieben.

Funktionen erweitern

- *Datenkonvertierungs-Exits*
Diese Exits erstellen Quellcodefragmente, die in Anwendungsprogramme eingefügt werden können, um Daten von einem Format in ein anderes zu konvertieren. Datenkonvertierungs-Exits werden im Handbuch *MQSeries Application Programming Guide* beschrieben.
- *Cluster-Workload-Exit*
Dieser Exit kann verwendet werden, um die Kriterien zu ändern, anhand derer ein WS-Manager in einem Cluster eines von mehreren Exemplaren einer fernen Warteschlange auswählt. Informationen zur Aufrufdefinition finden Sie im Handbuch *Cluster-Unterstützung in MQSeries*.

Anmerkung: Das Verfahren, mit dem Benutzer-Exits in MQSeries aktiviert werden, hat sich in Version 5.1 geändert. Lesen Sie die Beschreibung dieses Exit-Verfahrens sorgfältig durch, um festzustellen, welche Änderungen erforderlich sind, wenn Benutzer-Exits aus älteren Versionen von MQSeries for Compaq NSK migriert werden sollen.

Weitere Informationen zu diesen Exits finden Sie in „Anhang L. Benutzer-Exits“ auf Seite 373.

Installierbare Services

Installierbare Services bieten eine umfassendere Funktionalität als Benutzer-Exits, da sie über eine formalisierte Anwendungsprogrammierschnittstelle mit mehreren Eintrittspunkten verfügen.

Eine Implementierung eines installierbaren Services wird als *Servicekomponente* bezeichnet. Sie können die im Produkt integrierten Komponenten verwenden oder eigene Komponenten erstellen, von denen die von Ihnen benötigten Funktionen ausgeführt werden. Zurzeit stehen die folgenden installierbaren Services zur Verfügung:

- **Berechtigungsservice**

Der Berechtigungsservice ermöglicht die Erstellung eigener Sicherheitsfunktionen.

Dieser Service wird vom Objektberechtigungsmanager (OAM, Object Authority Manager), einer standardmäßig mit dem Produkt verfügbaren Servicekomponente implementiert. Der OAM ist standardmäßig aktiviert. Sie können die Berechtigungsservice-Schnittstelle verwenden, um andere Komponenten zu erstellen, die den OAM ersetzen oder erweitern. Weitere Informationen zum OAM finden Sie in „Kapitel 8. MQSeries-Objekte schützen“ auf Seite 141.

- **Namensservice**

Der Namensservice ermöglicht WS-Managern die gemeinsame Benutzung von Warteschlangen, indem Anwendungen ferne Warteschlangen auf dieselbe Weise wie lokale Warteschlangen angeben können.

Sie können einen eigenen Namensservice erstellen. Weitere Informationen hierzu finden Sie im Handbuch *MQSeries Programmable System Management*.

Sicherheit

MQSeries for Compaq NSK stellt über den Objektberechtigungsmanager (OAM, Object Authority Manager) Sicherheitsfunktionen zur Verfügung.

Objektberechtigungsmanager (OAM, Object Authority Manager)

Berechtigungen für die Verwendung von MQI-Aufrufen und -Befehlen sowie für den Zugriff auf Objekte werden vom Objektberechtigungsmanager (OAM, Object Authority Manager) zur Verfügung gestellt, der standardmäßig aktiviert ist. Der Zugriff auf MQSeries-Entitäten wird über MQSeries for Compaq NSK-Principals und -Benutzergruppen sowie über den Objektberechtigungsmanager gesteuert. Die vom Objektberechtigungsmanager unterstützten Principal- und Gruppennamen werden in Namen von Compaq NSK-Benutzer- und -Gruppen aufgelöst. In Version 5.1 müssen alle MQSeries-Benutzer über einen Principal-Namen verfügen, der einem Compaq NSK-Benutzernamen zugeordnet ist, unabhängig davon, ob der Objektberechtigungsmanager aktiviert ist. Über eine Befehlszeilenschnittstelle können Sie Principals hinzufügen und löschen sowie bei Bedarf Berechtigungen erteilen und widerrufen.

Darüber hinaus kann über Compaq NSK-Sicherheitsfunktionen der Zugriff auf MQSeries-Befehle und -Datenbankdateien gesteuert werden. Ist SAFEGUARD installiert, ist MQSeries mit einigen der erweiterten Tools dieses Sicherheitssystems kompatibel und kann diese nutzen. Weitere Informationen finden Sie in „Kapitel 8. MQSeries-Objekte schützen“ auf Seite 141.

Transaktionsunterstützung

Ein Anwendungsprogramm kann eine Gruppe von Aktualisierungen zu einer *Arbeitseinheit* zusammenfassen. Diese Aktualisierungen sind in der Regel logisch miteinander verbunden und müssen alle erfolgreich ausgeführt werden, damit die Datenintegrität erhalten bleibt. Wenn eine Aktualisierung erfolgreich ist, eine andere aber nicht, geht die Datenintegrität verloren.

Wurde eine Arbeitseinheit erfolgreich abgeschlossen, wird eine **COMMIT**-Operation ausgeführt, d. h., die Aktualisierungen werden festgeschrieben. Ab diesem Zeitpunkt sind alle Aktualisierungen, die in dieser Arbeitseinheit ausgeführt wurden, permanent bzw. nicht mehr rückgängig zu machen. Kann die Arbeitseinheit nicht erfolgreich abgeschlossen werden, werden alle Aktualisierungen *zurückgesetzt*. Der Prozess, von dem Arbeitseinheiten entweder festgeschrieben (COMMIT-Operation) oder zurückgesetzt werden, heißt Synchronisationspunktkoordinierung.

In einer *lokalen Arbeitseinheit* werden nur die Ressourcen aktualisiert, die zum MQSeries-WS-Manager gehören. In diesem Fall wird die Synchronisationspunktkoordinierung vom WS-Manager selbst mit Hilfe einer einphasigen Festschreibung bereitgestellt.

Sicherheit

In einer globalen Arbeitseinheit werden auch Ressourcen aktualisiert, die zu anderen Ressourcenmanagern, z. B. XA-kompatible Datenbanken, gehören. In diesem Fall muss eine zweiphasige Festschreibung verwendet werden, und die Arbeitseinheit kann entweder vom WS-Manager selbst oder extern von einem anderen XA-kompatiblen Transaktionsmanager wie z. B. IBM® CICS®, Transarc Encina oder BEA Tuxedo koordiniert werden.

In MQSeries for Compaq NSK werden alle Arbeitseinheiten über TM/MP koordiniert oder über ein kompatibles LP-Produkt (Layered Product), das selbst TM/MP verwendet (z. B. NonStop Tuxedo in der OSS-Umgebung). Anwendungen können TM/MP für die Koordination von Arbeitseinheiten verwenden, die Aktualisierungen von Ressourcen, die mit TM/MP kompatibel sind, und MQSeries-Nachrichtenübertragungen umfassen. In der Compaq NSK-Umgebung sind keine XA-kompatiblen Datenbanken vorhanden, es ist jedoch eine ähnliche Schnittstelle implementiert. Beispielsweise kann die Aktualisierung von NonStop-SQL-Datenbanken oder ENSCRIBE-Dateien mit MQSeries-Nachrichtenübertragungen koordiniert werden, um die Datenbankintegrität sicherzustellen.

Weitere Informationen finden Sie in „Kapitel 11. Transaktionale Unterstützung und transaktionale Nachrichtenübertragung“ auf Seite 183.

Durchsatzverbesserung, Zuverlässigkeit, Skalierbarkeit und Größenanpassung

Es stehen Funktionen zur Verfügung, mit deren Hilfe MQSeries for Compaq NSK die Merkmale der Compaq NSK-Hardware und -Software genutzt werden können, die für eine skalierbare, leistungsstarke und zuverlässige Anwendungsinfrastruktur sorgen. Bei diesen Merkmalen handelt es sich unter anderem um folgende:

- Die Möglichkeit, WS-Manager und ihre Objekte CPU- und plattenübergreifend zu konfigurieren und so durch Parallelbetrieb die Systemauslastung zu verbessern.
- Softwaremerkmale, die dafür sorgen, dass beim Ausfall einer einzelnen Komponente höchstens eine kurzfristige Unterbrechung des Betriebs auftritt und eine automatische Wiederherstellung des vollständigen Systembetrieb stattfindet.
- Datenbankintegrität, die durch Software und das zu Grunde liegende transaktionale Dateisystem geschützt wird.
- Kompatibilität mit Produkten für die Fehlerbehebung.

Weitere Informationen finden Sie in „Kapitel 15. Skalierbarkeit und Systemleistung“ auf Seite 225.

Kapitel 2. MQSeries for Compaq NSK V5.1 - Architektur

In diesem Kapitel wird die Architektur von MQSeries for Compaq NSK V5.1 beschrieben. Es enthält die folgenden Abschnitte:

- „WS-Managerprozesse - Übersicht“
- „Produktbindungen“ auf Seite 25
- „Ausführbare Module“ auf Seite 26
- „Vergleich der Architekturen von Version 5.1 und Version 2.2.0.1“ auf Seite 26
- „WS-Manager — Funktionsübersicht“ auf Seite 27
- „WS-Managerprozessmodell“ auf Seite 28
- „MQSeries-Dateien und -Teildatenträger“ auf Seite 29

Diese Informationen sind unter Umständen hilfreich für eine optimale Konfiguration dieses Produkts in Ihrer Betriebsumgebung.

In Ergänzung zu den Informationen in diesem Kapitel finden Sie in „Kapitel 15. Skalierbarkeit und Systemleistung“ auf Seite 225 und „Kapitel 16. Datenintegrität und Verfügbarkeit“ auf Seite 237 spezifische Konfigurationsangaben.

WS-Managerprozesse - Übersicht

In der Abbildung unten sind die Prozesse dargestellt, aus denen sich ein MQSeries-WS-Manager zusammensetzt (aus Gründen der Vollständigkeit wird der PATHMON-Prozess gezeigt, da der WS-Manager unter PATHWAY läuft). Die Abbildung enthält außerdem eine Übersicht über die Prozessinteraktionen über IPC.

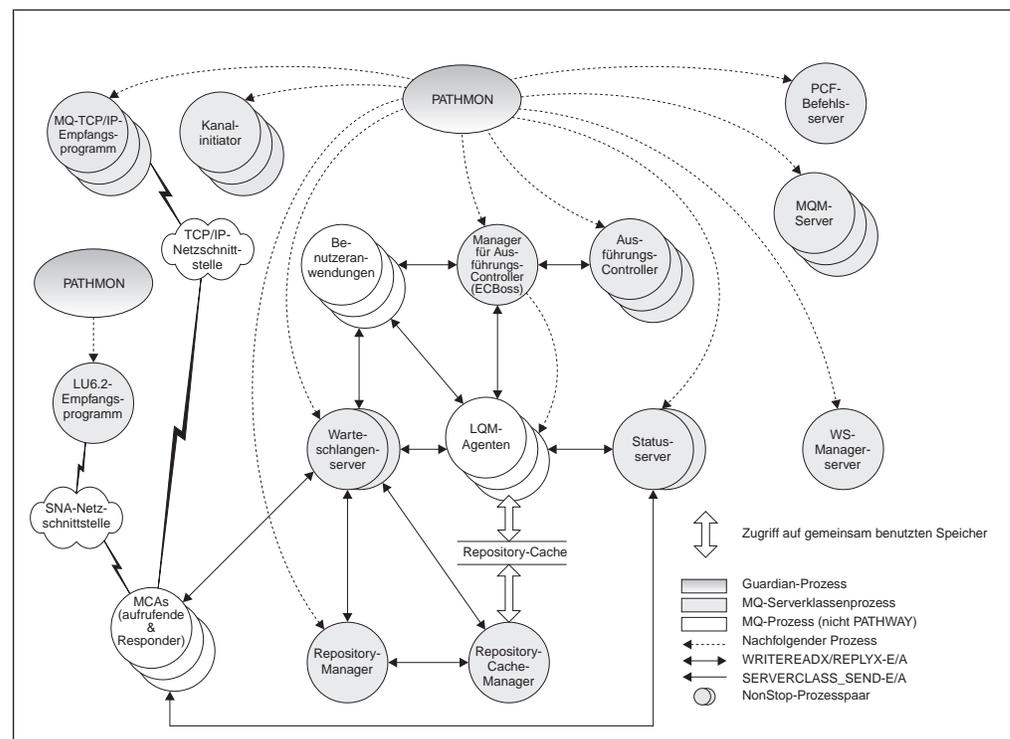


Abbildung 1. MQSeries for Compaq NSK - Prozesse

Beim Start eines WS-Managers erstellt PATHWAY die wichtigsten WS-Managerprozesse. In der Standardkonfiguration startet PATHMON den Manager der Ausführungs-Controller (ECBoss, Execution Controller Boss), die Ausführungs-Controller, die Status- und Warteschlangenserver und den Repository-Manager.

Der ECBoss bearbeitet alle eingehenden MQCONN-Anforderungen und verteilt sie auf die verfügbaren Ausführungs-Controller; dabei sorgt er für eine gleichmäßige Systemauslastung, indem er immer den Ausführungs-Controller mit der geringsten Auslastung auswählt.

Die Ausführungs-Controller steuern und überwachen die anderen WS-Managerprozesse und MQI-Anwendungen.

Die lokalen WS-Manageragenten (LQMA, Local Queue Manager Agents) führen die Operationen aus, die zur Verarbeitung der von den Anwendungen ausgegebenen MQI-Aufrufe erforderlich sind. Diese Agenten führen den größten Teil des Codes aus, der die MQI unterstützt. Die Agenten sollen hauptsächlich die Anwendungsprogramme von den kritischen Ressourcen des WS-Managers trennen und dienen so als Schutzschild vor unerwünschten Anwendungen oder Anwendungen, von denen ein Ressourcenmissbrauch zu befürchten ist.

Die Anzahl der Agentenprozesse hängt von der Anzahl der verbundenen Anwendungen ab und davon, wie viele Agentenprozesse gleichzeitig inaktiv sein können (wird über **MinIdleLQMAgents** in der Datei QMINI angegeben). Beim Start des WS-Managers startet jeder Ausführungs-Controller die angegebene Anzahl an Leerlaufagenten. Sobald ein Agent aktiviert wird, startet der Ausführungs-Controller einen weiteren Agenten, damit immer die Mindestanzahl an Leerlaufagenten verfügbar ist.

Eine wichtige Funktion in der Version 5.1 MQSeries-Architektur ist der *Warteschlangenserver*. Bei diesem handelt es sich um ein NonStop-Prozesspaar, das sämtliche Messaging-Operationen für lokale Warteschlangen unterstützt. Bei seiner erstmaligen Erstellung verfügt ein WS-Manager über einen einzigen Standardwarteschlangenserver. Je nach Systemkonfiguration und Leistungsanforderungen können Sie zusätzliche Warteschlangenserver konfigurieren und diesen lokale Warteschlangen zuordnen.

Der Statusserver handhabt die Statusinformationen für alle Objekte, bei denen es sich nicht um lokale Warteschlangen handelt. Dieser Server übernimmt die Funktionen des MQSS-Servers aus der Vorgängerversion 2.2.0.1. Darüber hinaus handhabt der Standardstatusserver auch noch Kanalstatusinformationen für den WS-Manager. Bei seiner erstmaligen Erstellung verfügt der WS-Manager über einen einzigen Standardstatusserver. Je nach Systemkonfiguration und Leistungsanforderungen können Sie zusätzliche Statusserver konfigurieren und diesen MQSeries-Objekte zuordnen.

Beim Start des WS-Managers wird automatisch die Kanalinitiator-Serverklasse aktiviert. Dadurch wird ein Clusterbetrieb ermöglicht, ohne dass der Kanalinitiator manuell gestartet werden muss.

Neu in Version 5.1 sind der *Repository-Managerprozess* und der *Repository-Cache-Serverprozess* für die Verarbeitung von Cluster-Warteschlangen. In jeder CPU, in der ein Ausführungs-Controller aktiv ist, ist auch ein Repository-Managerprozess aktiv. Der Prozess, der zuerst gestartet wird, übernimmt die Funktion des Repository-Managers, der die Repository-Aktivitäten im WS-Manager koordiniert. Die anschließend gestarteten Prozesse verwalten ein gemeinsam benutztes Speichersegment, das den Repository-Cache der CPU enthält, in dem die Prozesse ausgeführt werden. Es kann nur jeweils einen Repository-Manager pro CPU geben. Der Repository-Cache enthält Informationen über die MQSeries-Objekte im Cluster, wie z. B. Warteschlangen sowie weitere WS-Manager im Cluster.

Neu in Version 5.1 ist auch der *WS-Managerserver*. Dieser Server handhabt abgelaufene Nachrichten und die Berichterstellung.

Die Nachrichtenkanalagenten (MCA, Message Channel Agents) übertragen Nachrichten an bzw. von anderen WS-Managern. Um einen effizienteren Betrieb zu ermöglichen, werden in MQSeries for Compaq NSK V5.1 für die Nachrichtenkanalagenten FASTPATH-Bindungen verwendet. Weitere Informationen zu FASTPATH-Anwendungen finden Sie unter „FASTPATH-Bindungen für Anwendungsprogramme“ auf Seite 234.

Das LU6.2-Empfangsprogramm, das in einer eigenen PATHWAY-Umgebung ausgeführt wird, ist neu in Version 5.1. Ausführliche Informationen zum Betrieb und zur Konfiguration des Empfangsprogramms und dessen Umgebung finden Sie in „Anhang M. Kommunikation einrichten“ auf Seite 383.

Produktbindungen

MQSeries for Compaq NSK V5.1 ermöglicht drei Bindungsarten, nämlich das native dynamische, das native statische und das nicht native Binden.

Für das native dynamische Binden stellt MQSeries eine native gemeinsam benutzte Ressourcenbibliothek (MQSRLLIB) zur Verfügung. Eines der Ziele in MQSeries for Compaq NSK V5.1 war, möglichst große Teile des Produktcodes in der gemeinsam benutzten Ressourcenbibliothek (SRL, Shared Resource Library) unterzubringen, um den Umfang des Produkts zu verringern. Für MQSeries wird das native dynamische Binden bevorzugt, da es eine effizientere Nutzung der Systemressourcen ermöglicht. In der OSS-Umgebung wird nur diese Bindungsart unterstützt.

Für das native statische Binden stellt MQSeries for Compaq NSK eine Bibliothek zur Verfügung, die erneut gebunden werden kann (MQSRLLNK). Diese ist für Benutzer gedacht, die über eine eigene SRL verfügen und daher nicht die SRL in MQSeries verwenden können.

Aus Gründen der Kompatibilität mit traditionellen Anwendungen, für die keine nativen Bindungen möglich sind, stellt MQSeries eine nicht native, statische Bibliothek zur Verfügung.

Ausführbare Module

In der folgenden Tabelle sind die ausführbaren MQSeries-Module aufgeführt:

Tabelle 1. Ausführbare MQSeries-Module

Ausführbares MQSeries-Modul	Name
Ausführungs-Controller (EC, Execution Controller)	MQEC
Manager der Ausführungs-Controller (ECBoss, Execution Controller Boss)	MQECBOSS
Lokaler WS-Manager Agent (LMQA, Local Queue Manager Agent)	MQLQMAG
Statusserver	MQSTSVR
Warteschlangenserver	MQQSSVR
Repository-Manager	MQREPSVR
WS-Managerserver	MQMGRSVR
Befehlsserver	MQCMDSVR
MQSeries-Verwaltungsserver	MQMSVR
Kanalinitiator	RUNMQCHI
Aufrufender MCA (Caller MCA)	MQMCACAL
TCP/IP-Responder-MCA	MQTCPRES
LU6.2-Responder-MCA	MQLU6RES
Auslösemonitor	RUNMQTRM

Vergleich der Architekturen von Version 5.1 und Version 2.2.0.1

Ein grundlegender struktureller Unterschied zwischen MQSeries Version 2.2.0.1 und Version 5.1 ist die Einführung von Warteschlangenservern. Jeder Warteschlangenserver verwaltet die Daten und Dateien, die einem oder mehreren lokalen Warteschlangenobjekten zugeordnet sind. Bei seiner erstmaligen Erstellung verfügt ein WS-Manager über einen einzigen Standardwarteschlangenserver, der für alle lokalen Warteschlangen zuständig ist, die dem WS-Manager zugeordnet sind. Sie können zusätzliche Warteschlangenserver (unter Verwendung von PATHWAY) hinzufügen und diesen mit dem Befehl **altmqfls** Warteschlangen zuordnen.

Bei den Warteschlangenservern handelt es sich um NonStop-Prozesspaare, die alle Messaging-Operationen für lokale Warteschlangen unterstützen. Der Warteschlangenserver unterstützt außerdem auch nicht permanente, rein speicherbasierte Nachrichten und weist daher, was die Leistung angeht, deutliche Vorteile gegenüber Version 2.2.0.1 auf. MQSeries verwendet eine interne Schnittstelle zu TM/MP, über die diese speicherbasierten Nachrichten in die Transaktionen integriert werden; dies ermöglicht die Integration von permanenten und nicht permanenten Nachrichten innerhalb derselben Transaktionen, ohne dass die nicht permanenten Nachrichten auf Platte geschrieben werden müssen und damit Speicherplatz erforderlich ist. Die Warteschlangenserver übernehmen nur für lokale Warteschlangen die Funktion der MQSS-Server aus Version 2.2.0.1.

Die MQSS-Server wurden durch Statusserver ersetzt. Bei diesen Statusservern handelt es sich um NonStop-Prozesspaare, die die Statusinformationen für Objekte anderer lokaler Warteschlangen verwalten.

Eine zweite grundlegende Änderung an der Architektur ist die Einführung gemeinsam benutzter Speichersegmente. Die Initialisierungsdatei des WS-Managers wird über den WS-Manager verteilt, wobei nur der gemeinsam benutzte Speicher belegt wird, der schreibgeschützt ist. Dadurch ergibt sich eine Leistungsverbesserung, wenn eine Verbindung zu einem WS-Manager hergestellt wird. Der MQSeries-Repository-Cache, der zur Implementierung der Clusterfunktionen verwendet wird, verwendet in jeder CPU, in der MQSeries-Prozesse ausgeführt werden, ein gemeinsam benutztes Speichersegment, auf das Lese- und Schreibzugriff besteht.

Version 5.1 bietet mehr transparente Unterstützung für LU6.2-Kanäle. Die neue Version enthält ein LU6.2-Empfangsprogramm, das für den Start von LU6.2-Responder zuständig ist. Daher wurde das Attribut AUTOSTART für LU6.2-Kanäle entfernt.

WS-Manager — Funktionsübersicht

Im Folgenden die wichtigsten Komponenten eines WS-Managers:

Anwendungsschnittstelle

Stellt die Umgebung und Verfahren für die Ausführung von MQI-Aufrufen zur Verfügung.

WS-Manager-Kernel

Stellt die meisten MQI-Funktionen zur Verfügung, wie beispielsweise die Auslösefunktion.

Objektberechtigungsmanager (OAM, Object Authority Manager)

Ermöglicht die Zugriffssteuerung für den WS-Manager und dessen Ressourcen. Diese Komponente ermöglicht es, die Aktionen festzulegen, die Benutzer und Gruppen für bestimmte Ressourcen ausführen können.

Data Abstraction and Persistence (DAP)

Ermöglicht das Speichern und Wiederherstellen von Daten im WS-Manager. Diese Komponente enthält die Nachrichten.

Nachrichtenkanalagenten (MCA, Message Channel Agents)

Bei diesen handelt es sich um besondere Anwendungen, die die meisten Aktionen über die MQI ausführen. Sie sind für den zuverlässigen Nachrichtenaustausch zwischen WS-Managern zuständig. Für die Nachrichtenkanalagenten werden FASTPATH-Bindungen verwendet.

Befehlsserver

Bei dem Befehlsserver handelt es sich um eine besondere MQI-Anwendung, die für die Verarbeitung von Nachrichten zuständig ist, die Verwaltungsbefehle für den WS-Manager enthalten.

Allgemeine Services

Diese schirmen den restlichen WS-Managercode gegen das Betriebssystem ab. Dabei handelt es sich um eine Gruppe von Services, die ähnliche Funktionen wie ein Betriebssystem erfüllen (z. B. Speicherverwaltung, Serialisierung und Prozessverwaltung).

In Abb. 2 ist die Abhängigkeiten zwischen den verschiedenen Komponenten dargestellt.

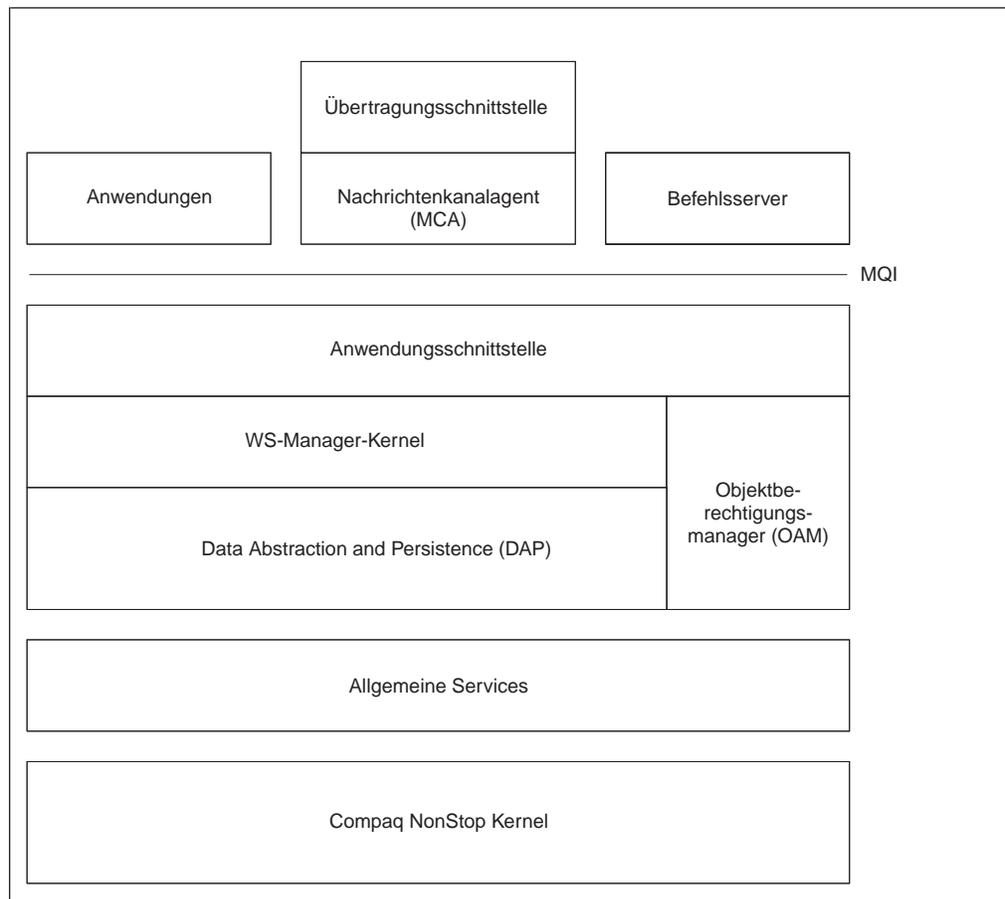


Abbildung 2. Komponenten von MQSeries for Compaq NSK V5.1

WS-Managerprozessmodell

Wenn eine Anwendung eine Verbindung zu einem Agenten benötigt, wendet sie sich an ECBoss, den Manager der Ausführungs-Controller. Dieser wählt den Ausführungs-Controller aus, der die geringste Auslastung aufweist (d. h., mit dem momentan nur wenige Anwendungen verbunden sind) und leitet die Verbindungsanforderung an diesen weiter. Der Ausführungs-Controller wiederum wählt einen lokalen WS-Manageragenten aus, der gerade nicht aktiv ist. Anschließend sendet der Ausführungs-Controller über den ECBoss eine Antwort an die Anwendung, die daraufhin eine Verbindung zu dem ausgewählten lokalen WS-Manageragenten herstellt.

Die Anwendungsschnittstelle setzt sich aus zwei Teilen zusammen:

- Der MQI-Anwendungs-Stub packt MQ-Anforderungen und übergibt sie über die Komponente für prozessübergreifende Kommunikation (IPCC, Inter-Process Communication Component) an den Agentenprozess.
- Die IPCC-Komponente stellt eine Schnittstelle für den Nachrichtenaustausch zwischen MQI-Anwendungen, den Agenten, den Ausführungs-Controllern und dem Manager der Ausführungs-Controller zur Verfügung. Die MQSeries-Komponente IPCC basiert auf standardmäßigen GUARDIAN-IPCs.

Die Anwendung kommuniziert über die IPCC-Komponente mit ihrem Agentenprozess. Dieser Agentenprozess führt die MQI-Aufrufe für die Anwendung aus. Die Kommunikation zwischen der Anwendung und dem Agenten über die IPCC-Komponente erfolgt über synchrone Anforderungs- und Antwortnachrichten. Neben der Kommunikation mit dem Agenten überträgt die Anwendung bei der Ausführung von MQPUT- und MQPUT-Vorgängen auch noch Nachrichtendaten direkt an den Warteschlangenserver, dem die Warteschlange zugeordnet ist.

Für Anwendungen, die über FASTPATH gebunden sind, wird der MQSeries-Code direkt mit der Benutzeranwendung verbunden. Dies hat Vorteile für die Leistung, allerdings besteht die Gefahr, dass unerwünschte Anwendungen den Betrieb des WS-Manager unterbrechen und Datenverluste oder andere Probleme verursachen könnten.

MQSeries-Dateien und -Teildatenträger

MQSeries verwendet eine Reihe von ENSCRIBE-Dateien von Compaq NonStop. Position und Namen dieser Dateien sind unten aufgeführt.

Die einem MQSeries-WS-Manager zugeordneten Dateien werden auf eine Reihe von Teildatenträgern verteilt.

<WS-Manager> D

Teildatenträger für Daten. Wird für Dateien verwendet, die allgemeine WS-Managerinformationen enthalten.

<WS-Manager> M

Teildatenträger für Nachrichtendaten. Wird für Dateien mit Nachrichtendaten verwendet.

<WS-Manager> L

Fehlerprotokolle.

<WS-Manager> S

Kanalsynchronisationsdateien.

<WS-Manager>

FFST™-Dateien.

Objektkatalog

Der Objektkatalog (OBJCAT) enthält Informationen zu MQSeries-Objekten. Er befindet sich im Teildatenträger für WS-Managerdaten.

Warteschlangendateien

Für jede lokale Warteschlange ist eine Warteschlangendatei (Queue), eine Datei für Warteschlangenüberlauf (Queue Overflow) sowie eine Änderungsdatei (Touch) vorhanden. Die Dateien haben das Präfix Q (Queue), O (Queue Overflow) oder T (Touch), bei den restlichen Zeichen handelt es sich entweder um einen Teil des MQSeries-Objektnamens oder um einen vom System generierten Namen. Über das Dienstprogramm **dsqmqls** können Sie die einer bestimmten Warteschlange zugeordneten Dateien ermitteln. Bei der Erstellung einer Warteschlange erstellt der WS-Manager auch die entsprechenden Dateien im Teildatenträger mit den Nachrichtendaten des WS-Managers (<qmgr>M). Über **altmqfls** können diese Dateien in einen anderen Datenträger verschoben werden, falls dies aus Leistungs- oder Platzgründen erforderlich sein sollte.

Der WS-Manager erstellt Nachrichtenüberlaufdateien für alle Nachrichten, deren Größe über dem Schwellenwert für Nachrichtenüberlauf liegt, der für die Warteschlange definiert wurde. Nachrichtenüberlaufdateien werden standardmäßig im Teildatenträger des Warteschlangenservers gespeichert. Der Datenträger, in dem der Warteschlangenserver die Nachrichtenüberlaufdateien erstellt, kann mit dem Dienstprogramm **altmqfls** geändert werden. Bei der Nachrichtenüberlaufdatei handelt es sich um eine unstrukturierte, nicht überprüfte Datei, die einer bestimmten Nachricht zugeordnet ist (weitere Informationen zur Konfiguration von Warteschlangendateien finden Sie unter „Warteschlangenserver und Warteschlangendateien“ auf Seite 228). Um eine effiziente Datenübertragung sicherzustellen, werden die Daten im großen Übertragungsmodus (SETMODE 141) in die Nachrichtenüberlaufdatei geschrieben; dieser Modus ermöglicht die direkte Übertragung aus dem Speicher des Anwendungsprozesses in 56-KB-Segmenten.

Bei permanenten Nachrichten, deren Größe unter 3 KB liegt, werden die Nachrichtendaten in der Warteschlangendatei gespeichert. Bei permanenten Nachrichten, deren Größe über 3 KB, aber unter dem für die Warteschlange definierten Schwellenwert für den Nachrichtenüberlauf liegt, schreibt der Warteschlangenserver zusätzliche Datensätze in die Warteschlangenüberlaufdatei. Bei permanenten Nachrichten, deren Größe über dem Schwellenwert für Nachrichtenüberlauf liegt, erstellt der Warteschlangenserver eine Nachrichtenüberlaufdatei, in der die Nachricht gespeichert wird.

Wie Sie Warteschlangen oder Warteschlangenüberlaufdateien aus Platz- oder Leistungsgründen partitionieren, erfahren Sie unter „Warteschlangendateien partitionieren“ auf Seite 230. Bei einer Partitionierung dieser Dateien verteilt MQSeries die Nachrichtendaten gleichmäßig auf die verschiedenen Partitionen, um eine optimale Leistung sicherzustellen.

Weitere Informationen zu den Möglichkeiten einer Leistungsoptimierung in Verbindung mit Warteschlangen und Warteschlangendateien finden Sie in „Kapitel 15. Skalierbarkeit und Systemleistung“ auf Seite 225.

Änderungsdateien für Aliaswarteschlangen und ferne Warteschlangen

Aliaswarteschlangen und fernen Warteschlangen ist eine Änderungsdatei (Touch) zugeordnet. Der Name dieser Datei beginnt mit dem Präfix T; die Datei befindet sich im Teildatenträger mit den WS-Managerdaten.

Namenslistendateien

Jeder Namensliste ist eine unstrukturierte Datei zugeordnet. Der Name dieser Dateien beginnt mit dem Präfix L, die Datei befindet sich im Teildatenträger mit den WS-Managerdaten. Vervollständigt wird der Name, indem entweder ein Teil des MQSeries-Objektnamens oder ein vom System generierter Name übernommen wird. Über das Dienstprogramm **dspmqls** können Sie die einer bestimmten Namensliste zugeordnete Datei ermitteln.

Kapitel 3. MQSeries-Befehlssätze verwenden

In diesem Kapitel werden die Befehle beschrieben, die für Systemverwaltungstasks für MQSeries-Objekte verwendet werden können. Zu diesen Tasks gehört das Erstellen, Starten, Ändern, Anzeigen, Stoppen und Löschen von WS-Managern, Warteschlangen, Prozessen und Kanälen. Diese Tasks werden jeweils mit bestimmten Befehlen ausgeführt.

In MQSeries for Compaq NSK V5.1 stehen für die Ausführung von Verwaltungstasks die folgenden Verwaltungsbefehlssätze zur Verfügung:

- MQSC-Befehle (MQSeries-Befehle)
- PCF-Befehle (PCF, Programmable Command Format)
- Steuerbefehle
- MQAI (MQSeries Administrator Interface)

Darüber hinaus steht noch Folgendes zur Verfügung:

- Einige TS/MP-Befehle (PATHWAY-Befehle) für Verwaltungszwecke.
- Die MQM-Funktion (Message Queue Management) unterstützt einige Verwaltungstasks. Eine Beschreibung dieser Funktion finden Sie unter „MQM (Message Queue Management) verwenden“ auf Seite 73.

Dieses Kapitel enthält eine Einführung in die MQSC-, PCF- und Steuerbefehlssätze sowie eine Übersicht über die Funktionen, die von den in „Anhang D. Befehlssätze - Übersicht“ auf Seite 333 aufgeführten Befehlssätzen unterstützt werden. Die Verwendung der TS/MP-Befehle wird unter „TS/MP-Verwaltung (PATHWAY-Verwaltung)“ auf Seite 34 beschrieben.

Verwaltungs-Tasks über Steuerbefehle ausführen

Folgende Steuerbefehle stehen zur Verfügung:

- Befehle zum Erstellen, Starten, Stoppen und Löschen von WS-Managern.
- Befehle zum Starten, Stoppen und Anzeigen von Befehlsservern.
- Befehle, mit denen Dienstprogramme aufgerufen werden, z. B. zum Ausführen von MQSC-Befehlen, für die Zugriffssteuerung auf MQSeries-Objekte, zum Starten bzw. Stoppen eines MQSeries-Trace oder zum Ausführen von Auslösemonitoren.

Steuerbefehle verwenden

Steuerbefehle können über die Compaq TACL-Eingabeaufforderung eingegeben werden. Für den Befehlsnamen muss die Groß-/Kleinschreibung nicht beachtet werden. Allerdings sollten Sie berücksichtigen, dass bei WS-Managernamen die Groß-/Kleinschreibung *beachtet werden muss*. Beispiel:

```
runmqsc
```

Eine Beschreibung der Syntax und der Funktion der einzelnen Befehle finden Sie in „Kapitel 17. MQSeries-Steuerbefehle“ auf Seite 257.

Verwaltungs-Tasks über MQSC-Befehle ausführen

MQSC-Befehle (MQSeries Commands) werden für die Verwaltung von WS-Managerobjekten verwendet; dazu gehören der WS-Manager, Kanäle, Warteschlangen und Prozessdefinitionen. Beispielsweise kann eine Warteschlange mit MQSC-Befehlen definiert, geändert, angezeigt und gelöscht werden.

Bei der Anzeige einer Warteschlange über den Befehl `DISPLAY QUEUE` werden die *Attribute* dieser Warteschlange abgerufen. Das Attribut `MAXMSGL` gibt beispielsweise die maximale Länge an, die eine Nachricht in dieser Warteschlange haben darf. Die Nachrichten, die sich in dieser Warteschlange befinden, können mit diesem Befehl nicht angezeigt werden. Eine Übersicht über MQSC-Befehle finden Sie in „Anhang D. Befehlssätze - Übersicht“ auf Seite 333. Ausführliche Informationen zu den einzelnen MQSC-Befehlen finden Sie im Handbuch *MQSeries MQSC - Befehle*.

MQSC-Befehle ausführen

MQSC-Befehlen können entweder interaktiv über die Eingabe des Steuerbefehls `runmqsc` an der TACL-Eingabeaufforderung von Compaq ausgeführt werden oder über die Ausführung eines Scripts, wenn ein lokaler WS-Manager aktiv ist. Für den Befehl `runmqsc` sind drei Ausführungsmodi möglich, abhängig von den im Befehl angegebenen Optionen:

- der *Prüfmodus*, bei dem MQSC-Befehle im lokalen WS-Manager überprüft, aber nicht ausgeführt werden,
- der *direkte Modus*, bei dem MQSC-Befehle in einem lokalen WS-Manager ausgeführt werden,
- der *indirekte Modus*, bei dem MQSC-Befehle in einem fernen WS-Manager ausgeführt werden.

Weitere Informationen zur Verwendung der MQSC-Befehle und -Textdateien finden Sie unter „Die MQSC-Funktion interaktiv verwenden“ auf Seite 97. Weitere Informationen zum Befehl `runmqsc` finden Sie unter „runmqsc (MQSeries-Befehle ausführen)“ auf Seite 302. Eine Zusammenfassung der MQSC-Befehle finden Sie in „Anhang D. Befehlssätze - Übersicht“ auf Seite 333.

Compaq NSK Fix Command verwenden

In der interaktiven Ausführung von `runmqsc` über die Compaq NSK-TACL-Eingabeaufforderung können Sie auch die Funktion Compaq NSK Fix Command verwenden, mit MQSC-Befehle abgerufen und bearbeitet werden können. Beispiel:

- Bei Eingabe von `history` oder `h` wird eine Liste mit den zehn Befehlen angezeigt, die zuletzt ausgeführt wurden.
- Bei Eingabe von `!n` (dabei steht *n* für die Befehlsnummer) wird der betreffende Befehl erneut ausgeführt.
- Bei Eingabe von `h n` oder `history n` (dabei steht *n* für eine Zahl) werden die *n* zuletzt eingegebenen Befehle angezeigt.
- Bei Eingabe von `fc` wird der zuletzt eingegebene Befehl angezeigt, der dann bearbeitet werden kann. Bei Eingabe von `fc n` (dabei steht *n* für die Befehlsnummer) wird der betreffende Befehl angezeigt und kann bearbeitet werden. Bei Eingabe von `fc Zeichenfolge` (dabei steht *Zeichenfolge* für den zuletzt eingegebenen Befehl oder den ersten Teil davon) wird der zuletzt eingegebene Befehl dieses Namens zur Bearbeitung angezeigt. Die Syntax entspricht dem NSK-Standard. Geben Sie beispielsweise `d` (delete) an, um ein Zeichen zu löschen, `i` (insert), um ein Zeichen einzufügen oder `r` (replace), um ein Zeichen zu ersetzen.

Verwaltungs-Tasks über PCF-Befehle ausführen

Mit PCF-Befehlen können Sie Verwaltungs-Tasks in Ihre Anwendungen oder in ein Verwaltungsprogramm integrieren. PCF-Befehle decken denselben Funktionsbereich wie MQSC-Befehle ab. Sie können ein Programm schreiben, das von einem einzelnen Knoten aus PCF-Befehle an einen beliebigen WS-Manager ausgibt. Darüber hinaus können Sie Verwaltungs-Tasks zentralisieren und automatisieren.

Jeder PCF-Befehl besteht aus einer Datenstruktur, die in den Bereich mit den Anwendungsdaten innerhalb einer MQSeries-Nachricht eingebettet wird. Jeder Befehl wird über die MQI-Funktion MQPUT an den Ziel-WS-Manager gesendet. Der Befehlsserver des WS-Managers, der die Nachricht empfängt, interpretiert sie als Befehlsnachricht und führt den Befehl aus. Um die Antworten abzurufen, gibt die Anwendung einen MQGET-Aufruf aus; daraufhin werden die Antwortdaten in Form einer Datenstruktur im Bereich mit den Anwendungsdaten in der MQSeries-Nachricht zurückgegeben. Die Anwendung kann die Antwort dann verarbeiten und entsprechend reagieren.

Anmerkung: Anders als MQSC-Befehle besitzen PCF-Befehle und deren Antworten kein lesbares Textformat.

Für die Erstellung einer PCF-Befehlsnachricht müssen Sie Folgendes angeben:

Nachrichtendeskriptor

Hierbei handelt es sich um einen standardmäßigen MQSeries-Nachrichtendeskriptor mit den folgenden Angaben:

Der Nachrichtenart (*MsgType*): MQMT_REQUEST.

Dem Nachrichtenformat (*Format*): MQFMT_ADMIN.

Anwendungsdaten

Sie enthalten die PCF-Nachricht einschließlich PCF-Header mit folgenden Angaben:

Der PCF-Nachrichtenart (*Type*): MQCFT_COMMAND.

Der Befehls-ID, die den Befehl angibt, z. B. *ChangeQueue* (MQCMD_CHANGE_Q).

Eine vollständige Beschreibung der PCF-Datenstrukturen sowie Anweisungen zu deren Implementierung finden Sie im Handbuch *MQSeries Programmable System Management*.

Attribute in MQSC- und PCF-Befehlen

Die in MQSC-Befehlen angegebenen Objektattribute werden in diesem Handbuch in Großbuchstaben angegeben (z. B. RQMNAME), obwohl bei ihnen die Groß-/Kleinschreibung nicht beachtet werden muss. Diese Attributnamen dürfen maximal acht Zeichen enthalten (z. B. QDPHIEV). Objektattribute in PCF-Befehlen werden kursiv geschrieben; die Länge ihrer Namen ist nicht auf acht Zeichen begrenzt. Der entsprechende PCF-Befehl für den MQSC-Befehl RQMNAME ist *RemoteQMgrName*, der entsprechende PCF-Befehl für QDPHIEV ist *QDepthHighEvent*.

PCF-Escape-Befehle

PCF-Escape-Befehle sind PCF-Befehle, deren Nachrichtentext MQSC-Befehle enthalten. Mit PCF-Befehlen können Sie Befehle an einen fernen WS-Manager senden. Weitere Informationen zu PCF-Escape-Befehlen finden Sie im Handbuch *MQSeries Programmable System Management*.

mit PCF-Befehlen

Bei MQAI (MQSeries Administration Interface) handelt es sich um eine Verwaltungsschnittstelle für MQSeries, die jetzt auf der Compaq NSK-Plattform zur Verfügung steht. Sie erleichtert die Verwaltung von MQSeries; da keine eigenen PCF-Nachrichten geschrieben werden müssen, werden Probleme, die in Zusammenhang mit komplexen Datenstrukturen auftreten, vermieden. Weitere Informationen zu MQAI finden Sie unter „Verwendung von PCF-Befehlen mit Hilfe von MQAI vereinfachen“ auf Seite 122.

TS/MP-Verwaltung (PATHWAY-Verwaltung)

Die meisten Operationen für WS-Manager werden über MQSeries-Steuerbefehle ausgeführt, die an der TACL-Eingabeaufforderung eingegeben werden. Bei einigen Operationen ist allerdings PATHWAY erforderlich, damit sie direkt für die TS/MP-Serverklassen ausgeführt werden. Darüber hinaus müssen unter Umständen einige Verwaltungs-Task auf Grund von Änderungen an der Systemkonfiguration an der TS/MP-Konfiguration selbst ausgeführt werden.

Dieser Abschnitt enthält eine Zusammenfassung dieser Tasks.

TCP/IP-Empfangsprogramme konfigurieren und steuern

TCP/IP-Empfangsprogramme können mit dem MQSeries-Steuerbefehl **runmqclsr** oder den PATHCOM-Befehlen **THAW SERVER** und **START SERVER** gestartet werden (eine Beschreibung des MQSC-Befehls finden Sie unter „runmqclsr (Empfangsprogramm ausführen)“ auf Seite 300). Gestoppt werden TCP/IP-Empfangsprogramme mit den PATHCOM-Befehlen **FREEZE SERVER** und **STOP SERVER**. Mit den PATHCOM-Befehlen **STATUS SERVER** können Sie die Anzahl der momentan aktiven TCP/IP-Empfangsprogramme und die Namen ihrer Prozesse anzeigen.

Standardmäßig verfügt jeder WS-Manager über ein Empfangsprogramm der Serverklasse **MQS-TCPLIS00**. Mit dem PATHCOM-Befehl **ADD SERVER** können Sie zusätzliche Serverklassen für TCP/IP-Empfangsprogramme erstellen, die an mehreren TCP/IP-Anschlüssen eingesetzt werden können. Um eine maximale Flexibilität zu gewährleisten, sollte jedes TCP/IP-Empfangsprogramm in einer eigenen TS/MP-Serverklasse konfiguriert werden. Werden zusätzliche TCP/IP-Empfangsprogramme erstellt, müssen Sie außerdem der Initialisierungsdatei des WS-Managers (**QMINI**) wie unter „Anzahl an TCP/IP-Ports, die vom WS-Manager überwacht werden“ auf Seite 57 beschriebenen TCP/IP-Anschlussdefinitionen hinzufügen. Das Empfangsprogramm, das zuerst gestartet wird, verwendet den ersten in der Datei **QMINI** definierten Anschluss, das nächste Empfangsprogramm verwendet den zweiten Anschluss usw. Ein Beispiel in Abschnitt „**QMINI** für die Unterstützung mehrerer TCP/IP-Empfangsprogramme konfigurieren“ auf Seite 401 zeigt die **QMINI**-Einträge, die bei Verwendung mehrerer Empfangsprogramme erforderlich sind.

Steuerung des Befehlsservers

Der Befehlsserver wird in der TS/MP-Serverklasse **MQS-CMDSERV00** erstellt. Anstelle der Steuerbefehle **strmqcsv**, **endmqcsv** und **dspmqcvs** können Sie auch die PATHCOM-Befehle **THAW SERVER**, **START SERVER**, **FREEZE SERVER**, **STOP SERVER** und **STATUS SERVER** verwenden.

Kanalinitiatoren konfigurieren und steuern

Der Standardkanalinitiator wird in der TS/MP-Serverklasse MQS-CHANINIT00 erstellt. Anstelle des unter „runmqchi (Kanalinitiator ausführen)“ auf Seite 296 beschriebenen Steuerbefehls **runmqchi** können Sie auch die PATHCOM-Befehle THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER und STATUS SERVER verwenden, um den Kanalinitiatorstatus zu steuern und abzurufen. Der Standardkanalinitiator ist für die standardmäßige Kanalinitialisierungswarteschlange SYSTEM.CHANNEL.INITQ zuständig.

Standardinitialisierungswarteschlange für Kanalinitiator ändern

In Version 5.1 wird der Standardkanalinitiator automatisch gestartet. Er ist für die Unterstützung von Clusteroperationen notwendig.

Soll eine andere als die dem Kanalinitiator standardmäßig zugeordnete Kanalinitialisierungswarteschlange (SYSTEM.CHANNEL.INITQ) verwendet werden, müssen Sie die PATHWAY-Konfiguration ändern.

Anmerkung: Gehört der WS-Manager zu einem Cluster, sollte die Initialisierungswarteschlange nicht geändert werden. Wenn Sie dem Standardkanalinitiator eine andere als die standardmäßig vorgegebene Initialisierungswarteschlange zuordnen, wird die Clusterunterstützung inaktiviert.

Die standardmäßige Initialisierungswarteschlange kann geändert werden, während der WS-Manager aktiv, die Serverklasse des Kanalinitiators muss jedoch zunächst gestoppt werden. Geben Sie für die PATHWAY-Konfiguration des WS-Managers den folgenden PATHCOM-Befehl ein:

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

Dabei steht <Initialisierungswarteschlange> für den Namen der Initialisierungswarteschlange, die anstelle der Standardwarteschlange verwendet werden soll. Anschließend können Sie den Kanalinitiator starten und PATHCOM beenden.

Auslösemonitore konfigurieren und steuern

In der TS/MP-Serverklasse MQS-TRIGMON00 wird ein einziger Standardauslösemonitor erstellt. Diese Serverklasse können Sie mit den PATHCOM-Befehlen THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER und STATUS SERVER steuern. Wenn zusätzliche Auslösemonitore erforderlich sind, können diese als zusätzliche Serverklassen auf der Basis der Serverklasse MQS-TRIGMON00 konfiguriert werden. Um eine maximale Flexibilität zu gewährleisten sollten Sie für jeden Auslösemonitor eine eigene Serverklasse erstellen. Die Steuerung der Auslösemonitore muss nicht über TS/MP erfolgen. Sie können Auslösemonitore auch starten, indem Sie an der TACL-Eingabeaufforderung den Steuerbefehl **runmqtrm** eingeben.

Dem Standardauslösemonitor ist standardmäßig die Initialisierungswarteschlange SYSTEM.DEFAULT.INITIATION.QUEUE zugeordnet. Sie können stattdessen eine andere Warteschlange zuordnen, indem Sie das Attribut STARTUP für die Serverklasse, die den Auslösemonitor enthält, hinzufügen oder ändern. Dies ist erforderlich, wenn mehrere Auslösemonitore für den WS-Manager konfiguriert werden. Sie können das Attribut mit dem PATHWAY-Befehl ALTER SERVER hinzufügen bzw. ändern.

Prozesse über mehrere CPUs verteilen

Ein wichtiger Aspekt für die Verteilung der Prozesse auf mehrere CPUs ist, dass jedem Ausführungs-Controller im WS-Manager je eine CPU zugeordnet ist. Die einzelnen Ausführungs-Controller erstellen und verwalten nur in der ihnen zugeordneten CPU eine Reihe von Agentenprozessen. Bei einer Verteilung der Ausführungs-Controller-Prozesse auf mehrere CPUs im System werden die Agentenprozesse ebenfalls entsprechend verteilt.

Bei Angabe mehrerer Ausführungs-Controller werden die Ausführungs-Controller-Prozesse (die jeweils in einer eigenen Serverklasse konfiguriert werden) so weit wie möglich gleichmäßig auf die im System vorhandenen CPUs verteilt. Es bestehen keine Vorgaben, die die Anzahl der Ausführungs-Controller-Prozesse in einem WS-Manager beschränkt; ihre Anzahl richtet sich lediglich nach der Arbeitslast, die ein WS-Manager übernehmen kann. Standardmäßig ist ein Ausführungs-Controller pro WS-Manager vorhanden.

Die Standardserverklasse für Ausführungs-Controller ist MQS-EC00. Soll ein WS-Manager mit mehreren Ausführungs-Controllern erstellt werden, geben Sie im Befehl `crtmqm` die Option `-e` an. Die Anzahl der Ausführungs-Controller-Prozesse kann nach der Erstellung des WS-Managers geändert werden, indem Serverklassen für Ausführungs-Controller-Prozesse hinzugefügt bzw. gelöscht werden und in der Datei QMINI der Eintrag `ExpectedNumECs` in der Zeilengruppe `ECBoss` entsprechend geändert wird.

Jeder Ausführungs-Controller-Prozess *muss* in einer eigenen Serverklasse konfiguriert werden. Für die manuelle Erstellung zusätzlicher Ausführungs-Controller-Prozesse kann die Serverklasse MQS-EC00 als Schablone verwendet werden.

In jeder CPU, die einen Ausführungs-Controller enthält, muss auch ein Repository-Manager vorhanden sein. Bei der Erstellung eines WS-Managers erstellt MQSeries für jeden Ausführungs-Controller eine Serverklasse für den Repository-Manager (deren Name das Format MQS-REPSVR00 hat). Werden in einer CPU, die zuvor keine Ausführungs-Controller enthielt, Serverklassen für Ausführungs-Controller manuell hinzugefügt, sollten Sie in der neuen CPU auf Basis der Serverklasse MQS-REPSVR00 eine neue Serverklasse für den Repository-Manager erstellen.

Für MQSeries ist in jeder CPU eine Serverklasse für den Repository-Manager erforderlich, wenn Folgendes zutrifft:

- Die CPU enthält einen Ausführungs-Controller ODER
- Die CPU enthält die Serverklasse MQS-QMGRSVR00 ODER
- In der CPU werden Anwendungen ausgeführt, für die FASTPATH-Bindungen verwendet werden ODER
- Benutzer der CPU führen eine der folgenden MQSeries-Anwendungen aus:
runmqsc, runmqchi, runmqchl oder **mqrepdmp**.

Die Standardzuordnung zwischen CPUs und Ausführungs-Controller-Prozessen oder anderen Serverklassen kann mit dem PATHCOM-Befehl `ALTER SERVER` unter Angabe des Attributs `CPU` geändert werden.

Der Standardstatusserver wird bei Erstellung eines WS-Managers (Befehl **crtmqm**) in der Serverklasse MQS-STATUS00 automatisch erstellt. Standardmäßig wird nur CPU 0 als primäre CPU und CPU 1 als Backup-CPU (oder CPUS(0:1)) festgelegt. Diese CPU-Zuordnung für die Serverklasse MQS-STATUS00 kann mit dem PATHCOM-Befehl ALTER SERVER geändert werden. Sie können für den Statusserver eine bestimmte CPU für den Backup vorgeben, und zwar durch die Angabe von zwei CPU-Nummern, die durch einen Doppelpunkt getrennt werden; Beispiel: CPUS(2:12). In diesem Fall verwendet PATHMON CPU 2 als primäre CPU und CPU 12 als Backup-CPU. Wird keine bestimmte Backup-CPU angegeben, wählt das Betriebssystem Compaq NSK die Backup-CPU aus.

Der Standardwarteschlangenserver wird ebenfalls bei der Erstellung eines WS-Managers (Befehl **crtmqm**) erstellt, und zwar in der Serverklasse MQS-QUEUE00. Standardmäßig wird nur CPU 0 als primäre CPU und CPU 1 als Backup-CPU (oder CPUS(0:1)) festgelegt. Diese CPU-Zuordnung für die Serverklasse MQS-QUEUE00 kann mit dem PATHCOM-Befehl ALTER SERVER geändert werden. Sie können für den Warteschlangenserver eine bestimmte CPU für den Backup vorgeben, und zwar durch die Angabe von zwei CPU-Nummern, die durch einen Doppelpunkt getrennt werden; Beispiel: CPUS(2:12). In diesem Fall verwendet PATHMON CPU 2 als primäre CPU und CPU 12 als Backup-CPU. Wird keine bestimmte Backup-CPU angegeben, wählt das Betriebssystem Compaq NSK die Backup-CPU aus.

Neue Statusserverprozesse hinzufügen

Sie können einem WS-Manager zusätzliche Statusserver hinzufügen, indem Sie auf Basis der standardmäßigen Statusserverklasse MQS-STATUS00 weitere Serverklassen erstellen.

Der Name der neuen Statusserverklasse sollte mit der Zeichenfolge MQS-STATUS beginnen. Wird bei der Benennung von Serverklassen diese Namenskonvention nicht beachtet, werden die Serverklassen beim Start des WS-Managers (mit Befehl **strmqm**) nicht automatisch gestartet; außerdem wird der Zugriff auf die für diese Statusserverklassen konfigurierten Objekte inaktiviert.

Jedem zusätzlich konfigurierten Statusserver muss ein eigener Prozessname zugewiesen werden. Außerdem sollten die einzelnen Statusserver in verschiedenen CPUs konfiguriert werden, um die Vorteile nutzen zu können, die die Skalierbarkeit der Statusserverarchitektur bietet.

Neue Warteschlangenserverprozesse hinzufügen

Sie können einem WS-Manager zusätzliche Warteschlangenserver hinzufügen, indem Sie auf Basis der standardmäßigen Warteschlangenserverklasse MQS-QUEUE00 weitere Serverklassen erstellen.

Der Name der neuen Warteschlangenserverklasse sollte mit der Zeichenfolge MQS-QUEUE beginnen. Wird bei der Benennung von Serverklassen diese Namenskonvention nicht beachtet, werden die Serverklassen beim Start des WS-Managers (mit Befehl **strmqm**) nicht automatisch gestartet; außerdem wird der Zugriff auf die für diese Warteschlangenserverklassen konfigurierten Objekte inaktiviert.

Jedem zusätzlich konfigurierten Warteschlangenserver muss ein eigener Prozessname zugewiesen werden. Außerdem sollten die einzelnen Warteschlangenserver in verschiedenen CPUs konfiguriert werden, um die Vorteile nutzen zu können, die die Skalierbarkeit der Warteschlangenserverarchitektur bietet.

Aktualisierungsintervall für MQM-Überwachungsanzeigen festlegen

Der PATHWAY-Parameter MQMQMREFRESHINT für MQS-MQMSVR00 gibt das Intervall an, in dem Überwachungsanzeigen für Kanäle und Warteschlangen aktualisiert werden. Als Standardintervall sind 30 Sekunden vorgegeben. Soll eine Aktualisierung dagegen beispielsweise alle 10 Sekunden erfolgen, müssen Sie für den WS-Manager folgenden PATHWAY-Befehl eingeben:

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

PATHWAY-Konfiguration für einen WS-Manager

Im Folgenden sehen Sie eine PATHWAY-Beispielkonfiguration für einen WS-Manager. Diese Beispielkonfiguration wurde durch die Eingabe einer Folge von INFO-Befehlen für die Objekte in der Standardkonfiguration des WS-Managers erstellt.

```
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
```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 1 von 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)

```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 2 von 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

```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 3 von 13)

TS/MP-Verwaltung

```
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
```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 4 von 13)

```
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
```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 5 von 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

```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 6 von 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

```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 7 von 13)

TS/MP-Verwaltung

```
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
```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 8 von 13)

```
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
```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 9 von 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

```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 10 von 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

```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 11 von 13)

TS/MP-Verwaltung

```
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.$DEV.ZMQSEX.MQMGRSVR
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D
```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 12 von 13)

```
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.$DEV.ZMQSEX.MQREPSVR
  SECURITY "N"
  TMF ON
  VOLUME \RAPTOR.$DATA1.P101D
```

Abbildung 3. PATHWAY-Beispielkonfiguration (Teile- 13 von 13)

Parameter von PATHWAY-Serverklassen ändern

So ändern Sie die Parameter von PATHWAY-Serverklassen:

1. Stoppen Sie den WS-Manager mit dem Befehl **endmqm**. Dadurch wird auch der PATHMON-Prozess beendet.
2. Wechseln Sie zum Teildatenträger *WS-ManagerD*, der die Datei PATHCTL enthält.
Beispiel:

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

3. Starten Sie einen PATHMON-Prozess mit dem Namen, den der PATHMON-Prozess des WS-Managers hat, und unter Angabe der Option NOWAIT. Beispiel:

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

4. Starten Sie einen PATHCOM-Prozess für den neuen PATHMON-Prozess.
Beispiel:

```
>PATHCOM $MT01
```

5. Laden Sie die vorhandene PATHWAY-Konfiguration des WS-Managers, indem Sie an der PATHCOM-Eingabeaufforderung den folgenden Befehl eingeben:

```
>START PATHWAY COOL
```

6. Nehmen Sie mit Hilfe von PATHCOM-Befehlen die notwendigen Änderungen vor.
7. Beenden Sie das PATHWAY-System mit dem folgenden Befehl:

```
>SHUTDOWN2
```

8. Starten Sie den WS-Manager mit dem Befehl **strmqm**.

Benutzerdefinierte Serverklassen einer PATHWAY-Konfiguration in MQSeries hinzufügen

Sie können der PATHWAY-Konfiguration in MQSeries mit Hilfe von PATHCOM eigene Serverklassendefinitionen hinzufügen. Hiervon wird jedoch abgeraten, da Server *well behaved* sein müssen, d. h., bestimmte Bedingungen erfüllen müssen, da **endmqm** andernfalls nicht korrekt ausgeführt wird. Außerdem gehen benutzerdefinierte Serverklassendefinitionen verloren, wenn ein WS-Manager gelöscht wird. Um dieses Problem zu vermeiden, sollten Sie ein wiederverwendbares Script verwenden.

Kapitel 4. WS-Manager verwalten

In diesem Kapitel werden die verschiedenen Aspekte der Verwaltung von WS-Managern in MQSeries erläutert.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Einführung“
- „Vorgehensweise beim Erstellen von WS-Managern“
- „WS-Managermerkmale ändern“ auf Seite 50
- „Name des Standardwarteschlangenservers“ auf Seite 62
- „Zusätzliche Warteschlangenserver hinzufügen und entfernen“ auf Seite 63
- „Datenträgerstruktur“ auf Seite 64
- „Mit WS-Managern arbeiten“ auf Seite 67
- „Standard-WS-Manager erstellen“ auf Seite 68
- „MQSeries-Principals erstellen“ auf Seite 69
- „'cleanrdf' für einen RDF-aktivierten WS-Manager ausführen“ auf Seite 70
- „WS-Manager starten“ auf Seite 70
- „Standard- und Systemobjekte wiederherstellen“ auf Seite 70
- „Objektdateien“ auf Seite 70
- „WS-Manager stoppen“ auf Seite 71
- „WS-Manager erneut starten“ auf Seite 72
- „Vorhandenen WS-Manager als Standard-WS-Manager definieren“ auf Seite 72
- „WS-Manager löschen“ auf Seite 73
- „MQM (Message Queue Management) verwenden“ auf Seite 73

Einführung

Vor der Verwendung von Nachrichten und Warteschlangen müssen Sie zunächst mindestens einen WS-Manager erstellen. Nach Abschluss der Installation können Sie mit den MQSeries-Steuerbefehlen WS-Manager und MQSeries-Principals erstellen und den WS-Manager starten.

Bei der Erstellung eines WS-Managers werden auch automatisch die erforderlichen Standardobjekte und Systemobjekte erstellt. Standardobjekte sind die Basis für alle Objektdefinitionen, die Sie erstellen; Systemobjekte sind für den Betrieb des WS-Managers erforderlich. Informationen zur Wiederherstellung der Standard-systemobjekte finden Sie unter „Standard- und Systemobjekte wiederherstellen“ auf Seite 70.

Weitere Informationen zu den Befehlen, die unter MQSeries for Compaq NSK verwendet werden können, sowie die zu den Möglichkeiten, diese Befehle aufzurufen, finden Sie in „Kapitel 3. MQSeries-Befehlsätze verwenden“ auf Seite 31.

Vorgehensweise beim Erstellen von WS-Managern

WS-Manager verwalten die ihnen zugeordneten Ressourcen, wie beispielsweise Warteschlangen. Der WS-Manager stellt Anwendungen Queuing-Services für MQI-Aufrufe sowie Befehle, mit denen MQSeries-Objekte erstellt, geändert, angezeigt und gelöscht werden können, zur Verfügung. Sie können WS-Manager mit dem Befehl **crtmqm** erstellen. Im Folgenden einige Punkte, die bei der Erstellung eines WS-Managers berücksichtigt werden sollten:

- **Geben Sie einen eindeutigen Namen für den WS-Manager an.**

WS-Manager erstellen

Bei der Erstellung eines WS-Managers müssen Sie sicherstellen, dass er einen Namen erhält, der keinem anderen WS-Manager im Netz zugeordnet ist. Die Namen der WS-Manager werden bei der Erstellung nicht überprüft, bei Vergabe von bereits vorhandenen Namen können jedoch keine Kanäle für die verteilte Steuerung von Warteschlangen erstellt werden.

Sie können sicherstellen, dass jeder WS-Manager einen eindeutigen Namen erhält, indem Sie diesem jeweils den eigenen Knotennamen voranstellen. Bei einem Knoten mit dem Namen `accounts` können Sie beispielsweise dem WS-Manager den Namen `accounts.saturn.queue.manager` zuweisen; dabei steht `saturn` für einen bestimmten WS-Manager, und `queue.manager` ist die Erweiterung, die allen WS-Managern zugewiesen wird. Sie können die Erweiterung auch weglassen, allerdings handelt es sich bei `accounts.saturn` und `accounts.saturn.queue.manager` um *verschiedene* WS-Managernamen.

Anmerkung: Bei der Angabe von WS-Managernamen in Steuerbefehlen muss die Groß-/Kleinschreibung beachtet werden. Beispielsweise können Sie zwei WS-Manager mit den Namen `jupiter.queue.manager` und `JUPITER.queue.manager` erstellen.

- **Die Anzahl der WS-Manager sollte begrenzt sein.**

In MQSeries for Compaq NSK können so viele WS-Manager erstellt werden, wie es die Ressourcen zulassen. Da jedoch für jeden WS-Manager eigene Ressourcen erforderlich sind, ist es in vielen Fällen vorteilhafter, einen WS-Manager mit 100 Warteschlangen zu konfigurieren, als zehn WS-Manager mit jeweils zehn Warteschlangen. Viele Knoten kommen mit einem einzigen WS-Manager aus, größere Server können jedoch mit mehreren WS-Managern betrieben werden. Besondere Anforderungen an die Systemleistung oder die Funktionalität können unter Umständen die Konfiguration mehrerer WS-Manager notwendig machen.

- **Geben Sie einen Standard-WS-Manager an.**

Jeder Knoten sollte über einen Standard-WS-Manager verfügen, obwohl es möglich ist, MQSeries auf einem Knoten ohne einen Standard-WS-Manager zu konfigurieren.

Ist in einem MQCONN-Aufruf kein WS-Manager angegeben, werden Anwendungen mit dem Standard-WS-Manager verbunden. Der Standard-WS-Manager verarbeitet außerdem die MQSC-Befehle, die über den Befehl `runmqsc` ohne Angabe eines WS-Managers ausgegeben werden.

Bei Angabe eines WS-Managers als Standard-WS-Manager *ersetzt* dieser einen eventuell vorhandenen Standard-WS-Manager.

Wird ein anderer Standard-WS-Manager angegeben, kann dies unter Umständen Auswirkungen für andere Benutzer oder Anwendungen haben. Die Änderung betrifft nicht die Anwendungen, die zum Zeitpunkt der Änderung bereits verbunden sind, da sie für weitere MQI-Aufrufe die Kennung aus dem ursprünglichen Verbindungsaufruf verwenden können. Diese Kennung stellt sicher, dass die Aufrufe an denselben WS-Manager weitergeleitet werden. Anwendungen, die nach der Änderung eine Verbindung anfordern, werden mit dem neuen Standard-WS-Manager verbunden.

Sie erstellen einen Standard-WS-Manager, indem Sie im Befehl `crtmqm` die Option `-q` angeben. Eine ausführliche Beschreibung dieses Befehls und seiner Parameter finden Sie unter „`crtmqm` (WS-Manager erstellen)“ auf Seite 273.

- **Geben Sie eine Warteschlange für nicht zustellbare Nachrichten an.**

Bei der Warteschlange für nicht zustellbare Nachrichten handelt es sich um eine lokale Warteschlange, in die Nachrichten eingereicht werden, die nicht an die eigentliche Zieladresse weitergeleitet werden können.

Achtung: Jeder WS-Manager in Ihrem Netz sollte über eine Warteschlange für nicht zustellbare Nachrichten verfügen. Andernfalls können Anwendungsprogrammfehler auftreten, die dazu führen, dass Kanäle beendet werden und die Ausführung von Verwaltungsbefehlen fehlschlägt. Versucht beispielsweise eine Anwendung, eine Nachricht in eine Warteschlange eines anderen WS-Managers einzureihen, und wurde die falsche Warteschlange angegeben, wird der Kanal gestoppt, und die Nachricht bleibt in der Übertragungswarteschlange. Andere Anwendungen können diesen Kanal nicht mehr für ihre Nachrichten verwenden.

Warteschlangen für nicht zustellbare Nachrichten in WS-Managern haben keine Auswirkungen auf die Kanäle. Wird eine nicht zustellbare Nachricht auf der Empfängerseite in die Warteschlange für nicht zustellbare Nachrichten eingereiht, bleiben der Kanal und die Übertragungswarteschlange weiterhin verfügbar.

Bei der Erstellung eines WS-Managers sollten Sie daher die Option **-u** verwenden, um den Namen der Warteschlange für nicht zustellbare Nachrichten anzugeben. Außerdem können Sie mit MQSC-Befehlen die Attribute eines WS-Managers ändern und damit die Warteschlange für nicht zustellbare Nachrichten angeben, die verwendet werden soll. Ein Beispiel für den MQSC-Befehl ALTER finden Sie unter „WS-Managerattribute ändern“ auf Seite 100.

Wenn Sie feststellen, dass eine Warteschlange für nicht zustellbare Nachrichten Nachrichten enthält, können Sie diese mit der in MQSeries verfügbaren Steueroutine der Warteschlange für nicht zustellbare Nachrichten verarbeiten. Weitere Informationen zu dieser Steueroutine und dazu, wie die Anzahl der Nachrichten, die in dieser Warteschlange eingereiht werden, reduziert werden kann, finden Sie in „Kapitel 9. MQSeries-Steueroutine der Warteschlange für nicht zustellbare Nachrichten“ auf Seite 163.

- **Geben Sie eine Standardübertragungswarteschlange an.**

Bei der Übertragungswarteschlange handelt es sich um eine lokale Warteschlange, in die vorübergehend eingereiht werden, bevor sie an einen fernen WS-Manager übertragen werden. Die Standardübertragungswarteschlange wird verwendet, wenn keine bestimmte Übertragungswarteschlange angegeben wird. Jedem WS-Manager kann eine Standardübertragungswarteschlange zugeordnet werden.

Bei der Erstellung eines WS-Managers sollten Sie die Option **-d** verwenden, um den Namen der Standardübertragungswarteschlange anzugeben. Mit dieser Option wird die Warteschlange jedoch noch nicht erstellt; dies müssen Sie zu einem späteren Zeitpunkt nachholen. Weitere Informationen finden Sie unter „Mit lokalen Warteschlangen arbeiten“ auf Seite 105.

- **Erstellen Sie nach der Erstellung eines WS-Managers eine Sicherungskopie der Konfigurationsdateien.**

Die MQSeries-Konfigurationsdatei (MQSINI) wird bei der Installation von MQSeries erstellt. Sie enthält eine Liste der WS-Manager und wird immer aktualisiert, wenn Sie einen WS-Manager erstellen oder löschen. Für jede Installation ist eine MQSINI-Datei vorhanden. Diese befindet sich standardmäßig im Verzeichnis \$SYSTEM.ZMQSSYS.

Die Konfigurationsdatei eines WS-Managers (QMINI) wird bei seiner Erstellung automatisch erstellt. Diese Datei enthält die Konfigurationsparameter des WS-Managers.

Sie sollten für diese Dateien eine Sicherungskopie erstellen. Wird ein neuer WS-Manager erstellt, bei dem es zu Problemen kommt, können Sie nach der

WS-Manager erstellen

Fehlerbehebung diese Sicherungskopie wiederherstellen. Die Konfigurationsdateien sollten jedes Mal gesichert werden, wenn ein neuer WS-Manager erstellt wird.

Weitere Informationen zu Konfigurationsdateien finden Sie in „Kapitel 13. Konfigurationsdateien“ auf Seite 193.

Sicherungskopien von Konfigurationsdateien nach Erstellung eines WS-Managers erstellen

Es müssen Sicherungskopien für zwei Konfigurationsdateien (MQSINI und QMINI) angelegt werden:

1. Die MQSeries-Konfigurationsdatei (MQSINI) wird bei der Installation von MQSeries erstellt. Sie enthält eine Liste der WS-Manager und wird immer aktualisiert, wenn Sie einen WS-Manager erstellen oder löschen. Für jede Installation ist eine MQSINI-Datei vorhanden. Diese befindet sich standardmäßig im Verzeichnis `$SYSTEM.ZMQSSYS`.
2. Die Konfigurationsdatei eines WS-Managers (QMINI) wird bei seiner Erstellung automatisch erstellt. Diese Datei enthält die Konfigurationsparameter des WS-Managers.

Sie sollten für diese Dateien eine Sicherungskopie erstellen. Wird ein neuer WS-Manager erstellt, bei dem es zu Problemen kommt, können Sie nach der Fehlerbehebung diese Sicherungskopie wiederherstellen. Die Konfigurationsdateien sollten jedes Mal gesichert werden, wenn ein neuer WS-Manager erstellt wurde.

WS-Managermerkmale ändern

Viele der Eigenschaften eines WS-Managers können bei seiner Erstellung geändert werden. Einige Eigenschaften der WS-Manager können auch noch nach ihrer Erstellung geändert werden, auch wenn der WS-Manager in der Regel zunächst gestoppt und anschließend neu gestartet werden muss, damit die Änderungen wirksam werden.

Im Folgenden werden einige der WS-Managermerkmale beschrieben, die Sie unter Umständen ändern müssen.

Ausgangsdatenträger des WS-Managers

Hierbei handelt es sich um den Datenträger, in dem alle Datenbanken einschließlich Warteschlangen erstellt werden. Einzelne Warteschlangen können jedoch nach ihrer Erstellung wie unter „`altmqfls` (Attribute der Warteschlangendateien ändern)“ auf Seite 260 beschrieben mit dem Steuerbefehl `altmqfls` in einen anderen Datenträger verschoben werden.

Die Standardeinstellung wird aus dem Eintrag `QMDefaultVolume` in der Zeilen-Gruppe `AllQueueManagers` der Datei `MQSINI` übernommen. Diese Einstellung kann mit dem Parameter `-p DefaultPrefix` des Befehls `crtmqm` überschrieben werden.

Der Ausgangsdatenträger kann *nur* bei der Erstellung eines WS-Managers angegeben werden. Nach der Erstellung kann er nicht mehr geändert werden.

Anzahl der Ausführungs-Controller pro WS-Manager

Standardmäßig ist ein Ausführungs-Controller-Prozess pro WS-Manager vorhanden. Die Anzahl der Ausführungs-Controller-Prozesse für einen WS-Manager kann über den Parameter `-e NumECs` im Befehl `crtmqm` angegeben werden.

Jeder Ausführungs-Controller ist für eine Reihe von Serverprozessen verantwortlich, die Messaging- und Queuing-Vorgänge für die Anwendungen und Kanäle ausführen, die sich in derselben CPU wie der Ausführungs-Controller selbst befinden. Es wird empfohlen, in jeder CPU nur jeweils einen Ausführungs-Controller zu konfigurieren; bei einer großen Anzahl von Anwendungen für einen Ausführungs-Controller ist es allerdings vorteilhafter, einen zusätzlichen Ausführungs-Controller in der CPU einzusetzen. In großen Installation sind beispielsweise mehrere Ausführungs-Controller-Prozesse sinnvoll (die oft auf mehrere CPUs verteilt sind), da so eine große Anzahl von Anwendungen und Kanälen gleichzeitig bearbeitet werden können.

Bei der Erstellung eines WS-Managers wird für jeden im Befehl `crtmqm` angegebenen Ausführungs-Controller eine TS/MP-Serverklasse erstellt. Die angegebenen Serverklassen der Ausführungs-Controller werden im Round-Robin-Verfahren auf alle CPUs im System verteilt. Werden beispielsweise in einem System mit vier CPUs acht Ausführungs-Controller-Prozesse angegeben, so würden in der Regel jeder CPU zwei Ausführungs-Controller-Prozesse zugeteilt werden.

Im Feld `ExpectedNumECs` in der Zeilengruppe `ECBoss` der Datei `QMINI` des WS-Managers wird bei dessen Erstellung die Anzahl der Ausführungs-Controller-Prozesse angegeben. Dieser Wert muss immer der TS/MP-Konfiguration entsprechen.

Die Anzahl der Ausführungs-Controller-Prozesse kann nach der Erstellung eines WS-Managers geändert werden, indem TS/MP-Serverklassen hinzugefügt bzw. gelöscht werden und der Eintrag `ExpectedNumECs` in der Zeilengruppe `ECBoss` der Datei `QMINI` entsprechend geändert wird. Dies ist nur möglich, wenn der WS-Manager zuvor gestoppt wird.

Gleichmäßige Systemauslastung

Der `ECBoss` (Execution Controller Boss, Manager der Ausführungs-Controller) ist für die gleichmäßige Verteilung der Arbeitslast eines WS-Managers auf die Ausführungs-Controller zuständig. Bei einer entsprechenden Konfiguration der Ausführungs-Controller kann die Arbeitslast eines WS-Managers gleichmäßig auf mehrere CPUs verteilt werden.

Trifft eine neue Verbindungsanforderung von einer lokalen Anwendung ein oder wird ein Kanal gestartet, ordnet der `ECBoss` die Anforderung dem Ausführungs-Controller zu, für den momentan die geringste Anzahl an lokalen WS-Manageragenten und Nachrichtenkanalagenten aktiv ist.

Ausgangsterminal des WS-Managers

Alle Compaq NSK-Prozesse einschließlich der Serverprozesse des WS-Managers verfügen über ein Ausgangsterminal. Dieses *muss* vorhanden sein und sich im Pausenstatus befinden. Im Allgemeinen wird das Ausgangsterminal des WS-Managers nicht für die Ausgabe verwendet. Bei dem Ausgangsterminal kann es sich um eine beliebige Terminaleinheit wie beispielsweise Compaq VHS (Virtual Hometerm Service) handeln.

Compaq NonStop Kernel unterstützt für jedes physisches Terminal maximal 255 primäre Prozesse. Daher ist eine sorgfältige Planung erforderlich, damit dieser Wert nicht überschritten wird. Ist eine Unterstützung für mehr als 255 primäre Prozesse erforderlich, sollten Sie Compaq VHS verwenden.

WS-Manager erstellen

Das Ausgangsterminal eines WS-Managers wird über den Parameter `-o HomeTerminalName` im Befehl `crtmqm` angegeben. Für diesen Parameter gibt es keinen Standardwert, seine Angabe ist obligatorisch.

Die Attribute `HOMETERM` und `OUT` aller TS/MP-Serverklassen werden auf die angegebene Terminaleinheit gesetzt. Diese Attribute können jederzeit geändert werden, wenn die Serverklassen zuvor gestoppt wurden; dies ist in der Regel der Fall, wenn der WS-Manager gestoppt wird.

Der Eintrag `HomeTerminalName` in der Zeilengruppe `Configuration` der Datei `QMINI` muss ebenfalls geändert werden, wenn für einen WS-Manager ein anderes Ausgangsterminal verwendet werden soll.

PATHMON-Prozessname für den WS-Manager

Jeder WS-Manager ist in einer eigenen TS/MP-Konfiguration (`PATHWAY`-Konfiguration) aktiv. Bei dem Steuerprozess handelt es sich um den `PATHMON`-Prozess. Für jeden WS-Manager muss ein eindeutiger Name angegeben werden. Dieser Name muss im gesamten System eindeutig sein.

Der `PATHMON`-Prozessname wird über den Parameter `-n PATHMONProcessName` im Befehl `crtmqm` angegeben. Für diesen Parameter gibt es keinen Standardwert, seine Angabe ist obligatorisch.

Verwenden Sie einen Prozessnamen, der systemweit eindeutig ist und problemlos dem WS-Manager zugeordnet werden kann, den dieser Prozess steuert.

So können Sie den Namen des `PATHMON`-Prozesses für einen WS-Manager ändern:

1. Stoppen Sie den WS-Manager.
2. Geben Sie für den Standard- und den Teildatenträger die Adresse der Dateien des WS-Managers (in der Regel `<WS-Manager>D`) an.
3. Ändern Sie den Eintrag `PathmonProcName` in der Datei `QMINI` des WS-Managers, indem Sie den neuen Prozessnamen angeben.
4. Führen Sie `PATHMON` für den WS-Manager unter Verwendung des *neuen* Prozessnamens aus.

Hierzu müssen Sie an der `TACL`-Eingabeaufforderung den folgenden Befehl eingeben:

```
PATHMON /name $<neuer_Name>, nowait/
```

5. Führen Sie `PATHCOM` für `PATHMON` unter dem neuen Prozessnamen aus. Hierzu müssen Sie an der `TACL`-Eingabeaufforderung den folgenden Befehl eingeben:

```
PATHCOM $<neuer_Name>
```

6. Laden Sie die `PATHWAY`-Konfiguration des WS-Managers, und bestätigen Sie die Änderung des `PATHMON`-Prozessnamens.

Führen Sie in PATHCOM den folgenden Befehl aus:

```
START PATHWAY COOL
```

Beim Laden der Konfiguration wird eine Meldung ausgegeben, die darauf hinweist, dass sich der Name des neuen PATHMON-Prozesses von dem Namen in der Konfigurationsdatei unterscheidet. Anschließend müssen Sie bestätigen, dass der Vorgang fortgesetzt werden soll. Geben `y` an der Eingabeaufforderung ein; der Ladevorgang der Konfiguration wird daraufhin abgeschlossen.

- Speichern Sie die PATHWAY-Konfigurationsdaten in der Datenbank.

Führen Sie dazu in PATHCOM die folgenden Befehle aus:

```
SHUTDOWN2  
EXIT
```

Der Eintrag `PathmonProcName` in der Zeilengruppe `Configuration` der Datei `QMINI` muss ebenfalls geändert werden.

Die Änderung des PATHMON-Prozessnamens ist damit abgeschlossen. Bei der nächsten Eingabe des Befehls `strmqm` wird der WS-Manager mit dem neuen PATHMON-Prozessnamen gestartet.

CCSID des WS-Managers

Bei der CCSID handelt es sich um die ID des codierten Zeichensatzes, den der WS-Manager zum Speichern von Informationen zu Nachrichten verwendet.

Die CCSID wird über den Parameter `-l CCSID` des Befehls `crmqm` angegeben. Der Standardwert ist 819.

Die CCSID kann nach der Erstellung eines WS-Managers jederzeit mit dem Befehl `runmqsc`, über die MQM-Funktion oder mit PCF-Befehlen geändert werden.

EBCDIC-Datenkonvertierung steuern

Die Handhabung von Zeilenumbruchzeichen (NL, New Line) bei der Konvertierung kann über den Eintrag `ConvEBCDICNewline` in der Zeilengruppe `AllQueueManagers` der `MQSINI`-Konfigurationsdatei vorgegeben werden.

ConvEBCDICNewline=NL_TO_LF|TABLE|ISO

EBCDIC-Codepages enthalten ein Zeilenumbruchzeichen, das von ASCII-Codepages nicht unterstützt wird (obwohl einige ISO-Varianten des ASCII-Codes ein entsprechendes Zeichen enthalten). Über das Attribut `ConvEBCDICNewline` können Sie angeben, wie MQSeries bei der Konvertierung des EBCDIC-Zeilenumbruchzeichens in das ASCII-Format vorgehen soll.

NL_TO_LF

Geben Sie `NL_TO_LF` an, wenn das EBCDIC-Zeilenumbruchzeichen (X'15') bei allen EBCDIC-ASCII-Konvertierungen in das ASCII-Zeichen für Zeilenvorschub (LF, Line Feed (X'0A')) umgesetzt werden soll. `NL_TO_LF` ist der Standardwert.

WS-Manager erstellen

TABLE

Geben Sie TABLE an, wenn das EBCDIC-Zeilenumbruchzeichen bei allen Konvertierungen von EBCDIC nach ASCII gemäß den auf Ihrer Plattform verwendeten Konvertierungstabellen konvertiert werden soll. Beachten Sie, dass das Ergebnis bei dieser Art der Konvertierung von Plattform zu Plattform und von Sprache zu Sprache unterschiedlich sein kann; auch auf derselben Plattform kann das Ergebnis unterschiedlich sein, wenn Sie unterschiedliche CCSIDs verwenden.

ISO

Geben Sie ISO an, wenn:

- ISSO-CCSIDs entsprechend der Methode TABLE konvertiert werden sollen,
- alle anderen CCSIDs mit der Methode NL_TO_CF konvertiert werden sollen.

Tabelle 2 enthält eine Auflistung möglicher ISO-CCSIDs.

Tabelle 2. Liste möglicher ISO-CCSIDs

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

Handelt es sich bei der ASCII-CCSID nicht um eine ISO-Untergruppe, wird für ConvEBCDICNewline als Standardwert NL_TO_LF übernommen.

Weitere Informationen zur Datenkonvertierung finden Sie im Handbuch *MQSeries Application Programming Guide*.

EMS-Ereignissammler für WS-Manager

Der WS-Manager kann bei Bedarf auch für die Verwendung eines anderen Ereignissammlers (Event Collector) konfiguriert werden. EMS-Ereignisse werden standardmäßig an \$0 gesendet. Der Eintrag `EMSCollectorName` in der Zeilengruppe `Configuration` der Datei `QMINI` gibt den Namen des EMS-Ereignissammlers für den betreffenden WS-Manager an.

Der EMS-Ereignissammler kann jederzeit geändert werden, indem für diesen Eintrag ein anderer Wert angegeben wird; Änderungen werden jedoch erst bei einem Neustart des WS-Managers wirksam.

Agenten-Pool der Ausführungs-Controller im WS-Manager

Für jeden der vier grundsätzlichen Agententypen verwaltet der Ausführungs-Controller einen Pool mit inaktiven Agentenprozessen, die jederzeit eingesetzt werden können. Die Größe dieser Pools kann so konfiguriert werden, dass ein angemessenes Verhältnis zwischen Antwortzeit bei neuen Verarbeitungsanforderungen

und Ressourcenbelegung erreicht wird. In den folgenden Felder in der Zeilengruppe Configuration der Datei QMINI kann die Anzahl der inaktiven Prozesse vorgegeben werden:

MinIdleMCALU62Responders

Gibt die Mindestanzahl inaktiver SNA-LU6.2-Responder-Nachrichtenkanalagenten an. Der Standardwert ist 0.

MinIdleMCATCPResponders

Gibt die Mindestanzahl inaktiver TCP/IP-Responder-Nachrichtenkanalagenten an. Der Standardwert ist 0.

MinIdleMCACallers

Gibt die Mindestanzahl inaktiver aufrufender Nachrichtenkanalagenten (protokollunabhängig) an. Der Standardwert ist 0.

MinIdleLQMAgents

Gibt die Mindestanzahl inaktiver lokaler WS-Manageragenten (LQMAs) an. Der Standardwert ist 1.

Die Angaben in diesen Feldern gelten pro Ausführungs-Controller, nicht pro WS-Manager. Für einen WS-Manager mit zwei Ausführungs-Controllern sind daher standardmäßig zwei inaktive lokale WS-Manageragenten vorgesehen.

Diese Einstellungen können jederzeit geändert werden; Änderungen werden jedoch erst bei einem Neustart des WS-Managers wirksam.

Maximale Anzahl inaktiver Agenten und Prozesswiederverwendung

Standardmäßig kann ein WS-Manager über maximal zehn inaktive Agentenprozesse pro Agententyp verfügen. Dieser Wert wird über den Eintrag `MaxIdleAgents` in der Zeilengruppe Configuration der Datei QMINI vorgegeben.

Über den Eintrag `MaxIdleAgentReuse` wird festgelegt, wie oft ein Agentenprozess verwendet werden kann, bevor ein neuer Agentenprozess erstellt wird. Der Standardwert für `MaxIdleAgentReuse` ist 10.

Priorität der WS-Managerprozesse

Die Prozesspriorität muss unter Umständen so geändert werden, dass eine gleichmäßige Ressourcenbelegung durch MQSeries und andere Anwendungen gegeben ist. Die Prozessprioritäten der TS/MP-Serverklassen können über die TS/MP-Objekte geändert werden, nachdem der WS-Manager zuvor gestoppt wurde.

Die Priorität der Agentenprozesse kann über die Felder `MCAAgentPriority` und `LQMAgentPriority` in der Zeilengruppe EC der Datei QMINI geändert werden.

Den Status- und Warteschlangenservern ist eine Standardpriorität von 176 vorgegeben, für die TS/MP-Prozesse eine Standardpriorität von 175. Für Nachrichtenkanalagenten und lokale WS-Manageragenten ist eine Standardpriorität von 165 festgelegt.

Stellen Sie sicher, dass Status- und Warteschlangenserver die höchste Priorität haben, gefolgt vom ECBoss und den Ausführungs-Controllern; zuletzt kommen die Nachrichtenkanalagenten und die lokalen WS-Manageragenten.

WS-Manager erstellen

Maximale Anzahl an Kanälen für den WS-Manager

Für jeden WS-Manager ist die Anzahl der vorhandenen Kanäle begrenzt. Ist dieser Grenzwert zu hoch angesetzt, kann sich dies nachteilig auf die Leistung auswirken, da dieser Parameter die Größe der Kanalstatustabelle vorgibt, die durchsucht wird. Ist der Grenzwert dagegen zu niedrig angesetzt, sind unter Umständen nicht genügend Kanäle für Ihre Anwendung vorhanden. Über das Feld `MaxChannels` in der Zeilengruppe `Channels` der Datei `QMINI` wird die maximale Anzahl an Kanälen angegeben, die gleichzeitig gesteuert werden können.

Der Standardwert bei der Erstellung eines WS-Managers ist 10. Dieser Wert kann bei der Erstellung nicht geändert werden.

Der Eintrag `MaxChannels` in der Zeilengruppe `Channels` der Datei `QMINI` kann später jedoch jederzeit geändert werden; Änderungen werden jedoch erst bei einem Neustart des WS-Managers wirksam.

Maximale Anzahl an aktiven Kanälen für den WS-Manager

Für jeden WS-Manager ist Anzahl der Kanäle, die gleichzeitig aktiv sein können, begrenzt. Dadurch kann die Belegung von Systemressourcen durch Kanäle bei Höchstbelastung gesteuert werden. Der Eintrag `MaxActiveChannels` in der Zeilengruppe `Channels` der Datei `QMINI` legt die maximale Anzahl aktiver Kanäle für den WS-Manager fest.

Der Standardwert bei der Erstellung eines WS-Managers ist 10. Dieser Wert kann bei der Erstellung nicht geändert werden.

Der Eintrag `MaxActiveChannels` in der Datei `QMINI` kann später jedoch jederzeit geändert werden; Änderungen werden jedoch erst bei einem Neustart des WS-Managers wirksam.

Guardian-Segment-IDs für MQSeries

In MQSeries werden Guardian-Speichersegmente sowohl den MQSeries-Prozessen als auch dem Prozess des Anwendungsprogramms zugeordnet. Der Bereich der Guardian-Segment-IDs, die MQSeries diesen Segmenten zuweist, wird in der Konfigurationsdatei `MQSINI` definiert. Der Eintrag `NSKSegidRange` in der Zeilengruppe `AllQueueManagers` der Konfigurationsdatei `MQSINI` gibt den Bereich von Segment-IDs vor, der MQSeries bei der Zuordnung von Speichersegmenten zur Verfügung steht.

Bei der Installation von MQSeries wird für diesen Bereich der Standardwert `NSKSegidRange=10-20` definiert.

Da diese Speichersegmente auch dem Anwendungsprogrammprozess zugeordnet werden, müssen Sie sicherstellen, dass keine Segmente aus diesem Bereich verwendet werden. Sie können vorgeben, dass die von MQSeries zugeordneten Segmente aus einem anderen Bereich stammen, indem Sie den Eintrag `NSKSegidRange` in der Datei `MQSINI` entsprechend festlegen. Die Änderung wird erst wirksam, nachdem Sie alle WS-Manager gestoppt und anschließend erneut gestartet haben.

Der über den Parameter `NSKSegidRange` vorgegebene Bereich muss in dem für Guardian-Segment-IDs vorgegebenen Bereich von 0 - 1023 liegen. Außerdem muss der in diesem Parameter für MQSeries vorgegebene Bereich mindestens zehn Guardian-Segment-IDs enthalten.

TCP/IP-Standard-Port

Der Eintrag `TCPPort` in der Zeilengruppe `TCPConfig` der Datei `QMINI` gibt die Standard-Port-Nummer für abgehende Kanäle an. Als Standard-Port wird 1414 verwendet. Dieser Wert wird durch den im Feld `CONNNAME` für einen Kanal angegebenen Wert überschrieben.

Anzahl an TCP/IP-Ports, die vom WS-Manager überwacht werden

Ein WS-Manager mit TCP/IP-Kanälen kann so konfiguriert werden, dass er einen oder mehr TCP/IP-Ports auf ankommende Verbindungen überwacht. Der Eintrag `TCPNumListenerPorts` in der Zeilengruppe `TCPConfig` der Datei `QMINI` gibt die Anzahl der Ports an, die überwacht werden, die Einträge `TCPLListenerPort` in derselben Zeilengruppe geben die Nummern dieser Ports an. Ein Beispiel für diese Einträge in der Datei `QMINI` finden Sie unter „TCP/IP-Kommunikationskonfiguration - Beispiel“ auf Seite 399.

In einem System können mehrere WS-Manager vorhanden sein. Jedem WS-Manager müssen mehrere TCP/IP-Ports zugewiesen werden, die auf ankommende Verbindungen überwacht werden, wobei es zu keinen Überschneidungen kommen darf. Werden nur wenige ankommende TCP/IP-Verbindungsanforderungen erwartet, ist unter Umständen jedem WS-Manager nur ein TCP/IP-Port zugewiesen; in größeren Konfigurationen können den einzelnen WS-Managern auch mehrere TCP/IP-Ports zugeordnet werden. Der Standard-TCP/IP-Port für WS-Manager ist 1414, und standardmäßig wird auch nur dieser Port vom WS-Manager auf ankommende Verbindungen überwacht.

Die Liste der Ports, die überwacht werden, kann in der Zeilengruppe `TCPConfig` der Datei `QMINI` geändert werden; der WS-Manager muss anschließend neu gestartet werden. Soll ein WS-Manager mehrere Ports überwachen, müssen für ihn unter Verwendung von `TS/MP` mehrere Serverklassen für TCP/IP-Serverklassen konfiguriert werden. Diese Definition erfolgt manuell mit `PATHCOM`.

Eine andere Möglichkeit besteht darin, eine Port-Nummer im Befehl `runmqclsr` anzugeben (siehe unter „`runmqclsr` (Empfangsprogramm ausführen)“ auf Seite 300). Die Werte der Einträge `TCPLListenerPort` werden von einem Empfangsprogramm überschrieben, falls in der Umgebung dieses Empfangsprogramms der Parameter `MLISTENPORTNUM` angegeben ist.

Der vom WS-Manager verwendete TCP/IP-Prozess

Die Schnittstelle zu Compaq TCP/IP erfolgt über einen Serverprozess, den so genannten TCP/IP-Prozess. Als Systemstandardwert ist `$ZTC0` vorgegeben. Dieser Wert kann bei der Erstellung des WS-Managers nicht geändert werden. Die Kanäle eines WS-Managers können so konfiguriert werden, dass ein bestimmter TCP/IP-Prozess verwendet wird, falls der vom System vorgegebene Prozess nicht ausreicht. Die Konfiguration der `TS/MP`-Serverklasse des TCP/IP-Empfangsprogramms muss manuell geändert werden, falls der vom System vorgegebene TCP/IP-Prozess `$ZTC0` nicht ausreicht oder falsch ist. Eine manuelle Änderung dieser Serverklassenkonfiguration ist auch möglich, wenn ein Empfangsprogramm einen bestimmten Port überwachen soll und damit die in der Datei `QMINI` definierten Ports außer Kraft gesetzt werden sollen.

Der Serverklasse `MQS-TCPLIS00` muss der Eintrag `DEFINE TCPIP^PROCESS^NAME` hinzugefügt werden, der auf den erforderlichen anderen TCP/IP-Prozessnamen verweist. Sind mehrere Ausführungs-Controller vorhanden, müssen alle

WS-Manager erstellen

aktualisiert werden. Weitere Informationen hierzu finden Sie in den Handbüchern zu dem TCP/IP-Produkt für Compaq NSK.

Änderungen an den TS/MP-Serverklassen sind nur möglich, wenn zuvor der WS-Manager gestoppt wurde.

MQSeries for Compaq NSK unterstützt auch mehrere TCP/IP-Prozesse für WS-Manager. Außerdem werden mehrere TCP/IP-Ports und Empfangsprogrammprozesse unterstützt.

Serverklasse MQS-TCPLISnn für die Verwendung eines anderen TCP/IP-Prozesses und Ports rekonfigurieren

Allgemeine Anweisungen, wie die Serverklasse MQS-TCPLISnn für die Verwendung eines anderen als des standardmäßigen TCP/IP-Prozesses und Ports rekonfiguriert wird, finden Sie unter „Parameter von PATHWAY-Serverklassen ändern“ auf Seite 45.

Mit den folgenden PATHCOM-Befehlen geben Sie einen bestimmten TCP/IP-Prozess an:

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

Dabei steht \$ZZZZ für den Namen des betreffenden TCP/IP-Prozesses.

Mit den folgenden PATHCOM-Befehlen geben Sie einen bestimmten Port an:

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

Dabei steht nnnn für die Nummer des Ports, der auf ankommende Verbindungen überwacht werden soll.

Bei der Verwendung von Empfangsprogrammen, die über TACL (**runmqlsru**) ausgeführt werden und für die ein anderer als der standardmäßig vorgegebene TCP/IP-Prozess \$ZTC0 erforderlich ist, muss der TACL-Umgebung, von der aus das Empfangsprogramm ausgeführt wird, Folgendes hinzugefügt werden:

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

Dabei steht \$ZZZZ für den Namen des betreffenden TCP/IP-Prozesses.

Anmerkung: Wenn einzelne Serverklassen für TCP/IP-Empfangsprogramme verschiedene TCP/IP-Prozesse für die einzelnen Ports verwenden sollen, muss in jedem Serverklassenexemplar die Port-Nummer und der TCP/IP-Prozessname angegeben werden, da andernfalls nicht vorherzusehen ist, welchen TCP/IP-Prozess ein in der Datei QMINI definierter Port verwenden wird.

Ist der Parameter MQLISTENPORTNUM in der Umgebung des Empfangsprogramms nicht definiert, entnimmt das Empfangsprogramm die Port-Nummer der Datei QMINI.

Zuordnung des Auslagerungsspeichers

MQSeries for Compaq NSK ordnet den verschiedenen ausführbaren Modulen in der Konfigurationsdatei QMINI einen Auslagerungsspeicher zu, dessen Größe entsprechend dem Eintrag `ExtPoolSize` festgelegt wird. Verwendet Ihr WS-Manager daher die Standardkonfigurationsdatei QMINI, in der für jedes ausführbare Modul 300 KB vorgesehen sind, und verfügt der WS-Manager über 10 abgehende Kanäle, 20 Agenten und 10 TCP-Responder, benötigen Sie mindestens 12 MB Auslagerungsspeicher (40 x 300 KB). Sie können diesen Wert verkleinern, indem Sie in der Datei QMINI niedrigere Werte angeben. Die `ExtPoolSize`-Werte geben den Mindestspeicher an, der dann zugeordnet wird, wenn der anfänglich vorgesehene Speicherplatz belegt ist. Dieser Wert muss nicht über der maximal für den WS-Manager zulässigen Nachrichtengröße liegen.

Eine Möglichkeit, den Auslagerungsspeicher für ausführbare MQSeries-Module zu ändern, besteht z. B. darin, die MQ-PATHWAY-Servertypen zu ändern; dazu wird Folgendes hinzugefügt:

```
DEFINE =_DEFAULTS, CLASS DEFAULTS, VOLUME Datenträger.WSManagerD Teildatenträger,  
SWAP Datenträger
```

Indem Sie DEFINE hinzufügen, übernehmen die vom Ausführungs-Controller-Server erstellten Agentenprozesse auch den für diese definierten Datenträger für den Auslagerungsspeicher; dies ermöglicht eine Skalierbarkeit der Auslagerungsspeicherbelegung, wenn mehrere Servertypen für Ausführungs-Controller verwendet werden.

Name des Standardstatusservers

Dem standardmäßigen Statusserverprozesspaar muss bei Erstellung eines WS-Managers ein eindeutiger Prozessname zugewiesen werden. Dieser Name wird über den obligatorischen Parameter `-s Statusservername` des Befehls `crtmqm` angegeben. Für diesen Parameter gibt es keinen Standardwert. Verwenden Sie einen Prozessnamen, der systemweit eindeutig ist und problemlos dem WS-Manager zugeordnet werden kann, dem er zugeordnet ist.

So können Sie den Namen des Standardstatusserverprozesses für einen WS-Manager ändern:

1. Stoppen Sie den WS-Manager.
2. Geben Sie für den Standard- und den Teildatenträger die Adresse der Daten-dateien des WS-Managers (in der Regel `<WS-Managername>D`) an.
3. Ändern Sie den Eintrag `DefaultStatusServerName` in der Zeilengruppe `Configuration` der Datei QMINI des WS-Managers, indem Sie den neuen Prozessnamen angeben.
4. Führen Sie PATHMON für den WS-Manager aus, indem Sie an der TACL-Eingabeaufforderung den folgenden Befehl eingeben:

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

Dabei steht `<pmon>` für den Namen des PATHMON-Prozesses für den WS-Manager.

WS-Manager erstellen

5. Führen Sie PATHCOM für den PATHMON-Prozess aus, indem Sie an der TACL-Eingabeaufforderung den folgenden Befehl eingeben:

```
PATHCOM $<pmon>
```

6. Laden Sie die PATHWAY-Konfiguration des WS-Managers, indem Sie in PATHCOM den folgenden Befehl eingeben:

```
START PATHWAY COOL
```

7. Ändern Sie die Serverklasse MQS-STATUS00, und setzen Sie den Prozess zurück, indem Sie in PATHCOM die folgenden Befehle eingeben:

```
ALTER SERVER MQS-STATUS00, DELETE PROCESS $<vorheriger_Name>  
ALTER SERVER MQS-STATUS00, PROCESS $<neuer_Name>
```

8. Speichern Sie die geänderte Konfiguration auf der Festplatte, indem Sie in PATHCOM die folgenden Befehle eingeben:

```
SHUTDOWN2  
EXIT
```

Der Namensänderung für den standardmäßigen Warteschlangenserverprozess ist damit abgeschlossen. Bei der nächsten Eingabe des Befehls **strmqm** wird der WS-Manager mit dem Namen des neuen Standardwarteschlangenservers gestartet.

Zusätzliche Statusserver hinzufügen und entfernen

Sie haben die Möglichkeit, zusätzliche Statusserver hinzuzufügen, wobei der vorhandene Standardstatusserver als Schablone verwendet werden kann. Der WS-Manager muss hierzu nicht erst gestoppt werden.

So fügen Sie einen neuen Statusserver hinzu:

Anmerkung: Bei der folgenden Beschreibung wird davon ausgegangen, dass der WS-Manager aktiv ist. Ist dies nicht der Fall, müssen Sie vor Beginn zunächst PATHMON starten und die PATHWAY-Konfiguration laden. Darüber hinaus fällt in diesem Fall Schritt 5 auf Seite 64 weg; speichern Sie stattdessen am Ende mit dem Befehl SHUTDOWN2 die PATHWAY-Konfiguration auf der Festplatte.

1. Führen Sie PATHCOM für den PATHMON-Prozess aus, indem Sie an der TACL-Eingabeaufforderung den folgenden Befehl eingeben:

```
PATHCOM $<pmon>
```

WS-Manager erstellen

- Erstellen Sie auf Basis der Standardstatusserverklasse eine Gruppe zulässiger Attribute, indem Sie in PATHCOM die folgenden Befehle eingeben:

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

- Ändern Sie die Attributgruppe für den neuen Server, indem Sie in PATHCOM die folgenden Befehle eingeben:

```
SET SERVER CPUS(n:m)
RESET SERVER PROCESS $<Name_des_standardmäßigen_Statusserverprozesses>
SET SERVER PROCESS $<Name_des_neuen_Statusservers>
```

- Fügen Sie den neuen Server unter einem neuen Serverklassennamen hinzu, indem Sie in PATHCOM den folgenden Befehl eingeben:

```
ADD SERVER MQS-STATUSxx
```

- Starten Sie den neuen Statusserver, damit er verwendet werden kann, und verlassen Sie PATHCOM, indem Sie die folgenden Befehle eingeben:

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

Die Namen für zusätzliche Serverklassen sollten dem Schema MQS-STATUS01, MQS-STATUS02 usw. folgen. Diese Namenskonvention ist jedoch nicht bindend. Sofern der Name einer Serverklasse mit der Zeichenfolge MQS-STATUS beginnt, wird sie bei Eingabe des Befehls **strmqm** gestartet.

Nachdem ein Statusserver hinzugefügt und gestartet wurde (entweder explizit unter Verwendung von PATHCOM oder implizit mit dem Befehl **strmqm**), können diesem neuen Server mit dem Befehl **altmqfls** Objekte zugewiesen werden. Weitere Information zur erneuten Zuordnung von Objekten finden Sie unter „Objekte Statusservern und Warteschlangenservern erneut zuordnen“ auf Seite 110.

Bevor ein Statusserver entfernt wird, sollten Sie zunächst sicherstellen, dass alle für diesen Server konfigurierten Objekte gelöscht oder einem anderen Statusserver zugeordnet wurden. Der Standardstatusserver darf **nicht** gelöscht werden, da andernfalls der WS-Manager nicht mehr funktionsfähig ist.

Name des Standardwarteschlangenservers

Dem standardmäßigen Warteschlangenserverprozesspaar muss bei Erstellung eines WS-Managers ein eindeutiger Prozessname zugewiesen werden. Dieser wird über den obligatorischen Parameter `-v Warteschlangenserver` des Befehls **crtmqm** angegeben. Für diesen Parameter gibt es keinen Standardwert. Verwenden Sie einen Prozessnamen, der systemweit eindeutig ist und problemlos dem WS-Manager zugeordnet werden kann, dem er zugeordnet ist.

So können Sie den Namen des standardmäßigen Warteschlangenserverprozesses für einen WS-Manager ändern:

1. Stoppen Sie den WS-Manager.
2. Geben Sie für den Standard- und den Teildatenträger die Adresse der Datendateien des WS-Managers (in der Regel `<WS-Managername>D`) an.
3. Ändern Sie den Eintrag `DefaultQueueServerName` in der Zeilengruppe `Configuration` der Datei `QMINI` des WS-Managers, indem Sie den neuen Prozessnamen angeben.
4. Führen Sie `PATHMON` für den WS-Manager aus, indem Sie an der `TACL`-Eingabeaufforderung den folgenden Befehl eingeben:

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

Dabei steht `<pmon>` für den Namen des `PATHMON`-Prozesses für den WS-Manager.

5. Führen Sie `PATHCOM` für den `PATHMON`-Prozess aus, indem Sie an der `TACL`-Eingabeaufforderung den folgenden Befehl eingeben:

```
PATHCOM $<pmon>
```

6. Laden Sie die `PATHWAY`-Konfiguration des WS-Managers, indem Sie in `PATHCOM` den folgenden Befehl eingeben:

```
START PATHWAY COOL
```

7. Ändern Sie die Serverklasse `MQS-QUEUEnn`, und setzen Sie den Prozess zurück, indem Sie in `PATHMON` die folgenden Befehle eingeben:

```
ALTER SERVER MQS-QUEUEnn, DELETE PROCESS $<vorheriger_Name>  
ALTER SERVER MQS-QUEUEnn, PROCESS $<neuer_Name>
```

- Speichern Sie die geänderte Konfiguration auf der Festplatte, indem Sie in PATHCOM die folgenden Befehle eingeben:

```
SHUTDOWN2  
EXIT
```

Der Namensänderung für den standardmäßigen Warteschlangenserverprozess ist damit abgeschlossen. Bei der nächsten Eingabe des Befehls **strmqm** wird der WS-Manager mit dem Namen des neuen Standardwarteschlangenservers gestartet.

Zusätzliche Warteschlangenserver hinzufügen und entfernen

Sie haben die Möglichkeit, zusätzliche Warteschlangen hinzuzufügen, wobei der vorhandene Standardwarteschlangenserver als Schablone verwendet werden kann. Der WS-Manager muss hierzu nicht erst gestoppt werden.

So fügen Sie einen neuen Warteschlangenserver hinzu:

Anmerkung: Bei der folgenden Beschreibung wird davon ausgegangen, dass der WS-Manager aktiv ist. Ist dies nicht der Fall, müssen Sie vor Beginn zunächst PATHMON starten und die PATHWAY-Konfiguration laden. Darüber hinaus fällt in diesem Fall Schritt 5 auf Seite 64 weg; speichern Sie stattdessen am Ende mit dem Befehl SHUTDOWN2 die PATHWAY-Konfiguration auf der Festplatte.

- Führen Sie PATHCOM für den PATHMON-Prozess aus, indem Sie an der TACL-Eingabeaufforderung den folgenden Befehl eingeben:

```
PATHCOM $<pmon>
```

- Erstellen Sie auf Basis der Serverklasse für den Standardwarteschlangenserver eine Gruppe zulässiger Attribute, indem Sie in PATHCOM die folgenden Befehle eingeben:

```
RESET SERVER  
SET LIKE MQS-QUEUEn
```

- Ändern Sie die Attributgruppe für den neuen Server, indem Sie in PATHCOM die folgenden Befehle eingeben:

```
SET SERVER CPUS(n:m)  
RESET SERVER PROCESS $<Name_des_Standard-Warteschlangenserverprozesses>  
SET SERVER PROCESS $<Name_des_neuen_Warteschlangenservers>
```

WS-Manager erstellen

4. Fügen Sie den neuen Server unter einem neuen Serverklassenamen hinzu, indem Sie in PATHCOM den folgenden Befehl eingeben:

```
ADD SERVER MQS-QUEUEnn
```

5. Starten Sie den neuen Warteschlangenserver, damit er verwendet werden kann, und verlassen Sie PATHCOM, indem Sie die folgenden Befehle eingeben:

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

Die Namen für zusätzliche Serverklassen sollten dem Schema MQS-QUEUE01, MQS-QUEUE02 usw. folgen. Diese Namenskonvention ist jedoch nicht bindend. Sofern der Name einer Serverklasse mit der Zeichenfolge MQS-QUEUE beginnt, wird sie bei Eingabe des Befehls **strmqm** gestartet.

Nachdem ein Warteschlangenserver hinzugefügt und gestartet wurde (entweder explizit unter Verwendung von PATHCOM oder implizit mit dem Befehl **strmqm**), können diesem neuen Server mit dem Befehl **altnmqfls** Objekte zugewiesen werden. Weitere Information zur erneuten Zuordnung von Objekten finden Sie unter „Objekte Statusservern und Warteschlangenservern erneut zuordnen“ auf Seite 110.

Bevor ein Warteschlangenserver entfernt wird, sollten Sie zunächst sicherstellen, dass alle für diesen Server konfigurierten Objekte gelöscht oder einem anderen Warteschlangenserver zugeordnet wurden. Der Standardwarteschlangenserver darf **nicht** gelöscht werden, da andernfalls der WS-Manager nicht mehr funktionsfähig ist.

Datenträgerstruktur

Die Dateien für MQSeries for Compaq NSK sind auf mehrere Teildatenträger verteilt. Der Datenträger, der diese Teildatenträger enthält, wird bei der Erstellung des WS-Managers ausgewählt; dazu wird entweder der in der Datei MQSINI angegebene Standarddatenträger oder der im Parameter *-p Standardpräfix* des Befehls **crtnmqm** angegebene Wert übernommen.

Für jeden WS-Manager sind fünf Teildatenträger vorhanden. Über den Inhalt dieser Datenträger entscheidet das letzte Zeichen des Teildatenträgernamens. Beispielsweise sind für einen WS-Manager mit dem Namen QMGR im Datenträger \$DATA folgende Teildatenträger vorhanden:

\$DATA.QMGR	Teildatenträger mit FFST-Dateien
\$DATA.QMGRD	Teildatenträger mit den Datendateien des WS-Managers
\$DATA.QMGRL	Teildatenträger mit den Fehlerprotokollen für den WS-Manager
\$DATA.QMGRM	Teildatenträger für Nachrichtenwarteschlangen
\$DATA.QMGRS	Teildatenträger für Kanalsynchronisation

Besteht der Name des WS-Managers aus mehr als sieben Zeichen, werden die Namen der Teildatenträger geändert oder gekürzt. In den Zeilengruppen QMVolume und QMSubvolume für den WS-Manager in der Datei MQSINI werden Adressen und Namen dieser Teildatenträger angegeben.

Teildatenträger mit den FFST-Dateien des WS-Managers

Der FFST-Teildatenträger enthält FFST-Dateien (First Failure Support Technology). Diesen Dateien beginnen mit dem Präfix FD. Sie weisen auf ernste Probleme im MQSeries-System hin, wie beispielsweise Ressourcenknappheit, interne MQSeries-Fehler oder Störungen im Compaq NSK-System.

Teildatenträger mit den Datendateien des WS-Managers

AMQRFNxx	Die Repository-Cache-Dateien im gemeinsam benutzten Speicher für die einzelnen CPUs (Nummer xx).
CCHDEFS	Die Datei mit den Definitionen für Client-Verbindungskanäle.
CCSIDMEM	Dateien im schreibgeschützten gemeinsam benutzten Speicherbereich für CCSID-Unterstützung bei Datenkonvertierungen.
CHDEFS	Die Datei mit den Kanaldefinitionen. Sie enthält Konfigurationsdaten zu den einzelnen Kanälen, die für den WS-Manager definiert sind.
Lxxxxxxx	Dateien mit Namenslisten. Lxxxxxxx wird vom Namen des Objekts abgeleitet; andernfalls handelt es sich um einen generierten Wert.
OAMDB	Die Berechtigungsdatenbank.
OBJCAT	Der Objektkatalog.
ABJCAT	Die Datei mit Alternativschlüsseln für den Objektkatalog OBJCAT, der einen Index enthält, der auf den Objektamen basiert.
PATHCTL	Die PATHWAY-Steuerdatei.
PRIDB	Die MQSeries-Principal-Datenbank (MQSeries-Benutzerdatenbank).
PRIDBA	Die Datei mit den Alternativschlüsseln für PRIDB.
QMINI	Die Initialisierungsdatei des WS-Managers.
QMINIMEM	Datei im schreibgeschützten gemeinsam benutzten Speicherbereich, die die allgemeinen statischen Konfigurationsdaten des WS-Managers enthält.
REPMGR	Eine Datei, mit der der Start mehrerer Repository-Serverprozesse koordiniert wird.
RDFPURGE	Die Datenbank, in die der WS-Manager logisch gelöschte Dateien einträgt, die nach einer gewissen Zeit mit dem Befehl cleanrdf gelöscht werden.
RUNTIME	Datei, die für die Koordination bei der Wiederherstellung des ECBOSS und der Ausführungs-Controller verwendet wird.
SHUTDOWN	Datei, über die die Ausführung von endmqm gesteuert wird.
STATABLE	Die Datei mit der Kanalstatustabelle. Diese Datei enthält dynamische Kanalstatusdaten. In ihr werden Kanalstatusdaten über den Abschluss und Neustart eines WS-Managers hinweg gespeichert.

Datenträgerstruktur

TRACEOPT	Diese Datei enthält die aktuellen Trace-Einstellungen für den WS-Manager in Form einer unformatierten Bitmap. Die Steuerbefehle strmqtrc und endmqtrc bewirken Änderungen am Inhalt dieser Datei, wobei alle Prozesse über den Mechanismus CONTROL 27 über die Aktualisierung informiert werden.
UMQSINI	Enthält eine Momentaufnahme (Snapshot) der unstrukturierten Datei MQSINI beim Start des WS-Managers.

Datenträger mit dem Fehlerprotokoll des WS-Managers

Dieser Teildatenträger enthält die Fehler- und Trace-Protokolldateien. Diese Dateien beginnen mit dem Präfix TR. (Das Präfix kann über den Eintrag TracePrefix in der Datei QMINI geändert werden.) Trace-Dateien enthalten Diagnoseinformationen und werden nur erstellt, wenn die Trace-Funktion mit der MQM-Funktion oder mit dem Steuerbefehl **strmqtrc** aktiviert wurde.

Die Namen der Fehlerprotokolldateien haben das Format MQERRLGn; dabei steht *n* für 1, 2 oder 3. Bei MQERRLG1 handelt es sich immer um das aktuelle Fehlerprotokoll. Ist diese Datei voll, wird der Inhalt in die Datei MQERRLG2 verschoben; wenn MQERRLG1 das nächste Mal gelehrt wird, wird der Inhalt aus Datei MQERRLG2 in Datei MQERRLG3 verschoben. MQERRLG3 wird bei Bedarf überschrieben. Es gibt nur jeweils drei Fehlerprotokolle; daher muss ihre Größe richtig gewählt werden, da andernfalls hilfreiche Fehlerinformationen verloren gehen.

Teildatenträger für Nachrichtenwarteschlangen des WS-Managers

Dieser Teildatenträger enthält Dateien, die zum Speichern von Nachrichten in lokalen Warteschlangen verwendet werden. Die Dateinamen haben das folgende Format:

Qxxxxxxx	Dabei handelt es sich um die Warteschlangendateien selbst, die permanente Nachrichten enthalten.
Txxxxxxx	Die Änderungsdateien. Wird ein Objekt geändert, ändern diese Dateien die Datumsmarke des Objekts. Txxxxxxx wird vom Namen des Objekts abgeleitet; andernfalls handelt es sich um einen generierten Wert.
Oxxxxxxx	Die Datei, die bei Warteschlangenüberlauf verwendet wird.
xxxxxxx	Name wird vom Namen der Warteschlange abgeleitet, falls es sich um einen eindeutigen, kurzen Namen handelt; andernfalls handelt es sich um einen generierten Wert (siehe „Umwandlung von Objektname“ auf Seite 67).

Zusätzlich zu den oben beschriebenen Dateien, deren Namen mit einem bestimmten Präfix beginnen, werden auf dem Teildatenträger mit den Nachrichtenwarteschlangen außerdem noch Nachrichtenüberlaufdateien erstellt. In diesen Dateien werden große Nachrichten gespeichert; ihre Namen enthalten zwar kein bestimmtes Präfix, sie sind aber leicht erkennbar, da es sich um unstrukturierte Dateien handelt, die nicht im Prüfprotokoll aufgezeichnet werden. Die Adresse neuer Nachrichtenüberlaufdateien kann für die einzelnen Warteschlangen mit dem Befehl **altmqfls** unter Angabe des Parameters `--msgofsubvol` geändert werden. (Weitere Informationen finden Sie unter „dspmqfls (MQSeries-Dateiattribute anzeigen)“ auf Seite 284.)

Teildatenträger für Kanalsynchronisation des WS-Managers

Dieser Teildatenträger enthält interne Datenbanken, in denen der Status von Arbeitseinheiten (oder Nachricht Stapel) aufgezeichnet wird, die über die dem WS-Manager zugeordneten Kanäle übertragen oder empfangen werden.

Nachdem Kanäle in einem WS-Manager einmal verwendet wurden, enthält dieser Teildatenträger die folgenden Dateien:

SYNCHIDX

Die Synchronisationsindexdatei. Sie enthält je einen Eintrag für jede vom WS-Manager erstellte Synchronisationsdatei.

Sxxxxxxx

Einzelne Synchronisationsdateien. Für jede Kombination aus lokalem und fernem Kanal, die im WS-Manager verwendet wurde, ist eine Datei vorhanden. Diese Dateien enthalten die IDs der Nachrichten, die innerhalb eines Nachrichtstapels übertragen oder empfangen wurden. Diese Informationen werden nach einem Ausfall und der Auflösung von Kanälen mit unbestätigtem Status zur Resynchronisation der Kanäle verwendet.

Umwandlung von Objektnamen

Bei Objektnamen handelt es sich nicht notwendigerweise um gültige Dateinamen. Daher müssen Objektnamen unter Umständen umgewandelt werden. Dieses Verfahren unterscheidet sich von dem für die Umwandlung von WS-Managernamen, da für jeden WS-Manager eine große Anzahl anderer Objekte vorhanden sein kann, selbst wenn jedes System nur wenige WS-Manager enthält. Dies betrifft nur Prozessdefinitionen, Namenslisten und Warteschlangen, für die jeweils eigene Dateien im Dateisystem verwendet werden; Kanäle und andere Objekte sind von diesen Überlegungen nicht betroffen, da sie in Form von Datensätzen in Datenbanken gespeichert werden, die eine Vielzahl von Objektdefinitionen enthalten.

Der durch den Umwandlungsprozess generierte Name hat keine Beziehung zu dem ursprünglichen Objektnamen. Mit dem Befehl **dspmqfls** können Sie den ursprünglichen Namen eines Objekts anzeigen; mit diesem Befehl werden die einem MQSeries-Objekt zugeordneten wichtigsten Dateien angezeigt.

Mit WS-Managern arbeiten

MQSeries stellt Steuerbefehle zum Erstellen, Starten, Beenden und Löschen von WS-Managern zur Verfügung. Darüber hinaus können Sie die Attribute eines WS-Managers mit dem MQSC-Befehl **DISPLAY QMGR** anzeigen und mit dem MQSC-Befehl **ALTER QMGR** ändern. (Siehe „WS-Managerattribute anzeigen“ auf Seite 99 und „WS-Managerattribute ändern“ auf Seite 100.)

Vor der Verwendung der Steuerbefehle sollten Sie zunächst sicherstellen, dass in der Umgebungsvariable **PMSEARCHLIST** die Adresse Ihrer ausführbaren MQSeries-Module angegeben ist. Weitere Informationen zu diesen Umgebungsvariablen finden Sie in „Anhang C. TACL-Umgebungsvariablen für MQSeries for Compaq NSK setzen“ auf Seite 329.

Standard-WS-Manager erstellen

Ein Standard-WS-Manager wird mit dem Befehl **crtmqm** erstellt. Die Eingabe des Befehls **crtmqm** mit der Option **-q** bewirkt Folgendes:

- Es wird ein Standard-WS-Manager mit dem Namen `saturn.queue.manager` erstellt.
- Es werden Standard- und Systemobjekte erstellt.
- Die dem WS-Manager zugeordnete Standardübertragungswarteschlange und Warteschlange für nicht zustellbare Nachrichten werden angegeben.

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

Dabei gilt:

- q** Gibt an, dass es sich bei diesem WS-Manager um den Standard-WS-Manager handelt.
- d MY.DEF.XMIT.Q**
Gibt den Namen der Standardübertragungswarteschlange an.
- u SYSTEM.DEAD.LETTER.QUEUE**
Gibt den Namen der Warteschlange für nicht zustellbare Nachrichten an.
- n \$PMON**
Gibt den Prozessnamen von PATHMON für den WS-Manager an.
- o \$TRM01**
Gibt den Namen des Ausgangsterminals an (dieses muss sich im Ruhestatus befinden).
- s \$MQSS**
Gibt den Prozessnamen des Standardstatusservers an.
- v \$MQQS**
Gibt den Prozessnamen des Standardwarteschlangenservers an.
- saturn.queue.manager**
Gibt den Namen dieses WS-Managers an. Im Befehl **crtmqm** muss es sich hierbei um den letzten Parameter handeln.

Es wird ein WS-Manager mit dem von Ihnen angegebenen Namen und den angegebenen Optionen erstellt. Bei der Erstellung eines WS-Managers wird automatisch auch Folgendes erstellt:

- der standardmäßige MQSeries-Principal für den Ersteller des WS-Managers,
- der Statusserver für den WS-Manager,
- der Warteschlangenserver für den WS-Manager,
- eine Konfigurationsdatei für den WS-Manager.

Sie können den WS-Manager jetzt starten. Weitere Informationen finden Sie in „WS-Manager starten“ auf Seite 70.

Unter Umständen müssen einige Attribute eines WS-Managers geändert werden. Dies ist mit der MQM-Funktion oder dem Steuerbefehl `strmqm` möglich. Weitere Informationen hierzu finden Sie in Abb. 5 auf Seite 75 oder unter „`strmqm` (WS-Manager starten)“ auf Seite 313.

Sie sollten für die bei der Erstellung des WS-Managers generierten beiden Konfigurationsdateien eine Sicherungskopie erstellen. Wird ein weiterer WS-Manager erstellt, bei dem es zu Problemen kommt, können Sie die Sicherungskopien wiederherstellen. Weitere Informationen finden Sie in „Sicherungskopien von Konfigurationsdateien nach Erstellung eines WS-Managers erstellen“ auf Seite 50.

MQSeries-Principals erstellen

Der `crtmqm` erstellt automatisch einen Principal für den Benutzer, der den WS-Manager erstellt. Aus Gründen der Kompatibilität mit anderen MQSeries-Implementierungen erhält dieser Principal, der so genannte Standard-Principal, immer den Namen `mqm`.

Nach der Erstellung eines WS-Managers können Sie Principals für weitere MQSeries-Benutzer erstellen. Dieser Schritt kann zu jedem beliebigen Zeitpunkt erfolgen, unabhängig davon, ob der WS-Manager gestartet wurde. Sind keine weiteren Benutzer für den WS-Manager erforderlich, kann dieser Schritt entfallen.

Um den MQSeries-Principal `MQPRINCIPAL` erstellen, der dem Compaq NSK-Benutzer `MQM.MQUSER` entspricht, müssen Sie den folgenden Befehl eingeben:

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

Um alle momentan vorhandenen Principals anzuzeigen, geben Sie den folgenden Befehl ein:

```
dspmqusr -m saturn.queue.manager
```

Wird für einen Benutzer kein Principal-Eintrag erstellt, führen alle Zugriffe dieses Benutzers auf den WS-Manager zu einem Berechtigungsfehler (unabhängig davon, ob der Objektberechtigungsmanager aktiviert ist oder nicht). Diese Änderung wurde in MQSeries for Compaq NSK Version 2.2.0.1 eingeführt und besteht in allen nachfolgenden Versionen.

'cleanrdf' für einen RDF-aktivierten WS-Manager ausführen

Ist ein WS-Manager in einer RDF-Umgebung aktiv, der über die Umgebungsvariable MQRDF RDF-aktiviert wurde, sollten Sie regelmäßig das Dienstprogramm **cleanrdf** wie folgt ausführen:

- Nach Änderungen an der Konfiguration (z. B. nachdem Objekte erstellt oder gelöscht wurden oder nach Änderungen an der Datei QMINI) sollte **cleanrdf** ausgeführt werden.
- Wenn Ihre Anwendung im Verlauf der Ausführung Objekte erstellt und löscht und vor allem dynamische Warteschlangen verwendet, sollte das Dienstprogramm **cleanrdf** während des normalen Betriebs ausgeführt werden; wie häufig dies geschehen sollte, hängt davon ab, wie oft Objekte gelöscht werden. Hierzu sollte NetBatch oder eine andere Scheduler-Software verwendet werden.

WS-Manager starten

Nach seiner Erstellung kann ein WS-Manager Befehle oder MQI-Aufrufe erst verarbeiten, nachdem er gestartet wurde. Starten Sie den WS-Manager, indem Sie den folgenden Befehl eingeben:

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

Der Befehl **strmqm** gibt die Steuerung erst zurück, nachdem der WS-Manager gestartet wurde und für Verbindungsanforderungen empfangsbereit ist.

Standard- und Systemobjekte wiederherstellen

Standard- und Systemobjekte werden bei der Erstellung eines WS-Managers automatisch erstellt; diese Definitionen können jedoch jederzeit durch andere Objektdefinitionen ersetzt werden. Um die Standard- und Systemobjekte für einen WS-Manager namens **saturn.queue.manager** wiederherzustellen, geben Sie den Befehl **strmqm** mit der Option **-c** ein:

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

Der WS-Manager wird daraufhin gestartet, und die Standard- und Systemobjekte, die bei der Erstellung des WS-Managers generiert wurden, werden wiederhergestellt; anschließend wird der WS-Manager gestoppt.

Objektdateien

Für jede MQSeries-Warteschlange, jeden WS-Manager und jedes Prozessobjekt ist eine eigene Datei vorhanden. Da es sich bei den Namen dieser Objekte nicht unbedingt um gültige Dateinamen handelt, konvertiert der WS-Manager den Objekt-namen bei Bedarf in einen gültigen Dateinamen. Dieser Vorgang wird unter „Umwandlung von Objekt-namen“ auf Seite 67 beschrieben.

WS-Manager stoppen

Um einen WS-Manager zu stoppen, geben Sie den Befehl **endmqm** ein. Beispielsweise wird ein WS-Manager mit dem Namen `saturn.queue.manager` wie folgt beendet:

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

Standardmäßig wird bei Eingabe dieses Befehls ein *gesteuerter* Abschluss des angegebenen WS-Managers ausgeführt. Dieser Vorgang kann einige Zeit beanspruchen, da der Abschluss erst ausgeführt wird, nachdem die Verbindung *aller* verbundenen Anwendungen beendet und alle aktiven Kanäle gestoppt wurden.

Eine Beschreibung der Optionen für den **endmqm**, über die angegeben wird, wie der Abschluss ausgeführt werden soll, finden Sie unter „Sofortiger und erzwungener Abschluss eines WS-Managers“.

Maßnahmen bei eventuellen Problemen

Probleme, die beim Beenden eines WS-Managers auftreten können, werden häufig von Anwendungen verursacht. So kommt es beispielsweise zu Problemen, wenn Anwendungen:

- den MQI-Rückkehrcode nicht sorgfältig überprüfen,
- keinen Hinweis auf einen gesteuerten Abschluss anfordern.

Sofortiger und erzwungener Abschluss eines WS-Managers

Dauert der Abschluss eines WS-Managers zu lange oder fährt er nicht herunter, können Sie ihn mit dem Befehl **endmqm** beenden, wobei Sie **BREAK** gefolgt von **STOP** verwenden. Anschließend können Sie den Befehl **endmqm** noch einmal eingeben, dieses Mal jedoch unter Angabe der Option für einen sofortigen oder erzwungenen Abschluss.

Bei einem *sofortigen Abschluss* können alle aktuellen MQI-Aufrufe noch abgeschlossen werden, neue Aufrufe schlagen jedoch fehl. Der Abschluss erfolgt sofort, unabhängig davon, ob noch Anwendungen mit dem WS-Manager verbunden sind oder nicht. Für einen sofortigen Abschluss geben Sie den folgenden Befehl ein:

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

Ist kein sofortiger Abschluss möglich, können Sie es mit einem *erzwungenen* Abschluss versuchen; hierzu wird die Option **-p** angegeben. Beispiel:

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

Mit WS-Managern arbeiten

Achtung

Ein erzwungener Abschluss sollte nur ausgeführt werden, wenn alle anderen Versuche, den WS-Manager mit dem Befehl **endmqm** zu beenden, fehlgeschlagen sind. Ein erzwungener Abschluss kann unvorhersehbare Auswirkungen auf Anwendungen haben, die noch mit dem WS-Manager verbunden sind.

Hat auch ein erzwungener Abschluss nicht das gewünschte Ergebnis, lesen Sie unter „WS-Manager manuell stoppen“ auf Seite 337 nach; dort werden weitere Möglichkeiten beschrieben, wie der WS-Manager beendet werden kann.

Ein ausführliche Beschreibung des Befehls **endmqm** und seiner Optionen finden Sie unter „endmqm (WS-Manager beenden)“ auf Seite 292.

WS-Manager erneut starten

Um einen WS-Manager erneut zu starten, geben Sie folgenden Befehl ein:

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

Vorhandenen WS-Manager als Standard-WS-Manager definieren

Bei der Erstellung eines Standard-WS-Managers wird der Name dieses Managers in die Zeilengruppe *DefaultQueueManager* der MQSeries-Konfigurationsdatei (MQSINI) eingefügt. Falls noch nicht vorhanden, werden diese Zeilengruppe und ihr Inhalt automatisch erstellt.

Diese Zeilengruppe muss unter Umständen geändert werden, beispielsweise in den folgenden Fällen:

- Ein vorhandener WS-Manager soll als Standard-WS-Manager eingerichtet werden. Dazu müssen Sie den in dieser Zeilengruppe angegebenen WS-Managernamen durch den Namen des neuen Standard-WS-Managers ersetzen. Diese Änderung muss manuell mit einem Texteditor vorgenommen werden.
- Auf dem Knoten ist kein Standard-WS-Manager vorhanden, daher soll ein vorhandener WS-Manager als Standard-WS-Manager definiert werden. Dazu müssen Sie die Zeilengruppe *DefaultQueueManager* (mit dem Namen des betreffenden WS-Managers) selbst erstellen.
- Ein anderer WS-Manager wurde versehentlich als Standard-WS-Manager definiert, und Sie möchten den ursprünglichen Standard-WS-Manager wieder einrichten. Dazu müssen Sie die Zeilengruppe *DefaultQueueManager* in der MQSeries-Konfigurationsdatei ändern, indem Sie den Namen des nicht erwünschten WS-Managers durch den Namen des WS-Managers ersetzen, der als Standard-WS-Manager eingesetzt werden soll.

Der angegebene WS-Manager ist jetzt der Standard-WS-Manager. Bei allen weiteren Verbindungsanforderungen, in denen die Anwendung keinen bestimmten WS-Manager angibt, wird eine Verbindung zu diesem neuen Standard-WS-Manager hergestellt.

Weitere Informationen zu Konfigurationsdateien finden Sie in „Kapitel 13. Konfigurationsdateien“ auf Seite 193.

Nachdem die erforderlichen Änderungen an der Zeilengruppe vorgenommen wurde, stoppen Sie den WS-Manager, und starten Sie ihn anschließend erneut.

WS-Manager löschen

Um einen WS-Manager zu löschen, müssen Sie ihn zunächst stoppen und anschließend den folgenden Befehl eingeben:

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

Achtung

Wenn Sie einen WS-Manager löschen, werden dabei auch alle ihm zugeordneten Ressourcen wie Warteschlangen und die darin enthaltenen Nachrichten sowie alle Objektdefinitionen gelöscht. Außerdem wird der Inhalt aller Teildatenträger gelöscht, die für den WS-Manager angelegt wurden (auch die, die nicht von MQSeries erstellt wurden).

Ein Beschreibung des Befehls **dltmqm** und seiner Optionen finden Sie unter „dltmqm (WS-Manager löschen)“ auf Seite 277. Stellen Sie sicher, dass nur vertrauenswürdige Administratoren über eine Berechtigung zur Ausführung dieses Befehls verfügen.

Kann ein WS-Manager nicht auf normalem Weg beendet werden, lesen Sie unter „WS-Manager manuell entfernen“ auf Seite 337 nach; dort werden weitere Möglichkeiten beschrieben, wie ein WS-Manager beendet werden kann.

MQM (Message Queue Management) verwenden

Die Funktion MQM (Message Queue Management) in MQSeries for Compaq NSK, V5.1 wird als PATHWAY-SCOBOL-Requester unter TCP (Terminal Control Process) ausgeführt. Dieses Tool verwendet einen Server der Serverklasse MQM, der eine C-Anwendungsprogrammierschnittstelle aufruft.

Für jeden auf einem System konfigurierten WS-Manager ist ein eigenes MQM-Exemplar vorhanden, da jeder WS-Manager über eine eigene PATHWAY-Konfiguration verfügt. Daher kann mit MQM nur der jeweils zugeordnete WS-Manager verwaltet werden.

Anmerkung: Standardmäßig können bis zu 10 Benutzer gleichzeitig die MQM-Funktion verwenden. Soll die Zahl beispielsweise auf 20 Benutzer erhöht werden, geben Sie in der PATHWAY-Umgebung des WS-Managers Folgendes ein:

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

Weitere Informationen finden Sie in „Kapitel 3. MQSeries-Befehlsätze verwenden“ auf Seite 31.

Message Queue Management (MQM)

Anmerkung:

Um MQM aufzurufen, geben Sie an der PATHCOM-Eingabeaufforderung des WS-Managers den Befehl `run mqmc` ein.

Daraufhin wird das Hauptmenü von MQM (MQM Main Menu) geöffnet:

```
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.
```

Abbildung 4. MQSeries for Compaq NSK - MQM-Hauptmenü

In diesem Menü können Sie eines der folgenden Untermenüs auswählen:

1. Queue Manager (WS-Manager)
2. Queues (Warteschlangen)
3. Channels (Kanäle)

Im Folgenden sollen diese Untermenüs beschrieben werden. Über **Alt+F6** können Sie jederzeit zum MQM-Hauptmenü zurückkehren. Mit der Rückkehrtaste (F16) kehren Sie in die vorherige Anzeige zurück. Bei Betätigen der Funktionstaste F16 im **MQM Main Menu** wird MQM beendet.

'Queue Manager Menu' (WS-Managermenü)

Um die Option **Queue Manager** auszuwählen, geben Sie 1 im Feld **Enter Choice** (Auswahl eingeben) im **MQM Main Menu** ein, und drücken Sie die Taste F1 (Eingabetaste). Daraufhin wird die Anzeige **Queue Manager Menu** (WS-Managermenü) geöffnet:

```

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
    
```

Abbildung 5. Die Anzeige 'Queue Manager Menu'

Drücken Sie die Taste **Bild-Ab**, um den zweiten Teil dieser Anzeige anzuzeigen.

```

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
    
```

Abbildung 6. Die Anzeige 'Queue Manager Menu', Teil 2

In dieser Anzeige haben Sie folgende Möglichkeiten:

- Sie können einige Attribute des WS-Managers ändern.
- Sie können die Trace-Funktion für MQSeries-Objekte steuern.

WS-Managerattribute ändern

Nehmen Sie in der Anzeige **Queue Manager Menu** die gewünschten Änderungen vor, indem Sie die betreffenden Werte überschreiben, und drücken Sie die Taste F1 (Modify, Ändern). Die Werte der Attribute, die nicht geändert werden können, können auch nicht überschrieben werden.

Message Queue Management (MQM)

Trace für MQSeries-Objekte durchführen

Drücken Sie die Taste F2 (Trace), um das Menü **QUEUE MANAGER TRACE MENU** (Menü 'WS-Manager-Trace') zu öffnen:

```
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
```

Abbildung 7. Die Anzeige 'QUEUE MANAGER TRACE MENU'

Folgende Trace-Optionen stehen zur Auswahl:

- API** Für Trace-Punkte, die der MQI und Hauptkomponenten des WS-Managers zugeordnet sind, werden Daten ausgegeben.
- COMMS** Für Trace-Punkte, die Datenflüssen in Kommunikationsnetzen zugeordnet sind, werden Daten ausgegeben.
- CSFLOWS** Für Trace-Punkte, die Verarbeitungsabläufen in allgemeinen Services zugeordnet sind, werden Daten ausgegeben.
- LQMFLows** Für Trace-Punkte, die Verarbeitungsabläufen im lokalen WS-Manager zugeordnet sind, werden Daten ausgegeben.
- REMOTEFLows** Für Trace-Punkte, die Verarbeitungsabläufen in der DFV-Komponente zugeordnet sind, werden Daten ausgegeben.
- ADMINFLow** Für Trace-Punkte, die internen Datenpuffern für Verwaltungszwecke zugeordnet sind, werden Daten ausgegeben.
- OTHERFLows** Für Trace-Punkte, die den Verarbeitungsabläufen anderer Komponenten zugeordnet sind, werden Daten ausgegeben.
- CSDATA** Für Trace-Punkte, die internen Datenpuffern in allgemeinen Services zugeordnet sind, werden Daten ausgegeben.
- LQMData** Für Trace-Punkte, die internen Datenpuffern im lokalen WS-Manager zugeordnet sind, werden Daten ausgegeben.

Message Queue Management (MQM)

REMOTEDATA

Für Trace-Punkte, die internen Datenpuffern in der DFV-Komponente zugeordnet sind, werden Daten ausgegeben.

ADMINDATA

Für Trace-Punkte, die internen Datenpuffern in der DFV-Komponente zugeordnet sind, werden Daten ausgegeben.

OTHERDATA Für Trace-Punkte, die internen Datenpuffern anderer Komponenten zugeordnet sind, werden Daten ausgegeben.

VERSIONDATA

Für Trace-Punkte, die der aktiven MQSeries-Version zugeordnet sind, werden Daten ausgegeben.

COMMENTARY

Für Trace-Punkte, die Kommentaren in den MQSeries-Komponenten zugeordnet sind, werden Daten ausgegeben.

ALL

Trace-Punkte werden aktiviert und es wird ein vollständiger Trace durchgeführt.

Um einen Trace für bestimmte Komponenten zu starten bzw. zu stoppen, geben Sie zunächst vor dem Namen der betreffenden Komponente(n) ein beliebiges Zeichen ein.

Drücken Sie die Taste F1 (Start Trace, Trace starten), um einen Trace für die ausgewählten Komponenten zu starten. Drücken Sie die Taste F2 (Stop Trace, Trace stoppen), um den Trace für die ausgewählten Komponenten zu beenden.

Anzeige 'Queues Menu'

Um die Option **Queues** auszuwählen, geben Sie 2 im Feld **Enter Choice** (Auswahl eingeben) im **MQM Main Menu** ein, und drücken Sie die Taste F1 (Eingabetaste). Daraufhin wird die Anzeige **Search Criteria** (Suchkriterien) geöffnet:

```
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
```

Abbildung 8. Die Anzeige 'Search Criteria' (Warteschlangen)

Message Queue Management (MQM)

Geben Sie in dieser Anzeige im Feld **Queue Name** (Name der Warteschlange) den Namen der betreffenden Warteschlange vollständig oder teilweise ein. Sie können auch eine ID für den Warteschlangentyp eingeben, wenn nur nach bestimmten Warteschlangen gesucht werden soll. Drücken Sie anschließend die Taste F1 (Eingabetaste). Daraufhin wird die Anzeige **Queue Menu** (Warteschlangenmenü) geöffnet, in der Sie MQSeries-Warteschlangen anzeigen, ändern, erstellen, kopieren, löschen und überwachen können.

```
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.SDRC.LOCAL                       QLOCAL
- ANNE.ET01.SDRC.REMOTE                      QREMOTE
- ANNE.M401.RQSD.LOCAL                       QLOCAL
- ANNE.M401.RQSD.REMOTE                      QREMOTE
- ANNE.M401.RQSV.LOCAL                      QLOCAL
- ANNE.M401.RQSV.REMOTE                     QREMOTE
- ANNE.M401.SDRC.LOCAL                      QLOCAL
- ANNE.M401.SDRC.REMOTE                     QREMOTE

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

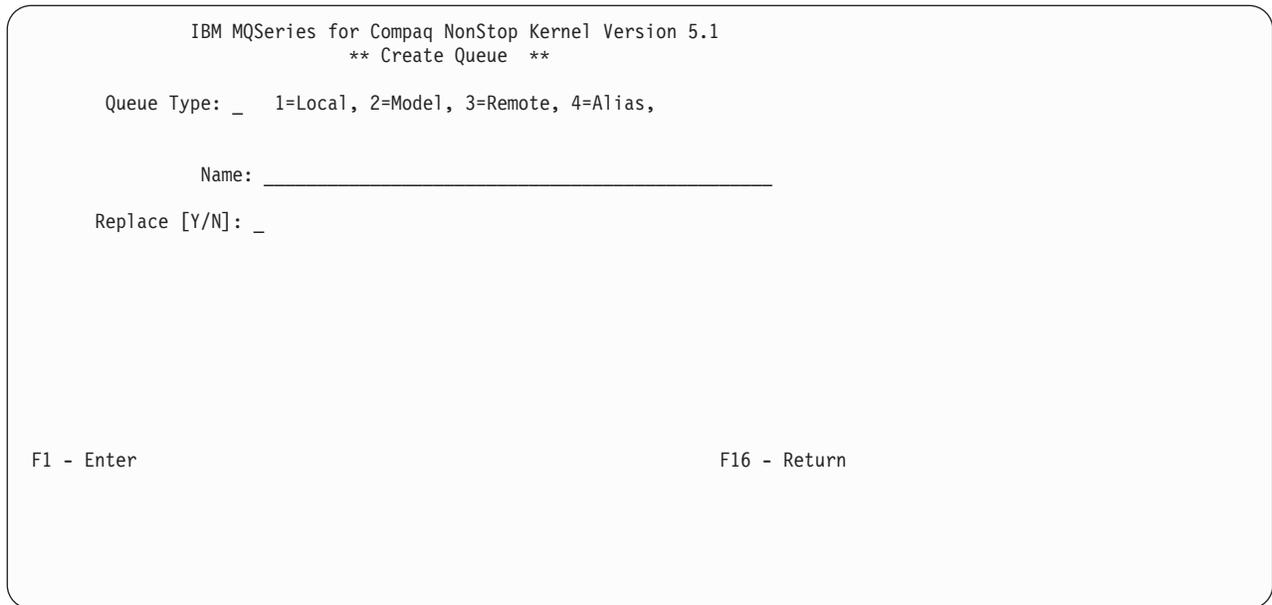
Abbildung 9. Die Anzeige 'Queue Menu'

Anmerkung: Es können nur Warteschlangen für den WS-Manager erstellt, geändert und gelöscht werden, dem der von Ihnen momentan verwendete MQM-Requester zugeordnet ist.

Mit den Tasten **Bild-Auf** und **Bild-Ab** können Sie die Liste mit Warteschlangen nach unten bzw. oben verschieben.

Warteschlange erstellen

Drücken Sie in der Anzeige **Queues Menu** (Warteschlangenmenü) die Taste F2 (Create, Erstellen), um die Anzeige **Create Queue** zu öffnen:



```
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                                F16 - Return
```

Abbildung 10. Die Anzeige 'Create Queue'

So erstellen Sie eine neue Warteschlangendefinition:

1. Geben Sie im Feld **Queue Type** (Warteschlangentyp) eine 1 (lokale Warteschlange), eine 2 (Modellwarteschlange), eine 3 (ferne Warteschlange) oder eine 4 (Aliaswarteschlange) ein.
2. Geben Sie im Feld **Name** den Namen des WS-Managers ein.
3. Soll die Warteschlange eine bereits vorhandene Warteschlange desselben Namens und desselben Typs ersetzen, geben Sie Y (Ja) im Feld **Replace** (Ersetzen) ein.
4. Drücken Sie anschließend die Taste F1 (Eingabetaste).

Wenn Sie eine lokale Warteschlange erstellen, wird die Anzeige **Create Local Queue** (Lokale Warteschlange erstellen) geöffnet:

Message Queue Management (MQM)

```
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
```

Abbildung 11. Die Anzeige 'Create Local Queue'

Drücken Sie die Taste **Bild-Ab**, um den zweiten Teil dieser Anzeige anzuzeigen.

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

Cluster Name : _____
Cluster Name List : _____
Distribution List Y/N : _      Default Binding [O/N] : _

PGUP - Return
```

Abbildung 12. Die Anzeige 'Create Local Queue', Teil 2

Füllen Sie in dieser Anzeige die erforderlichen Felder aus, und drücken Sie anschließend die Taste F1 (Eingabetaste).

Wenn Sie eine ferne Warteschlange erstellen, wird die Anzeige **Create Remote Queue** (Ferne Warteschlange erstellen) geöffnet:

```

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                                F16 - Return

```

Abbildung 13. Die Anzeige 'Create Remote Queue'

Füllen Sie in dieser Anzeige die erforderlichen Felder aus, und drücken Sie anschließend die Taste F1 (Eingabetaste).

Warteschlange kopieren

Drücken Sie in der Anzeige **Queue Menu** (Warteschlangenmenü) die Taste F3 (Copy, Kopieren), um eine neue Warteschlange durch Kopieren einer vorhandenen Definition zu erstellen. Daraufhin wird die Anzeige **Copy Queue** geöffnet:

```

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

```

Abbildung 14. Die Anzeige 'Copy Queue'

Message Queue Management (MQM)

Geben Sie im Feld **Name** den Namen der neuen Warteschlangendefinition ein; soll die neue Warteschlange eine vorhandene Warteschlange desselben Namens und desselben Typs ersetzen, geben Sie Y (Ja) im Feld **Replace** (Ersetzen) ein; geben Sie außerdem im Feld **Like Queue** (Wie Warteschlange) den Namen der Definition, die kopiert werden soll, und im Feld **Queue Type** (Warteschlangentyp) den Typ ein. Drücken Sie anschließend die Taste F1 (Eingabetaste).

Warteschlange ändern

Drücken Sie in der Anzeige **Queue Menu** (Warteschlangenmenü) die Taste F1 (Modify, Ändern), um die Anzeige **Display/Modify Local Queue** zu öffnen:

```
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
```

Abbildung 15. Die Anzeige 'Display/Modify Local Queue'

Überschreiben Sie die Werte, die geändert werden sollen, und drücken Sie die Taste F1 (Modify, Ändern). Die Werte der Attribute, die nicht geändert werden können, können auch nicht überschrieben werden.

Warteschlange löschen

Geben Sie in der Anzeige **Queue Menu** (Warteschlangenmenü) ein beliebiges Zeichen vor dem Namen der Warteschlange ein, die gelöscht werden soll. Drücken Sie die Taste F4 (Delete, Löschen) und anschließend F4, um den Löschvorgang zu bestätigen.

Warteschlange überwachen

Drücken Sie in der Anzeige **Queue Menu** die Taste F5 (Monitor, Überwachen), um die Anzeige **Monitor Local Queues** (Lokale Warteschlangen überwachen) zu öffnen:

```

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
    
```

Abbildung 16. Die Anzeige 'Monitor Local Queues'

In diesem Beispiel sind die Warteschlangen weder für Eingaben noch für Ausgaben geöffnet. Die Warteschlange ANNE_ME02_SDRD_LOCAL enthält zehn Nachrichten.

Der PATHWAY-Parameter MQMQMREFRESHINT für MQS-MQMSVR00 gibt das Intervall an, in dem Überwachungsanzeigen für Kanäle und Warteschlangen aktualisiert werden. Als Standardintervall sind 30 Sekunden vorgegeben. Soll eine Aktualisierung dagegen beispielsweise alle 10 Sekunden erfolgen, müssen Sie für den WS-Manager folgenden PATHWAY-Befehl eingeben:

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

Anzeige 'Channels Menu'

Um die Option **Channels** (Kanäle) auszuwählen, geben Sie 3 im Feld **Enter Choice** im **MQM Main Menu** ein, und drücken Sie anschließend die Taste F1 (Eingabetaste). Daraufhin wird die Anzeige **Search Criteria** (Suchkriterien) für Kanäle geöffnet:

Message Queue Management (MQM)

```
IBM MQSeries for Compaq NonStop Kernel Version 5.1

** Search Criteria **

Channel Name: _____
Enter a channel name or part of one:

Channel Type: _
choose one or leave blank:      1. Sender
                                2. Server
                                3. Receiver
                                4. Requester
                                5. SvrConn
                                6. Cluster Sender
                                7. Cluster Receiver

F1 - Enter                      F16 - Return
```

Abbildung 17. Die Anzeige 'Search Criteria' (für Kanäle)

Geben Sie im Feld **Channel Name** (Kanalname) den Namen des Kanals teilweise oder vollständig ein. Geben Sie im Feld **Channel Type** (Kanaltyp) eine Zahl zwischen 1 und 5 für den Kanaltyp an, der ausgewählt werden soll. Drücken Sie anschließend die Taste F1 (Eingabetaste), um die Anzeige **Channel Menu** zu öffnen:

```
IBM MQSeries for Compaq NonStop Kernel Version 5.1

** Channel Menu **

Channel Name      TYPE      STATUS
- MA02.MT01.SDRC.0001    RECEIVER
- MA02.MT01.RQSD_0001    REQUESTER
- MA02.MT01.RQSV_0001    REQUESTER
- MA02.MT01.SDRC_0001    RECEIVER
- MD01.MT01.RQSD_0001    REQUESTER
- MD01.MT01.RQSV_0001    REQUESTER
- MD01.MT01.SDRC_0001    RECEIVER
- MD01.MT01.SVRC_0001    RECEIVER
- ME02.MT01.RQSD_0001    REQUESTER
- ME02.MT01.RQSV_0001    REQUESTER
- ME02.MT01.SDRC_0001    RECEIVER
- ME02.MT01.SVRC_0001    RECEIVER

F1 - Enter/Display/Modify    F2 - Create    F3 - Copy    F4 - Delete
F5 - Monitor    F6 - Resolve    F7 - Reset MSN    F8 - Start/Stop    F10 - Status
F12 - Refresh          PGDN          PGUP          F16 - Return
```

Abbildung 18. Die Anzeige 'Channel Menu'

Message Queue Management (MQM)

Die Anzeige **Channel Menu** (Kanalmenü) enthält eine Liste der Kanäle, die Ihren Suchkriterien entsprechen. In dieser Anzeige haben Sie folgende Auswahlmöglichkeiten:

- Kanalstatus anzeigen und ändern.
- Neue Kanaldefinition erstellen.
- Kanaldefinition kopieren.
- Kanaldefinition löschen.
- Kanalstatus überwachen.
- Kanal auflösen.
- Nachrichtenfolgennummer (MSN, Message Sequence Number) zurücksetzen.
- Kanal starten bzw. stoppen.

Kanal ändern

Geben Sie in der Anzeige **Channel Menu** (Kanalmenü) vor dem Kanal, der geändert werden soll, ein beliebiges Zeichen ein, und drücken Sie die Taste F1 (Enter/Display/Modify, Eingeben/Anzeigen/Ändern). Die entsprechende Anzeige wird geöffnet. Wenn Sie beispielsweise einen Senderkanal ausgewählt haben, wird die Anzeige **Display/Modify Sender Channel** (Senderkanal anzeigen/ändern) geöffnet:

```
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 F16 - Return
```

Abbildung 19. Die Anzeige 'Display/Modify Sender Channel' (Teil 1)

Message Queue Management (MQM)

Drücken Sie die Taste **Bild-Ab**, um den zweiten Teil dieser Anzeige anzuzeigen:

```
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

PGUP - Return
```

Abbildung 20. Die Anzeige 'Display/Modify Sender Channel' (Teil 2)

Überschreiben Sie die Werte, die geändert werden sollen, und drücken Sie die Taste F1 (Modify, Ändern). Die Werte der Attribute, die nicht geändert werden können, können auch nicht überschrieben werden.

Kanaldefinition erstellen

Drücken Sie in der Anzeige **Channel Menu** (Kanalmenü) die Taste F2 (Create, Erstellen), um die Anzeige **Create Channel** (Kanal erstellen) zu öffnen:

```
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                                F16 - Return
```

Abbildung 21. Die Anzeige 'Create Channel'

Message Queue Management (MQM)

So erstellen Sie eine neue Definition:

1. Geben Sie im Feld **Channel Type** (Kanaltyp) eine 1 (Senderkanal), eine 2 (Serverkanal), eine 3 (Empfängerkanal), eine 4 (Requester-Kanal) oder eine 5 (Serververbindung) ein.
2. Geben Sie im Feld **Name** den Namen der Kanaldefinition ein.
3. Drücken Sie anschließend die Taste F1 (Eingabetaste).
4. Soll die Definition eine bereits vorhandene Definition desselben Namens und desselben Typs ersetzen, geben Sie Y (Ja) im Feld **Replace** (Ersetzen) ein.

Wenn Sie 1 im Feld **Channel Type** (Kanaltyp) eingegeben haben, wird die Anzeige **Create Sender Channel** (Senderkanal erstellen) geöffnet:

```
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
```

Abbildung 22. Die Anzeige 'Create Sender Channel'

Message Queue Management (MQM)

Wenn Sie 3 im Feld **Channel Type** (Kanaltyp) eingeben, wird die Anzeige **Create Receiver Channel** (Empfängerkanal erstellen) geöffnet:

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

Channel Name      : SOLARIS_T0_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   :          99999999

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

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

Abbildung 23. Die Anzeige 'Create Receiver Channel'

Wenn Sie 5 im Feld **Channel Type** (Kanaltyp) eingeben, wird die Anzeige **Create Server Connection Channel** (Kanal für Serververbindung erstellen) geöffnet:

```
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                               F16 - Return
```

Abbildung 24. Die Anzeige 'Create Server Connection Channel'

Um eine neue Kanaldefinition zu erstellen, füllen Sie die erforderliche Anzeige aus, und drücken Sie anschließend die Taste F1 (Eingabetaste).

Message Queue Management (MQM)

Wenn Sie 6 im Feld **Channel Type** (Kanaltyp) eingeben, wird die Anzeige **Create Cluster Sender Channel** (Cluster-Senderkanal erstellen) geöffnet:

```
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  :          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 : _____
Local LU Name    : _____ Remote LU Name   : _____
Local TP Name    : _____ Mode Name       : _____
Remote TP Name   : _____
F1 - Enter      PGDN - Exits                      F16 - Return
```

Abbildung 25. Die Anzeige 'Create Cluster Sender Channel'

Wenn Sie 7 im Feld **Channel Type** (Kanaltyp) eingeben, wird die Anzeige **Create Cluster Receiver Channel** (Cluster-Empfängerkanal erstellen) geöffnet:

```
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                      F16 - Return
```

Abbildung 26. Die Anzeige 'Create Cluster Receiver Channel'

Message Queue Management (MQM)

Kanal überwachen

Drücken Sie in der Anzeige **Channel Menu** (Kanalmenü) die Taste F5 (Monitor, Überwachen), um die Anzeige **Monitor Channels** (Kanäle überwachen) zu öffnen:

```
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
MT01.VM03.SDRC.0002  RUNNING      6266      6266  RUNNING     NO

F12 - Refresh          PGDN          PGUP          F16 - Return

Refreshing.....
```

Abbildung 27. Die Anzeige 'Monitor Channels'

Der PATHWAY-Parameter MQMQMREFRESHINT für MQS-MQMSVR00 gibt das Intervall an, in dem Überwachungsanzeigen für Kanäle und Warteschlangen aktualisiert werden. Als Standardintervall sind 30 Sekunden vorgegeben. Soll eine Aktualisierung dagegen beispielsweise alle 10 Sekunden erfolgen, müssen Sie für den WS-Manager folgenden PATHWAY-Befehl eingeben:

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

Kanal löschen

Wählen Sie in der Anzeige **Channel Menu** den Kanal aus, der gelöscht werden soll, indem Sie ein beliebiges Zeichen vor dem Namen des betreffenden Kanals eingeben. Drücken Sie die Taste F4 (Delete, Löschen), um den Kanal zu löschen, und anschließend F4, um den Löschvorgang zu bestätigen.

Kanalstatus anzeigen

Drücken Sie in der Anzeige **Channel Menu** (Kanalmenü) die Taste F10 (Status), um die Anzeige **Channel Status** (Kanalstatus) zu öffnen:

```

IBM MQSeries for Compaq NonStop Kernel Version 5.1
Channel Status

Channel Name : MT01.VM03.SDRC.0002_
Xmit Queue Name: VM03NCM.TQ.SDRC.0001_
Connection Name: $BP01.IYAHT080.IYCNVM03_

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

```

Abbildung 28. Die Anzeige 'Channel Status'

Kanäle starten und stoppen

Drücken Sie in der Anzeige **Channel Menu** (Kanalmenü) die Taste F8 (Start/Stop, Starten/Stoppen), um die Anzeige **Start/Stop Channel** (Kanal starten/stoppen) zu öffnen:

```

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

Name: MT01_MA02_SDRC_0001_

Status:

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

F1 - Enter                      F16 - Return

```

Abbildung 29. Die Anzeige 'Start/Stop Channel'

Geben Sie im Feld **Name** den Namen des Kanals ein und im Feld **Action** (Aktion) eine Zahl zwischen 1 und 3. Drücken Sie anschließend die Taste F1 (Eingabetaste).

Message Queue Management (MQM)

Nachrichtenfolgennummer zurücksetzen

Drücken Sie in der Anzeige **Channel Menu** (Kanalmenü) die Taste F7 (Reset MSN, Nachrichtenfolgennummer zurücksetzen), um die Anzeige **Reset Channel** (Kanal zurücksetzen) zu öffnen:

```
IBM MQSeries for Compaq NonStop Kernel Version 5.1
Reset Channel
      Name: MT01_M401_RQSD_0001_____
Sequence Number:      1

F1 - Enter                                F16 - Return
```

Abbildung 30. Die Anzeige 'Reset Channel'

Durch die Nachrichtenfolgennummer wird sichergestellt, dass Nachrichten nicht doppelt vorkommen und dass sie in der Reihenfolge gespeichert werden, in der sie übertragen werden. In dieser Anzeige können Sie bei Bedarf die Folgenummer eines Kanals zurücksetzen.

Kanal auflösen

Drücken Sie in der Anzeige **Channel Menu** (Kanalmenü) die Taste F6 (Resolve, Auflösen), um die Anzeige **Resolve Channel** (Kanal auflösen) zu öffnen.

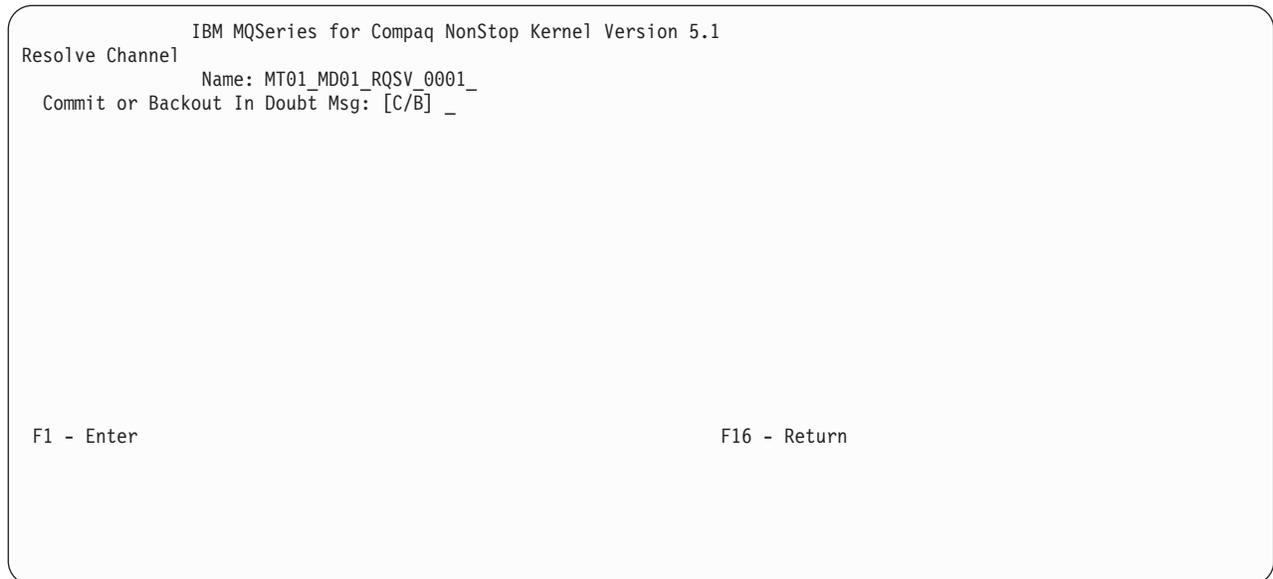


Abbildung 31. Die Anzeige 'Resolve Channel'

Sie haben folgende Auswahlmöglichkeiten:

- Unbestätigten Nachrichtenstapel zurücksetzen (B, Backout)
- Unbestätigten Nachrichtenstapel festschreiben (C, Commit)

Message Queue Management (MQM)

Kanal kopieren

Drücken Sie in der Anzeige **Channel Menu** (Kanalmenü) die Taste F3 (Copy, Kopieren). Daraufhin wird die Anzeige **Copy Channel** (Kanal kopieren) geöffnet:

```
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
```

Abbildung 32. Die Anzeige 'Copy Channel'

Geben Sie im Feld **Name** den Namen des neuen Kanals ein, im Feld **Like Name** (Wie Name) den Namen der Kanaldefinition, die kopiert werden soll, und im Feld **Channel Type** (Kanaltyp) den Kanaltyp. Drücken Sie anschließend die Taste F1 (Eingabetaste), um die Kanaldefinition zu kopieren.

Kapitel 5. Lokale MQSeries-Objekte verwalten

In diesem Kapitel wird beschrieben, wie lokale MQSeries-Objekte verwaltet werden, um Anwendungsprogramme zu unterstützen, die MQI (Message Queuing Interface) unterstützen. Diese Schnittstelle ermöglicht Anwendungsprogrammen den Zugriff auf Message-Queuing-Services.

Unter 'lokaler Verwaltung' versteht man das Erstellen, Anzeigen, Ändern, Kopieren und Löschen von MQSeries-Objekten.

Dieses Kapitel enthält folgende Abschnitte:

- „Unterstützung von Anwendungsprogrammen, die MQI verwenden“
- „Lokale Verwaltungs-Tasks mit Hilfe von MQSC-Befehlen ausführen“ auf Seite 96
- „MQSC-Befehle aus Textdateien ausführen“ auf Seite 100
- „MQSC-Probleme lösen“ auf Seite 104
- „Mit lokalen Warteschlangen arbeiten“ auf Seite 105
- „Mit Aliaswarteschlangen arbeiten“ auf Seite 114
- „Mit Modellwarteschlangen arbeiten“ auf Seite 116
- „Objekte für Auslösefunktion verwalten“ auf Seite 117

Unterstützung von Anwendungsprogrammen, die MQI verwenden

MQI-Anwendungsprogramme benötigen für die erfolgreiche Ausführung bestimmte Objekte. MQI-Anwendungen können Nachrichten aus einer Warteschlange entfernen, diese verarbeiten und das Ergebnis an eine andere Warteschlange in demselben WS-Manager senden.

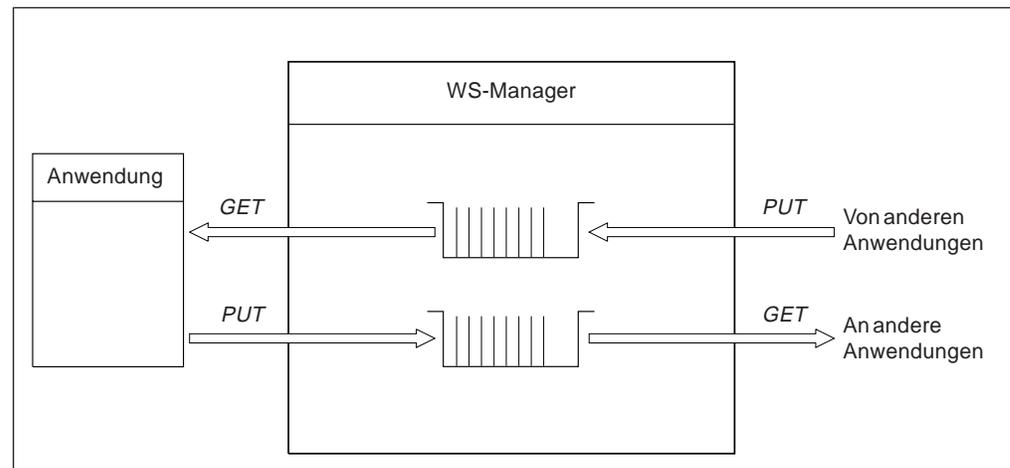


Abbildung 33. Warteschlangen, Nachrichten und Anwendungen

Anwendungen können Nachrichten zwar in lokale und in ferne Warteschlangen einreihen (mit MQPUT), Nachrichten direkt abrufen können sie jedoch nur aus lokalen Warteschlangen (mit MQGET).

Anwendungsunterstützung

Folgende Bedingungen müssen erfüllt sein, damit diese Anwendung ausgeführt werden kann:

- Der WS-Manager muss vorhanden und aktiv sein.
- Die erste Anwendungswarteschlange, aus der die Nachrichten abgerufen werden, muss definiert sein.
- Die zweite Warteschlange, in die die Anwendung Nachrichten einreicht, muss ebenfalls definiert sein, sofern es sich nicht um eine dynamische Warteschlange handelt.
- Die Anwendung muss eine Verbindung mit dem WS-Manager herstellen können. Dazu muss sie mit dem Produktcode verbunden sein. Weitere Informationen finden Sie unter „Anhang I. Anwendungen erstellen und ausführen“ auf Seite 355.
- Die Anwendungen, die die Nachrichten in die erste Warteschlange einreihen, müssen ebenfalls mit einem WS-Manager verbunden sein. Handelt es sich um ferne Anwendungen, müssen für sie außerdem Übertragungswarteschlangen und Kanäle konfiguriert sein.

Lokale Verwaltungs-Tasks mit Hilfe von MQSC-Befehlen ausführen

Mit MQSC-Befehlen (MQSeries Commands) können verschiedene Aktionen in Zusammenhang mit MQSeries-Objekten ausgeführt werden. Diese Befehle können über die Eingabe des Steuerbefehls **runmqsc** an der Eingabeaufforderung aufgerufen werden.

Weitere Informationen zur Verwendung von MQSC-Befehlen in der MQSeries for Compaq NSK-Umgebung finden Sie in „Anhang G. In MQSeries for Compaq NSK unterstützte MQSC-Befehle“ auf Seite 341.

Mit den MQSC-Befehlen können WS-Managerobjekte verwaltet werden; dazu gehören neben dem WS-Manager selbst Cluster, Kanäle, Warteschlangen, Namenslisten und Prozessdefinitionen. Dieser Abschnitt behandelt WS-Manager, Warteschlangen und Prozessdefinitionen. Informationen zur Verwaltung von Kanalobjekten finden Sie unter 'DQM implementation' im Handbuch *MQSeries Intercommunication*.

MQSC-Befehle für einen WS-Manager werden mit Hilfe des Befehls **runmqsc** ausgegeben. Dies kann interaktiv erfolgen, durch Eingeben der Befehle über die Tastatur, oder indem Sie die Standardeingabe umleiten, um eine Befehlsfolge aus einer ASCII-Textdatei auszuführen. Das Format der Befehle ist in beiden Fällen dasselbe.

Sie können den Befehl **runmqsc** in drei Modi ausführen, abhängig von den mit dem Befehl angegebenen Optionen:

- *Prüfmodus*, bei dem MQSC-Befehle in einem lokalen WS-Manager geprüft, aber nicht wirklich ausgeführt werden.
- *direkter Modus*, bei dem MQSC-Befehle in einem lokalen WS-Manager ausgeführt werden.
- *indirekter Modus*, bei dem MQSC-Befehle in einem fernen WS-Manager ausgeführt werden.

Objektattribute, die in MQSC angegeben werden, werden in diesem Buch in Großbuchstaben geschrieben (z. B. RQMNAME), obwohl bei ihnen die Groß-/Kleinschreibung nicht beachtet werden muss. Namen von MQSC-Attributen dürfen maximal acht Zeichen lang sein.

Vor dem Start

Vor Verwendung der MQSC-Befehle müssen Sie zunächst den WS-Manager, in dem die MQSC-Befehle ausgeführt werden, erstellen und starten. Weitere Informationen finden Sie unter „Standard-WS-Manager erstellen“ auf Seite 68.

MQSeries-Objektnamen

In den Beispielen werden lange Namen für Objekte verwendet. Dies soll Ihnen helfen, den Objekttyp, mit dem Sie es zu tun haben, zu identifizieren.

Wenn Sie MQSC-Befehle ausgeben, müssen Sie nur den lokalen Namen der Warteschlange angeben. In unseren Beispielen werden Namen verwendet, die folgendes Format haben: ORANGE.LOCAL.QUEUE

Der Teil LOCAL.QUEUE des Namens wurde gewählt, um deutlich zu machen, dass es sich um eine lokale Warteschlange handelt. Es bedeutet *nicht*, dass die Namen lokaler Warteschlangen diesen Teil enthalten müssen.

Als Name des WS-Managers wird saturn.queue.manager verwendet.

Der Teil queue.manager des Namens wurde gewählt, um deutlich zu machen, dass es sich bei diesem Objekt um einen WS-Manager handelt. Es bedeutet *nicht*, dass Namen von WS-Managern diesen Teil enthalten müssen.

Sie können auch andere Namen verwenden, allerdings müssen Sie dann in den Beispielen alle Befehle mit diesen Namen entsprechend ändern.

Groß-/Kleinschreibung bei MQSC-Befehlen

Bei der Eingabe von MQSC-Befehlen und ihren Attributen muss die Groß-/Kleinschreibung nicht beachtet werden. Bei Objektnamen hingegen ist die Groß-/Kleinschreibung wichtig; so handelt es sich beispielsweise bei **Eingabewarteschlange** und **EINGABEWARTESCHLANGE** um zwei verschiedene Objekte. Objektnamen werden in MQSC-Befehlen in Großbuchstaben umgesetzt, d. h., es wird kein Unterschied zwischen **Warteschlange** und **WARTESCHLANGE** gemacht, wenn die Namen nicht in einfache Anführungszeichen gesetzt sind. Fehlen die Anführungszeichen, werden Objektnamen in Großbuchstaben umgesetzt. Weitere Informationen hierzu finden Sie im Handbuch *MQSeries MQSC - Befehle*.

Bei einigen der Argumente, die im Befehl **runmqsc** (mit dem die MQSC-Funktion aufgerufen wird) verwendet werden, muss die Groß-/Kleinschreibung allerdings beachtet werden (siehe „Steuerbefehle verwenden“ auf Seite 31).

Die MQSC-Funktion interaktiv verwenden

Um Befehle interaktiv zu verwenden, öffnen Sie eine TACL-Sitzung geben Sie Folgendes ein:

```
runmqsc
```

In diesem Beispiel wurde kein WS-Manager angegeben, daher werden die MQSC-Befehle vom Standard-WS-Manager verarbeitet. Sie können einen beliebigen MQSC-Befehl eingeben.

MQSC-Befehle ausgeben

Beispiel:

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

Um anzuzeigen, dass ein Befehl in der nächsten Zeile fortgesetzt wird, müssen Fortsetzungszeichen verwendet werden:

- Ein Minuszeichen (-) gibt an, dass der Befehl am Anfang der nächsten Zeile fortgesetzt wird.
- Ein Pluszeichen (+) gibt an, dass der Befehl mit dem ersten belegten Zeichen in der nächsten Zeile fortgesetzt wird.

Die Befehlseingabe endet mit dem Schlusszeichen einer belegten Zeile, bei dem es sich nicht um ein Fortsetzungszeichen handelt.

Der Befehl **runmqsc** unterstützt auch das standardmäßige Compaq NSK-Protokoll und die Compaq NSK Fix Command-Funktion. Beispiel:

- Bei Eingabe von `history` oder `h` wird eine Liste mit den zehn Befehlen angezeigt, die zuletzt ausgeführt wurden.
- Bei Eingabe von `!n` (dabei steht `n` für die Befehlsnummer) wird der betreffende Befehl erneut ausgeführt.
- Bei Eingabe von `h n` oder `history n` (dabei steht `n` für eine Zahl) werden die `n` zuletzt eingegebenen Befehle angezeigt.
- Bei Eingabe von `fc` wird der zuletzt eingegebene Befehl angezeigt, der dann bearbeitet werden kann. Bei Eingabe von `fc n` (dabei steht `n` für die Befehlsnummer) wird der betreffende Befehl angezeigt und kann bearbeitet werden. Bei Eingabe von `fc Zeichenfolge` (dabei steht *Zeichenfolge* für den zuletzt eingegebenen Befehl oder den ersten Teil davon) wird der zuletzt eingegebene Befehl dieses Namens zur Bearbeitung angezeigt. Die Syntax entspricht dem NSK-Standard. Geben Sie beispielsweise `d` (delete) an, um ein Zeichen zu löschen, `i` (insert), um ein Zeichen einzufügen oder `r` (replace), um ein Zeichen zu ersetzen.

Rückmeldung auf MQSC-Befehle

Auf die Eingabe von MQSC-Befehlen hin gibt der WS-Manager Bestätigungsnachrichten oder Fehlnachrichten zurück. Beispiel:

```
AMQ8006: Die MQSeries-Warteschlange wurde erstellt.  
.  
.  
.  
AMQ8405: Syntaxfehler in oder bei folgendem Befehlssegment:-
```

Die erste Nachricht bestätigt die erfolgreiche Erstellung einer Warteschlange. Die zweite Nachricht weist auf einen Syntaxfehler hin. Wenn Sie den Befehl nicht korrekt eingegeben haben, schlagen Sie im Handbuch *MQSeries MQSC - Befehle* die richtige Syntax nach.

Interaktive Eingabe von MQSC-Befehlen beenden

Bei der interaktiven Verwendung der MQSC-Funktion kann diese durch Eingabe des EOF-Zeichens (Strg+Y) beendet werden oder indem Sie `exit`, `quit` oder `end` eingeben und anschließend die Eingabetaste drücken.

Erfolgt die Eingabe aus einer anderen Quelle, beispielsweise aus einer Textdatei, wird die MQSC-Funktion am Ende der Datei automatisch beendet.

WS-Managerattribute anzeigen

Mit dem folgenden MQSeries Befehl zeigen Sie die Attribute des im Befehl `runmqsc` angegebenen WS-Managers an:

```
MQSC>DISPLAY QMGR ALL
```

Eine typische Ausgabe sehen Sie in Abb. 34:

```
1 : dis qmgr all
AMQ8408: Details zu DISPLAY QMGR werden angezeigt.
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)
ALTDATA(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)
```

Abbildung 34. Ausgabe für QMGR ALL - Beispiel

Bei Angabe des Parameters ALL im Befehl `DISPLAY QMGR` werden alle Attribute des WS-Managers angezeigt. Die Ausgabe enthält den Namen des WS-Managers (`saturn.queue.manager`), den Namen der Warteschlange für nicht zustellbare Nachrichten (`SYSTEM.DEAD.LETTER.QUEUE`) und den Namen der Befehlswarteschlange (`SYSTEM.ADMIN.COMMAND.QUEUE`). Wird im Befehl `crtmqmc` keine Warteschlange für nicht zustellbare Nachrichten angegeben, muss der WS-Manager geändert werden, indem ihm eine Warteschlange für nicht zustellbare Nachrichten zugeordnet wird.

Sie sollten sicherstellen, dass diese Warteschlangen erstellt wurden; dies ist mit dem folgenden Befehl möglich:

```
DISPLAY QUEUE (*)
```

MQSC-Befehle ausgeben

Einen WS-Manager verwenden, der nicht der Standard-WS-Manager ist

Sie können im Befehl **runmqsc** einen WS-Manager angeben, damit MQSC-Befehle nicht im Standard-WS-Manager, sondern von einem anderen lokalen WS-Manager ausgeführt werden. Sollen MQSC-Befehle beispielsweise von einem WS-Manager des Namens `jupiter.queue.manager` ausgeführt werden, geben Sie folgenden Befehl ein:

```
runmqsc jupiter.queue.manager
```

Alle anschließend eingegebenen MQSC-Befehle werden von diesem WS-Manager ausgeführt, vorausgesetzt er befindet sich auf demselben Knoten und ist aktiv.

Sie können MQSC-Befehle auch in einem fernen WS-Manager ausführen (siehe „Ferne Ausführung von MQSC-Befehlen“ auf Seite 131).

WS-Managerattribute ändern

Die im Befehl **runmqsc** angegebenen Attribute eines WS-Managers können geändert werden, indem Sie den Befehl **MQSC ALTER QMGR** unter Angabe der Attribute und ihrer Werte, die geändert werden sollen, eingeben. Geben Sie zum Beispiel folgende Befehle ein, um die Attribute von `jupiter.queue.manager` zu ändern:

```
runmqsc jupiter.queue.manager  
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

Mit dem Befehl **ALTER QMGR** wird die verwendete Warteschlange für nicht zustellbare Nachrichten geändert, und es können Sperrereignisse aktiviert werden.

MQSC-Befehle aus Textdateien ausführen

Die interaktive Ausführung von MQSC-Befehlen ist für rasche Tests geeignet; bei langen Befehlen oder bei Befehlsfolgen, die wiederholt ausgeführt werden sollen, sollte die Eingabe aus einer Textdatei erfolgen.

Erstellen Sie hierzu mit einem Texteditor eine Textdatei mit den MQSC-Befehlen, die ausgeführt werden sollen. Bei Verwendung des Befehls **runmqsc** sollten Sie für diesen die TACL-Umleitungsoperatoren **IN** und **OUT** bzw. oder die Optionen **-i** und **-o** verwenden. Mit dem folgenden Befehl werden zum Beispiel die in der Textdatei `mymqscin` enthaltenen Befehle ausgeführt:

```
runmqsc /IN mymqscin/
```

oder

```
runmqsc -i mymqscin
```

MQSC-Befehle ausführen

Ebenso kann die Ausgabe in eine Datei umgeleitet werden. Die Datei mit den MQSC-Befehlen, die ausgeführt werden sollen, wird als *MQSC-Datei* bezeichnet. Die Ausgabedatei, die die Rückmeldung des WS-Managers enthält, wird als *Berichtsdatei* bezeichnet.

Um die Ein- und Ausgabe im Befehl **runmqsc** umzuleiten, geben Sie folgenden Befehl ein:

```
runmqsc /IN mymqscin, OUT mymqscou/
```

ODER

```
runmqsc -i mymqscin -o mymqscou
```

Mit diesem Befehl werden die in der Datei *mymqscin* enthaltenen Befehle ausgeführt. Da kein WS-Manager angegeben wurde, werden die MQSC-Befehle vom Standard-WS-Manager verarbeitet. Die Ausgabe wird in die Berichtsdatei *mymqscou* geschrieben. In Abb. 35 auf Seite 102 ist ein Auszug der MQSC-Befehlsdatei *mymqscin*, in Abb. 36 auf Seite 103 ein Auszug der entsprechenden Berichtsdatei *mymqscou* dargestellt.

Soll die Ein- und Ausgabe des Befehls **runmqsc** für einen WS-Manager (*saturn.queue.manager*) umgeleitet werden, bei dem es sich nicht um den Standard-WS-Manager handelt, geben Sie folgenden Befehl ein:

```
runmqsc /IN mymqscin, OUT mymqscou/ saturn.queue.manager
```

ODER

```
runmqsc -i mymqscin -o mymqscou saturn.queue.manager
```

MQSC-Befehlsdateien verwenden

MQSC-Befehlsdateien werden als EDIT-Dateien erstellt (Compaq-Dateityp 101). Abb. 35 auf Seite 102 zeigt einen Auszug aus einer MQSC-Datei mit dem MQSeries-Befehl **DEFINE QLOCAL** und den zugehörigen Attributen. Eine Beschreibung der einzelnen MQSC-Befehle und der jeweiligen Syntax finden Sie im Handbuch *MQSeries MQSC - Befehle*.

MQSC-Befehle ausführen

```
.  
. .  
. .  
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  
. .  
. .  
. .
```

Abbildung 35. Auszug aus der MQSC-Befehlsdatei *mymqscin*

Die Zeilen dürfen maximal 72 Zeichen enthalten. Das Pluszeichen (+) zeigt an, dass der Befehl in der nächsten Zeile fortgesetzt wird. Dem Pluszeichen muss ein Leerzeichen vorangestellt sein.

MQSC-Berichtsdateien verwenden

Der Befehl *runmqsc* gibt einen *Bericht* zurück, der an den aktuellen OUT-Strom gesendet wird; dieser Bericht enthält Folgendes:

- einen Header, der auf MQSC als Quelle des Berichts hinweist:

```
MQSeries-Befehle werden gestartet.
```

- optional eine nummerierte Liste der ausgegebenen MQSC-Befehle. Standardmäßig wird der Text der Eingabe in der Ausgabe zurückgegeben. In der Ausgabe steht vor jedem Befehl eine Folgenummer (siehe Abb. 36 auf Seite 103). Sie können die Ausgabe jedoch unterdrücken, indem Sie im Befehl *runmqsc* die Option *-e* angeben.
- eine Syntaxfehlnachricht für jeden fehlerhaften Befehl.
- eine *Bedienernachricht* zu jedem ausgeführten Befehl. Die Bedienernachricht zu einem erfolgreich ausgeführten DEFINE QLOCAL-Befehl lautet zum Beispiel:

```
AMQ8006: Die MQSeries-Warteschlange wurde erstellt.
```

- andere Nachrichten zu allgemeinen Fehlern während der Ausführung der Scriptdatei.
- eine kurze statistische Zusammenfassung des Berichts mit der Anzahl der gelesenen Befehle, der Anzahl der Befehle mit Syntaxfehlern und der Anzahl der Befehle, die nicht verarbeitet werden konnten.

Anmerkung: Der WS-Manager verarbeitet nur Befehle, die keine Syntaxfehler enthalten.

```

MQSeries-Befehle werden gestartet.
.
.
12:   DEFINE QLOCAL('RED.LOCAL.QUEUE') REPLACE +
:     DESCR(' ') +
:     PUT(ENABLED) +
:     DEFPRTY(0) +
:     DEFPSIST(NO) +
:     GET(ENABLED) +
:     MAXDEPTH(5000) +
:     MAXMSGL(1024) +
:     DEFSOPT(SHARED) +
:     USAGE(NORMAL) +
:     NOTRIGGER
AMQ8006: Die MQSeries-Warteschlange wurde erstellt.
:
.
.
15 MQSC-Befehle gelesen.
Bei 0 Befehlen liegt ein Syntaxfehler vor.
0 Befehl(e) kann (können) nicht verarbeitet werden.

```

Abbildung 36. Auszug aus der MQSC-Berichtsdatei mymqscou

Im Lieferumfang enthaltene MQSC-Befehlsdatei ausführen

Bei der Installation von MQSeries for Compaq NSK wird die MQSC-Datei **AMQSCOS0** zur Verfügung gestellt. Diese Datei enthält die Definitionen der Objekte, die von den Beispielprogrammen verwendet werden. Diese Datei befindet sich im Teildatenträger mit den Beispielen (standardmäßig in `$SYSTEM.ZMQSSMPL`).

Befehle mit Hilfe von `runmqsc` prüfen

Mit dem Befehl `runmqsc` können Sie MQSC-Befehle in einem lokalen Standard-WS-Manager überprüfen, ohne sie tatsächlich auszuführen. Geben Sie dazu den Befehl `runmqsc` mit der Option `-v` ein. Beispiel:

```
runmqsc -i mymqscin -o mymqscou -v
```

Wenn Sie `runmqsc` für eine MQSC-Befehlsdatei ausführen, prüft der WS-Manager jeden Befehl und gibt einen Bericht zurück, ohne die MQSC-Befehle wirklich auszuführen. Auf diese Weise können Sie die Syntax aller Befehle in einer Befehlsdatei überprüfen. Dies ist in folgenden Fällen wichtig:

- Bei der Ausführung einer Befehlsdatei, die eine große Anzahl von Befehlen enthält.
- Wenn eine MQSC-Befehlsdatei wiederholt ausgeführt wird.

Dieser Bericht ähnelt dem Bericht, der in Abb. 36 gezeigt wird.

MQSC-Befehle ausführen

Mit dieser Methode können Sie jedoch keine fernen MQSC-Befehle prüfen. Beispielsweise wird bei Eingabe des Befehls:

```
runmqsc -i mymqscin -o mymqscou -w 30 -v jupiter.queue.manager
```

die Option `-w`, die auf einen fernen WS-Manager hinweisen soll, ignoriert; der Befehl wird lokal im Prüfmodus ausgeführt.

MQSC-Probleme lösen

Wenn MQSC-Befehle nicht ordnungsgemäß ausgeführt werden, stellen Sie anhand der folgenden Prüfliste fest, ob eines der dort beschriebenen Probleme zutrifft.

Bei Verwendung des Befehls **runmqsc** sollten Sie Folgendes beachten:

- Stellen Sie sicher, dass in der Datei TACL CSTM der Teildatenträger \$SYSTEM.ZMQSEXE in der Umgebungsvariable PMSEARCH angegeben ist.
- Geben Sie den Operator IN oder die Option `-i` an, wenn die Eingabe aus einer Datei erfolgen soll. Andernfalls interpretiert der WS-Manager den Dateinamen als den Namen eines WS-Managers und gibt die folgende Fehlermeldung aus:

```
AMQ8118: MQSeries-Warteschlangenmanager nicht vorhanden.
```

- Verwenden Sie den Operator OUT oder die Option `-o`, wenn die Ausgabe in eine Datei umgeleitet werden soll. Standardmäßig wird die Ausgabedatei auf Basis der TACL-Standardwerte erstellt, die zu dem Zeitpunkt gültig sind, zu dem der Befehl eingegeben wird. Geben Sie einen vollständig qualifizierten Dateinamen an, wenn die Ausgabe an eine bestimmte Datei geschickt werden soll.
- Überprüfen Sie, ob der WS-Manager, der die Befehle ausführen soll, vorhanden ist. Überprüfen Sie hierzu die Konfigurationsdatei MQSINI, die sich standardmäßig in dem Teildatenträger befindet, die auch die Installation enthält (\$SYSTEM.ZMQSSYS). Diese Datei enthält die Namen der WS-Manager und den Namen des Standard-WS-Managers (sofern vorhanden).
- Der WS-Manager muss aktiv sein; ist er nicht aktiv, muss er wie unter „WS-Manager starten“ auf Seite 70 beschrieben gestartet werden. Ist er bereits aktiv, erhalten Sie eine Fehlermeldung.
- Wurde kein Standard-WS-Manager definiert, müssen Sie im Befehl **runmqsc** einen WS-Manager angeben; andernfalls erhalten Sie eine Fehlermeldung:

```
AMQ8146: MQSeries-Warteschlangenmanager nicht verfügbar.
```

Weitere Informationen zur Behebung dieses Fehlers finden Sie unter „Vorhandenen WS-Manager als Standard-WS-Manager definieren“ auf Seite 72.

- Sie können im Befehl **runmqsc** keine MQSC-Befehle als Parameter angeben. Beispielsweise ist der folgende Befehl ungültig:

```
runmqsc DEFINE QLOCAL(FRED)
```

- Sie können MQSC-Befehle an einer TACL-Eingabeaufforderung erst nach Eingabe des Befehls **runmqsc** eingeben. Beispiel:

```
DEFINE QLOCAL(Queue1)
```

```
* Error Name of Variable, built-in, or file needed.
```

- Sie können über **runmqsc** keine Steuerbefehle ausführen. Zum Beispiel können Sie keinen WS-Manager starten, wenn Sie MQSC-Befehle interaktiv ausführen:

```
runmqsc(C) Copyright IBM Corp. 1993, 2001. Alle Rechte vorbehalten  
MQSeries-Befehle werden gestartet.
```

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

```
1 : strmqm saturn.queue.manager
```

```
AMQ8405: Syntaxfehler in oder bei folgendem Befehlssegment:
```

Siehe auch „Probleme bei der Verwendung von MQSC-Befehlen für die Fernverwaltung“ auf Seite 133.

Mit lokalen Warteschlangen arbeiten

Dieser Abschnitt enthält Beispiele für einige der MQSC-Befehle, die Sie verwenden können. Eine vollständige Beschreibung dieser Befehle finden Sie im Handbuch *MQSeries MQSC - Befehle*.

Eine lokale Warteschlange definieren

Für eine Anwendung ist der lokale WS-Manager derjenige WS-Manager, mit dem sie verbunden ist. Warteschlangen, die vom lokalen WS-Manager verwaltet werden, werden als lokale Warteschlangen dieses WS-Managers bezeichnet.

Mit dem MQSC-Befehl `MQSC DEFINE QLOCAL` können Sie die Definition für eine lokale Warteschlange sowie eine Datenstruktur, die als Warteschlange bezeichnet wird, erstellen. Außerdem haben Sie die Möglichkeit, die Eigenschaften der Warteschlange, die von der standardmäßigen lokalen Warteschlange übernommen wurden, zu ändern.

Die in diesem Beispiel definierte Warteschlange `ORANGE.LOCAL.QUEUE` hat folgende Eigenschaften:

- Sie ist für GET-Aufrufe aktiviert, nicht jedoch für PUT-Aufrufe; Abrufe erfolgen nach dem FIFO-Verfahren (First In/First Out).
- Es handelt sich um eine 'normale' Warteschlange. Es ist also keine Initialisierungs- oder Übertragungswarteschlange, und sie generiert auch keine Auslösenachrichten.
- Die maximale Warteschlangenlänge umfasst 1000 Nachrichten; die maximale Nachrichtenlänge beträgt 2000 Bytes.

Mit lokalen Warteschlangen arbeiten

Diese Warteschlange wird mit dem folgenden MQSC-Befehl erstellt:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +
  DESCR('Warteschlange für Nachrichten von anderen Systemen') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE(NORMAL)
```

Anmerkungen:

1. Bei den meisten dieser Attribute handelt es sich um die vom Produkt voreingestellten Standardattribute. Sie werden hier zur Veranschaulichung jedoch ebenfalls angezeigt. Sie müssen nur angegeben werden, wenn Sie nicht die Standardwerte verwenden wollen oder wenn die Standardwerte geändert werden müssen.
2. USAGE (NORMAL) gibt an, dass es sich bei dieser Warteschlange um eine Übertragungswarteschlange handelt.
3. Wenn in demselben WS-Manager bereits eine Warteschlange mit dem Namen ORANGE.LOCAL.QUEUE gibt, führt der Befehl zu einem Fehler. Verwenden Sie das Attribut REPLACE, wenn Sie die vorhandene Definition einer Warteschlange überschreiben wollen, beachten Sie jedoch die Informationen unter „Attribute einer lokalen Warteschlange ändern“ auf Seite 108.

Größe der physischen Datei für Warteschlangen ändern

Standardmäßig erstellt der WS-Manager für Warteschlangen Datendateien, die eine Speicherkapazität von bis zu 100 MB haben. Ist dieser Grenzwert erreicht, wird der Rückkehrcode MQRC_Q_SPACE_NOT_AVAILABLE ausgegeben. Sie können den einer Warteschlange zugeordneten maximalen Speicher ändern, indem Sie zunächst mit dem Befehl **dspmqfls** die physischen Dateien ermitteln, die die Daten der betreffenden Warteschlange enthalten, und anschließend mit dem Befehl **altmqfls** die Größe der Primär- und Sekundärbereiche sowie die maximale Größe ändern. Ist die Datei bereits partitioniert, wird mit dem Befehl **altmqfls** die Größe aller Partitionen geändert. Weitere Informationen finden Sie in „altmqfls (Attribute der Warteschlangendateien ändern)“ auf Seite 260. Wird mehr Speicherplatz benötigt, können Sie die Datei über mehrere Datenträger partitionieren.

Eine Warteschlange für nicht zustellbare Nachrichten definieren

Jeder WS-Manager sollte über eine lokale Warteschlange verfügen, die als Warteschlange für nicht zustellbare Nachrichten verwendet wird, damit Nachrichten, die nicht an ihre eigentliche Zieladresse zugestellt werden können, für spätere Abrufe gespeichert werden können.

Sie müssen den WS-Manager auf die Warteschlange für nicht zustellbare Nachrichten hinweisen. Dies können Sie tun, indem Sie eine Warteschlange für nicht zustellbare Nachrichten im Befehl **crtmqm** angeben, Sie können sie aber auch später mit dem Befehl ALTER QMGR angeben. Die Warteschlange für nicht zustellbare Nachrichten muss jedoch definiert werden, bevor sie verwendet werden kann.

Das Produkt verfügt standardmäßig über eine Warteschlange für nicht zustellbare Nachrichten mit dem Namen SYSTEM.DEAD.LETTER.QUEUE. Diese Warte-

Mit lokalen Warteschlangen arbeiten

schlange wird automatisch bei der Erstellung des WS-Managers erstellt. Bei Bedarf können Sie diese Definition ändern. Sie muss jedoch nicht umbenannt werden.

Für eine Warteschlange für nicht zustellbare Nachrichten gelten nur die folgenden Bedingungen:

- Es muss sich um eine lokale Warteschlange handeln.
- Das Attribut MAXMSGL (maximale Nachrichtenlänge) muss so festgelegt werden, dass die Warteschlange die längsten vom WS-Manager zu bearbeitenden Nachrichten **sowie** die Größe des Headers der Warteschlange für nicht zustellbare Nachrichten (MQDLH) aufnehmen kann.

MQSeries stellt eine Steueroutine für die Warteschlange für nicht zustellbare Nachrichten zur Verfügung, mit deren Hilfe Sie vorgeben können, wie die Nachrichten in einer solchen Warteschlange verarbeitet oder entfernt werden. Weitere Informationen finden Sie in „Kapitel 9. MQSeries-Steueroutine der Warteschlange für nicht zustellbare Nachrichten“ auf Seite 163.

Standardobjektattribute anzeigen

Wenn Sie ein MQSeries-Objekt definieren, werden alle Attribute, die Sie nicht angeben, vom Standardobjekt übernommen. Wenn Sie zum Beispiel eine lokale Warteschlange definieren, werden alle Attribute, die Sie in der Definition nicht angeben, von der standardmäßigen lokalen Warteschlange SYSTEM.DEFAULT.LOCAL.QUEUE übernommen. Mit dem folgenden Befehl können Sie diese Attribute abrufen:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Anmerkung: Die Syntax dieses Befehls unterscheidet sich von der des entsprechenden DEFINE-Befehls.

Sie können einzelne Attribute zur Anzeige auswählen. Beispiel:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
MAXDEPTH +  
MAXMSGL +  
CURDEPTH
```

Mit diesem Befehl werden die drei angegebenen Attribute wie folgt angezeigt:

```
AMQ8409: Warteschlangendetails werden angezeigt.  
QUEUE (ORANGE.LOCAL.QUEUE)  
MAXDEPTH (1000)  
MAXMSGL (2000)  
CURDEPTH (0)
```

Das Attribut CURDEPTH gibt die aktuelle Warteschlangenlänge an, d. h. die Anzahl der Nachrichten in der Warteschlange. Durch Anzeigen dieses Attributs können Sie die Warteschlangenlänge ständig überwachen, um sicherzustellen, dass immer Platz für neue Nachrichten verfügbar ist.

Lokale Warteschlangendefinition kopieren

Eine lokale Warteschlangendefinition kopieren

Mit Hilfe des Attributs LIKE im Befehl DEFINE können Sie eine Warteschlangendefinition kopieren. Beispiel:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
  LIKE (ORANGE.LOCAL.QUEUE)
```

Dieser Befehl erstellt eine Warteschlange mit denselben Attributen wie die Warteschlange ORANGE.LOCAL.QUEUE, statt mit den Attributen der standardmäßigen lokalen Warteschlange.

Mit diesem DEFINE-Befehl können Sie auch eine Warteschlangendefinition kopieren, wobei jedoch eine oder mehrere Änderungen an den ursprünglichen Attributen vorgenommen werden. Beispiel:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
  LIKE (ORANGE.LOCAL.QUEUE) +  
  MAXMSGL(1024)
```

Dieser Befehl kopiert die Attribute der Warteschlange ORANGE.LOCAL.QUEUE in die Warteschlange THIRD.QUEUE, legt aber gleichzeitig die maximale Nachrichtenlänge auf 1024 statt auf 2000 Bytes fest.

Anmerkungen:

1. Mit dem Attribut LIKE in einem DEFINE-Befehl kopieren Sie nur die Warteschlangenattribute. Die Nachrichten in der Warteschlange werden nicht kopiert.
2. Wenn Sie eine lokale Warteschlange definieren, ohne das Attribut LIKE anzugeben, entspricht das dem Befehl DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE).

Attribute einer lokalen Warteschlange ändern

Sie haben zwei Möglichkeiten zum Ändern von Warteschlangenattributen: den Befehl ALTER QLOCAL oder den Befehl DEFINE QLOCAL mit dem Attribut REPLACE. Die Warteschlange ORANGE.LOCAL.QUEUE wurde unter „Eine lokale Warteschlange definieren“ auf Seite 105 definiert. Angenommen, Sie möchten die maximale Nachrichtenlänge für diese Warteschlange auf 10 000 Bytes erweitern.

- Befehl ALTER:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Dieser Befehle ändert ein einzelnes Attribut, nämlich die maximale Nachrichtenlänge. Alle anderen Attribute bleiben unverändert.

- Befehl DEFINE unter Angabe der Option REPLACE; Beispiel:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Dieser Befehl ändert nicht nur die maximale Nachrichtenlänge, sondern auch alle anderen Attribute, für die die Standardwerte eingestellt werden. Die Warteschlange ist jetzt für PUT-Aufrufe aktiviert, was zuvor nicht der Fall war. Dies

Attribute lokaler Warteschlangen ändern

ist die Standardeinstellung, die durch die Warteschlange SYSTEM.DEFAULT.LOCAL.QUEUE festgelegt wurde (sofern Sie dieses Attribut dort nicht geändert haben).

Wenn Sie die maximale Nachrichtenlänge einer vorhandenen Warteschlange *verringern*, hat dies keine Auswirkungen auf die Nachrichten, die sich bereits in der Warteschlange befinden. Alle neuen Nachrichten müssen jedoch den neuen Kriterien entsprechen.

Datenträger einer lokalen Warteschlange ändern

Mit dem Befehl **altmqfls** können Sie den Datenträger, in dem eine lokale, vordefinierte Warteschlange gespeichert ist, ändern. Dies kann sich zum Beispiel als notwendig erweisen, wenn Platten-E/A auf verschiedene Datenträger zu verteilt werden sollen, um eine optimale Systemleistung zu erreichen. Voraussetzung für die Ausführung dieses Befehls ist, dass der WS-Manager aktiv und Warteschlange selbst nicht geöffnet ist. Im Befehl **altmqfls** kann nur jeweils eine Warteschlange angegeben werden. Eine Beschreibung der Syntax diesen Befehl finden Sie unter „altmqfls (Attribute der Warteschlangendateien ändern)“ auf Seite 260.

Optionen einer lokalen Warteschlange ändern

Mit dem Befehl **altmqfls** können Sie Folgendes ändern:

- Ob die Warteschlange beim Start des WS-Managers von der Festplatte geladen wird. Geben Sie hierzu die Option `--qsoptions S` an.
- Ob die Warteschlange, nachdem sie geladen wurde, im Speicher bleibt, solange der Warteschlangenserver aktiv ist. Ist diese Option gesetzt, behält der Warteschlangenserver die Warteschlangenstrukturen und -daten im Speicher, nachdem sie geladen wurden. Wird diese Option nicht gesetzt, kann der einer Warteschlange zugeordnete Speicherplatz aus dem Adressbereich des Warteschlangenservers entfernt werden, wenn auf ihn nicht mehr zugegriffen wird. Geben Sie hierzu die Option `--qsoptions L` ein.
- Ob für nicht permanente Nachrichten Prüfpunktinformationen im Backup-Warteschlangenserver aufgezeichnet werden sollen; dadurch wird eine Fehlertoleranz ermöglicht, allerdings auf Kosten der CPU-Auslastung, die sich durch die zusätzliche Aufzeichnung von Prüfpunktinformationen, die zusätzlichen IPC-Nachrichten und den zusätzlichen Speicher für die Nachrichten erhöht. Geben Sie hierzu die Option `--qsoptions C` ein.

Anmerkung: Die Option `--qsoptions` kann nur jeweils einmal in einer Befehlszeile angegeben werden, daher werden bei jedem Befehl immer *alle* drei Optionen (S, L und C) gesetzt. Bei Angabe von `--qsoptions S` beispielsweise wird der Schalter S gesetzt und die Schalter L und C (da sie nicht angegeben wurden) zurückgesetzt.

- Die Anzahl an Datenbytes, die maximal pro permanenter Nachricht im Cache des Warteschlangenservers (und auf der Platte) gespeichert werden kann. Die angegebene Anzahl an Datenbytes wird im Speicher behalten; diese Daten können bei einem Abrufvorgang (Browse) an die Anwendung zurückgegeben werden, ohne dass ein Zugriff auf die Platte erforderlich ist. Dadurch werden vom Warteschlangenserver mehr Speicherressourcen belegt. Geben Sie hierzu den Parameter `--browse` ein.
- Gibt die Mindestgröße der Nachrichten an, die in einer Nachrichtenüberlaufdatei gespeichert werden. Permanente Nachrichten, die kleiner sind, werden in der Überlaufdatei der Warteschlange gespeichert. Die Massendaten permanenter Nachrichten, die größer sind, werden in einer dedizierten Nachrichtenüberlaufdatei gespeichert. Geben Sie hierzu den Parameter `--msgofthresh` an.

Größenänderungen, Aktualisierung und Abruf

- Den Teildatenträger in dem Datenträger, der die Warteschlange enthält; in diesem Teildatenträger erstellt der Warteschlangenserver neue Nachrichtenüberlaufdateien. Geben Sie hierzu den Parameter `--msgofsubvol` ein.
- Den Namen eines Längenzählers, der (wenn er für eine aktive Messung verwendet wird) mit der aktuellen Warteschlangenlänge initialisiert wird und vom Warteschlangenserver, dem die Warteschlange zugeordnet ist, beim Einreihen bzw. Abrufen von Nachrichten entsprechend erhöht bzw. vermindert wird. Geben Sie hierzu den Parameter `--meascount` ein.
- Die Größe des Primär- und des Sekundärspeicherbereichs sowie die maximale Anzahl an Speicherbereichen für die Warteschlangendatei. Geben Sie hierzu den Parameter `--qsize (primaryextent,secondaryextent,maxextents)` ein. In diesem Parameter müssen alle Werte angegeben werden.
- Die Größe des Primär- und des Sekundärspeicherbereichs sowie die maximale Anzahl an Speicherbereichen für die Warteschlangenüberlaufdatei. Geben Sie hierzu den Parameter `--osize (primaryextent,secondaryextent,maxextents)` ein. In diesem Parameter müssen alle Werte angegeben werden.

Objekte Statusservern und Warteschlangenservern erneut zuordnen

Statusserver handhaben alle Objekte mit Ausnahme von lokalen Warteschlangen und von Modellwarteschlangen. Für lokale Warteschlangen und Modellwarteschlangen sind die Warteschlangenserver zuständig.

Für alle Objekte wird bei ihrer Erstellung festgelegt, dass sie je nachdem den Standardwarteschlangenserver oder den Standardstatusserver verwenden. *Nach* der Erstellung eines Objekts können Sie mit dem Befehl **altmqfls** angeben, dass das Objekt einen anderen als den Standardserver verwenden soll. Das Objekt kann jedoch erst verwendet werden, nachdem Sie eine neue PATHWAY-Serverklasse für den Server erstellt und diesen gestartet haben.

Sie können in der Befehlszeile für **altmqfls** entweder einen Prozessnamen oder DEFAULT eingeben. Bei der Rekonfigurierung des Objekts wird nicht überprüft, ob der neue Server aktiv ist oder konfiguriert wurde.

Mit dem Befehl **dspmqls** können Sie den von einem Objekt aktuell verwendeten Statusserver anzeigen.

Der Warteschlangenserver kann für lokale Warteschlangen und für Modellwarteschlangen festgelegt werden. Dynamische Warteschlangen übernehmen den Warteschlangenserver der Modellwarteschlange, auf deren Basis sie erstellt wurden.

Inhalt einer lokalen Warteschlange löschen

Mit folgendem Befehl löschen Sie die Nachrichten in einer lokalen Warteschlange mit dem Namen MAGENTA.QUEUE:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

In folgenden Fällen können Sie den Inhalt einer Warteschlange nicht löschen:

- Die Warteschlange enthält nicht festgeschriebene Nachrichten, die an einem Synchronisationspunkt in die Warteschlange eingereiht wurden.
- Eine Anwendung hat die Warteschlange geöffnet.

Warteschlangen durchsuchen

```
PutDate : '19941124' PutTime : '11240694'
ApplOriginData : ' '

**** Nachricht ****

length - 36 bytes

00000000: 0000 0002 0000 0024 0000 0001 0000 0015 '.....$......'
00000010: 0000 0001 0000 0001 0000 0000 0000 0000 '.....'
00000020: 0000 0000 '.....'

Keine weiteren Nachrichten
MQCLOSE
MQDISC
```

Mit Aliaswarteschlangen arbeiten

Eine Aliaswarteschlange ermöglicht das Umleiten von MQI-Aufrufen. Bei einer Aliaswarteschlange handelt es sich nicht um eine tatsächliche Warteschlange, sondern um eine Definition, deren Name in den Namen einer tatsächlichen Warteschlange aufgelöst wird. Die Definition einer Aliaswarteschlange enthält den Namen einer Zielwarteschlange, der durch das Attribut TARGQ (*BaseQName* in PCF) angegeben wird. Wenn eine Anwendung in einem MQI-Aufruf den Namen einer Aliaswarteschlange angibt, löst der WS-Manager diesen Namen zur Laufzeit in den Namen der tatsächlichen Warteschlange auf.

Es wurde zum Beispiel eine Anwendung entwickelt, die Nachrichten in eine Warteschlange mit dem Namen MY.ALIAS.QUEUE einreicht. Bei einem MQOPEN-Aufruf und einem PUT zum Einreihen einer Nachricht gibt die Anwendung den Namen dieser Warteschlange an. Der Anwendung ist nicht bekannt, dass es sich bei der Warteschlange um eine Aliaswarteschlange handelt. Bei jedem MQI-Aufruf mit diesem Aliasnamen löst der WS-Manager den Aliasnamen in den Namen der tatsächlichen Warteschlange auf, bei der es sich sowohl um eine lokale als auch um eine ferne Warteschlange dieses WS-Managers handeln kann.

Indem Sie den Wert des Attributs TARGQ ändern, können Sie MQI-Aufrufe an eine andere Warteschlange, gegebenenfalls auch die eines anderen WS-Managers, umleiten. Dies ist hilfreich für Verwaltungs- und Migrationsaufgaben und um eine gleichmäßige Systemauslastung zu erreichen.

Eine Aliaswarteschlange definieren

Der folgende Befehl erstellt eine Aliaswarteschlange:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (YELLOW.QUEUE)
```

Dieser Befehl leitet MQI-Aufrufe, in denen die Warteschlange MY.ALIAS.QUEUE angegeben ist, an die Warteschlange YELLOW.QUEUE um. Der Befehl erstellt die Zielwarteschlange jedoch nicht, d. h., die MQI-Aufrufe schlagen fehl, wenn die Warteschlange YELLOW.QUEUE zur Laufzeit nicht existiert.

Aliaswarteschlange definieren

Sie können die MQI-Aufrufe an eine andere Warteschlange umleiten, indem Sie die Definition der Aliaswarteschlange ändern. Beispiel:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (MAGENTA.QUEUE) REPLACE
```

Dieser Befehl leitet MQI-Aufrufe an eine andere Warteschlange mit dem Namen MAGENTA.QUEUE weiter.

Mit Hilfe von Aliaswarteschlangen können Sie einer einzelnen Warteschlange (der Zielwarteschlange) verschiedene Attribute für verschiedene Anwendungen zuweisen. Definieren Sie zu diesem Zweck zwei Aliasnamen, für jede Anwendung einen. Angenommen, es gibt zwei Anwendungen:

- Anwendung ALPHA kann Nachrichten in Warteschlange YELLOW.QUEUE einreihen, darf jedoch keine Nachrichten daraus abrufen.
- Anwendung BETA kann Nachrichten aus der Warteschlange YELLOW.QUEUE abrufen, darf jedoch keine Nachrichten einreihen.

Hier können Sie den folgenden Befehl eingeben:

```
* Dieser Aliasname ist PUT-aktiviert und GET-inaktiviert für Anwendung ALPHA
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +
    TARGQ (YELLOW.QUEUE) +
    PUT (ENABLED) +
    GET(DISABLED)

* Dieser Aliasname ist PUT-inaktiviert und GET-aktiviert für Anwendung BETA
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +
    TARGQ (YELLOW.QUEUE) +
    PUT (DISABLED) +
    GET(ENABLED)
```

ALPHA verwendet in ihren MQI-Aufrufen den Warteschlangennamen ALPHAS.ALIAS.QUEUE, und BETA verwendet den Warteschlangennamen BETAS.ALIAS.QUEUE. Beide greifen auf dieselbe Warteschlange zu, jedoch auf unterschiedliche Weise.

Die Attribute LIKE und REPLACE können Sie beim Definieren von Aliaswarteschlangen auf dieselbe Weise verwenden wie beim Definieren von lokalen Warteschlangen.

Andere Befehle für Aliaswarteschlangen verwenden

Zum Anzeigen oder Ändern der Attribute von Aliaswarteschlangen bzw. zum Löschen des Aliaswarteschlangenobjekts können Sie die entsprechenden MQSC-Befehle verwenden.

Andere Befehle für Aliaswarteschlangen verwenden

Beispiel:

```
* Attribute des Warteschlangenalias anzeigen
* ALL = Alle Attribute anzeigen

DISPLAY QUEUE (ALPHAS.ALIAS.QUEUE) ALL

* ALTER = Basiswarteschlangenname ändern, in den der Aliasname aufgelöst wird.
* FORCE = Änderung erzwingen, auch wenn Warteschlange geöffnet ist.

ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGQ(ORANGE.LOCAL.QUEUE) FORCE

* Diese Warteschlange löschen, falls möglich.

DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Eine Aliaswarteschlange kann zum Beispiel in folgenden Fällen nicht gelöscht werden: Eine Anwendung hat die Aliaswarteschlange geöffnet, oder sie hat eine Warteschlange geöffnet, in deren Name der Name der Aliaswarteschlange aufgelöst wird. Weitere Informationen hierzu und zu anderen Befehlen für Aliaswarteschlangen finden Sie im Handbuch *MQSeries MQSC - Befehle*.

Mit Modellwarteschlangen arbeiten

Ein WS-Manager erstellt eine *dynamische Warteschlange*, wenn er von einer Anwendung einen MQI-Aufruf empfängt, in dem der Name einer Warteschlange angegeben ist, die als Modellwarteschlange definiert wurde. Der Name der neuen dynamischen Warteschlange wird beim Erstellen der Warteschlange vom WS-Manager generiert. Eine *Modellwarteschlange* ist eine Vorlage, in der die Attribute für alle dynamischen Warteschlangen, die auf ihrer Basis erstellt werden, angegeben sind.

Mit Hilfe von Modellwarteschlangen können Anwendungen bei Bedarf sehr einfach Warteschlangen erstellen.

Eine Modellwarteschlange definieren

Eine Modellwarteschlange mit bestimmten Attributen wird auf dieselbe Weise definiert wie eine lokale Warteschlange. Modellwarteschlangen und lokale Warteschlangen verfügen über dieselben Attribute, außer dass Sie für Modellwarteschlangen angeben können, ob die dynamischen Warteschlangen als temporäre oder permanente Warteschlangen erstellt werden sollen. (Permanente Warteschlangen bleiben beim Neustart des WS-Managers erhalten, temporäre Warteschlangen gehen verloren). Beispiel:

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
  DESCR('Warteschlange für Nachrichten von Anwendung X') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL) +
  DEFTYPE (PERMDYN)
```

Modellwarteschlange definieren

Dieser Befehl erstellt eine Modellwarteschlangendefinition. Das Attribut DEFTYPE legt fest, dass die auf Basis dieser Vorlage erstellten Warteschlangen als permanente dynamische Warteschlangen erstellt werden.

Anmerkung: Die nicht angegebenen Attribute werden automatisch aus der Standardwarteschlange SYSYTEM.DEFAULT.MODEL.QUEUE kopiert.

Die Attribute LIKE und REPLACE können Sie beim Definieren von Modellwarteschlangen auf dieselbe Weise verwenden wie beim Definieren von lokalen Warteschlangen.

Andere Befehle für Modellwarteschlangen verwenden

Zum Anzeigen oder Ändern der Attribute einer Modellwarteschlange bzw. zum Löschen des Modellwarteschlangenobjekts können Sie die entsprechenden MQSC-Befehle verwenden. Beispiel:

```
* Attribute der Modellwarteschlange anzeigen
* ALL = Alle Attribute anzeigen

DISPLAY QUEUE (GREEN.MODEL.QUEUE) ALL

* ALTER = Modell ändern, um PUT-Aufrufe für alle dynamischen Warteschlangen
* zu aktivieren, die aus diesem Modell erstellt werden.

ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)

* Diese Modellwarteschlange löschen:

DELETE QMODEL (RED.MODEL.QUEUE)
```

Objekte für Auslösefunktion verwalten

MQSeries bietet eine Funktion zum automatischen Starten einer Anwendung, wenn in einer Warteschlange bestimmte Bedingungen zutreffen. Ein Beispiel für eine solche Bedingung ist, wenn die Anzahl der Nachrichten in einer Warteschlange eine bestimmte Größe erreicht. Diese Funktion, die so genannte *Auslösefunktion* (Triggering), wird im Handbuch *MQSeries Application Programming Guide* ausführlich beschrieben. In diesem Abschnitt wird beschrieben, wie die erforderlichen Objekte zur Unterstützung der Auslösefunktion in MQSeries for Compaq NSK definiert werden.

Anwendungswarteschlange für die Auslösefunktion definieren

Eine Anwendungswarteschlange ist eine lokale Warteschlange, die von Anwendungen über MQI-Aufrufe zum Übertragen von Nachrichten verwendet wird. Für die Auslösefunktion müssen mehrere Warteschlangenattribute für die Anwendungswarteschlange definiert werden. Die Auslösefunktion selbst wird durch das Attribut *Trigger* (TRIGGER in MQSC) aktiviert.

Anwendungswarteschlange für Auslösefunktion

Im folgenden Beispiel wird ein Auslöseereignis generiert, wenn die lokale Warteschlange MOTOR.INSURANCE.QUEUE 100 Nachrichten mit der Priorität 5 oder höher enthält:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +  
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +  
MAXMSGL (2000) +  
DEFPSIST (YES) +  
INITQ (MOTOR.INS.INIT.QUEUE) +  
TRIGGER +  
TRIGTYPE (DEPTH) +  
TRIGDPTH (100)+  
TRIGMPRI (5)
```

Dabei gilt:

QLOCAL (MOTOR.INSURANCE.QUEUE)

Gibt den Namen der zu definierenden Anwendungswarteschlange an.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

Gibt den Namen der Anwendung an, die von einem Auslösemonitorprogramm gestartet wird.

MAXMSGL (2000)

Gibt die maximale Länge für Nachrichten in der Warteschlange an.

DEFPSIST (YES)

Gibt an, dass Nachrichten in dieser Warteschlange permanent sind.

INITQ (MOTOR.INS.INIT.QUEUE)

Dies ist der Name der Initialisierungswarteschlange, in die der WS-Manager die Auslösenachrichten einreicht.

TRIGGER

Dies ist der Wert für das Attribut 'Trigger'.

TRIGTYPE (DEPTH)

Gibt an, dass ein Auslöseereignis generiert wird, sobald die Anzahl der Nachrichten mit der in TRIMPRI angegebenen Priorität den in TRIGDPTH angegebenen Wert erreicht.

TRIGDPTH (100)

Gibt die Anzahl der Nachrichten an, die zum Generieren eines Auslöseereignisses erforderlich sind.

TRIGMPRI (5)

Dies ist die Priorität der Nachrichten, die der WS-Manager zählen muss, um zu entscheiden, ob ein Auslöseereignis generiert wird. Es werden nur Nachrichten mit Priorität 5 oder höher gezählt.

Eine Initialisierungswarteschlange definieren

Sobald ein Auslöseereignis eintritt, reißt der WS-Manager eine Auslösenachricht in die Initialisierungswarteschlange ein, die in der Definition der Anwendungswarteschlange angegeben wurde. Für Initialisierungswarteschlangen gibt es keine besonderen Einstellungen, Sie können jedoch die folgende Definition der lokalen Warteschlange MOTOR.INS.INIT.QUEUE als Schablone verwenden:

```
DEFINE QLOCAL(MOTOR.INS.INIT.QUEUE) +
  GET (ENABLED) +
  NOSHARE +
  NOTRIGGER +
  MAXMSGL (2000) +
  MAXDEPTH (10)
```

Eine Prozessdefinition erstellen

Eine Prozessdefinition wird mit dem Befehl DEFINE PROCESS erstellt. Eine Prozessdefinition ordnet einer Anwendungswarteschlange die Anwendung zu, von der Nachrichten aus der Warteschlange verarbeitet werden. Dazu wird das Attribut PROCESS für die Anwendungswarteschlange MOTOR.INSURANCE.QUEUE verwendet. Der folgende MQSC-Befehl erstellt den in diesem Beispiel angegebenen Prozess MOTOR.INSURANCE.QUOTE.PROCESS:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
  DESCR ('Nachricht mit Versicherungsantrag verarbeiten') +
  APPLTYPE (NSK) +
  APPLICID ('$DATA1.TEST.IRMPO1') +
  USERDATA ('öffnen, schließen, 235')
```

Dabei gilt:

MOTOR.INSURANCE.QUOTE.PROCESS

Name der Prozessdefinition.

DESCR ('Nachricht mit Versicherungsantrag verarbeiten')

Der beschreibende Text (hinter dem Schlüsselwort) für das Anwendungsprogramm, auf das sich die Definition bezieht. Dieser Text wird angezeigt, wenn Sie den Befehl DISPLAY PROCESS verwenden. Anhand des Textes können Sie die jeweilige Funktion des Prozesses erkennen. Wenn die Zeichenfolge Leerzeichen enthält, müssen Sie sie in einfache Anführungszeichen setzen.

APPLTYPE(NSK)

Der Anwendungstyp, der unter Compaq NSK ausgeführt wird.

APPLICID ('\$DATA1.TEST.IRMPO1')

Der Name des ausführbaren Programms der Anwendung auf dem lokalen System.

USERDATA ('öffnen, schließen, 235')

Benutzerdefinierte Daten, die von der Anwendung verwendet werden können.

Prozessdefinition anzeigen

Prozessdefinition anzeigen

Mit dem Befehl DISPLAY PROCESS und dem Schlüsselwort ALL können Sie die Ergebnisse Ihrer Definition prüfen. Beispiel:

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL

      24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Details zu DISPLAY PROCESS werden angezeigt.
DESCR (Nachrichten mit Versicherungsantrag verarbeiten)
APPLICID ($DATA1.TEST.IRMP01)
ENVRDATA ( )
USERDATA (öffnen, schließen, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (NSK)
```

Bei der Zeichenfolge USERDATA handelt es sich um die Argumente, die an die ausgelöste Anwendung übergeben werden. Anhand der Beispielprogramme AMQSTRG0 und AMQINQA (im Teildatenträger ZMQSSMPL) können Sie sehen, wie Auslösemonitore und Anwendungen, die ausgelöst werden, geschrieben werden.

Mit dem Befehl MQSC ALTER PROCESS können Sie eine vorhandene Prozessdefinition ändern, und mit dem Befehl DELETE PROCESS können Sie Prozessdefinitionen löschen.

Kapitel 6. Verwaltungs-Tasks automatisieren

Dieses Kapitel setzt voraus, dass Sie bereits über Erfahrungen mit dem Verwalten von MQSeries-Objekten verfügen.

Irgendwann werden Sie zu der Überzeugung kommen, dass es gut für Ihre Installation ist, einige Verwaltungs- und Überwachungs-Tasks zu automatisieren. Sie können Verwaltungs-Tasks sowohl für lokale als auch ferne WS-Manager mit Hilfe von PCF-Befehlen (Programmable Command Format) automatisieren.

In diesem Kapitel wird Folgendes beschrieben:

- Die Verwendung von PCF-Befehlen zum Automatisieren von Verwaltungs-Tasks (unter „Verwaltungs-Tasks über PCF-Befehle ausführen“ auf Seite 33).
- Die Verwendung des Befehlsservers (unter „Befehlsserver für Fernverwaltung verwalten“ auf Seite 123).

PCF-Befehle

PCF-Befehle in MQSeries dienen dazu, Verwaltungs-Tasks innerhalb eines Verwaltungsprogramms auszuführen. Auf diese Weise können Sie in einem Programm Warteschlangen, Prozessdefinitionen, Kanäle und Namenslisten erstellen sowie WS-Manager ändern.

PCF-Befehle decken denselben Funktionsbereich ab, der auch von der MQSC-Funktion bereitgestellt wird.

Daher können Sie ein Programm schreiben, das PCF-Befehle von einem einzelnen Knoten aus an jeden beliebigen WS-Manager im Netz ausgibt. Auf diese Weise können Sie Verwaltungs-Tasks sowohl zentralisieren als auch automatisieren.

Jeder PCF-Befehl besteht aus einer Datenstruktur, die in den Bereich der Anwendungsdaten einer MQSeries-Nachricht eingebettet wird. Jeder einzelne Befehl wird, wie jede andere Nachricht auch, mit dem MQI-Aufruf MQPUT an den Ziel-WS-Manager gesendet. Der Befehlsserver des WS-Managers, der die Nachricht empfängt, interpretiert sie als Befehlsnachricht und führt den Befehl aus. Zum Abrufen der Antwort gibt die Anwendung einen MQGET-Aufruf aus, und die Daten werden in einer anderen Datenstruktur zurückgegeben. Die Anwendung kann die Antwort dann verarbeiten und entsprechend reagieren.

Anmerkung: Anders als MQSC-Befehle besitzen PCF-Befehle und deren Antworten kein lesbares Textformat.

Im Folgenden ist kurz zusammengefasst, was der Anwendungsprogrammierer beim Erstellen einer PCF-Befehlsnachricht angeben muss:

Nachrichtendeskriptor

Hierbei handelt es sich um einen standardmäßigen MQSeries-Nachrichtendeskriptor mit folgenden Angaben:

Die Nachrichtenart (*MsgType*) ist MQMT_REQUEST.

Das Nachrichtenformat (*Format*) ist MQFMT_ADMIN.

mit PCF-Befehlen

Anwendungsdaten

Sie enthalten die PCF-Nachricht einschließlich PCF-Header mit folgenden Angaben:

Als PCF-Nachrichtenart (*Type*) ist MQCFT_COMMAND angegeben.

Die Befehls-ID gibt den Befehl an, z. B. *Warteschlange ändern* (MQCMD_CHANGE_Q).

Eine vollständige Beschreibung der PCF-Datenstrukturen sowie Anweisungen zu deren Implementierung finden Sie im Handbuch *MQSeries Programmable System Management*.

Attribute in MQSC- und PCF-Befehlen

Objektattribute, die in MQSC angegeben werden, werden in diesem Buch in Großbuchstaben geschrieben (z. B. RQMNAME), obwohl bei ihnen die Groß-/Kleinschreibung nicht beachtet werden muss. Namen von MQSC-Attributen dürfen maximal acht Zeichen lang sein.

Objektattribute in PCF-Befehlen, die nicht auf acht Zeichen begrenzt sind, werden kursiv dargestellt. Beispiel: Dem MQSC-Attribut RQMNAME entspricht das PCF-Attribut *RemoteQMgrName*.

PCF-Escape-Befehle

PCF-Escape-Befehle sind PCF-Befehle, deren Nachrichtentext MQSC-Befehle enthalten. Mit PCF-Befehlen können Sie Befehle an einen fernen WS-Manager senden. Weitere Informationen zu PCF-Escape-Befehlen finden Sie im Handbuch *MQSeries Programmable System Management*.

Verwendung von PCF-Befehlen mit Hilfe von MQAI vereinfachen

Bei MQAI (MQSeries Administration Interface) handelt es sich um eine Verwaltungsschnittstelle für MQSeries, die jetzt auf der Compaq NSK-Plattform zur Verfügung steht.

Sie führt Verwaltungs-Tasks in einem WS-Manager mit Hilfe von *Datenbehältern* (Data Bags) aus. Mit diesen Datenbehältern können Eigenschaften (oder Parameter) von Objekten leichter bearbeitet werden als mit PCF-Befehlen.

Verwendungsmöglichkeiten der Verwaltungsschnittstelle MQAI:

- **Einfachere Verwendung von PCF-Nachrichten.** Die Verwaltungsschnittstelle MQAI erleichtert die Verwaltung von MQSeries. Sie müssen keine eigenen PCF-Nachrichten erstellen, wodurch die Probleme, die üblicherweise mit komplexen Datenstrukturen verbunden sind, vermieden werden.

Um Parameter in Programmen, die mit MQI-Aufrufen geschrieben wurden, zu übergeben, muss die PCF-Nachricht den Befehl und Einzelangaben der Zeichenfolge bzw. ganzzahlige Daten enthalten. Zu diesem Zweck muss das Programm für jede Struktur mehrere Anweisungen enthalten, darüber hinaus muss Speicherbereich zugeordnet werden. Dies ist eine langwierige und arbeitsintensive Aufgabe.

Dagegen übergeben Programme, die MQAI verwenden, Parameter an den entsprechenden Datenbehälter (Data Bag) und müssen nur eine Anweisung für jede Struktur enthalten. Durch die Verwendung von MQAI-Datenbehältern entfällt die Notwendigkeit, Feldgruppen zu bearbeiten und Speicher anzulegen; darüber hinaus wird dadurch eine gewisse Isolierung von den Einzelangaben des PCF-Befehls erreicht.

- **Einfachere Handhabung von Fehlerbedingungen.** Es ist schwierig, Rückkehr-codes von MQSC-Befehlen zu erhalten; durch die Verwendung von MQAI ist es für das Programm jedoch einfacher, Fehlerbedingungen zu bearbeiten.

Nachdem Sie einen Datenbehälter erstellt und mit Daten gefüllt haben, können Sie mit dem Aufruf 'mqExecute' eine Nachricht mit einem Verwaltungsbefehl an den Befehlsserver eines WS-Managers senden. Der Aufruf wartet anschließend auf Antwortnachrichten. Der Aufruf 'mqExecute' übernimmt den Datenaustausch mit dem Befehlsserver und gibt Antworten in einem Datenbehälter zurück.

Weitere Informationen zur Verwendung der Schnittstelle MQAI finden Sie im Handbuch *MQSeries Administration Interface Programming Guide and Reference*.

Weitere allgemeine Informationen zu PCF-Befehlen finden Sie im Handbuch *MQSeries Programmable System Management*.

Befehlsserver für Fernverwaltung verwalten

Jedem WS-Manager kann ein Befehlsserver zugeordnet werden. Ein Befehlsserver verarbeitet alle von fernen WS-Managern eingehenden Befehle bzw. PCF-Befehle von Anwendungen. Er übergibt die Befehle zur Verarbeitung an den WS-Manager und gibt, abhängig vom Absender des Befehls, einen Beendigungscode oder eine Bedienernachricht zurück.

Ein Befehlsserver ist für alle Verwaltungsfunktionen, die PCF-Befehle oder MQAI verwenden, sowie für die Fernverwaltung zwingend erforderlich.

Anmerkung: Bei einer Fernverwaltung müssen Sie sicherstellen, dass der Ziel-WS-Manager aktiv ist. Andernfalls können die Nachrichten mit den Befehlen den WS-Manager, von dem sie ausgegeben werden, nicht verlassen. Stattdessen werden diese Nachrichten in die lokale Übertragungswarteschlange eingereiht, die den fernen WS-Manager versorgt. Diese Situation sollte unter allen Umständen vermieden werden.

Befehlsserver starten

Der Befehlsserver wird mit folgendem Befehl gestartet:

```
strmqcsv saturn.queue.manager
```

Dabei steht saturn.queue.manager für den WS-Manager, für den der Befehlsserver gestartet wird.

Der Befehlsserver kann auch gestartet werden, indem die PATHCOM-Befehle THAW SERVER und START SERVER für seine Serverklasse ausgeführt werden.

Befehlsserver für Fernverwaltung

Status des Befehlsservers anzeigen

Bei einer Fernverwaltung müssen Sie sicherstellen, dass der Befehlsserver des Ziel-WS-Managers aktiv ist. Ist er nicht aktiv, können ferne Befehle nicht verarbeitet werden. Alle Nachrichten mit Befehlen werden in die Befehlswarteschlange des Ziel-WS-Managers eingereiht.

Der Befehl zum Anzeigen des Status des Befehlsservers eines WS-Managers, in diesem Beispiel `saturn.queue.manager`, lautet wie folgt:

```
dspmqcsv saturn.queue.manager
```

Diesen Befehl müssen Sie auf der Zielmaschine ausgeben. Wenn der Befehlsserver aktiv ist, wird folgende Nachricht zurückgegeben:

```
AMQ8027 Status des MQSeries-Befehlsservers: Aktiv
```

Befehlsserver stoppen

Der Befehl zum Beenden eines Befehlsservers lautet in diesem Beispiel:

```
endmqcsv saturn.queue.manager
```

Sie haben zwei Möglichkeiten, den Befehlsserver zu stoppen:

- Verwenden Sie für einen gesteuerten Stopp den Befehl **endmqcsv** mit der Option `-c` (dies ist der Standard).
- Verwenden Sie für einen sofortigen Stopp den Befehl **endmqcsv** mit der Option `-i`.

Anmerkung: Beim Stoppen eines WS-Managers wird auch der ihm zugeordnete Befehlsserver gestoppt (sofern er gestartet wurde).

Kapitel 7. Ferne MQSeries-Objekte verwalten

In diesem Kapitel wird beschrieben, wie MQSeries-Objekte in einem anderen WS-Manager verwaltet werden. Außerdem wird hier beschrieben, wie über ferne Warteschlangenobjekte die Zieladressen für Nachrichten und Antwortnachrichten gesteuert werden können.

Es enthält die folgenden Abschnitte:

- „Einen fernen WS-Manager verwalten“ auf Seite 127
- „Eine lokale Definition für eine ferne Warteschlange erstellen“ auf Seite 133
- „Definitionen ferner Warteschlangen als Aliasnamen verwenden“ auf Seite 136

Weitere Informationen zu Kanälen und Kanalattributen sowie zu deren Konfiguration finden Sie im Handbuch *MQSeries Intercommunication*.

Kanäle, Cluster und fernes Queuing

Ein WS-Manager kommuniziert mit einem anderen WS-Manager, indem er eine Nachricht sendet und, falls erforderlich, ein Antwort erhält. Für den empfangenden WS-Manager gibt es in Bezug auf seinen Standort folgende Möglichkeiten:

- Er befindet sich auf derselben Maschine.
- Er befindet sich auf einer anderen Maschine an demselben Standort oder am anderen Ende der Welt.
- Er wird auf derselben Plattform als lokaler WS-Manager ausgeführt.
- Er wird auf einer anderen Plattform, die von MQSeries unterstützt wird, ausgeführt.

Die Nachrichten können folgende Absender haben:

- Benutzerdefinierte Anwendungsprogramme, die Daten von einem Knoten an einen anderen übertragen.
- Benutzerdefinierte Anwendungsprogramme, die PCF-Befehle oder die Schnittstelle MQAI verwenden.
- WS-Manager, die Folgendes senden:
 - Instrumentierungsereignisnachrichten an andere WS-Manager
 - MQSC-Befehle, die von einem Befehl **runmqsc** im indirekten Modus ausgegeben werden (wobei die Befehle in einem anderen WS-Manager ausgeführt werden).

Damit eine Nachricht an einen fernen WS-Manager gesendet werden kann, benötigt der lokale WS-Manager einen Mechanismus, mit dem er den Eingang von Nachrichten erkennen und sie transportieren kann. Dieser Mechanismus muss folgende Komponenten umfassen:

- mindestens einen Kanal
- eine Übertragungswarteschlange
- einen Nachrichtenkanalagenten (MCA)
- ein Kanalempfangsprogramm
- einen Kanalinitiator

Ein Kanal ist eine einseitige Übertragungsverbindung zwischen zwei WS-Managern, über die Nachrichten an beliebig viele Warteschlangen des fernen WS-Managers übertragen werden können.

Ferne Objekte verwalten

Für jeden Ausgang des Kanals gibt es eine separate Definition. Wenn zum Beispiel an dem einen Ausgang ein Sender oder Server ist, muss am anderen Ausgang ein Empfänger oder Requester sein. Ein einfacher Kanal besteht aus einer *Definition eines Senderkanals* am Ausgang mit dem lokalen WS-Manager und einer *Definition eines Empfängerkanals* am Ausgang mit dem fernen WS-Manager. Die beiden Definitionen müssen denselben Namen haben und zusammen einen einfachen Kanal bilden.

Wenn der ferne WS-Manager auf Nachrichten, die vom lokalen WS-Manager gesendet werden, antworten soll, muss ein zweiter Kanal konfiguriert werden, über den Antworten zurück an den lokalen WS-Manager gesendet werden.

Kanäle werden mit dem MQSC-Befehl DEFINE CHANNEL definiert. Für die Beispiele in diesem Kapitel gelten die Standardkanalattribute, sofern nicht anders angegeben.

An jedem Kanalausgang gibt es einen Nachrichtenkanalagenten (MCA), der das Senden und Empfangen von Nachrichten steuert. Seine Aufgabe besteht darin, Nachrichten aus der Übertragungswarteschlange abzurufen und sie an die Kommunikationsverbindung zwischen den WS-Managern zu übergeben. Der Nachrichtenkanalagent auf der Empfängerseite dagegen ruft Nachrichten aus der Kommunikationsverbindung ab und reiht sie in die Zielwarteschlangen ein.

Eine Übertragungswarteschlange ist eine lokale Warteschlange mit einer besonderen Funktion, in der Nachrichten temporär aufbewahrt werden, bevor sie vom Nachrichtenkanalagenten abgerufen und an den fernen WS-Manager gesendet werden. Der Name der Übertragungswarteschlange wird in einer *Definition einer fernen Warteschlange* angegeben.

„Kanäle und Übertragungswarteschlangen für Fernverwaltung vorbereiten“ auf Seite 128 zeigt, wie diese Definitionen zum Konfigurieren einer Fernverwaltung verwendet werden.

Weitere allgemeine Informationen zum Konfigurieren eines verteilten Queuing finden Sie im Handbuch *MQSeries Intercommunication*.

Fernverwaltung mit Hilfe von Clustern

In einem herkömmlichen MQSeries-Netz arbeitet bei einem verteilten Message-Queuing jeder WS-Manager unabhängig. Damit ein WS-Manager Nachrichten an einen anderen WS-Manager senden kann, muss für ihn eine Übertragungswarteschlange, ein Kanal zum fernen WS-Manager und eine Definition einer fernen Warteschlange definiert werden, und das für jede Warteschlange, an die der WS-Manager Nachrichten senden soll.

Bei einem *Cluster* handelt es sich um eine Gruppe von WS-Managern, die so konfiguriert sind, dass sie über ein einziges Netz direkt miteinander kommunizieren können, ohne dass dazu komplexe Übertragungswarteschlangen, Kanäle und Warteschlangendefinitionen erforderlich sind. Cluster können einfach konfiguriert werden und bestehen in der Regel aus WS-Managern, zwischen denen logische Beziehungen bestehen und die Daten oder Anwendungen gemeinsam benutzen.

Nachdem ein Cluster erstellt wurde, können die WS-Manager miteinander kommunizieren, *ohne dass komplizierte Definitionen für Kanäle oder ferne Warteschlangen benötigt werden*. Selbst der kleinste Cluster trägt dazu bei, den Aufwand für die Systemverwaltung zu senken.

Für den Aufbau eines Netzes von WS-Managern in einem Cluster sind weniger Definitionen erforderlich als für den Aufbau einer herkömmlichen Umgebung für verteiltes Queuing. Je weniger Definitionen Sie erstellen, desto schneller und einfacher können Sie ein Netz installieren oder ändern. Darüber hinaus wird das Risiko von Fehlern in den Definitionen verringert.

Zum Konfigurieren eines Clusters benötigen Sie normalerweise pro WS-Manager eine Definition für einen Clustersender (CLUSSDR) und eine Definition für einen Clusterempfänger (CLUSRCVR). Sie benötigen keine Definitionen für Übertragungswarteschlangen oder ferne Warteschlangen. Die Regeln der Fernverwaltung gelten auch in einem Cluster, die Definitionen selbst sind jedoch wesentlich einfacher.

Weitere Informationen zu Clustern und Clusterattributen sowie zu deren Konfiguration finden Sie im Handbuch *Cluster-Unterstützung in MQSeries*.

Einen fernen WS-Manager verwalten

In diesem Abschnitt wird erläutert, wie ein ferner WS-Manager über einen lokalen WS-Manager verwaltet wird. Sie können die Fernverwaltung von einem lokalen Knoten aus folgendermaßen implementieren:

- mit MQSC-Befehlen
- mit PCF-Befehlen

Die Vorbereitung der Warteschlangen und Kanäle ist in beiden Fällen nahezu dieselbe. Die Beispiele in diesem Buch enthalten MQSC-Befehle, da sie einfacher zu verstehen sind. Sie können die Beispiele jedoch bei Bedarf ohne weiteres auf PCF-Befehle übertragen. Weitere Informationen zum Erstellen von Verwaltungsprogrammen mit Hilfe von PCF-Befehlen finden Sie im Handbuch *MQSeries Programmable System Management*.

Bei der Fernverwaltung senden Sie MQSC-Befehle an einen fernen WS-Manager, entweder interaktiv oder aus einer Textdatei mit den Befehlen. Der ferne WS-Manager kann sich auf derselben Maschine oder, was typischer ist, auf einer anderen Maschine befinden. Die Fernverwaltung von WS-Managern ist in verschiedenen MQSeries-Umgebungen möglich, einschließlich UNIX[®], Compaq NSK, AS/400[®], MVS/ESA, Windows[®] 2000 und OS/390.

Zum Implementieren der Fernverwaltung müssen Sie bestimmte Objekte erstellen. Sofern Sie keine besonderen Anforderungen stellen, können die Standardwerte (z. B. für die Nachrichtenlänge) unverändert übernommen werden.

WS-Manager für Fernverwaltung vorbereiten

Abb. 37 auf Seite 128 zeigt die Konfiguration von WS-Managern und Kanälen, die für die Fernverwaltung erforderlich sind. `source.queue.manager` ist der *Quellen-WS-Manager*, von dem aus Sie MQSC-Befehle eingeben können und an den, wenn möglich, das Ergebnis dieser Befehle (nämlich Bedienernachrichten) zurückgegeben wird. `target.queue.manager` ist der Ziel-WS-Manager, der die Befehle verarbeitet und Bedienernachrichten generiert.

Anmerkung: `source.queue.manager` muss der Standard-WS-Manager auf der Maschine, die Sie verwenden, sein. Weitere Informationen zum Erstellen eines WS-Managers finden Sie unter „`crtmqm` (WS-Manager erstellen)“ auf Seite 273.

Fernen WS-Manager verwalten

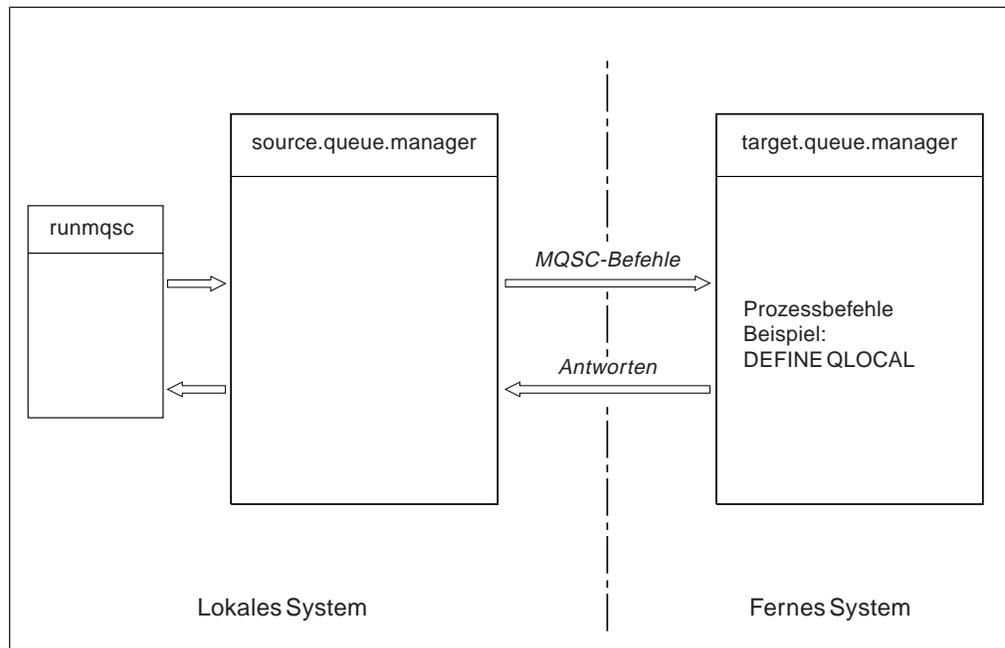


Abbildung 37. Fernverwaltung

Sofern noch nicht geschehen, müssen Sie auf beiden Systemen Folgendes tun:

- den WS-Manager mit dem Befehl **crtmqm** erstellen.
- den WS-Manager mit dem Befehl **strmqm** starten.

Weitere Informationen hierzu finden Sie unter „Standard- und Systemobjekte wiederherstellen“ auf Seite 70. Diese Befehle müssen lokal oder über eine Netzfunktion, z. B. Telnet, ausgeführt werden.

Für den Ziel-WS-Manager gilt Folgendes:

- Die Befehlswarteschlange `SYSTEM.ADMIN.COMMAND.QUEUE` muss vorhanden sein. Diese Warteschlange wird standardmäßig zusammen mit einem WS-Manager erstellt.
- Der Befehlsserver muss gestartet werden (mit dem Befehl **strmqcsv**).

Kanäle und Übertragungswarteschlangen für Fernverwaltung vorbereiten

Für die ferne Ausführung von MQSC-Befehlen müssen Sie zwei Kanäle, einen für jede Richtung, und die ihnen zugeordneten Übertragungswarteschlangen definieren. In diesem Beispiel wird davon ausgegangen, dass als Transportprotokoll TCP/IP verwendet wird und die TCP/IP-Adresse bekannt ist.

Über den Kanal `source.to.target` werden MQSC-Befehle vom Quellen-WS-Manager an die Zieladresse gesendet. Der Sender an diesem Kanal ist der WS-Manager `source.queue.manager`, und der Empfänger ist der WS-Manager `target.queue.manager`. Über den Kanal `target.to.source` werden die Ausgaben von Befehlen und alle für den Quellen-WS-Manager generierten Bedienernachrichten zurückgegeben. Außerdem müssen Sie für jeden Sender eine Übertragungswarteschlange definieren. Dabei handelt es sich um eine lokale Warteschlange mit dem Namen des empfangenden WS-Managers.

Fernen WS-Manager verwalten

Der Name der Übertragungswarteschlange (XMITQ) muss dem Namen des fernen WS-Managers entsprechen, damit eine Fernverwaltung möglich ist (außer Sie verwenden einen WS-Manageraliasnamen). Abb. 38 zeigt eine Übersicht über diese Konfiguration. Beachten Sie jedoch, dass es sich bei SYSTEM.MQSC.REPLY.QUEUE um den Namen der Modellwarteschlange handelt, auf deren Basis die MQSC-Funktion eigene dynamische Warteschlangen für Antwortnachrichten erstellt. Dieser Warteschlangennamen variiert und wird intern in der MQSC-Funktion verwendet.

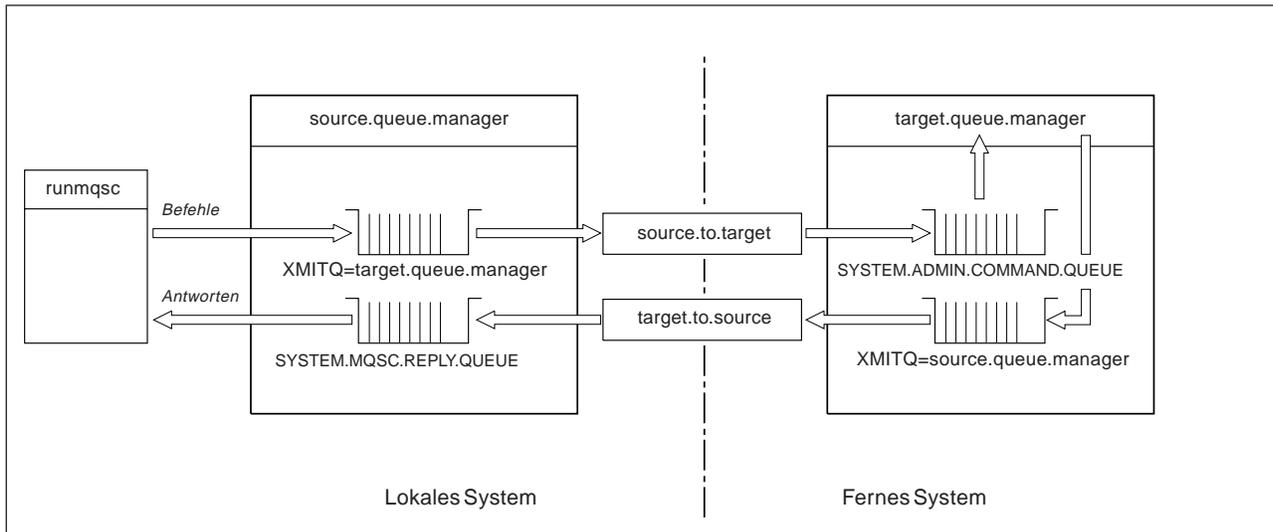


Abbildung 38. Kanäle und Warteschlangen für Fernverwaltung konfigurieren

Weitere Informationen zum Konfigurieren ferner Kanäle finden Sie im Handbuch *MQSeries Intercommunication*.

Kanäle und Übertragungswarteschlangen definieren

Geben Sie im Quellen-WS-Manager die folgenden MQSC-Befehle aus, um die Kanäle und die Übertragungswarteschlange zu definieren:

```
* Sendekanal im Quellen-WS-Manager definieren
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)

* Empfangskanal im Quellen-WS-Manager definieren
DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Übertragungswarteschlange im Quellen-WS-Manager definieren
DEFINE QLOCAL ('target.queue.manager') +
  USAGE(XMITQ)
```

Fernen WS-Manager verwalten

Geben Sie die folgenden Befehle im Ziel-WS-Manager (target.queue.manager) aus, um dort die Kanäle und die Übertragungswarteschlange zu erstellen:

```
* Sendekanal im Ziel-WS-Manager definieren

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ ('source.queue.manager') +
  TRPTYPE(TCP)

* Empfangskanal im Ziel-WS-Manager definieren

DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Übertragungswarteschlange im Ziel-WS-Manager definieren

DEFINE QLOCAL ('source.queue.manager') +
  USAGE(XMITQ)
```

Anmerkung: Die TCP/IP-Verbindungsnamen, die für das Attribut CONNAME in den Definitionen des Senderkanals angegeben sind, dienen nur der Veranschaulichung. Dabei handelt es sich um die IP-Adresse bzw. den Netznamen der Maschine am anderen Ende der Verbindung. Verwenden Sie die entsprechenden Werte für Ihr Netz.

Kanäle starten

In der folgenden Beschreibung wird angenommen, dass beide Ausgänge des Kanals in MQSeries for Compaq NSK aktiv sind. Ist dies nicht der Fall, schlagen Sie in der Dokumentation für den Ausgang des Kanals nach, der nicht unter Compaq NSK ausgeführt wird.

Stellen Sie vor dem Starten der beiden Kanäle zunächst sicher, dass die TCP-Empfangsprogrammprozesse von Compaq NSK auf beiden Knoten für MQSeries konfiguriert sind und an beiden Enden der Verbindung aktiv sind. Starten Sie anschließend die Kanäle in **runmqsc**.

- Geben Sie im Quellen-WS-Manager den folgenden Befehl ein:

```
start channel ('source.to.target')
```

- Geben Sie im Ziel-WS-Manager den folgenden Befehl ein:

```
start channel ('target.to.source')
```

Automatische Definition von Kanälen

Die automatische Kanaldefinition ist nur möglich, wenn der Ziel-WS-Manager in MQSeries Version 5.1 oder höher betrieben wird. Wenn eine eingehende Zuordnungsanforderung empfangen wird und in der Datei mit den Kanaldefinitionen (CDF, Channel Definition File) keine Definition für einen entsprechenden Empfänger oder eine entsprechende Serververbindung gefunden wird, erstellt MQSeries automatisch eine Definition und fügt sie der CDF hinzu. Automatische Definitionen basieren auf zwei von MQSeries bereitgestellten Standarddefinitionen: SYSTEM.AUTO.RECEIVER und SYSTEM.AUTO.SVRCONN.

Sie können die automatische Erstellung von Empfänger- und Serververbindungsdefinitionen aktivieren, indem Sie das WS-Managerobjekt mit dem MQSC-Befehl ALTER QMGR (oder dem PCF-Befehl 'Change Queue Manager') aktualisieren.

Weitere Informationen zum automatischen Erstellen von Kanaldefinitionen finden Sie im Handbuch *MQSeries Intercommunication*.

Informationen zur automatischen Definition von Kanälen für Cluster finden Sie im Handbuch *Cluster-Unterstützung in MQSeries*.

Ferne Ausführung von MQSC-Befehlen

Der Befehlsserver *muss* im Ziel-WS-Manager aktiv sein, wenn er eine Fernverarbeitung von MQSC-Befehlen ausführen soll. (Im Quellen-WS-Manager ist dies nicht erforderlich.)

- Geben Sie im Ziel-WS-Manager folgenden Befehl ein:

```
strmqcsv target.queue.manager
```

- Im Quellen-WS-Manager können Sie MQSC-Befehle interaktiv im Warteschlangenmodus ausführen, indem Sie den folgenden Befehl eingeben:

```
runmqsc -w 30 target.queue.manager
```

Bei diesem Format des Befehls **runmqsc** mit der Option **-w** werden die MQSC-Befehle im Warteschlangenmodus ausgeführt, d. h., Befehle werden (in einem veränderten Format) in die Eingabewarteschlange des Befehlsservers eingereiht und der Reihe nach ausgeführt.

Wenn Sie einen MQSC-Befehl eingeben, wird er an den fernen WS-Manager, in diesem Fall `target.queue.manager`, umgeleitet. Das Zeitlimit wird auf 30 Sekunden festgelegt. Wird innerhalb dieser 30 Sekunden keine Antwort empfangen, wird im lokalen (Quellen-)WS-Manager die folgende Nachricht generiert:

```
AMQ8416: MQSC-Zeitüberschreitung bei Warten auf Antwort vom Befehlsserver.
```

Am Ende der MQSC-Sitzung zeigt der lokale WS-Manager alle eingegangenen Antworten mit Zeitlimitüberschreitung an. Wenn die MQSC-Sitzung beendet wird, werden alle weiteren Antworten gelöscht.

Fernen WS-Manager verwalten

Im Warteschlangenmodus können Sie auch eine MQSC-Befehlsdatei in einem fernen WS-Manager ausführen. Beispiel:

```
runmqsc /IN mycmds, OUT report/ -w 60 target.queue.manager
```

Dabei steht mycmds für eine Datei mit MQSC-Befehlen und report für die Berichtsdatei.

Mit WS-Managern unter MVS/ESA arbeiten

Sie können von einem WS-Manager in MQSeries for Compaq NSK MQSC-Befehle an einen WS-Manager in einem MQV/ESA-System ausgeben. Dazu müssen Sie jedoch den Befehl **runmqsc** und die Kanaldefinitionen auf dem Sender ändern.

Insbesondere müssen Sie im Befehl **runmqsc** auf einem Compaq NSK-Knoten die Option **-x** angeben:

```
runmqsc -w 30 -x QMRI
```

Die Kanaldefinition sieht wie folgt aus:

```
* Sendekanal im Quellen-WS-Manager unter Compaq NSK definieren
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ (QMRI) +
  TRPTYPE(TCP) +
```

Auch hier gilt, dass Sie außerdem den Empfängerkanal und die Übertragungswarteschlange für den Quellen-WS-Manager definieren müssen. Es sei noch einmal erwähnt, dass in diesem Beispiel von TCP/IP als Übertragungsprotokoll ausgegangen wird.

Empfehlungen für fernes Queuing

Gehen Sie beim Implementieren des fernen Queuings folgendermaßen vor:

1. Erstellen Sie eine Befehlsdatei mit den MQSC-Befehlen, die auf dem fernen System ausgeführt werden sollen.
2. Überprüfen Sie die MQSC-Befehle lokal, indem Sie im Befehl **runmqsc** die Option **-v** angeben.
Sie können mit dem Befehl **runmqsc** keine MQSC-Befehle in einem anderen WS-Manager überprüfen.
3. Überprüfen Sie so genau wie möglich, ob die Befehlsdatei lokal fehlerfrei ausgeführt wird.
4. Führen Sie dann die Befehlsdatei für das ferne System aus.

Probleme bei der Verwendung von MQSC-Befehlen für die Fernverwaltung

Wenn beim Ausführen ferner MQSC-Befehle Probleme auftreten, überprüfen Sie, ob die folgenden Punkte zutreffen:

- Der Befehlsserver im Ziel-WS-Manager wurde gestartet.
- Eine gültige Übertragungswarteschlange wurde definiert.
- Beide Ausgänge der Nachrichtenkanäle wurden definiert, und zwar für beide Kanäle:
 - Den Kanal, über den die Befehle gesendet werden.
 - Den Kanal, über den die Antworten zurückgegeben werden.
- In der Kanaldefinition wurde der richtige Verbindungsname (CONNNAME) angegeben.
- Die Empfangsprogramme wurden vor den Nachrichtenkanälen gestartet.
- Es wurde überprüft, dass das Zeitintervall für die Verbindungsunterbrechung nicht abgelaufen ist, z. B., wenn ein Kanal gestartet, aber nach kurzer Zeit wieder gestoppt wurde. Dies ist besonders wichtig, wenn Sie die Kanäle manuell starten.
- Es wurde sichergestellt, dass von einem Quellen-WS-Manager keine Anforderungen gesendet werden, die der Ziel-WS-Manager nicht verstehen kann (z. B. Anforderungen, die neue Parameter enthalten).

Siehe auch „MQSC-Probleme lösen“ auf Seite 104.

Eine lokale Definition für eine ferne Warteschlange erstellen

Sie können die Definition einer fernen Warteschlange als lokale Definition einer fernen Warteschlange verwenden. Sie erstellen ein Objekt für eine ferne Warteschlange in Ihrem lokalen WS-Manager, das auf eine lokale Warteschlange in einem anderen WS-Manager verweist.

Funktionsweise von lokalen Definitionen ferner Warteschlangen

Eine Anwendung stellt eine Verbindung mit einem lokalen WS-Manager her und gibt dann einen MQOPEN-Aufruf aus. Der in diesem Aufruf angegebene Warteschlangename ist der Name der Definition einer fernen Warteschlange im lokalen WS-Manager. Die Definition der fernen Warteschlange liefert die Namen der Zielwarteschlange, des Ziel-WS-Managers und, optional, einer Übertragungswarteschlange. Um eine Nachricht in die ferne Warteschlange einzureihen, gibt die Anwendung einen MQPUT-Aufruf aus, in dem die vom MQOPEN-Aufruf zurückgegebene Kennung angegeben ist. Der WS-Manager hängt den Namen der fernen Warteschlange und den Namen des fernen WS-Managers an einen Übertragungs-Header in der Nachricht an. Anhand dieser Informationen wird die Nachricht an ihre richtige Zieladresse im Netz weitergeleitet.

Als Administrator können Sie die Zieladresse der Nachricht steuern, indem Sie die Definition der fernen Warteschlange ändern.

Beispiel

Eine Anwendung soll eine Nachricht in eine Warteschlange in einem fernen WS-Manager einreihen.

Lokale Definition für eine ferne Warteschlange erstellen

Funktionsweise

Die Anwendung stellt eine Verbindung mit einem WS-Manager her, z. B. mit saturn.queue.manager. Die Zielwarteschlange ist einem anderen WS-Manager zugeordnet.

Im MQOPEN-Aufruf gibt die Anwendung Werte in den folgenden MQOD-Feldern an:

Feldwert	Beschreibung
<i>ObjectName</i> CYAN.REMOTE.QUEUE	Gibt den lokalen Namen des Objekts für die ferne Warteschlange an. Dieses Objekt definiert die Zielwarteschlange und den Ziel-WS-Manager.
<i>ObjectType</i> (Warteschlange)	Identifiziert dieses Objekt als Warteschlange.
<i>ObjectQmgrName</i> Leerzeichen oder saturn.queue.manager	Dieses Feld ist optional. Ist es leer, wird der Name des lokalen WS-Managers verwendet. Dabei handelt es sich um den WS-manager, in dem die ferne Warteschlange definiert wurde und mit der die Anwendung verbunden ist. Ist es nicht leer, muss der Name des lokalen WS-Managers angegeben werden.

Als Nächstes gibt die Anwendung einen MQPUT-Aufruf aus, um eine Nachricht in die Warteschlange einzureihen.

Auf dem lokalen WS-Manager können Sie mit folgenden MQSC-Befehlen eine lokale Definition einer fernen Warteschlange erstellen:

```
DEFINE QREMOTE ('CYAN.REMOTE.QUEUE') +
  DESCR ('Warteschlange für Autoversicherungsanträge aus den Filialen') +
  RNAME ('AUTOMOBILE.INSURANCE.QUOTE.QUEUE') +
  RQMNAME ('jupiter.queue.manager') +
  XMITQ ('INQUOTE.XMIT.QUEUE')
```

Dabei gilt:

QREMOTE ('CYAN.REMOTE.QUEUE')

Der lokale Name des fernen Warteschlangenobjekts. Diesen Namen müssen Anwendungen, die mit diesem WS-Manager verbunden sind, im MQOPEN-Aufruf angeben, um die Warteschlange AUTOMOBILE.INSURANCE.QUOTE.QUEUE im fernen WS-Manager jupiter.queue.manager zu öffnen.

DESCR ('Warteschlange für Autoversicherungsanträge aus den Filialen')

Zusätzlicher Text, der die Verwendung der Warteschlange beschreibt.

RNAME ('AUTOMOBILE.INSURANCE.QUOTE.QUEUE')

Der Name der Zielwarteschlange im fernen WS-Manager. Dabei handelt es sich um die eigentliche Zielwarteschlange für Nachrichten, die von den Anwendungen, die die Warteschlange 'CYAN.REMOTE.QUEUE' angeben, gesendet werden. Die Warteschlange 'AUTOMOBILE.INSURANCE.QUOTE.QUEUE' muss im fernen WS-Manager als lokale Warteschlange definiert werden.

Lokale Definition für eine ferne Warteschlange erstellen

RQMNAME ('jupiter.queue.manager')

Der Name des fernen WS-Managers, dem die Zielwarteschlange 'AUTO-MOBILE.INSURANCE.QUOTE.QUEUE' zugeordnet ist.

XMITQ ('INQUOTE.XMIT.QUEUE')

Der Name der Übertragungswarteschlange. Diese Angabe ist optional; wird kein Name angegeben, wird eine Warteschlange mit demselben Namen wie der ferne WS-Manager verwendet.

In beiden Fällen muss die entsprechende Übertragungswarteschlange als lokale Warteschlange mit dem Attribut *Usage* definiert werden, das angibt, dass es sich um eine Übertragungswarteschlange handelt (USAGE(XMITQ) in MQSC).

Ein alternative Möglichkeit, Nachrichten in eine ferne Warteschlange einzureihen

Die Verwendung einer lokalen Definition einer fernen Warteschlange stellt nicht die einzige Möglichkeit dar, Nachrichten in eine ferne Warteschlange einzureihen. Anwendungen können den vollständigen Warteschlangennamen, der den Namen des fernen WS-Managers einschließt, als Teil des MQOPEN-Aufrufs angeben. In diesem Fall wird keine lokale Definition einer fernen Warteschlange benötigt. Diese Alternative setzt jedoch voraus, dass Anwendungen entweder den Namen des fernen WS-Managers kennen oder zur Laufzeit Zugriff darauf haben.

Weitere Befehle für ferne Warteschlangen verwenden

Zum Anzeigen oder Ändern der Attribute eines Objekts für eine ferne Warteschlange bzw. zum Löschen des Objekts für eine ferne Warteschlange können Sie die entsprechenden MQSC-Befehle verwenden. Beispiel:

```
* Attribute der fernen Warteschlange anzeigen
* ALL = Alle Attribute anzeigen

DISPLAY QUEUE (CYAN.REMOTE.QUEUE) ALL

* ALTER = Ferne Warteschlange ändern, um PUT-Aufrufe zu ermöglichen.
* Dies wirkt sich nicht auf die Zielwarteschlange aus, sondern
* nur auf Anwendungen, die diese ferne Warteschlange angeben.

ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)

* Diese ferne Warteschlange löschen.
* Dies wirkt sich nicht auf die Zielwarteschlange aus,
* sondern nur auf ihre lokale Definition

DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

Anmerkung: Wenn Sie eine ferne Warteschlange löschen, löschen Sie nur die lokale Darstellung der fernen Warteschlange. Sie löschen nicht die ferne Warteschlange selbst und auch keine darin enthaltenen Nachrichten.

Eine Übertragungswarteschlange erstellen

Eine Übertragungswarteschlange ist eine lokale Warteschlange, die verwendet wird, wenn ein WS-Manager Nachrichten über einen Nachrichtenkanal an einen fernen WS-Manager weiterleitet. Der Kanal stellt eine einseitige Verbindung mit dem fernen WS-Manager zur Verfügung. Nachrichten werden so lange in der Übertragungswarteschlange aufbewahrt, bis der Kanal sie entgegennehmen kann. Beim Definieren eines Kanals müssen Sie den Namen einer Übertragungswarteschlange für den Sendeausgang des Nachrichtenkanals angeben.

Das Attribut *Usage* (USAGE in MQSC) definiert, ob eine Warteschlange eine Übertragungswarteschlange oder eine normale Warteschlange ist.

Standardübertragungswarteschlangen

Optional können Sie mit Hilfe des Attributs *XmitQName* (XMITQ in MQSC) eine Übertragungswarteschlange in einem Objekt für eine ferne Warteschlange angeben. Wenn keine Übertragungswarteschlange definiert ist, wird eine Standardwarteschlange verwendet. Wenn Anwendungen Nachrichten in eine ferne Warteschlange einreihen und zu diesem Zeitpunkt eine Übertragungswarteschlange mit demselben Namen wie dem der Zielwarteschlange existiert, wird diese Warteschlange verwendet. Wenn diese Warteschlange nicht existiert, wird die Warteschlange verwendet, die durch das Attribut *DefaultXmitQ* (DEFXMITQ in MQSC) im lokalen WS-Manager angegeben wurde.

Der folgende MQSC-Befehl erstellt beispielsweise eine Standardübertragungswarteschlange im WS-Manager `source.queue.manager` für Nachrichten, die an den WS-Manager `target.queue.manager` gerichtet sind:

```
DEFINE QLOCAL ('target.queue.manager') +
DESCR ('Standardübertragungswarteschlange für Ziel-WS-Manager') +
USAGE(XMITQ)
```

Anwendungen können Nachrichten direkt in eine Übertragungswarteschlange einreihen oder indirekt, z. B. über eine ferne Warteschlangendefinition. Siehe auch „Eine lokale Definition für eine ferne Warteschlange erstellen“ auf Seite 133.

Definitionen ferner Warteschlangen als Aliasnamen verwenden

Sie können eine lokale Definition einer fernen Warteschlange nicht nur als Verweis auf eine Warteschlange in einem anderen WS-Manager verwenden, sondern auch als:

- Aliasname eines WS-Managers
- Aliasname einer Warteschlange für zu beantwortende Nachrichten

Beide Aliasnamen werden über die lokale Definition einer fernen Warteschlange aufgelöst.

Wie beim fernen Queuing üblich, müssen die entsprechenden Kanäle konfiguriert werden, wenn die Nachricht ihre Zieladresse erreichen soll.

Aliasname eines WS-Managers

Über einen Aliasnamen kann der in einer Nachricht angegebene Name des Ziel-WS-Managers während der Nachrichtenweiterleitung von einem WS-Manager

geändert werden. Aliasnamen von WS-Managern sind wichtig, weil Sie mit ihrer Hilfe die Zieladresse von Nachrichten in einem Netz von WS-Managern steuern können.

Diese Steuerung erfolgt dadurch, dass Sie die Definition der fernen Warteschlange im WS-Manager am Steuerungspunkt ändern. Die sendende Anwendung weiß nicht, dass es sich bei dem angegebenen WS-Managernamen um einen Aliasnamen handelt.

Weitere Informationen zu Aliasnamen für WS-Manager finden Sie im Handbuch *MQSeries Intercommunication*.

Aliasnamen für Warteschlangen für zu beantwortende Nachrichten

Eine Anwendung kann optional den Namen einer Warteschlange für zu beantwortende Nachrichten angeben, wenn sie eine *Anforderungsnachricht* in eine Warteschlange einreicht. Wenn die Anwendung, von der die Nachricht verarbeitet wird, den Namen der Warteschlange für zu beantwortende Nachrichten extrahiert, weiß sie, wohin die *Antwortnachricht* (falls eine erforderlich ist) gesendet werden muss.

Über den Aliasnamen einer Warteschlange für zu beantwortende Nachrichten kann ein WS-Manager die in einer Anforderungsnachricht angegebene Warteschlange für zu beantwortende Nachrichten während der Nachrichtenübertragung ändern. Die sendende Anwendung weiß nicht, dass es sich bei dem angegebenen Namen der Warteschlange für zu beantwortende Nachrichten um einen Aliasnamen handelt.

Mit Hilfe eines Aliasnamens einer Warteschlange für zu beantwortende Nachrichten können Sie den Namen der Warteschlange für zu beantwortende Nachrichten und optional deren WS-Manager ändern. Auf diese Weise haben Sie die Möglichkeit, den Weg für die Weiterleitung von Antwortnachrichten zu steuern.

Weitere Informationen zu Anforderungsnachrichten, Antwortnachrichten und Warteschlangen für zu beantwortende Nachrichten finden Sie im Handbuch *MQSeries Application Programming Reference*. Weitere Informationen zu Aliasnamen von Warteschlangen für zu beantwortende Nachrichten finden Sie im Handbuch *MQSeries Intercommunication*.

Datenkonvertierung

Nachrichtendaten in MQSeries-definierten Formaten (auch als integrierte Formate bekannt) können vom WS-Manager aus einem codierten Zeichensatz in einen anderen konvertiert werden, wenn beide Zeichensätze sich auf eine einzige Sprache oder auf eine Gruppe verwandter Sprachen beziehen.

Zum Beispiel wird eine Konvertierung zwischen codierten Zeichensätzen mit den Kennungen (CCSIDs) 850 und 500 unterstützt, weil beide auf westeuropäische Sprachen angewendet werden.

Informationen zur Konvertierung von EBCDIC-Zeilenvorschubzeichen nach ASCII finden Sie unter „Inhalt der MQSeries-Konfigurationsdatei“ auf Seite 194.

Die unterstützten Konvertierungen sind in Appendix F. "Code page conversion tables" im Handbuch *MQSeries Application Programming Reference* aufgeführt.

Ein WS-Manager kann Nachrichten nicht in integrierte Formate konvertieren

Der WS-Manager kann Nachrichten nicht automatisch in integrierte Formate konvertieren, wenn deren CCSIDs zu verschiedenen Sprachengruppen gehören. Eine Konvertierung zwischen CCSID 850 und CCSID 1025 (ein EBCDIC-codierter Zeichensatz für Sprachen mit kyrillischen Zeichen) wird beispielsweise nicht unterstützt, weil viele Zeichen aus dem einen codierten Zeichensatz im anderen codierten Zeichensatz nicht dargestellt werden können. Wenn in Ihrem Netz WS-Manager mit verschiedenen Landessprachen aktiv sind, eine Konvertierung zwischen einigen der codierten Zeichensätze aber nicht unterstützt wird, können Sie eine Standardkonvertierung aktivieren. Eine Beschreibung der Standarddatenkonvertierung finden Sie unter „Standarddatenkonvertierung“.

Datei CCSID

Die Datei CCSID enthält zusätzliche Zeichensätze sowie Standarddatenkonvertierungen. Die Daten in dieser Datei können aktualisiert werden; dies kann zum Beispiel erforderlich sein, wenn ein späteres Release Ihres Betriebssystems zusätzliche codierte Zeichensätze unterstützt. Um diese anzugeben, müssen Sie die Datei CCSID entsprechend ändern. Informationen zur Vorgehensweise finden Sie in dieser Datei.

Standarddatenkonvertierung

Die Standarddatenkonvertierung wird implementiert, indem Sie in der Datei CCSID eine Standard-EBCDIC-CCSID und eine Standard-ASCII-CCSID sowie Standard-CCSIDs angeben. Anweisungen dazu finden Sie in der Datei.

Wenn Sie die Datei CCSID bearbeiten, um die Standarddatenkonvertierung zu implementieren, müssen Sie den WS-Manager anschließend erneut starten, damit die Änderungen wirksam werden.

Die Standarddatenkonvertierung läuft folgendermaßen ab:

- Wenn zwischen der Quellen- und der Ziel-CCSID keine Konvertierung unterstützt wird, es sich bei den CCSIDs der Quellen- und Zielumgebung aber um zwei EBCDIC- bzw. zwei ASCII-Kennungen handelt, werden die Zeichendaten ohne Konvertierung an die Zielanwendung übergeben.
- Handelt es sich bei einer CCSID um eine ASCII-CCSID, bei der anderen um eine EBCDIC-CCSID, konvertiert MQSeries die Daten entsprechend den in der Datei CCSID angegebenen CCSIDs.

Anmerkung: Es wird empfohlen, die Konvertierung nur für die Zeichen zuzulassen, für die im codierten Zeichensatz der Nachricht und im codierten Zeichensatz für die Standardkonvertierung dieselben Codewerte stehen. Wenn Sie nur den Zeichensatz verwenden, der für MQSeries-Objektnamen gültig ist, ist diese Anforderung in der Regel erfüllt. Ausnahmen gelten für die in Japan verwendeten EBCDIC-CCSIDs 290, 930, 1279 und 5026, weil sich die Codes für Kleinbuchstaben in diesen Zeichensätzen von denen in anderen EBCDIC-CCSIDs unterscheiden.

Konvertierung von Nachrichten in benutzerdefinierte Formate

Nachrichten in benutzerdefinierten Formaten können vom WS-Manager nicht von einem codierten Zeichensatz in einen anderen konvertiert werden. Wenn Daten in einem benutzerdefinierten Format konvertiert werden müssen, müssen Sie für jedes benutzerdefinierte Format einen Datenkonvertierungs-Exit bereitstellen. Es wird nicht empfohlen, Standard-CCSIDs für die Konvertierung von Daten in benutzerdefinierten Formaten zu verwenden, obwohl dies möglich ist. Weitere Informationen zum Konvertieren von Daten in benutzerdefinierten Formaten und zum Erstellen von Datenkonvertierungs-Exits finden Sie im Handbuch *MQSeries Application Programming Guide*.

CCSID des WS-Managers ändern

Es wird empfohlen, den WS-Manager nach einer Änderung der CCSID des WS-Managers mit Hilfe des Attributs CCSID im Befehl ALTER QMGR zu stoppen und erneut zu starten.

Dadurch wird sichergestellt, dass alle aktiven Anwendungen, einschließlich Befehlsserver und Kanalprogramme, gestoppt und erneut gestartet werden.

Dies ist notwendig, weil alle Anwendungen, die während der Änderung der CCSID des WS-Managers aktiv sind, sonst weiter die vorhandene CCSID verwenden.

Kapitel 8. MQSeries-Objekte schützen

In diesem Kapitel werden die Funktionen der Zugriffssteuerung in MQSeries for Compaq NSK erläutert, und wie Sie diese Sicherheitsfunktion implementieren können.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Konzept der Benutzer-IDs in der Benutzergruppe MQM“
- „Gründe für den Schutz von MQSeries-Ressourcen“
- „Objektberechtigungsmanager (OAM, Object Authority Manager)“ auf Seite 142
- „OAM-Befehle verwenden“ auf Seite 146
- „Zugriffsberechtigungen“ auf Seite 149
- „Befehl zum Anzeigen von Berechtigungen“ auf Seite 149
- „Hinweise zur Verwendung des OAM“ auf Seite 150
- „Erläuterungen zu den Tabellen mit den Berechtigungsspezifikationen“ auf Seite 153
- „Erläuterung der Berechtigungsdateien“ auf Seite 159

Gründe für den Schutz von MQSeries-Ressourcen

Da WS-Manager von MQSeries mit der Übertragung von Daten befasst sind, die möglicherweise einen hohen Wert darstellen, benötigen Sie den Schutz durch ein Berechtigungssystem. Dadurch wird sichergestellt, dass die einem WS-Manager zugeordneten und von diesem verwalteten Ressourcen vor unbefugten Zugriffen geschützt sind, die den Verlust oder die unerwünschte Veröffentlichung von Daten zur Folge haben können. Für ein sicheres System ist es entscheidend, dass die folgenden Ressourcen vor einem Zugriff oder einer Änderung durch unbefugte Benutzer oder Anwendungen geschützt sind:

- Verbindungen mit einem WS-Manager
- MQSeries-Objekte wie Warteschlangen, Cluster, Kanäle und Prozesse
- Befehle für die Verwaltung von WS-Managern, einschließlich MQSC- und PCF-Befehle
- MQSeries-Nachrichten
- Kontextinformationen von Nachrichten

Erstellen Sie eine eigene Sicherheitsrichtlinie, in der Sie festlegen, welche Benutzer auf welche Ressourcen zugreifen dürfen.

Konzept der Benutzer-IDs in der Benutzergruppe MQM

Alle WS-Managerressourcen werden unter der Gruppen-ID MQM ausgeführt.

Damit Sie Zugriff auf MQSeries for Compaq NSK erhalten, muss Ihre Benutzer-ID einem MQSeries-Principal entsprechen. Zunächst wird nur für die Benutzer-ID der MQSeries-Principal **mqm** erstellt, unter der der WS-Manager erstellt wurde. Für jeden weiteren Benutzer, der auf MQSeries zugreift, müssen Sie mit dem Befehl **altmqsr** einen Principal erstellen. Principal und Benutzer müssen eindeutig sein. Mit dem Befehl **dspmqs** können Sie die Principals eines WS-Managers und ihre Merkmale anzeigen.

Konzept der Benutzer-IDs

Wenn Ihre Benutzer-ID zur MQSeries for Compaq NSK-Gruppe **MQM** gehört und für diese ID ein MQSeries-Principal erstellt wurde, verfügen Sie über die Zugriffsberechtigung für alle MQSeries-Ressourcen. Ihre Benutzer-ID *muss* zur Gruppe **MQM** gehören, damit Sie alle MQSeries for Compaq NSK-Steuerbefehle (mit Ausnahme des Befehls **crtmqcvx**) verwenden können. Vor allem benötigen Sie diese Berechtigung für:

- den Befehl **runmqsc** zum Ausführen von MQSC-Befehlen,
- die Verwaltung von Berechtigungen in MQSeries for Compaq NSK mit Hilfe des Befehls **setmqaut**.

Wenn Sie Kanalbefehle an WS-Manager auf einem fernen Compaq NSK-System senden, muss Ihre Benutzer-ID auf dem Zielsystem auf jeden Fall zur Compaq NSK-Gruppe **MQM** gehören. Eine Liste der PCF- und MQSC-Kanalbefehle finden Sie unter „Sicherheit für Kanalbefehle“ auf Seite 152.

Für folgende Vorgänge ist es nicht erforderlich, dass Ihre Benutzer-ID zur Gruppe **MQM** gehört:

- Ausgeben von PCF-Befehlen, einschließlich PCF-Escape-Befehlen, aus einem Verwaltungsprogramm
- Ausgeben von MQI-Aufrufen aus einem Anwendungsprogramm

Anmerkung: Die Berechtigungen für den Principal **mqm** (und die MQM-Gruppen) sind wichtig, da dieser Principal von den internen WS-Managerkomponenten selbst für den Zugriff auf geschützte Ressourcen verwendet wird. Wenn Sie für die Objekte im WS-Manager den Principal **mqm** bzw. die Gruppe **MQM** oder gar den Principal **mqm** selbst entfernen, kann der WS-Manager unter Umständen nicht mehr verwaltet oder überhaupt nicht mehr verwendet werden.

Weitere Informationen

Weitere Informationen zu:

- MQSeries for Compaq NSK-Befehlssätzen siehe „Kapitel 3. MQSeries-Befehlssätze verwenden“ auf Seite 31
- MQSeries for Compaq NSK-Steuerbefehlen siehe „Kapitel 17. MQSeries-Steuerbefehle“ auf Seite 257
- PCF-Befehlen und PCF-Escape-Befehlen siehe das Handbuch *MQSeries Programmable System Management*
- MQI-Aufrufen siehe die Handbücher *MQSeries Application Programming Guide* und *MQSeries Application Programming Reference*

Objektberechtigungsmanager (OAM, Object Authority Manager)

Der Zugriff auf WS-Managerressourcen wird standardmäßig durch eine installierbare Komponente für die Berechtigungsverwaltung gesteuert. Diese Komponente, der Objektberechtigungsmanager (OAM, Object Authority Manager) gehört zum Lieferumfang von MQSeries for Compaq NSK und wird, falls nicht anders angegeben, bei der Erstellung eines WS-Managers automatisch erstellt und aktiviert. In diesem Kapitel wird für den im Produkt enthaltenen Objektberechtigungsmanager die Abkürzung OAM (Object Authority Manager) verwendet.

Der OAM ist eine *installierbare Komponente* des Berechtigungsservices. Dies ermöglicht einen flexiblen Einsatz:

- Sie können den standardmäßigen OAM über die zu diesem Zweck bereitgestellte Schnittstelle durch eine eigene Berechtigungsservicekomponente ersetzen.

Objektberechtigungsmanager (OAM)

- Sie können die Funktionalität des standardmäßigen OAM um Funktionen aus Ihrer eigenen Berechtigungsservicekomponente erweitern (ebenfalls über die bereitgestellte Schnittstelle).
- Sie können den OAM entfernen oder löschen und ohne Berechtigungsservice arbeiten.

Weitere Informationen zu installierbaren Services finden Sie im Handbuch *MQSeries Programmable System Management*.

Der OAM verwaltet Benutzerberechtigungen zum Bearbeiten von MQSeries-Objekten einschließlich Warteschlangen, Prozessdefinitionen und Kanälen. Er stellt außerdem eine Befehlsschnittstelle zur Verfügung, über die Sie einer bestimmten Gruppe oder bestimmten Benutzern Zugriffsberechtigung für ein Objekt erteilen oder sie widerrufen können. Die Entscheidung, ob Zugriff auf eine Ressource ermöglicht wird, trifft der OAM. Der WS-Manager richtet sich nach dieser Entscheidung. Kann der OAM keine Entscheidung treffen, verhindert der WS-Manager den Zugriff auf diese Ressource.

Funktionsweise des OAM

Der Objektberechtigungsmanager verwendet die Benutzer- und Gruppen-IDs sowie die Sicherheitsfunktionen des Betriebssystems Compaq NSK. Benutzer können nur auf WS-Managerobjekte zugreifen, wenn sie über die erforderliche Berechtigung verfügen.

Zugriffsverwaltung über Benutzergruppen

Die Verwaltung der Zugriffsberechtigungen auf MQSeries-Ressourcen erfolgt über Compaq NSK-Gruppen. Der OAM verwaltet die Berechtigungen auf Gruppenebene.

In der Befehlsschnittstelle werden MQSeries-Principals anstelle von Benutzer-IDs verwendet. Dies liegt daran, dass die Berechtigungen, die einer Benutzer-ID erteilt sind, auch anderen Definitionseinheiten erteilt werden können. So können beispielsweise einem Anwendungsprogramm, das MQI-Aufrufe ausgibt oder einem Anwendungsprogramm, das PCF-Befehle absetzt, Berechtigungen erteilt werden. In diesen Fällen handelt es sich bei dem Principal, der dem Programm zugeordnet ist, nicht notwendigerweise um die Benutzer-ID, unter der das Programm gestartet wurde.

Compaq NSK-Benutzer-IDs können das Format <Gruppe>.<Name> haben; dabei können 'Gruppe' und 'Name' jeweils bis zu 8 Zeichen umfassen; die Namen von MQSeries-Principals können dagegen bis zu 12 Zeichen umfassen. Darüber hinaus ist die Verwendung von Punkten (.) in Benutzer-IDs auf einigen Plattformen nicht zulässig. In MQSeries for Compaq NSK enthält die Principal-Datenbank die Zuordnung von Compaq NSK-Benutzer-IDs zu MQS-Principals, deren Namen bis zu 12 Zeichen umfassen.

Benutzer, die zu mehreren Benutzergruppen gehören

Die Berechtigungen eines Benutzers umfassen die Berechtigungen aller Gruppen, zu denen der Benutzer gehört, und die Standardberechtigungen für alle Benutzer. Mit dem Befehl **setmqaut** können Sie die Berechtigungen für eine bestimmte Gruppe festlegen.

Anmerkung: Alle Änderungen über den Befehl **setmqaut** werden sofort wirksam, es sei denn, das Objekt wird gerade verwendet. In diesem Fall werden die Änderungen wirksam, wenn das Objekt das nächste Mal geöffnet wird.

Objektberechtigungsmanager (OAM)

Gruppen und Primärgruppe

Die Verwaltung von Zugriffsberechtigungen auf MQSeries-Ressourcen erfolgt über Compaq NSK-Benutzergruppen. Ist SAFEGUARD aktiv, kann eine Compaq NSK-Benutzer-ID mehreren Gruppen zugeordnet werden, d. h. der entsprechende MQSeries-Principal wird ebenfalls diesen Gruppen zugeordnet. Bei der Primärgruppe handelt es sich immer um die Compaq-Verwaltungsgruppe. Sekundärgruppen werden über die Erstellung von SAFEGUARD-Gruppen mit gemeinsamem Dateizugriff konfiguriert, denen anschließend eine Compaq NSK-Benutzer-ID zugeordnet wird.

Der OAM verwaltet die Berechtigungen auf Gruppenebene und nicht anhand einzelner Principals. Die Zuordnung von Principals zu Gruppennamen erfolgt innerhalb des OAM unter Verwendung der Principal-Datenbank sowie der Compaq NSK- und SAFEGUARD-Funktionen; die OAM-Vorgänge werden auf Gruppenebene ausgeführt. Sie können jedoch die Berechtigungen eines einzelnen Principals anzeigen.

Ressourcen über den OAM schützen

Über den OAM können Sie Folgendes steuern:

- Zugriff auf MQSeries-Objekte über MQI. Bei dem Versuch eines Anwendungsprogramms, auf ein Objekt zuzugreifen, überprüft der OAM zunächst, ob die Benutzer-ID, unter der der Zugriff erfolgt, auf Grund ihrer Gruppenzugehörigkeit über die Berechtigung für diese Operation verfügt.
Dies bedeutet vor allem, dass Warteschlangen und die Nachrichten in Warteschlangen vor unbefugtem Zugriff geschützt werden können.
- Berechtigung zum Ausführen von MQSC-Befehlen. Nur Benutzer, die zur Benutzergruppe **mqm** gehören oder über **setmqaut** entsprechend berechtigt wurden, können Verwaltungsbefehle für WS-Manager ausführen, beispielsweise um eine Warteschlange zu erstellen.
- Berechtigung zur Verwendung von Steuerbefehlen. Nur Benutzer, die zur Gruppe **mqm** gehören, können Steuerbefehle ausführen, beispielsweise um einen WS-Manager zu erstellen oder einen Befehlsserver zu starten.
- Berechtigung zum Ausführen von PCF-Befehlen

Verschiedenen Benutzergruppen können unterschiedliche Zugriffsberechtigungen für ein und dasselbe Objekt erteilt werden. Beispielsweise kann eine Gruppe für eine bestimmte Warteschlange über PUT- und GET-Berechtigungen verfügen, während eine andere Gruppe nur den Inhalt dieser Warteschlange anzeigen kann (MQGET-Aufruf mit Option **BROWSE**). Ebenso können Gruppen für eine Warteschlange die GET- und PUT-Berechtigung erhalten, jedoch nicht die Berechtigung, diese Warteschlange zu ändern oder zu löschen.

Berechtigungsverteilung über Gruppen

Durch die Berechtigungsverteilung für Gruppen anstatt für einzelne Principals wird der erforderliche Verwaltungsaufwand verringert. In der Regel wird eine einzelne Zugriffsart von mehreren Principals angefordert. Beispielsweise können Sie eine Gruppe mit Endbenutzern definieren, die eine bestimmte Anwendung ausführen möchten. Neuen Benutzern kann dieser Zugriff ebenfalls erteilt werden, indem diese Gruppe ihrer Compaq NSK-Benutzer-ID hinzugefügt wird. Sofern MQSeries nicht auf einem System mit SAFEGUARD installiert ist, das die Erstellung von Gruppen für die gemeinsame Benutzung von Daten ermöglicht, kann jede Benutzer-ID nur jeweils einer Primärgruppe zugeordnet werden.

Objektberechtigungsmanager (OAM)

Ohne SAFEGUARD kann eine Gruppe aus höchstens 255 Principals bestehen. Durch die Verwendung des gemeinsamen Dateizugriffs mit SAFEGUARD sind bis zu 65535 Principals pro Gruppe möglich beschränkt. Dieses Konzept ermöglicht es außerdem, jedem Principal eigene Berechtigungen zu erteilen. Verfügt beispielsweise die SAFEGUARD-Verwaltungsgruppe über keine Berechtigungen, können jedem Principal, der der Gruppe mit gemeinsamem Dateizugriff gehört, eigene Berechtigungen erteilt werden (Principal A, Mitglied TESTSEC1 der Gruppe MQSEC verfügt nur über die PUT-Berechtigung, Principal B, Mitglied TESTSEC2 der Gruppe MQSEC verfügt nur über die GET-Berechtigung).

Ohne SAFEGUARD müssten in diesem Fall zwei Gruppen erstellt werden, d. h. zwei Compaq NSK-Benutzer-IDs in verschiedenen Gruppen (z. B. MQ.TEST mit den Berechtigungen **connect/put** und MQ1.TEST mit den Berechtigungen **connect/get**). Dies ist erforderlich, da Benutzer alle Berechtigungen der Gruppen erhalten, zu denen sie gehören (z. B. Compaq NSK-Benutzer-ID MQM.MANAGER, Principal mqm, Gruppe MQM (Berechtigung **connect**) und Compaq NSK-Benutzer-ID MQM.APPS, Principal apps, Gruppe MQM (Berechtigung **get/put**)). Die für die Gruppe eingerichtete Berechtigung (**connect** für mqm) und die für den Principal definierten Berechtigungen (**get/put** für apps) werden zusammengefasst (**connect/get/put**). Benutzer werden auf diese Berechtigungen hin überprüft. Versucht eine unter der Benutzer-ID MQ.TEST angemeldete Anwendung einen MQGET-Aufruf, wird dieser mit MQRC_NOT_AUTHORIZED (2035) zurückgewiesen (diese Benutzer-ID verfügt nicht über die Berechtigung **get**). Für eine Anwendung, die unter der Benutzer-ID MQ1.TEST ausgeführt wird, wäre der MQGET-Aufruf erfolgreich, ebenso für alle Compaq-Benutzer-IDs, die zur Gruppe MQM gehören. Wird einem bestimmten Principal die Berechtigung erteilt, erhalten alle Principals in der Gruppe dieselbe Berechtigung.

Principal	Compaq-ID	Gruppe	Gruppentyp	Berechtigung
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

Abbildung 39. Verwendung von Gruppen mit SAFEGUARD zur Erteilung von Berechtigungen

In Abb. 39 verfügt jeder angemeldete Benutzer, der zur Gruppe MQM gehört, über die Berechtigungen **connect**, **get**, **put** und **inq**. Der Benutzer MQ.TEST verfügt über die Berechtigungen **connect** und **put**, der Benutzer MQ1.TEST über die Berechtigungen **connect** und **get**. Benutzer, die zur Gruppe MQSEC gehören, verfügen über die Berechtigungen, die der SAFEGUARD-Gruppe mit gemeinsamem Dateizugriff erteilt wurden.

Sie sollten die Anzahl der Gruppen so gering wie möglich halten. Beispielsweise können Sie die Benutzer in eine Gruppe für Anwendungsbenutzer und eine Gruppe für Administratoren aufteilen.

Objektberechtigungsmanager (OAM)

OAM inaktivieren

Der OAM ist standardmäßig aktiviert. Sie können ihn inaktivieren, indem Sie vor der Erstellung des WS-Managers die Compaq NSK-Umgebungsvariable MQSNOAUT wie folgt setzen:

```
PARAM MQSNOAUT 1
```

Wenn Sie den OAM für einen WS-Manager allerdings inaktivieren, kann der OAM später nicht mehr erneut gestartet werden. Daher sollte der OAM nicht inaktiviert werden, und Sie sollten sicherstellen, dass alle Benutzer und Anwendungen Zugriff über eine entsprechende Benutzer-ID haben. Sie haben die Möglichkeit, den OAM für Testzwecke zu inaktivieren, indem Sie die Zeilengruppe für den Berechtigungsservice aus der Konfigurationsdatei des WS-Managers (QMINI) entfernen, oder indem Sie in dieser Datei MQAUTH in der Zeilengruppe Authority auf 'off' setzen (siehe „Konfigurationsdatei der WS-Manager (QMINI)“ auf Seite 195).

Anmerkung: Über die Angabe PARAM MQSNOAUT 0 kann der OAM nicht inaktiviert werden. Soll der OAM wieder erneut aktiviert werden, darf diese Umgebungsvariable in der Umgebung nicht vorhanden sein.

OAM-Befehle verwenden

Der OAM stellt eine Befehlschnittstelle zum Erteilen und Widerrufen von Berechtigungen zur Verfügung. Damit Sie diese Befehle verwenden können, müssen Sie über die entsprechende Berechtigung verfügen, d. h., Ihre Benutzer-ID muss zur Compaq NSK-Gruppe MQM gehören. Diese Gruppe muss vor der Installation von MQSeries for Compaq NSK eingerichtet werden.

Gehört Ihre Benutzer-ID zur Gruppe MQM, verfügen Sie für den WS-Manager über die Berechtigung eines 'Superusers'. Damit haben Sie die Berechtigung, unter Ihrer Benutzer-ID MQI-Anforderungen und Steuerbefehle auszugeben.

Der OAM stellt vier Befehle zur Verfügung, die Sie über die TAACL-Eingabeaufforderung zur Verwaltung von Benutzerberechtigungen eingeben können. Dabei handelt es sich um folgende Befehle:

- **altmqusr** (MQSeries-Principal erstellen, entfernen oder ändern)
- **dspmqusr** (Principal anzeigen)
- **setmqaut** (Berechtigung setzen oder zurücksetzen)
- **dspmqaut** (Berechtigung anzeigen)

Bei den folgenden Aufrufen wird die Berechtigung überprüft: MQCONN, MQOPEN, MQPUT1 und MQCLOSE. Eine Berechtigungsprüfung wird nur ausgeführt, wenn einer dieser Aufrufe zum ersten Mal erfolgt; Änderungen an der Berechtigung werden erst wirksam, wenn Sie das Objekt zurücksetzen (d. h. schließen oder öffnen). Deshalb werden Änderungen an den Berechtigungen für ein Objekt über den Befehl **setmqaut** erst wirksam, nachdem Sie ein Objekt zurückgesetzt haben.

Angaben in OAM-Befehlen

OAM-Befehle gelten für den angegebenen WS-Manager; wird kein WS-Manager angegeben, wird der Standard-WS-Manager verwendet. Bei diesen Befehlen

müssen Sie das Objekt eindeutig angeben, d. h., Sie müssen den Objektnamen und die Objektart angeben. Darüber hinaus müssen Sie den Benutzer oder die Gruppe angeben, für die die Berechtigung gilt.

Berechtigungslisten

Sie geben im Befehl **setmqaut** eine Liste mit Berechtigungen an. Dies ist eine einfache Methode, Ressourcen Berechtigungen zu erteilen bzw. Berechtigungen für Ressourcen zu widerrufen. Die Berechtigungen in dieser Liste werden als Schlüsselwörter in Kleinbuchstaben aufgeführt, denen je nachdem ein Pluszeichen (+) oder Minuszeichen (-) vorangestellt ist. Mit dem Pluszeichen wird die angegebene Berechtigung erteilt, mit dem Minuszeichen dagegen wird die angegebene Berechtigung entfernt. Sie können in einem einzigen Befehl beliebig viele Berechtigungen angeben. Beispiel:

```
+browse -get +put
```

Der Befehl 'altmqsr'

Sofern Sie über die entsprechende Berechtigung verfügen, können Sie mit dem Befehl **altmqsr** einen MQSeries-Principal erstellen und diesen einer Compaq NSK-Benutzer-ID oder einem SAFEGUARD-Aliasnamen zuordnen. Das folgende Beispiel veranschaulicht die Verwendung des Befehls **altmqsr**:

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

In diesem Beispiel gilt Folgendes:

Angabe	Bedeutung
saturn.queue.manager	Name des WS-Managers
MQPRINCIPAL	Name des Principals, der erstellt werden soll
MQM.MQUSER	Name der Compaq NSK-Benutzer-ID

Eine Beschreibung dieses Befehls finden Sie unter „altmqsr (MQSeries-Benutzerinformationen ändern)“ auf Seite 265.

Mit dem Befehl **altmqsr** kann ein Principal auch entfernt werden und damit auch dessen Zugriffsrechte auf MQSeries. Beispiel:

```
altmqsr -m saturn.queue.manager -p MQPRINCIPAL -remove
```

In diesem Beispiel gilt Folgendes:

Angabe	Bedeutung
saturn.queue.manager	Name des WS-Managers
MQPRINCIPAL	Name des Principals, der entfernt werden soll
-remove	Anweisung, dass der Principal entfernt werden soll

OAM-Befehle verwenden

Der Befehl 'dspmqusr'

Mit dem Befehl **dspmqusr** können Sie zusätzlich zu den Compaq NSK-Gruppen für die Verwaltung und den gemeinsamen Dateizugriff, zu denen die einzelnen MQSeries-Principals entsprechenden Benutzer-IDs gehören, auch den Inhalt der Principal-Datenbank anzeigen. Über den Parameter **-p** können diese Informationen auf den angegebenen Principal beschränkt werden. Beispiel:

```
dspmqusr -m saturn.queue.manager -p MQPRINCIPAL
```

In diesem Beispiel gilt Folgendes:

Angabe	Bedeutung
saturn.queue.manager	Name des WS-Managers
MQPRINCIPAL	Name des Principals, der angezeigt werden soll

Eine Beschreibung dieses Befehls finden Sie unter „dspmqusr (MQSeries-Benutzerinformationen anzeigen)“ auf Seite 288.

Der Befehl 'setmqaut'

Sofern Sie über die entsprechende Berechtigung verfügen, können Sie mit dem Befehl **setmqaut** einem Principal bzw. einer Benutzergruppe die Zugriffsberechtigung für ein bestimmtes Objekt erteilen bzw. diese Berechtigung widerrufen. Das folgende Beispiel zeigt die Verwendung des Befehls **setmqaut**:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE -g GroupA +browse -get +put
```

In diesem Beispiel gilt Folgendes:

Angabe	Bedeutung
saturn.queue.manager	Name des WS-Managers
Warteschlange	Objektart
RED.LOCAL.QUEUE	Name des Objekts
GroupA	Die ID der Gruppe, der die Berechtigungen erteilt werden
+browse -get +put	Gibt die Berechtigungsliste für die angegebene Warteschlange an. Zwischen Plus- bzw. Minuszeichen und Schlüsselwort darf kein Leerzeichen stehen.

Die Berechtigungsliste enthält die Berechtigungen, die hinzugefügt oder entfernt werden, wobei Folgendes gilt:

Angabe	Bedeutung
+browse	Fügt die Berechtigung zum Anzeigen von Nachrichten hinzu (MQGET-Aufruf mit der Option browse).
-get	Entfernt die Berechtigung zum Abrufen (MQGET) von Nachrichten aus der Warteschlange.
+put	Fügt die Berechtigung zum Einreihen (MQPUT) von Nachrichten in die Warteschlange hinzu.

Anwendungen, die unter einer Benutzer-ID, die zur Compaq NSK-Benutzergruppe GroupA gehört, gestartet wurden, verfügen über diese Berechtigungen.

Mit dem folgenden Befehl wird die Berechtigung zum Einreihen von Nachrichten in die Warteschlange MyQueue für die Gruppen GroupA und GroupB widerrufen.

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -g GroupA -g GroupB -put
```

Eine formale Definition des Befehls und seiner Syntax finden Sie unter „setmqaut (Berechtigung setzen/zurücksetzen)“ auf Seite 306.

Berechtigungsbeefehle und installierbare Services

Im Befehl **setmqaut** ist ein weiterer Parameter möglich, der den Namen der Berechtigungsservicekomponente angibt, für die Aktualisierung gilt. Dieser Parameter muss angegeben werden, wenn mehrere Berechtigungskomponenten gleichzeitig aktiv sind. Dies ist normalerweise nicht der Fall. Erfolgt in diesem Fall keine Angabe, wird die Aktualisierung für die erste Berechtigungskomponente (falls vorhanden), die gefunden wird, ausgeführt. Dies ist standardmäßig der mitgelieferte OAM.

Zugriffsberechtigungen

Berechtigungen, die in der Berechtigungsliste des Befehls **setmqaut** definiert werden, können in folgende Kategorien eingeteilt werden:

- Berechtigungen für MQI-Aufrufe
- Berechtigungen für Verwaltungsbefehle
- Kontextberechtigungen
- allgemeine Berechtigungen, d. h. für MQI-Aufrufe, Befehle oder beides

Jeder Berechtigung entspricht ein Schlüsselwort, das in den Befehlen **setmqaut** und **dspmqa** angegeben wird. Diese Schlüsselwörter werden unter „setmqaut (Berechtigung setzen/zurücksetzen)“ auf Seite 306 beschrieben.

Befehl zum Anzeigen von Berechtigungen

Mit dem Befehl **dspmqa** können Sie die Berechtigungen anzeigen, über die ein bestimmter Principal oder eine bestimmte Gruppe für ein bestimmtes Objekt verfügt. Die Optionen haben dieselbe Bedeutung wie im Befehl **setmqaut**. Es können immer nur die Berechtigungen für eine Gruppe bzw. einen Principal angezeigt werden. Eine formale Spezifikation dieses Befehls finden Sie unter „dspmqa (Berechtigung anzeigen)“ auf Seite 279.

Befehl zum Anzeigen von Berechtigungen

Der folgende Befehl zeigt zum Beispiel die Berechtigungen an, über die die Gruppe GpAdmin für eine Prozessdefinition mit dem Namen Rentenansprüche im WS-Manager WS-Man1 verfügt.

```
dspmqaout -m QueueMan1 -t process -n Rentenansprüche -g GpAdmin
```

Die Schlüsselwörter, die als Ergebnis dieses Befehls angezeigt werden, geben die aktiven Berechtigungen an.

Hinweise zur Verwendung des OAM

Einige Operationen sind besonders kritisch und sollten deshalb auf privilegierte Benutzer beschränkt sein. Beispiel:

- Erstellen, Löschen, Starten und Stoppen von WS-Managern
- Zugriff auf bestimmte Warteschlangen, wie beispielsweise Übertragungswarteschlangen oder die Befehlswarteschlange SYSTEM.ADMIN.COMMAND.QUEUE
- Programme, die vollständige MQI-Kontextoptionen verwenden
- Erstellen und Kopieren von Anwendungswarteschlangen

Benutzer-IDs

Zur Sondergruppe MQM, die erstellt wird, sollten nur Produktadministratoren gehören. Ihr sollten nur privilegierte Benutzer zugeordnet werden.

Datenträger des WS-Managers

Der Datenträger mit den Warteschlangen und anderen WS-Managerdaten ist ein produktspezifischer Datenträger. Objekte in diesem Verzeichnis verfügen über Compaq NSK-Benutzerberechtigungen, die in Beziehung zu ihren OAM-Berechtigungen stehen. Aus den folgenden Gründen können keine Compaq NSK-Standardbefehle verwendet werden, um Berechtigungen für MQI-Ressourcen zu erteilen oder zu widerrufen:

- MQSeries-Objektnamen stimmen nicht notwendigerweise mit den entsprechenden Systemobjektnamen überein. Weitere Informationen hierzu finden Sie unter „Datenträgerstruktur“ auf Seite 64.
- Die Sicherheitseinstellungen von MQSeries-Objekten entsprechen nicht notwendigerweise den Sicherheitseinstellungen des Objekts unter Compaq NSK.

Warteschlangen

Die Zugriffsberechtigung für eine dynamische Warteschlange basiert auf der Zugriffsberechtigung für die Modellwarteschlange, auf deren Basis sie erstellt wurde, ist aber nicht notwendigerweise dieselbe.

Für Aliaswarteschlangen und ferne Warteschlangen gilt die Berechtigung des Objekts selbst, und nicht die Berechtigung der Warteschlange, in deren Name der Aliasname oder der Name der fernen Warteschlange aufgelöst wird. Daher ist es möglich, einem Principal Zugriff auf eine Aliaswarteschlange zu erteilen, die in eine lokale Warteschlange aufgelöst wird, auf die der Principal eigentlich keinen Zugriff hat.

Begrenzen Sie daher die Berechtigung zum Erstellen von Warteschlangen auf privilegierte Benutzer. Andernfalls können Benutzer die normale Zugriffssteuerung umgehen, indem sie einfach Aliaswarteschlangen erstellen.

Alternative Benutzerberechtigung

Über die alternative Benutzerberechtigung wird gesteuert, ob eine Benutzer-ID beim Zugriff auf ein MQSeries-Objekt die Berechtigung einer anderen Benutzer-ID verwenden kann. Dies ist wichtig, wenn ein Server Anforderungen von einem Programm empfängt und sicherstellen muss, dass das Programm für diese Anforderung berechtigt ist. Der Server selbst verfügt möglicherweise über die erforderliche Berechtigung, muss jedoch wissen, ob das Programm zur Ausführung der angeforderten Aktionen berechtigt ist.

Beispiel:

- Ein Serverprogramm, das unter der Benutzer-ID PAYSERV ausgeführt wird, ruft aus einer Warteschlange eine Anforderungsnachricht ab, die von der Benutzer-ID USER1 in diese Warteschlange eingereicht wurde.
- Nachdem das Serverprogramm die Anforderungsnachricht abgerufen hat, verarbeitet es die Anforderung und reiht die Antwort in die Warteschlange für zu beantwortende Nachrichten ein, die mit der Anforderungsnachricht angegeben wurde.
- Statt die eigene Benutzer-ID (PAYSERV) zu verwenden, um seine Berechtigung zum Öffnen der Warteschlange für zu beantwortende Nachrichten nachzuweisen, kann der Server eine andere Benutzer-ID angeben, in diesem Fall USER1. In diesem Beispiel können Sie mit Hilfe der alternativen Benutzerberechtigung steuern, ob PAYSERV zum Öffnen der Warteschlange für zu beantwortende Nachrichten die Benutzer-ID USER1 als alternative Benutzer-ID angeben darf.

Die alternative Benutzer-ID wird im Feld *AlternateUserId* des Objektdeskriptors angegeben.

Damit ihnen Berechtigungen erteilt werden, müssen für die Benutzer-ID und die alternative Benutzer-ID Principals angegeben werden, die Einträgen in der Principal-Datenbank entsprechen, denen eine Compaq NSK-Benutzer-ID zugeordnet ist.

Anmerkung:

Sie können alternative Benutzer-IDs für alle MQSeries-Objekte verwenden. Alternative Benutzer-IDs haben keine Auswirkung auf die Benutzer-ID, die von einem anderen Ressourcenmanager verwendet wird.

Kontextberechtigung

Als Kontext werden die Informationen bezeichnet, die einer einzelnen Nachricht zugeordnet sind und im Nachrichtendeskriptor (MQMD) als einem Teil der Nachricht enthalten sind. Die Kontextinformationen bestehen aus zwei Abschnitten:

Identität

Dieser Abschnitt gibt an, von wem die Nachricht gesendet wurde. Er besteht aus folgenden Feldern:

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

Ursprung

Dieser Abschnitt gibt an, woher die Nachricht kommt und wann sie in die Warteschlange eingereicht wurde. Er besteht aus folgenden Feldern:

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

OAM-Richtlinien

Anwendungen können die Kontextdaten entweder bei einem MQOPEN- oder einem MQPUT-Aufruf angeben. Diese Daten können von der Anwendung generiert, von einer anderen Nachricht übernommen oder standardmäßig vom WS-Manager generiert werden. Kontextdaten können zum Beispiel von Serverprogrammen verwendet werden, um die Identität des Requesters zu überprüfen, indem sie testen, ob die Nachricht von einer Anwendung kommt, die unter der ID eines berechtigten Benutzers ausgeführt wird.

Ein Serverprogramm kann über das Feld *UserIdentifier* die Benutzer-ID eines alternativen Benutzers feststellen. In diesem Feld muss ein Principal angegeben werden, dem ein Eintrag in der Principal-Datenbank entspricht.

Mit der Kontextberechtigung steuern Sie, ob der Benutzer eine der Kontextoptionen bei einem MQOPEN- oder MQPUT1-Aufruf angeben kann. Informationen zu Kontextoptionen finden Sie im Handbuch *MQSeries Application Programming Guide*. Eine Beschreibung der Felder des Nachrichtendeskriptors, die im Zusammenhang mit Kontextinformationen stehen, finden Sie im Handbuch *MQSeries Application Programming Reference*.

Sicherheitsüberlegungen bei fernen WS-Managern

In Bezug auf die Sicherheit bei fernen WS-Managern sollten Sie Folgendes beachten:

PUT-Berechtigung

Um Sicherheit zwischen WS-Managern herzustellen, können Sie die PUT-Berechtigung angeben, die verwendet wird, wenn ein Kanal eine Nachricht von einem anderen WS-Manager empfängt.

Geben Sie das Kanalattribut PUTAUT wie folgt an:

DEF Standard-Benutzer-ID. Unter dieser Benutzer-ID ist der Nachrichtenkanalagent aktiv.

CTX Die Benutzer-ID im Nachrichtenkontext.

In beiden Fällen muss es sich bei der Benutzer-ID um einen Principal handeln, dem ein Eintrag in der Principal-Datenbank entspricht.

Übertragungswarteschlangen

WS-Manager reihen ferne Nachrichten automatisch in eine Übertragungswarteschlange ein; hierfür ist keine spezielle Berechtigung erforderlich. Um eine Nachricht direkt in eine Übertragungswarteschlange einzureihen, wird jedoch eine spezielle Berechtigung benötigt (siehe Tabelle 3 auf Seite 154).

Kanal-Exits

Kanal-Exits können für eine zusätzliche Sicherheit verwendet werden.

Weitere Informationen finden Sie im Handbuch *MQSeries Intercommunication*.

Sicherheit für Kanalbefehle

Kanalbefehle können als PCF-Befehle über die Schnittstelle MQAI, über MQSC-Befehle und über Steuerbefehle ausgegeben werden.

PCF-Befehle

Sie können PCF-Kanalbefehle ausgeben, indem Sie eine PCF-Nachricht an die Warteschlange SYSTEM.ADMIN.COMMAND.QUEUE auf einem fernen Compaq NSK-System senden. Bei der im Nachrichtendeskriptor angegebenen Benutzer-ID muss es sich um einen Principal handeln, dem ein Eintrag in der Principal-Datenbank

entspricht und der einer Compaq NSK-Benutzer-ID zugeordnet ist, die auf dem Zielsystem zur Gruppe **mqm** gehört. Hierbei handelt es sich um folgende Befehle:

- ChangeChannel
- CopyChannel
- CreateChannel
- DeleteChannel
- PingChannel
- ResetChannel
- StartChannel
- StopChannel
- ResolveChannel

Informationen zu PCF-Sicherheitsanforderungen finden Sie im Handbuch *MQSeries Programmable System Management*.

MQSC-Kanalbefehle

Sie können MQSC-Kanalbefehle an ein fernes Compaq NSK-System ausgeben, indem Sie den Befehl entweder direkt in einer PCF-Escape-Nachricht senden oder ihn über den Befehl **runmqsc** im indirekten Modus eingeben. Bei der im Nachrichtendeskriptor angegebenen Benutzer-ID muss es sich um einen Principal handeln, dem ein Eintrag in der Principal-Datenbank entspricht und der einer Compaq NSK-Benutzer-ID zugeordnet ist, die auf dem Zielsystem zur Gruppe **mqm** gehört. (PCF-Befehle sind implizit in MQSC-Befehlen enthalten, die über **runmqsc** im indirekten Modus ausgegeben werden.) Hierbei handelt es sich um folgende Befehle:

- ALTER CHANNEL
- DEFINE CHANNEL
- DELETE CHANNEL
- PING CHANNEL
- RESET CHANNEL
- START CHANNEL
- START CHINIT
- STOP CHANNEL
- RESOLVE CHANNEL

Bei MQSC-Befehlen, die über den Befehl **runmqsc** ausgeführt werden, entspricht die Benutzer-ID in der PCF-Nachricht normalerweise der des aktuellen Benutzers.

Erläuterungen zu den Tabellen mit den Berechtigungsspezifikationen

In den Tabellen mit den Berechtigungsspezifikationen ab auf Seite 154 werden die Funktionen der Berechtigungen und die jeweiligen Einschränkungen genau definiert. Die Tabellen gelten für:

- Anwendungen, die MQI-Aufrufe ausgeben
- Verwaltungsprogramme, die MQSC-Befehle als PCF-Escape-Befehle ausgeben
- Verwaltungsprogramme, die PCF-Befehle ausgeben

Die Informationen in diesem Abschnitt sind in mehreren Tabellen zusammengefasst, die folgende Angaben enthalten:

Auszuführende Aktion	MQI-Option, MQSC-Befehl oder PCF-Befehl
Zugriffssteuerungsobjekt	Warteschlange, Prozess oder WS-Manager
Erforderliche Berechtigung	Dargestellt als eine 'MQZAO_'-Konstante

In den Tabellen entsprechen die Konstanten mit dem Präfix MQZAO_ den Schlüsselwörtern in der Berechtigungsliste des Befehls **setmqaut** für die jeweilige Definitionseinheit. Zum Beispiel entspricht MQZAO_BROWSE dem Schlüsselwort

Tabellen mit Berechtigungsspezifikationen

+browse, MQZAO_SET_ALL_CONTEXT dem Schlüsselwort +setall usw. Diese Konstanten sind in der Header-Datei CMQZCH im Teildatenträger ZMQSLIB definiert, die mit dem Produkt geliefert wird.

MQI-Berechtigungen

Eine Anwendung kann bestimmte MQI-Aufrufe und -Optionen nur verwenden, wenn die Benutzer-ID, unter der die Anwendung ausgeführt wird (bzw. deren Berechtigungen sie verwenden darf), über die entsprechende Berechtigung verfügt.

Bei vier MQI-Aufrufen sind Berechtigungsprüfungen erforderlich: MQCONN, MQOPEN, MQPUT1 und MQCLOSE.

Bei MQOPEN und MQPUT1 erfolgt die Berechtigungsprüfung für den Namen des Objekts, das geöffnet wird, und nicht für den Namen (oder die Namen), in den ein Name aufgelöst wurde. Eine Anwendung kann zum Beispiel über die Berechtigung verfügen, eine Aliaswarteschlange zu öffnen, ohne jedoch berechtigt zu sein, die Basiswarteschlange, in deren Name der Aliasname aufgelöst wird, zu öffnen. Die Regel besagt, dass die Überprüfung für die erste Definition ausgeführt wird, die bei der Namensauflösung gefunden wird und bei der es sich nicht um den Aliasnamen eines WS-Managers handelt, es sei denn, die Definition des WS-Manageralias wird direkt geöffnet; d. h., ihr Name steht im Feld *ObjectName* des Objektdeskriptors. Eine Berechtigung ist immer für das jeweilige Objekt erforderlich, das geöffnet wird; in einigen Fällen sind zusätzliche, warteschlangenabhängige Berechtigungen erforderlich, die durch eine Berechtigung für das WS-Managerobjekt erlangt werden.

Tabelle 3 enthält eine Zusammenfassung der für die einzelnen Aufrufe benötigten Berechtigungen.

Tabelle 3. Erforderliche Sicherheitsberechtigungen für MQI-Aufrufe

Erforderliche Berechtigung für:	Warteschlangenobjekt	Prozessobjekt	WS-Manager	Namenslisten
Option MQCONN	Nicht zutreffend	Nicht zutreffend	MQZAO_CONNECT	Nicht zutreffend
Option MQOPEN				
MQOO_INQUIRE	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)
MQOO_BROWSE	MQOO_BROWSE	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQOO_INPUT_*	MQZAO_INPUT	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQOO_SAVE_ALL_CONTEXT (3)	MQZAO_INPUT	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQOO_OUTPUT (normale Warteschlange) (4)	MQOO_OUTPUT	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQOO_PASS_IDENTITY_CONTEXT (5)	MQZAO_PASS_IDENTITY_CONTEXT	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQOO_PASS_ALL_CONTEXT (5, 6)	MQZAO_PASS_ALL_CONTEXT	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQOO_SET_IDENTITY_CONTEXT (5, 6)	MQZAO_SET_IDENTITY_CONTEXT	Nicht zutreffend	MQZAO_SET_IDENTITY_CONTEXT	Nicht zutreffend

Tabellen mit Berechtigungsspezifikationen

Table 3. Erforderliche Sicherheitsberechtigungen für MQI-Aufrufe (Forts.)

Erforderliche Berechtigung für:	Warteschlangenobjekt	Prozessobjekt	WS-Manager	Namenslisten
MQOO_SET_ALL_CONTEXT (5, 8)	MQZAO_SET_ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ALL_CONTEXT (7)	Nicht zutreffend
MQOO_OUTPUT (Übertragungswarteschlange) (9)	MQZAO_SET_ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ALL_CONTEXT (7)	Nicht zutreffend
MQOO_SET	MQZAO_SET	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQOO_ALTERNATE_USER_AUTHORITY	(10)	(10)	MQZAO_ALTERNATE_USER_AUTHORITY (10, 11)	Nicht zutreffend
Option MQPUT1				
MQPMO_PASS_IDENTITY_CONTEXT	MQZAO_PASS_IDENTITY_CONTEXT (12)	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT (12)	Nicht zutreffend	Keine Überprüfung	Nicht zutreffend
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT (12)	Nicht zutreffend	MQZAO_SET_IDENTITY_CONTEXT (7)	Nicht zutreffend
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT (12)	Nicht zutreffend	MQZAO_SET_ALL_CONTEXT (7)	Nicht zutreffend
(Übertragungswarteschlange) (9)	MQZAO_SET_ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ALL_CONTEXT (7)	Nicht zutreffend
MQPMO_ALTERNATE_USER_AUTHORITY	(13)	Nicht zutreffend	MQZAO_ALTERNATE_USER_AUTHORITY (11)	Nicht zutreffend
Option MQCLOSE				
MQCO_DELETE	MQZAO_DELETE (14)	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
MQCO_DELETE_PURGE	MQZAO_DELETE (14)	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend

Besondere Hinweise:

- Beim Öffnen einer Modellwarteschlange gilt Folgendes:
 - Für die Modellwarteschlange wird die Berechtigung MQZAO_DISPLAY benötigt, und zwar zusätzlich zu allen anderen Berechtigungen (ebenfalls für die Modellwarteschlange), die möglicherweise für die angegebenen Öffnungsoptionen erforderlich sind.
 - Zum Erstellen der dynamischen Warteschlange wird die Berechtigung MQZAO_CREATE nicht benötigt.
 - Der Benutzer-ID, unter der die Modellwarteschlange geöffnet wird, werden automatisch alle warteschlangenspezifischen Berechtigungen (entspricht MQZAO_ALL) für die dynamische Warteschlange erteilt.
- Abhängig von der Art des Objekts, das geöffnet wird, wird entweder das Warteschlangen-, Prozess-, Namenslisten- oder WS-Managerobjekt überprüft.
- MQOO_INPUT_* muss ebenfalls angegeben werden. Dies gilt für eine lokale, Modell- oder Aliaswarteschlange.

Tabellen mit Berechtigungsspezifikationen

4. Diese Überprüfung wird für alle Ausgaben ausgeführt (Ausnahme siehe Hinweis 9).
5. MQOO_OUTPUT muss ebenfalls angegeben werden.
6. MQOO_PASS_IDENTITY_CONTEXT ist in dieser Option enthalten.
7. Diese Berechtigung ist sowohl für das WS-Managerobjekt als auch für die betreffende Warteschlange erforderlich.
8. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT und MQOO_SET_IDENTITY_CONTEXT sind in dieser Option enthalten.
9. Diese Überprüfung wird für eine lokale oder Modellwarteschlange ausgeführt, für die das Warteschlangenattribut *Usage* auf MQUS_TRANSMISSION gesetzt ist und die direkt für eine Ausgabe geöffnet wurde. Dies gilt nicht, wenn eine ferne Warteschlange geöffnet wurde (indem entweder die Namen des fernen WS-Managers und der fernen Warteschlange oder der Name einer lokalen Definition der fernen Warteschlange angegeben wird).
10. Eine der folgenden Berechtigungen muss ebenfalls angegeben werden: MQOO_INQUIRE (für jede Objektart) oder (für Warteschlangen) MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT bzw. MQOO_SET. Die ausgeführte Überprüfung entspricht der für die anderen angegebenen Optionen, wobei die bereitgestellte alternative Benutzer-ID für die Überprüfung der speziell angegebenen Objektberechtigung und die aktuelle Anwendungsberechtigung für die Überprüfung der Berechtigung MQZAO_ALTERNATE_USER_IDENTIFIER verwendet wird.
11. Diese Berechtigung lässt die Angabe einer beliebigen alternativen Benutzer-ID (*AlternateUserId*) zu.
12. Eine MQZAO_OUTPUT-Überprüfung wird ebenfalls durchgeführt, wenn das Warteschlangenattribut *Usage* für die Warteschlange nicht auf MQUS_TRANSMISSION gesetzt ist.
13. Die ausgeführte Überprüfung entspricht der für die anderen angegebenen Optionen, wobei die bereitgestellte alternative Benutzer-ID für die Überprüfung der speziell angegebenen Warteschlangenberechtigung und die aktuelle Anwendungsberechtigung für die Überprüfung der Berechtigung MQZAO_ALTERNATE_USER_IDENTIFIER verwendet wird.
14. Die Überprüfung wird nur ausgeführt, wenn die beiden folgenden Bedingungen zutreffen:
 - Eine permanente dynamische Warteschlange wird geschlossen und gelöscht.
 - Die Warteschlange wurde nicht von dem MQOPEN-Aufruf geöffnet, der die verwendete Objektkennung zurückgegeben hat.

Andernfalls findet keine Überprüfung statt.

Allgemeine Hinweise:

1. Die Sonderberechtigung MQZAO_ALL_MQI schließt alle folgenden ein, die sich auf die Objektart beziehen:
 - 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

Tabellen mit Berechtigungsspezifikationen

2. MQZAO_DELETE (siehe Hinweis 14 auf Seite 156) und MQZAO_DISPLAY sind als Verwaltungsberechtigungen klassifiziert. Sie sind daher nicht in MQZAO_ALL_MQI eingeschlossen.
3. 'Keine Überprüfung' bedeutet, dass keine Berechtigungsprüfung durchgeführt wird.
4. 'Nicht zutreffend' bedeutet, dass Berechtigungsprüfungen für diese Operation nicht relevant sind. Sie können zum Beispiel keinen MQPUT-Aufruf für ein Prozessobjekt ausgeben.

Verwaltungsberechtigungen

Benutzer mit diesen Berechtigungen können Verwaltungsbefehle ausführen. Dabei kann es sich um einen MQSC-Befehl in Form einer PCF-Escape-Nachricht oder eines PCF-Befehls handeln. Mit Hilfe dieser Methoden kann ein Programm einen Verwaltungsbefehl in Form einer Nachricht an einen WS-Manager senden, der dann im Namen des betreffenden Benutzers ausgeführt wird.

Berechtigungen für MQSC-Befehle in PCF-Escape-Befehlen

Tabelle 4 enthält eine Zusammenfassung der benötigten Berechtigungen für einzelne MQSC-Befehle, die in PCF-Escape-Befehlen enthalten sind.

Tabelle 4. MQSC-Befehle und erforderliche Sicherheitsberechtigungen

(2) Erforderliche Berechtigung für:	Warteschlangenobjekt	Prozessobjekt	WS-Managerobjekt	Namenslisten
ALTER Objekt	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
CLEAR QLOCAL	MQZAO_CLEAR	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
DEFINE Objekt NOREPLACE (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Nicht zutreffend	MQZAO_CREATE (4)
DEFINE Objekt REPLACE (3, 5)	MQZAO_CHANGE	MQZAO_CHANGE	Nicht zutreffend	MQZAO_CHANGE
DELETE Objekt	MQZAO_DELETE	MQZAO_DELETE	Nicht zutreffend	MQZAO_DELETE
DISPLAY Objekt	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY

Besondere Hinweise:

1. Die Benutzer-ID, unter der das Programm (z. B. **runmqsc**) ausgeführt wird, das den Befehl übergibt, muss außerdem über die Berechtigung MQZAO_CONNECT für den WS-Manager verfügen.
2. Abhängig von der Art des Objekts wird entweder das Warteschlangen-, Prozess-, Namenslisten- oder WS-Managerobjekt überprüft.
3. Für DEFINE-Befehle wird außerdem die Berechtigung MQZAO_DISPLAY für das LIKE-Objekt (falls eins angegeben ist) oder für das entsprechende Objekt SYSTEM.DEFAULT.xxx (falls LIKE nicht angegeben ist) benötigt.
4. Die Berechtigung MQZAO_CREATE bezieht sich nicht auf ein bestimmtes Objekt oder eine bestimmte Objektart. Die Berechtigung zum Erstellen wird für alle Objekte (für einen angegebenen WS-Manager) erteilt, indem im Befehl SETMQAUT die Objektart QMGR angegeben wird.
5. Dies gilt, wenn das zu ersetzende Objekt tatsächlich bereits vorhanden ist. Wenn nicht, entspricht die Überprüfung der für 'DEFINE Objekt NOREPLACE'.

Tabellen mit Berechtigungsspezifikationen

Allgemeine Hinweise:

1. Um einen PCF-Befehl ausführen zu können, müssen Sie über die Berechtigung DISPLAY für den WS-Manager verfügen.
2. Die Berechtigung zum Ausführen eines PCF-Escape-Befehls ist von dem MQSC-Befehl abhängig, der im Text der PCF-Escape-Nachricht enthalten ist.
3. 'Nicht zutreffend' bedeutet, dass Berechtigungsprüfungen für diese Operation nicht relevant sind. Sie können zum Beispiel keinen Befehl CLEAR QLOCAL für ein WS-Managerobjekt ausgeben.

Berechtigungen für PCF-Befehle

Tabelle 5 enthält eine Zusammenfassung der für die einzelnen PCF-Befehle benötigten Berechtigungen.

Tabelle 5. PCF-Befehle und erforderliche Sicherheitsberechtigungen

(2) Erforderliche Berechtigung für:	Warteschlangenobjekt	Prozessobjekt	WS-Managerobjekt	Namenslisten
Change object	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
Clear Queue	MQZAO_CLEAR	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
Copy object (ohne 'replace') (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Nicht zutreffend	MQZAO_CREATE (4)
Copy object (mit 'replace') (3, 6)	MQZAO_CHANGE	MQZAO_CHANGE	Nicht zutreffend	MQZAO_CHANGE
Create object (ohne 'replace') (5)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Nicht zutreffend	MQZAO_CREATE (4)
Create object (mit 'replace') (5, 6)	MQZAO_CHANGE	MQZAO_CHANGE	Nicht zutreffend	MQZAO_CHANGE
Delete object	MQZAO_DELETE	MQZAO_DELETE	Nicht zutreffend	MQZAO_DELETE
Inquire object	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY
Inquire object names	Keine Überprüfung	Keine Überprüfung	Keine Überprüfung	Keine Überprüfung
Reset queue statistics	MQZAO_DISPLAY und MQZAO_CHANGE	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend

Besondere Hinweise:

1. Die Benutzer-ID, unter der das Programm, das den Befehl übergibt, ausgeführt wird, muss außerdem berechtigt sein, eine Verbindung mit seinem lokalen WS-Manager herzustellen und die Admin-Befehlswarteschlange für Ausgaben zu öffnen.
2. Je nach Objektart wird das Warteschlangen-, das Prozess- oder das WS-Managerobjekt überprüft.
3. Für Copy-Befehle wird außerdem die Berechtigung MQZAO_DISPLAY für das From-Objekt benötigt.
4. Die Berechtigung MQZAO_CREATE bezieht sich nicht auf ein bestimmtes Objekt oder eine bestimmte Objektart. Die Berechtigung zum Erstellen wird für alle Objekte (für einen angegebenen WS-Manager) erteilt, indem im Befehl SETMQAUT die Objektart QMGR angegeben wird.
5. Für Create-Befehle wird außerdem die Berechtigung MQZAO_DISPLAY für das entsprechende SYSTEM.DEFAULT.*-Objekt benötigt.
6. Dies gilt, wenn das zu ersetzende Objekt bereits vorhanden ist. Wenn nicht, entspricht die Überprüfung der für 'Copy' oder 'Create' ohne 'replace'.

Allgemeine Hinweise:

1. Um einen PCF-Befehl ausführen zu können, müssen Sie über die Berechtigung DISPLAY für den WS-Manager verfügen.
2. Die Sonderberechtigung MQZAO_ALL_ADMIN schließt alle folgenden ein, die sich auf die Objektart beziehen:
 - MQZAO_CHANGE
 - MQZAO_CLEAR
 - MQZAO_DELETE
 - MQZAO_DISPLAY

MQZAO_CREATE ist nicht eingeschlossen, weil diese Berechtigung sich nicht speziell auf ein bestimmtes Objekt oder eine bestimmte Objektart bezieht.

3. 'Keine Überprüfung' bedeutet, dass keine Berechtigungsprüfung durchgeführt wird.
4. 'Nicht zutreffend' bedeutet, dass Berechtigungsprüfungen für diese Operation nicht relevant sind. Beispielsweise kann der Befehl **Clear Queue** nicht für ein Prozessobjekt verwendet werden.

Erläuterung der Berechtigungsdateien

In MQSeries for Compaq NSK V5.1 werden alle Daten in Zusammenhang mit Berechtigungen in den folgenden TM/MP-Prüflistendateien gespeichert (die sich im Verzeichnis \$VOL.<QMgrSubVol>D befinden):

OAMDB Die OAM-Datenbank
PRIDB Die Principal-Datenbank
PRIDBA Die Datei mit den Alternativschlüsseln für die Principal-Datenbank

Die Principal-Datenbank

Jeder Datensatz in der Principal-Datenbank ordnet eine Compaq NSK-Benutzer-ID einem Principal zu. Bei dieser Datenbank handelt es sich um eine ENSCRIBE-Datei in Schlüsselfolge, die OAM-Principals Compaq NSK-Anmelde-IDs zuordnet.

OAM-Principal MQPRINCIPAL
Compaq-Anmelde-ID 0x2CFF

Der Primärschlüssel ist der OAM-Principal (12 Zeichen). Der Alternativschlüssel ist die Compaq-Anmelde-ID (eine kurze 2-Byte-Integer). Bei der Angabe des OAM-Principals muss die Groß-/Kleinschreibung beachtet werden. Die Bytes im Feld der Anmelde-ID stehen für <Gruppe>.<Benutzer> (im Beispiel oben handelt es sich um die Compaq NSK-ID 44.255).

Die OAM-Datenbank

Jeder Datensatz in der neuen OAM-Berechtigungsdatenbank entspricht einem bestimmten WS-Managerobjekt oder einer bestimmten Objektklasse. Bei dem Primärschlüssel handelt es sich um den Objektnamen plus Objektart. Die Datensätze haben eine variable Länge und folgendes Format:

Object Name	Type	#Auth Entries	Auth Entries
QUEUE.AUTH	1	2	PAYROLL 0x00000004, ADMIN 0xFFFFFFFF

Berechtigungsdateien

Das Feld **Object Name** enthält den vollständigen Namen des Objekts; es können Namen mit bis zu 48 Zeichen eingegeben werden; bei kürzeren Namen werden die restlichen Stellen mit Leerzeichen aufgefüllt. Im Feld **Type** (4 Byte) wird zwischen MQSeries-Objekten und den vom OAM benötigten Objektklassen unterschieden.

In diesem Feld sind folgende Werte zulässig:

- 1 Name der Warteschlange
- 2 Name des Prozesses
- 4 Name des WS-Managers
- 128 Klasse

Das Feld **#Auth Entries** (4 Byte) gibt die Anzahl der Berechtigungen an, die im Feld **Auth Entries** dieses Datensatzes enthalten sind. Jeder dieser Berechtigungseinträge besteht aus einem Gruppennamen und der Berechtigung, über die die Gruppe für dieses Objekt verfügt:

```
Group Name  PAYROLL
Auth        0x00000004
```

Das Feld **Group Name** hat eine Länge von 12 Bytes und enthält den Namen der Compaq NSK-Verwaltungsgruppe bzw. der SAFEGUARD-Gruppe für Dateizugriff (nur die ersten 12 Zeichen). Das Feld **Auth** mit einer Länge von 4 Bytes (ULONG) ist eine Bitmaske mit der Berechtigung für diese Gruppe. In jedem Datensatz können bis zu 250 Berechtigungseinträge (die sich jeweils aus Gruppenname und Berechtigung zusammensetzen) enthalten sein.

Für ein und dasselbe Objekt kann es bei Bedarf mehrere Datensätze geben, die Informationen zu Berechtigungen für mehr als 250 Gruppen enthalten.

Die Berechtigungsspezifikation ergibt sich aus den einzelnen Bitmustern auf Basis der folgenden Zuordnungen:

Authorization keyword	Formal name	Hexadecimal Value
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

Diese Definitionen erfolgen in der Header-Datei CMQZCH. Im folgenden Beispiel wurden groupB Berechtigungen entsprechend der Hexadezimalzahl 0x40007 erteilt. Dies entspricht:

MQZAO_CONNECT	0x00000001
MQZAO_BROWSE	0x00000002
MQZAO_INPUT	0x00000004
MQZAO_DISPLAY	0x00040000
Authority is:	----- 0x00040007

Diese Zugriffsrechte bedeuten, dass jeder Benutzer in groupB die folgenden MQI-Aufrufe ausgeben kann:

- MQCONN
- MQGET (mit 'browse')

Alle Mitglieder verfügen außerdem über die Berechtigung DISPLAY für das Objekt, das dieser Berechtigungsdatei zugeordnet ist.

Datensätze für Klassenberechtigungen

Datensätze für Klassenberechtigungen enthalten Berechtigungen für die gesamte Klasse. Die Felder für den Objektname und die Klassenart enthalten Folgendes:

Objektname	Art
@QMGRCLASS	0x80
@PROCESSCLASS	0x80
@QUEUECLASS	0x80

Der Eintrag MQZAO0_CRT im Berechtigungsfeld erteilt der Klasse die Berechtigung zum Erstellen eines Objekts. Dies ist die einzige Klassenberechtigung.

Datensatz mit Berechtigungen für alle Klassen

Dieser Datensatz enthält die Berechtigungen, die für den gesamten WS-Manager gelten. Die Felder für den Objektname und die Klassenart enthalten Folgendes:

Objektname	Art
@ALLCLASSES	0x80

Die folgenden Berechtigungen gelten für den gesamten WS-Manager und sind in der Berechtigungsdatei für alle Klassen definiert:

Eintrag	Berechtigung
MQZAO_ALTUSER	Übernehmen der Identität eines anderen Benutzers bei der Bearbeitung von MQSeries-Objekten
MQZAO_SET_ALL_CONTEXT	Angaben des Kontextes einer Nachricht beim Ausgeben eines MQPUT-Aufrufs
MQZAO_SET_IDENTITY_CONTEXT	Angaben des Identitätskontextes einer Nachricht beim Ausgeben eines MQPUT-Aufrufs

Berechtigungsdateien

Kapitel 9. MQSeries-Steuerroutine der Warteschlange für nicht zustellbare Nachrichten

In MQSeries for Compaq NSK steht eine Warteschlange für nicht zustellbare Nachrichten (Dead-Letter Queue, DLQ), manchmal auch als Warteschlange für nicht zugestellte Nachrichten bezeichnet, zur Verfügung, die die Nachrichten enthält, die nicht an die betreffenden Zielwarteschlangen übergeben werden können. Jeder WS-Manager in einem Netz sollte über eine Warteschlange für nicht zustellbare Nachrichten verfügen.

In diese Nachrichtenschlange werden von WS-Managern, Nachrichtenkanalagenten (Message Channel Agents, MCA) und Anwendungen Nachrichten eingereiht. Allen Nachrichten in der Warteschlange für nicht zustellbare Nachrichten muss ein *DLQ-Header* (MQDLH) vorangestellt werden. Nachrichten, die von einem WS-Manager oder einem Nachrichtenkanalagenten in die DLQ eingereiht wurden, müssen immer diesen Header enthalten. Nachrichten, die von Anwendungen in die DLQ eingereiht werden, sollten ebenfalls die MQDLH-Struktur enthalten. Das Feld *Reason* in der MQDLH-Struktur enthält einen Ursachencode mit einer Information, warum sich die Nachricht in der DLQ befindet.

Sie sollten über eine Routine verfügen, die in regelmäßigen Abständen die Nachrichten in der DLQ verarbeitet. MQSeries stellt eine Standardroutine, die so genannte *Steuerroutine für die Warteschlange für nicht zustellbare Nachrichten* (DLQ-Steuerroutine) zur Verfügung, die mit dem Befehl **runmqdlq** aufgerufen wird.

Anweisungen für die Verarbeitung von Nachrichten aus der DLQ werden über eine benutzerdefinierte *Regeltabelle* an die DLQ-Steuerroutine übergeben. Dabei vergleicht die DLQ-Steuerroutine Nachrichten in der DLQ mit den Einträgen in der Regeltabelle. Stimmt eine DLQ-Nachricht mit einem Eintrag in der Regeltabelle überein, führt die DLQ-Steuerroutine die diesem Eintrag zugeordnete Aktion aus.

Dieses Kapitel enthält die folgenden Abschnitte:

- „DLQ-Steuerroutine aufrufen“
- „Regeltabelle für die DLQ-Steuerroutine“ auf Seite 164
- „Verarbeitung der Regeltabelle“ auf Seite 171
- „Regeltabelle für DLQ-Steuerroutine - Beispiel“ auf Seite 173

DLQ-Steuerroutine aufrufen

Die DLQ-Steuerroutine wird mit dem Befehl **runmqdlq** aufgerufen. Sie haben folgende Möglichkeiten, den Namen der DLQ, die verarbeitet werden soll, und den Namen des WS-Managers, der verwendet werden soll, anzugeben:

- Über die Eingabeaufforderung unter Angabe von Parametern. Beispiel:

```
runmqdlq /IN qru1e/ ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER
```

DLQ-Steueroutine aufrufen

- In der Regeltabelle. Beispiel:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Diese Beispiele betreffen die DLQ mit dem Namen ABC1.DEAD.LETTER.QUEUE, die von dem WS-Manager ABC1.QUEUE.MANAGER verwaltet wird.

Wenn Sie keine DLQ oder keinen WS-Manager angeben, werden der bei der Installation definierte Standard-WS-Manager und die diesem zugeordnete DLQ verwendet.

Der Befehl **runmqdlq** liest die Eingabe aus der Regeltabelle, die für die Standard-eingabedatei zur Verfügung gestellt wird. Die Regeltabelle wird dem Befehl **runmqdlq** zugeordnet, indem IN an die Regeldatei (rules file) umgeleitet wird.

Damit die DLQ-Steueroutine ausgeführt werden kann, müssen Sie über Zugriff auf die DLQ selbst sowie auf alle Nachrichtenwarteschlangen, an die Nachrichten aus der DLQ weitergeleitet werden können, verfügen. Wenn die DLQ-Steueroutine in der Lage ist, Nachrichten in Warteschlangen mit der Berechtigung der im Nachrichtenkontext angegebenen Benutzer-ID einzureihen, müssen Sie darüber hinaus über die Berechtigung verfügen, die Identität anderer Benutzer anzunehmen.

Weitere Informationen zum Befehl **runmqdlq** finden Sie unter „runmqdlq (Steueroutine der DLQ ausführen)“ auf Seite 298

Regeltabelle für die DLQ-Steueroutine

Die Regeltabelle der DLQ-Steueroutine definiert, wie die DLQ-Steueroutine Nachrichten, die in der DLQ ankommen, verarbeitet. Diese Tabelle enthält zwei Arten von Einträgen:

- Der erste Eintrag in der Tabelle ist optional und enthält *Steuerdaten*.
- Bei allen anderen Einträgen in der Tabelle handelt es sich um *Regeln* für die DLQ-Steueroutine. Jede Regel besteht aus einem *Muster* (einer Gruppe von Nachrichtenmerkmalen), gegen das eine Nachricht abgeglichen wird, und einer *Aktion*, die ausgeführt wird, wenn eine Nachricht in der DLQ mit dem angegebenen Muster übereinstimmt. Eine Regeltabelle muss mindestens eine Regel enthalten.

Jeder Eintrag in der Regeltabelle besteht aus einem oder mehreren Schlüsselwörtern.

Eine Beschreibung der Syntaxregeln für Regeltabellen finden Sie unter „Konventionen für die Regeltabelle“ auf Seite 170.

Steuerdaten

In diesem Abschnitt werden die Schlüsselwörter beschrieben, die ein Steuerdateneintrag in einer Regeltabelle der DLQ-Steueroutine enthalten kann. Folgendes sollten Sie beachten:

- Der Standardwert für ein Schlüsselwort ist unterstrichen.
- Die vertikale Linie (|) trennt alternative Werte, d. h., es kann jeweils nur einer der Werte angegeben werden.
- Alle Schlüsselwörter sind optional.

INPUTQ (*Name der Warteschlange* | ' ')

Dieses Schlüsselwort gibt den Namen der DLQ an, der die Regeltabelle zugeordnet ist. Über dieses Schlüsselwort können Sie den Namen der DLQ angeben, die verarbeitet werden soll:

1. Wenn Sie im Befehl **runmqdlq** den Parameter *Name der Warteschlange* angeben, überschreibt dieser einen eventuell in der Regeltabelle für INPUTQ angegebenen Wert.
2. Wird im Befehl **runmqdlq** der Parameter *Name der Warteschlange* nicht angegeben, in der Regeltabelle jedoch ein Wert für INPUTQ angegeben, so wird dieser Wert übernommen.
3. Wenn Sie keine DLQ oder in der Regeltabelle INPUTQ(' ') angeben, wird die DLQ des WS-Managers verarbeitet, dessen Name über den Parameter *WS-Managername* im Befehl **runmqdlq** oder über das Schlüsselwort INPUTQM in der Regeltabelle angegeben wird.

INPUTQM (*WS-Managername* | ' ')

Dieses Schlüsselwort gibt den Namen des WS-Managers an, dem die DLQ zugeordnet ist. Über dieses Schlüsselwort können Sie den WS-Manager angeben, zu dem die über das Schlüsselwort INPUTQ angegebene DLQ gehört.

1. Wenn Sie im Befehl **runmqdlq** den Parameter *WS-Managername* angeben, überschreibt dieser einen eventuell in der Regeltabelle für INPUTQM angegebenen Wert.
2. Wird im Befehl **runmqdlq** der Parameter *WS-Managername* nicht angegeben, wird der in der Regeltabelle für INPUTQM angegebene Wert verwendet.
3. Wenn Sie keinen WS-Manager oder in der Regeltabelle INPUTQM(' ') angeben, wird der bei der Installation definierte Standard-WS-Manager verwendet.

RETRYINT (*Intervall* | 60)

Dieses Schlüsselwort gibt das Intervall (in Sekunden) an, in dem die DLQ-Steuerroutine versuchen soll, die Nachrichten in der DLQ erneut zu verarbeiten, die beim ersten Versuch nicht verarbeitet werden konnten und für die Wiederholungen angefordert wurden. Das Wiederholungsintervall ist standardmäßig auf 60 Sekunden eingestellt.

WAIT (YES | NO | *nnn*)

Dieses Schlüsselwort gibt an, ob die DLQ-Steuerroutine auf weitere Nachrichten in der DLQ warten soll, wenn sie feststellt, dass keine weiteren Nachrichten zum Verarbeiten anliegen.

- YES** Bei Angabe dieses Schlüsselwortes wartet die DLQ-Steuerroutine eine unbestimmte Zeit.
- NO** Bei Angabe dieses Schlüsselwortes wird die DLQ-Steuerroutine beendet, wenn sie feststellt, dass die DLQ leer ist oder keine Nachrichten enthält, die sie verarbeiten kann.
- nnn* Bei Angabe dieses Schlüsselwortes wartet die DLQ-Steuerroutine *nnn* Sekunden lang auf neue zu verarbeitende Nachrichten, bevor sie beendet wird, nachdem sie feststellt, dass die Warteschlange leer ist oder keine Nachrichten enthält, die sie verarbeiten kann.

Für stark ausgelastete DLQs sollten Sie WAIT (YES), für DLQs mit einer geringen Auslastung WAIT (NO) oder WAIT (*nnn*) angeben. Wenn die DLQ-Steuerroutine auf diese Weise beendet wird, sollte sie mit Hilfe der Auslösefunktion erneut aufgerufen werden.

Regeltabelle

Aus den Steuerdaten in Abb. 40 geht hervor, dass die Regeltabelle für die DLQ gilt, die dem WS-Manager QM1 zugeordnet ist. Das Pluszeichen (+) am Ende von Zeile 1 weist darauf hin, dass die Steuerdaten in Zeile 2 ab dem ersten belegten Zeichen fortgesetzt werden.

```
INPUTQ' ' +  
INPUTQM'QM1'
```

Abbildung 40. Steuerdaten - Beispiel

Als Alternative zum Einfügen von Steuerdaten in der Regeltabelle können Sie die Namen der DLQ und des WS-Managers als Eingabeparameter des Befehls **runmqdlq** angeben. Wenn sowohl in der Regeltabelle als auch mit dem Befehl **runmqdlq** ein Wert angegeben wird, hat der mit dem Befehl **runmqdlq** angegebene Wert Vorrang.

Anmerkung: Wird in der Regeltabelle ein Steuerdateneintrag eingefügt, muss es sich um den ersten Eintrag in der Tabelle handeln.

Regeln (Muster und Aktionen)

Abb. 41 zeigt eine Beispielregel aus einer Regeltabelle der DLQ-Steuerroutine. Diese Regel weist die DLQ-Steuerroutine an, drei Versuche zu unternehmen, permanente Nachrichten, die auf Grund inaktivierter MQPUT- und MQPUT1-Aufrufe in die DLQ eingereicht wurden, an ihre Zielwarteschlangen zuzustellen.

```
PERSIST(MQPER_PERSISTENT) REASON(MQRC_PUT_INHIBITED) +  
ACTION(RETRY) RETRY(3)
```

Abbildung 41. Beispielregel. Das Pluszeichen (+) am Ende von Zeile 1 weist darauf hin, dass die Regel in Zeile 2 ab dem ersten belegten Zeichen fortgesetzt wird.

Alle Schlüsselwörter, die Sie in einer Regel verwenden können, werden später in diesem Abschnitt beschrieben. Folgendes sollten Sie beachten:

- Der Standardwert für ein Schlüsselwort ist unterstrichen. Der Standardwert für die meisten Schlüsselwörter ist der Stern (*), der für einen beliebigen Wert steht.
- Die vertikale Linie (|) trennt alternative Werte, d. h., es kann jeweils nur einer der Werte angegeben werden.
- Alle Schlüsselwörter mit Ausnahme von ACTION sind optional.

Dieser Abschnitt beginnt mit einer Beschreibung der Schlüsselwörter für die Mustererkennung (mit denen die Nachrichten aus der DLQ verglichen werden), gefolgt von einer Beschreibung der Aktionsschlüsselwörter (die festlegen, wie die DLQ-Steuerroutine eine Nachricht, für die eine Übereinstimmung festgestellt wurde, verarbeiten soll).

Schlüsselwörter für die Mustererkennung

Die Schlüsselwörter für die Mustererkennung, über die Sie die Werte angeben, mit denen Nachrichten in der DLQ verglichen werden, werden im Folgenden beschrieben. Alle Schlüsselwörter für die Mustererkennung sind optional.

APPLIDAT (*ApplIdentityData* | *)

Dieses Schlüsselwort gibt den Wert an, der im Feld *ApplIdentityData* des Nachrichtendeskriptors (MQMD) der Nachricht in der DLQ angegeben ist.

APPLNAME (*PutApplName* | *)

Dieses Schlüsselwort gibt den Namen der Anwendung an, die den MQPUT- oder MQPUT1-Aufruf ausgegeben hat und die im Feld *PutApplName* des Nachrichtendeskriptors (MQMD) der Nachricht in der DLQ angegeben ist.

APPLTYPE (*PutApplType* | *)

Dieses Schlüsselwort gibt den Wert an, der im Feld *PutApplType* des Nachrichtendeskriptors (MQMD) der Nachricht in der DLQ angegeben ist.

DESTQ (*Name der Warteschlange* | *)

Dieses Schlüsselwort gibt den Namen der Nachrichtenwarteschlange an, an die die Nachricht gerichtet ist.

DESTQM (*WS-Managername* | *)

Dieses Schlüsselwort gibt den Namen des WS-Managers an, dem die Zielwarteschlange der Nachricht zugeordnet ist.

FEEDBACK (*Rückmeldung* | *)

Wenn MQFB_REPORT für *MsgType* angegeben ist, gibt das Schlüsselwort *Feedback* die Berichtart an.

Es können symbolische Namen verwendet werden. Sie können zum Beispiel den symbolischen Namen MQFB_COA verwenden, um diejenigen Nachrichten in der DLQ zu identifizieren, für die eine Bestätigung des Eingangs in ihren Zielwarteschlangen angefordert wird.

FORMAT (*Format* | *)

Dieses Schlüsselwort gibt den Namen an, den der Sender der Nachricht für das Format der Nachrichtendaten verwendet.

MSGTYPE (*Nachrichtenart* | *)

Dieses Schlüsselwort gibt die Art der Nachricht in der DLQ an.

Es können symbolische Namen verwendet werden. Sie können zum Beispiel den symbolischen Namen MQMT_REQUEST verwenden, um diejenigen Nachrichten in der DLQ zu identifizieren, für die Antworten angefordert werden.

PERSIST (*Permanenz* | *)

Dieses Schlüsselwort gibt an, ob die Nachricht permanent ist. (Die Permanenz einer Nachricht legt fest, ob die Nachricht bei Neustarts des WS-Managers erhalten bleibt.)

Es können symbolische Namen verwendet werden. Sie können zum Beispiel den symbolischen Namen MQPER_PERSISTENT verwenden, um diejenigen Nachrichten in der DLQ zu identifizieren, die permanent sind.

REASON (*Ursachencode* | *)

Dieses Schlüsselwort gibt den Ursachencode an, der beschreibt, warum eine Nachricht in die DLQ eingereicht wurde.

Es können symbolische Namen verwendet werden. Sie können zum Beispiel den symbolischen Namen MQRC_Q_FULL verwenden, um diejenigen Nachrichten zu identifizieren, die in die DLQ eingereicht wurden, weil ihre Zielwarteschlangen voll waren.

REPLYQ (*Name der Warteschlange* | *)

Dieses Schlüsselwort gibt den Namen der Warteschlange für zu beantwortende Nachrichten an, die im Nachrichtendeskriptor (MQMD) der Nachricht in der DLQ angegeben ist.

Regeltabelle

REPLYQM (*WS-Managername* | *)

Dieses Schlüsselwort gibt den Namen des WS-Managers an, dem die Warteschlange für zu beantwortende Nachrichten zugeordnet ist und der im Nachrichtendeskriptor (MQMD) der Nachricht in der DLQ angegeben ist.

USERID (*Benutzer-ID* | *)

Dieses Schlüsselwort gibt die Benutzer-ID des Benutzers (wie im Nachrichtendeskriptor MQMD angegeben) an, von dem die Nachricht stammt.

Aktionsschlüsselwörter

Im Folgenden werden die Aktionsschlüsselwörter beschrieben, über die Sie angeben, wie Nachrichten, für die eine Übereinstimmung gefunden wurde, verarbeitet werden sollen:

ACTION (**DISCARD** | **IGNORE** | **RETRY** | **FWD**)

Dieses Schlüsselwort gibt die Aktion an, die für jede Nachricht in der DLQ ausgeführt werden soll, die mit dem in dieser Regel definierten Muster übereinstimmt.

DISCARD Dieses Schlüsselwort gibt an, dass die Nachricht aus der DLQ gelöscht werden soll.

IGNORE Dieses Schlüsselwort gibt an, dass die Nachricht in der DLQ bleiben soll.

RETRY Dieses Schlüsselwort gibt an, dass die DLQ-Steuerroutine erneut versuchen soll, die Nachricht in die Zielwarteschlange einzureihen.

FWD Dieses Schlüsselwort gibt an, dass die DLQ-Steuerroutine die Nachricht an die über das Schlüsselwort FWDQ angegebene Warteschlange weiterleiten soll.

Sie müssen einen Wert für das Schlüsselwort ACTION angeben. Die Anzahl der Versuche zum Implementieren einer Aktion wird durch das Schlüsselwort RETRY festgelegt. Das Intervall zwischen Versuchen wird durch das Schlüsselwort RETRYINT in den Steuerdaten festgelegt.

FWDQ (*Name der Warteschlange* | **&DESTQ** | **&REPLYQ**)

Dieses Schlüsselwort gibt den Namen der Nachrichtenwarteschlange an, an die die Nachricht bei Anforderung von ACTION (FWD) weitergeleitet werden soll.

Name der Warteschlange

Dieses Schlüsselwort gibt den Namen einer Nachrichtenwarteschlange an. FWDQ(' ') ist nicht gültig.

&DESTQ

Bei Angabe dieses Schlüsselwortes wird der Name der Warteschlange aus dem Feld *DestQName* in der MQDLH-Struktur übernommen.

&REPLYQ

Bei Angabe dieses Schlüsselwortes wird der Name der Warteschlange aus dem Feld *ReplyToQ* im Nachrichtendeskriptor (MQMD) übernommen.

Um Fehlermeldungen zu vermeiden, wenn eine Regel, in der FWDQ (&REPLYQ) angegeben ist, mit einer Nachricht übereinstimmt, in der das Feld *ReplyToQ* leer ist, können Sie REPLYQ (?) im Nachrichtemuster angeben.

FWDQM (WS-Managername | &DESTQM | &REPLYQM | ' _')

Dieses Schlüsselwort gibt den WS-Manager der Warteschlange an, an die die Nachricht weitergeleitet werden soll.

Name des WS-Managers

Dieses Schlüsselwort gibt den WS-Manager der Warteschlange an, an den die Nachricht bei Anforderung von ACTION (FWD) weitergeleitet werden soll.

&DESTQM

Bei Angabe dieses Schlüsselwortes wird der Name des WS-Managers aus dem Feld *DestQMgrName* in der MQDLH-Struktur übernommen.

&REPLYQM

Bei Angabe dieses Schlüsselwortes wird der Name des WS-Managers dem Feld *ReplyToQMgr* des Nachrichtendesktors (MQMD) übernommen.

' ' Der Standardwert FWDQM(' ') gibt den lokalen WS-Manager an.

HEADER (YES | NO)

Dieses Schlüsselwort gibt an, ob der Header MQDLH in der Nachricht bleiben soll, für die die ACTION (FWD) angefordert wird. Der Header MQDLH bleibt standardmäßig in der Nachricht stehen. Das Schlüsselwort HEADER ist nur für die Aktion FWD gültig.

PUTAUT (DEF | CTX)

Dieses Schlüsselwort gibt die Berechtigung an, mit der Nachrichten von der DLQ-Steueroutine eingereicht werden sollen:

DEF Bei Angabe dieses Schlüsselwortes werden die Nachrichten mit der Berechtigung der DLQ-Steueroutine eingereicht.

CTX Bei Angabe dieses Schlüsselwortes werden die Nachrichten mit der Berechtigung der im Nachrichtenkontext angegebenen Benutzer-ID eingereicht. Wenn Sie PUTAUT (CTX) angeben, müssen Sie über die Berechtigung verfügen, die Identität anderer Benutzer anzunehmen.

RETRY (Wiederholungszähler | 1)

Dieses Schlüsselwort (zulässige Werte sind 1 bis 999) gibt an, wie oft eine Aktion wiederholt werden soll (in Abständen, die über das Schlüsselwort RETRYINT in den Steuerdaten angegeben sind).

Anmerkung: Die Anzahl der Wiederholungen, die von der DLQ-Steueroutine zum Implementieren einer bestimmten Regel ausgeführt werden, gilt nur für das aktuelle Exemplar der DLQ-Steueroutine. Der Zähler bleibt bei Neustarts nicht erhalten. Wenn die DLQ-Steueroutine erneut gestartet wird, wird der Zähler für die Anzahl der Wiederholungen zum Anwenden einer Regel auf null zurückgesetzt.

Konventionen für die Regeltabelle

Die Regeltabelle muss in Bezug auf ihre Syntax, Struktur und ihren Inhalt den folgenden Konventionen entsprechen:

- Eine Regeltabelle muss mindestens eine Regel enthalten.
- Die Reihenfolge der Schlüsselwörter ist beliebig.
- Ein Schlüsselwort darf nur ein Mal in einer Regel vorkommen.
- Bei Schlüsselwörtern wird die Groß-/Kleinschreibung nicht beachtet.
- Ein Schlüsselwort und seine Parameterwerte müssen durch mindestens ein Leerzeichen oder Komma von anderen Schlüsselwörtern getrennt werden.
- Vor und nach einer Regel sowie zwischen Schlüsselwörtern, Satzzeichen und Werten können beliebig viele Leerzeichen stehen.
- Jede Regel muss in einer neuen Zeile beginnen.
- Aus Gründen der Portierbarkeit sollte die signifikante Länge einer Zeile 72 Zeichen nicht überschreiten.
- Verwenden Sie das Pluszeichen (+) als letztes belegtes Zeichen in einer Zeile, um anzugeben, dass die Regel in der nächsten Zeile mit dem ersten Zeichen, bei dem es sich nicht um ein Leerzeichen handelt, fortgesetzt wird. Verwenden Sie das Minuszeichen (-) als letztes belegtes Zeichen in einer Zeile, um anzugeben, dass die Regel am Beginn der nächsten Zeile fortgesetzt wird. Fortsetzungszeichen können zwischen Schlüsselwörtern und Parametern stehen.
- Kommentarzeilen, die mit einem Stern (*) beginnen, können an jeder beliebigen Stelle in der Regeltabelle stehen.
- Leerzeilen werden ignoriert.
- Jeder Eintrag in der Regeltabelle der DLQ-Steuerroutine besteht aus einem oder mehreren Schlüsselwörtern und den zugehörigen Parametern. Die Parameter müssen den folgenden Syntaxregeln entsprechen:
 - Jeder Parameterwert muss mindestens ein signifikantes Zeichen enthalten. Die Anführungszeichen, die als Begrenzungszeichen für Werte verwendet werden, gelten nicht als signifikante Zeichen. Folgende Parameter sind beispielsweise gültig:
 - FORMAT('ABC')** 3 signifikante Zeichen
 - FORMAT(ABC)** 3 signifikante Zeichen
 - FORMAT('A')** 1 signifikantes Zeichen
 - FORMAT(A)** 1 signifikantes Zeichen
 - FORMAT(' ')** 1 signifikantes Zeichen

Die folgenden Parameter sind ungültig, da sie keine signifikanten Zeichen enthalten:

FORMAT('')
FORMAT()
FORMAT()
FORMAT

- Platzhalterzeichen werden unterstützt: Das Fragezeichen (?) kann anstelle eines einzelnen Zeichens, außer für ein abschließendes Leerzeichen, und der Stern (*) anstelle einer Null oder mehrerer aufeinanderfolgender Zeichen verwendet werden. Der Stern (*) und das Fragezeichen (?) werden in Parameterangaben *immer* als Platzhalterzeichen interpretiert.

Konventionen für die Regeltabelle

- Platzhalterzeichen sind in Parametern der folgenden Schlüsselwörter nicht zulässig: ACTION, HEADER, RETRY, FWDQ, FWDQM und PUTAUT.
- Abschließende Leerzeichen in Parameterwerten und in den entsprechenden Feldern der Nachricht in der DLQ gelten nicht als signifikante Zeichen, wenn Platzhalterzeichenabgleiche ausgeführt werden. Führende und eingebettete Leerzeichen innerhalb von Zeichenfolgen in Anführungszeichen gelten bei Platzhalterzeichenabgleichen jedoch als signifikante Zeichen.
- In numerischen Parametern darf das Fragezeichen (?) nicht als Platzhalterzeichen verwendet werden. Der Stern (*) kann anstelle eines vollständigen numerischen Parameters verwendet werden, darf jedoch nicht Teil eines numerischen Parameters sein. Folgende numerische Parameter sind beispielsweise gültig:

MSGTYPE(2) Es sind nur Antwortnachrichten auswählbar.

MSGTYPE(*) Alle Nachrichtenarten sind auswählbar.

MSGTYPE('*') Alle Nachrichtenarten sind auswählbar.

MSGTYPE('2*') ist jedoch nicht gültig, da es einen Stern (*) als Teil eines numerischen Parameters enthält.

- Numerische Parameter müssen im Bereich 0 bis 999 liegen. Ein Parameterwert, der in diesem Bereich liegt, wird akzeptiert, und zwar auch dann, wenn er in dem Feld, auf das sich das Schlüsselwort bezieht, eigentlich nicht gültig ist. Für numerische Parameter können symbolische Namen verwendet werden.
- Wenn ein Zeichenfolgewart kleiner als das Feld im Header MQDLH oder MQMD ist, auf das sich das Schlüsselwort bezieht, wird das Feld bis zum Ende mit Leerzeichen aufgefüllt. Ist der Wert (ohne Sterne) länger als das Feld, wird ein Fehler erkannt. Bei den folgenden Beispielen handelt es sich um gültige Zeichenfolgewartwerte für ein Feld mit einer Länge von 8 Zeichen:
'ABCDEFGH' 8 Zeichen
'A*C*E*G*I' 5 Zeichen (ohne die Sterne)
'*A*C*E*G*I*K*M*O*'
8 Zeichen (ohne die Sterne)
- Zeichenfolgen mit Leerzeichen, Kleinbuchstaben oder Sonderzeichen, außer Punkt (.), Schrägstrich (/), Unterstreichung (_) und Prozentzeichen (%), müssen in einfachen Anführungszeichen stehen. Kleinbuchstaben, die nicht in Anführungszeichen stehen, werden in Großbuchstaben umgesetzt. Wenn die Zeichenfolge ein Zitat enthält, müssen sowohl am Anfang als auch am Ende des Zitats zwei einfache Anführungszeichen stehen. Bei der Berechnung der Zeichenfolgelänge wird jedes Vorkommen von doppelten Anführungszeichen als ein einfaches Anführungszeichen gezählt.

Verarbeitung der Regeltabelle

Die DLQ-Steuerroutine durchsucht die Regeltabelle nach einer Regel, deren Muster mit einer Nachricht in der DLQ übereinstimmt. Die Suche beginnt mit der ersten Regel in der Tabelle und geht dann die Regeln sequenziell bis zum Ende der Tabelle durch. Wenn eine Regel mit einem übereinstimmenden Muster gefunden wird, wird die Aktion der betreffenden Regel ausgeführt. Die DLQ-Steuerroutine erhöht den Wiederholungszähler für eine Regel um 1, wenn sie versucht, die Regel anzuwenden. Schlägt der erste Versuch fehl, wird der Versuch so lange wiederholt, bis die Anzahl der Versuche mit dem im Schlüsselwort RETRY angegebenen Wert übereinstimmt. Wenn alle Versuche fehlschlagen, sucht die DLQ-Steuerroutine nach der nächsten übereinstimmenden Regel in der Tabelle.

Verarbeitung der Regeltabelle

Dieser Prozess wird so lange für weitere übereinstimmende Regeln wiederholt, bis eine Aktion erfolgreich ausgeführt werden kann. Wenn jede übereinstimmende Regel so oft versucht wurde, wie im jeweils zugehörigen Schlüsselwort `RETRY` angegeben, aber kein Versuch erfolgreich war, wird `ACTION (IGNORE)` angenommen. `ACTION (IGNORE)` wird auch angenommen, wenn keine übereinstimmende Regel gefunden wird.

Anmerkungen:

1. Übereinstimmende Regelmuster werden nur für Nachrichten in der DLQ gesucht, die mit einem Header `MQDLH` beginnen. Nachrichten, die nicht mit einem `MQDLH` beginnen, werden in regelmäßigen Abständen als fehlerhaft gemeldet und verbleiben auf unbestimmte Zeit in der DLQ.
2. Für alle Musterschlüsselwörter können Standardwerte verwendet werden, so dass eine Regel nur aus einer Aktion bestehen kann. Beachten Sie jedoch, dass Regeln, die nur aus einer Aktion bestehen, auf alle Nachrichten in der Warteschlange angewendet werden, die `MQDLH`-Strukturen enthalten und die noch nicht anhand anderer Regeln in der Tabelle verarbeitet wurden.
3. Die Gültigkeit der Regeltabelle wird beim Starten der DLQ-Steuerroutine überprüft, und Fehler werden markiert. Eine Beschreibung der von der DLQ-Steuerroutine ausgegebenen Fehlernachrichten finden Sie im Handbuch *MQSeries Messages*. Sie können die Regeltabelle jederzeit ändern, die Änderungen werden jedoch erst nach einem Neustart der DLQ-Steuerroutine wirksam.
4. Die DLQ-Steuerroutine ändert weder den Inhalt von Nachrichten noch den des Headers `MQDLH` oder des Nachrichtendeskriptors. Die DLQ-Steuerroutine reiht Nachrichten immer mit der Nachrichtenoption `MQPMO_PASS_ALL_CONTEXT` in andere Warteschlangen ein.
5. Die DLQ-Steuerroutine öffnet die DLQ mit der Option `MQOO_INPUT_AS_Q_DEF`.
6. Mehrere Exemplare der DLQ-Steuerroutine können gleichzeitig unter Verwendung derselben Regeltabelle für dieselbe Warteschlange ausgeführt werden. Im Normalfall besteht jedoch eine Eins-zu-eins-Beziehung zwischen einer DLQ und einer DLQ-Steuerroutine.

Verarbeitung aller DLQ-Nachrichten sicherstellen

Die DLQ-Steuerroutine führt ein Protokoll über die Nachrichten in der DLQ, die angezeigt, aber nicht entfernt wurden. Wenn Sie die DLQ-Steuerroutine als ein Filter zum Extrahieren einer kleinen Untermenge von Nachrichten aus der DLQ verwenden, muss die DLQ-Steuerroutine für die Nachrichten in der DLQ, die nicht von ihr verarbeitet wurden, weiter ein Protokoll führen. Außerdem kann DLQ-Steuerroutine auch nicht sicherstellen, dass neu in der DLQ eintreffende Nachrichten angezeigt werden, selbst wenn für die DLQ das FIFO-Verfahren (First In/First Out) definiert wurde. Wenn die Warteschlange nicht leer ist, wird die DLQ deshalb in regelmäßigen Abständen neu durchsucht, um alle Nachrichten zu überprüfen. Daher sollten Sie dafür sorgen, dass die DLQ immer möglichst wenige Nachrichten enthält. Wenn sich Nachrichten, die nicht gelöscht oder an andere Warteschlangen weitergeleitet werden können (aus welchen Gründen auch immer), in der Warteschlange ansammeln, erhöht sich die Arbeitsbelastung der DLQ-Steuerroutine, und es besteht die Gefahr, dass die DLQ vollständig gefüllt wird.

Verarbeitung der Regeltabelle

Sie können besondere Maßnahmen ergreifen, um der DLQ-Steerroutine zu ermöglichen, die DLQ zu leeren. Vermeiden Sie beispielsweise die Verwendung der ACTION (IGNORE), bei der Nachrichten in der DLQ beibehalten werden. Diese Aktion wird für Nachrichten verwendet, auf die keine andere Regel in der Tabelle zutrifft. Verwenden Sie für Nachrichten, die ignoriert werden sollen, eine Aktion, mit der die Nachrichten in eine andere Warteschlange verschoben werden. Beispiel:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Die letzte Regel in der Tabelle sollte so definiert werden, dass sie alle Nachrichten erfasst, auf die keine der vorausgehenden Regeln in der Tabelle angewendet werden konnten. Die letzte Regel in der Tabelle könnte zum Beispiel folgendermaßen lauten:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Mit dieser Aktion werden Nachrichten, auf die keine der vorausgehenden Regeln in der Tabelle angewendet werden konnte, an die Warteschlange REALLY.DEAD.QUEUE weitergeleitet, wo sie manuell verarbeitet werden können. Wenn eine solche letzte Regel nicht vorhanden ist, bleiben Nachrichten auf unbestimmte Zeit in der DLQ stehen.

Regeltabelle für DLQ-Steerroutine - Beispiel

Es folgt ein Beispiel für eine Regeltabelle, die einen einzigen Steuerdateneintrag und mehrere Regeln enthält:

```
*****
*       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
```

Regeltabelle - Beispiel

- * 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
- * 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)
```

Kapitel 10. Instrumentierungs- und EMS-Ereignisse

Dieses Kapitel enthält Folgendes:

- Eine kurze Einführung in MQSeries-Instrumentierungsereignisse, anhand deren Sie den Betrieb der WS-Manager überwachen können. Siehe hierzu „MQSeries-Instrumentierungsereignisse“. Ausführliche Informationen zu Instrumentierungsereignissen finden Sie im Handbuch *MQSeries Programmable System Management*.
- Eine Beschreibung der Verwendung von EMS-Ereignissen in MQSeries for Compaq NSK (EMS, Event Management Service). Siehe hierzu „EMS-Ereignisse“ auf Seite 178.

MQSeries-Instrumentierungsereignisse

Instrumentierungsereignisse sind der Anlass für die Generierung von *Ereignisnachrichten*, wenn ein WS-Manager erkennt, dass eine Reihe vordefinierter Bedingungen eingetreten sind. Das Ereignis **Queue Full** (Warteschlange voll) tritt beispielsweise unter folgenden Bedingungen ein:

- Für eine bestimmte Warteschlange ist das Ereignis **Queue Full** (Warteschlange voll) aktiviert.
- Eine Anwendung gibt einen MQPUT-Aufruf aus, um eine Nachricht in diese Warteschlange einzureihen, der Aufruf schlägt jedoch fehl, weil die Warteschlange voll ist.

Die folgenden Bedingungen können ebenfalls zu Instrumentierungsereignissen führen:

- Die in einer Warteschlange maximal zulässige Anzahl an Nachrichten wurde erreicht.
- Auf eine Warteschlange erfolgt innerhalb eines bestimmten Zeitraums kein Zugriff.
- Ein Kanalexemplar wird gestartet oder gestoppt.
- Eine Anwendung versucht, eine Warteschlange unter einer Benutzer-ID zu öffnen, die über keine entsprechende Berechtigung verfügt.

Mit Ausnahme von Kanalereignissen müssen alle Instrumentierungsereignisse aktiviert werden, bevor sie generiert werden können.

Die Ereignisnachricht enthält Informationen zu den Bedingungen, die zu dem Ereignis geführt haben. Diese Nachricht wird in die *Ereigniswarteschlange* einge-reiht. Die Ereignisnachricht kann von einer Anwendung zu Analysezwecken aus dieser Warteschlange abgerufen werden.

Werden Ereigniswarteschlangen als ferne Warteschlangen definiert, können Sie alle Ereigniswarteschlangen in einem einzigen WS-Manager unterbringen (für Knoten, die Instrumentierungsereignisse unterstützen). Mit Hilfe der generierten Ereignisse können Sie dann ein Netz von WS-Managern von einem einzigen Knoten aus überwachen.

Instrumentierungsereignisse

Ereignisarten

Es gibt vier Arten von Instrumentierungsereignissen:

WS-Managerereignisse

Diese Ereignisse beziehen sich auf Ressourcendefinitionen in den WS-Managern. So wird beispielsweise ein WS-Managerereignis generiert, wenn eine Anwendung versucht, eine Nachricht in eine nicht vorhandene Warteschlange einzureihen.

Leistungsereignisse

Diese Ereignisse weisen darauf hin, dass ein für eine Ressource definierter Schwellenwert erreicht wurde. Ein Leistungsereignis wird beispielsweise generiert, wenn ein Grenzwert für die Länge einer Warteschlange erreicht wurde oder wenn nach einem MQGET-Aufruf innerhalb eines bestimmten Zeitraums kein Zugriff auf eine Warteschlange erfolgt.

Kanalereignisse

Diese Ereignisse werden von Kanälen ausgegeben, wenn während ihres Betriebs bestimmte Bedingungen auftreten. Ein Kanalereignis wird beispielsweise generiert, wenn ein Kanalexemplar gestoppt wird.

Auslöseereignis

Dieses Ereignis tritt auf, wenn ein WS-Manager feststellt, dass die für das Auslöseereignis definierten Bedingungen eingetreten sind. Eine Warteschlange kann zum Beispiel so konfiguriert werden, dass sie bei jedem Eingang einer Nachricht ein Auslöseereignis generiert. (Es bestehen erhebliche Unterschiede zwischen den Bedingungen für Auslöseereignisse und denen für Instrumentierungsereignisse.)

Nach einem Auslöseereignis wird eine Auslösenachricht in eine Initialisierungswarteschlange eingereiht und, optional, ein Anwendungsprogramm gestartet.

Ereignisaufzeichnung durch Ereigniswarteschlangen

Wenn ein Ereignis eintritt, reiht der WS-Manager eine Ereignisnachricht in die entsprechende Ereigniswarteschlange (falls definiert) ein. Die Ereignisnachricht enthält Informationen zu dem betreffenden Ereignis; diese Ereignisse können Sie abrufen, indem Sie ein geeignetes MQI-Anwendungsprogramm mit den folgenden Funktionen erstellen:

- Abrufen der Nachricht aus der Warteschlange.
- Verarbeiten der Nachricht zum Extrahieren der Ereignisdaten. Eine Beschreibung der Ereignisnachrichtenformate finden Sie im Handbuch *MQSeries Programmable System Management*.

Für jede Ereigniskategorie gibt es eine eigene Ereigniswarteschlange. Alle Ereignisse einer Kategorie haben zur Folge, dass eine Ereignisnachricht in die zugehörige Warteschlange eingereiht wird.

Diese Warteschlange...	enthält Nachrichten zu...
SYSTEM.ADMIN.QMGR.EVENT	WS-Managerereignissen
SYSTEM.ADMIN.PERFM.EVENT	Leistungsereignissen
SYSTEM.ADMIN.CHANNEL.EVENT	Kanalereignissen

Ereigniswarteschlangen mit Auslösern verwenden

Sie können Auslöser für die Ereigniswarteschlangen so konfigurieren, dass die Ereignisnachricht, die bei der Generierung eines Ereignisses in die Ereigniswarteschlange eingereicht wird, eine benutzerdefinierte Überwachungsanwendung startet. Diese Anwendung kann die Ereignisnachrichten verarbeiten und entsprechende Aktionen ausführen. Bei einigen Ereignissen ist es beispielsweise erforderlich, einen Bediener zu informieren; andere Ereignisse wiederum können eine Anwendung starten, die verschiedene Verwaltungs-Tasks automatisch ausführt.

Instrumentierungsereignisse aktivieren

Die Aktivierung von Instrumentierungsereignissen hängt von der Ereignisart ab:

- WS-Managerereignisse werden über Attribute des WS-Managers aktiviert.
- Leistungsereignisse müssen insgesamt auf dem WS-Manager aktiviert werden. Darüber hinaus müssen Sie bestimmte Leistungsereignisse über das entsprechende Warteschlangenattribut aktivieren und dabei die Bedingungen (z. B. eine Obergrenze für die Warteschlangenlänge) angeben, bei deren Eintreten das Ereignis generiert werden soll.
- Kanalereignisse treten automatisch ein, müssen also nicht aktiviert werden. Wenn Sie Kanalereignisse nicht überwachen wollen, können Sie PUT-Aufrufe für die Kanalereigniswarteschlange inaktivieren.

Sie haben folgende Möglichkeiten, die Generierung der Instrumentierungsereignisse zu aktivieren bzw. inaktivieren:

- Über MQSC-Befehle. Weitere Informationen hierzu finden Sie im Handbuch *MQSeries MQSC - Befehle*.
- Über PCF-Befehle für WS-Manager. Weitere Informationen hierzu finden Sie im Handbuch *MQSeries Programmable System Management*.
- Über MQAI-Befehle. Weitere Informationen hierzu finden Sie im Handbuch *MQSeries Administration Interface Programming Guide and Reference*.

Ereignisnachrichten

Ereignisnachrichten enthalten Informationen zum Ursprung eines Ereignisses, einschließlich Ereignisart und Name der Anwendung, die das Ereignis verursacht hat; bei Leistungsereignissen wird außerdem eine kurze Statistik für die Warteschlange erstellt.

Das Format von Ereignisnachrichten ist dem von PCF-Antwortnachrichten ähnlich. Die Nachrichtendaten können von benutzerdefinierten Verwaltungsprogrammen mit Hilfe der im Handbuch *MQSeries Programmable System Management* beschriebenen Datenstrukturen aus den Nachrichten abgerufen werden.

EMS-Ereignisse

MQSeries for Compaq NSK generiert EMS-Ereignisse (EMS, Event Management Service), die den WS-Manager-, Kanal- und Leistungsereignissen von MQSeries entsprechen. Ebenso können EMS-Nachrichten generiert werden, die FFST-Berichten und den Nachrichteneinträgen in MQSeries-Protokollen entsprechen. Diese Ereignisnachrichten können Systembediener und Administratoren auf Softwarebedingungen hinweisen, die sich nachteilig auf die MQSeries-Betriebsumgebung auswirken könnten.

EMS-Schablonendateien in MQSeries for Compaq NSK

Im Teildatenträger ZMQSSYS stehen die folgenden Dateien zur Verfügung:

ZMQSTMPL (Dateicode 839)

Eine EMS-Schablonenobjektdatei, die Formatierschablonen für die von MQSeries generierten EMS-Ereignisse enthält.

ZMQSDDL (Dateicode 101)

Das DDL-Schema (Data Definition Language, Datendefinitionssprache) für die von MQSeries generierten EMS-Ereignisse.

ZMQSC (Dateicode 101)

Kompilierte Ausgabe (in C) des DDL-Compilers für die Definitionen der vom Produkt generierten EMS-Ereignisse.

ZMQSCOB (Dateicode 101)

Kompilierte Ausgabe (in COBOL) des DDL-Compilers für die Definitionen der vom Produkt generierten EMS-Ereignisse.

ZMQSPAS (Dateicode 101)

Kompilierte Ausgabe (in PASCAL) des DDL-Compilers für die Definitionen der vom Produkt generierten EMS-Ereignisse.

ZMQSTACL (Dateicode 101)

Kompilierte Ausgabe (in TACL) des DDL-Compilers für die Definitionen der vom Produkt generierten EMS-Ereignisse.

ZMQSTAL (Dateicode 101)

Kompilierte Ausgabe (in TAL) des DDL-Compilers für die Definitionen der vom Produkt generierten EMS-Ereignisse.

Der Teildatenträger ZMQSSYS enthält die EMS-Schablonendatei SMQSTMPL, auf deren Basis die Schablonendatei ZMQSTMPL generiert wird. Die Datei ZMQSTMPL kann mit den Befehlen COUP und SYSGEN in die Ereignisschablonen Ihres Systems integriert werden. Der Quellcode der Ereignisschablonen steht zur Verfügung, so dass Sie die Formatierung der Ereignisse in Ihrer Umgebung anpassen können.

Beispielsweise legen Sie vielleicht keinen Wert darauf, dass alle Informationen in einem Ereignis angezeigt werden, oder Sie möchten den Text, der zusammen mit den Informationen im Ereignis angezeigt wird, vielleicht erweitern oder ändern. Eine Beschreibung der Quellsprache der EMS-Ereignisschablonen und der Vorgehensweise beim Kompilieren der Definitionen, um eine alternative ZMQSTMPL-Datei zu erstellen, finden Sie in der Compaq-Dokumentation.

MQSeries-EMS-Ereignisschablonen integrieren

Die Schablonenobjektdatei muss in die residenten und nicht residenten EMS-Schablonendateien Ihres Systems integriert werden, damit Programme wie z. B. VIEWPOINT und EMSDIST MQSeries-EMS-Ereignisse formatieren und anzeigen können.

Ein Verfahren für die Integration der EMS-Schablonen von MQSeries in die Systemschablonen wird im Folgenden beschrieben. Unter Umständen ist für Ihre Installation ein anderes Verfahren erforderlich.

1. Ermitteln Sie mit Hilfe des COUP-Befehls INFO ALLPROCESSORS die Namen der aktuellen Systemschablonen, und notieren Sie sich die für den Parameter EMS^TEMPLATES angegebenen Werte. Beispiel:

```
$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. Ermitteln Sie den Namen der aktuellen Systemschablone für ein Betriebssystem der G-Serie mit dem SCF-Befehl ASSUME SUBSYS \$ZZKRN; INFO und beachten Sie die für NONRESIDENT_TEMPLATES und RESIDENT_TEMPLATES angezeigten Werte. Beispiel:

```
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
```

EMS-Ereignisse

3. Führen Sie den Compiler TEMPLI aus, um neue Systemschablonendateien zu erstellen, indem die neuen MQSeries-Schablonen in die aktuellen Systemschablonen integriert werden. Dieser Vorgang vollzieht sich in zwei Schritten:
 - a. Erstellen Sie eine Textdatei, die die folgenden Befehle enthält:

```
FILE <current NONRESIDENT system template file>
FILE <MQSeries install volume>.ZMQSSYS.ZMQSTMPL
EXIT
```

Beispiel:

```
FILE $SYSTEM.SYS06.TEMPLATE
FILE $DEV2.ZMQSSYS.ZMQSTMPL
EXIT
```

- b. Führen Sie den Compiler TEMPLI aus, wobei Sie die neue Textdatei als Eingabe angeben:

```
TEMPLI /IN <Befehlsdatei>/<neue residente Schablonendatei>, <new
nonresident template file>
```

Wenn Sie beispielsweise die Befehlsdatei TEMGUIDE erstellt haben und neue Schablonendateien in \$SYSTEM.EMS erstellen möchten, müssen Sie den folgenden Befehl eingeben:

```
TEMPLI /IN TEMGUIDE/$SYSTEM.EMS.NEWRES, $SYSTEM.EMS.NEWNRES
```

Die Kompilierung der neuen Schablonendateien kann mehrere Minuten in Anspruch nehmen, da alle auf Ihrem System erforderlichen EMS-Ereignisschablonen verarbeitet werden.

4. Konfigurieren Sie mit dem COUP-Befehl Ihr System, so dass anstelle der aktuellen Schablonen die neuen EMS-Ereignisschablonen verwendet werden:

```
ASSUME ALLPROCESSORS
ALTER EMS^TEMPLATES(RESIDENT <new resident template file>,
                    NONRESIDENT <new nonresident template file>)
EXIT
```

Bei Betriebssystemen der G-Serie müssen Sie mit SCF-Befehlen Ihr System für die Verwendung der neuen EMS-Ereignisschablonen verwenden:

```
ALTER $ZZKRN, RESIDENT_TEMPLATES $SYSTEM.SYS01.NEWRES
ALTER $ZZKRN, NONRESIDENT_TEMPLATES $SYSTEM.SYS01.NEWNRES
EXIT
```

Anmerkung: Um eine permanente Änderung zu erreichen, müssen Sie das System mit dem SYSGEN aktualisieren.

Weitere Informationen zu EMS-Schablonen finden Sie im Handbuch *DSM Template Services Manual* von Compaq. In diesem Handbuch wird auch die Verwendung von SYSGEN hierzu erläutert.

Umgebungsvariable MQEMSEVENTS definieren

Um die EMS-Ereignisse von MQSeries endgültig zu aktivieren, müssen Sie sicherstellen, dass die Umgebungsvariable MQEMSEVENTS korrekt definiert wird. Bei diesem Wert handelt es sich um eine Zeichenfolge mit 4 Zeichen, die wie folgt als Bitmap interpretiert wird:

EMS-Nachricht	Bitmap-Eintrag	Wert für 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

Damit also alle EMS-Ereignisse für MQSeries aktiviert werden, müssen Sie in der TACL-Umgebung (in der alle Verwaltungsbefehle eingegeben werden) die folgende Umgebungsvariable definieren:

```
PARAM MQEMSEVENTS 127
```

Diese Definition ist auch in den Serverklassendefinitionen aller MQSeries-Serverklassen erforderlich. Für jede Serverklasse können eigene Optionen verwendet werden. Weitere Informationen finden Sie unter „Parameter von PATHWAY-Serverklassen ändern“ auf Seite 45.

Die Standardeinstellung gibt vor, dass keine EMS-Ereignisse generiert werden (d. h., die Umgebungsvariablen werden nicht definiert).

Alternativen Ereignissammler (Event Collector) verwenden

Unter Compaq NSK wird \$0 standardmäßig als EMS-Ereignissammler verwendet, der immer vorhanden ist. Alle EMS-Ereignisse, die von einem MQSeries-WS-Manager generiert werden, werden an diesen Standardsammler gesendet. Soll für den WS-Manager ein anderer Sammler für die EMS-Ereignisse verwendet werden, müssen Sie den Eintrag EMSCollector in der Zeilengruppe Configuration der Konfigurationsdatei QMINI entsprechend ändern und anschließend den WS-Manager erneut starten. Sie können für jeden WS-Manager einen eigenen EMS-Ereignissammler angeben.

Programme für die Verarbeitung von MQSeries-EMS-Ereignissen erstellen

Sie können eine eigene Anwendung zur Überwachung eines MQSeries-WS-Managers durch die Verarbeitung von EMS-Ereignisnachrichten erstellen. Eine solche Anwendung könnte auch Einfluss auf den Betrieb des WS-Managers nehmen, indem es auf die generierten EMS-Ereignisnachrichten hin PCF-Befehle ausgibt.

In den in MQSeries for Compaq NSK zur Verfügung gestellten Dateien ZMQSC, ZMQSTAL, ZMQSCOB, ZMQSPAS und ZMQSTACL im Teildatenträger ZMQSSYS sind die in den MQSeries-EMS-Ereignisnachrichten enthaltenen Token in C, TAL, COBOL, PASCAL und TACL definiert. Diese Definitionen können von einem Verwaltungsprogramm verwendet werden, damit das Format der Nachrichten verständlich wird.

Weitere Informationen zu den von MQSeries generierten EMS-Ereignissen finden Sie in „Anhang P. In MQSeries for Compaq NSK verwendete EMS-Ereignisschablone“ auf Seite 413.

Kapitel 11. Transaktionale Unterstützung und transaktionale Nachrichtenübertragung

Bei Anwendungen, die die Schnittstelle MQI (Message Queue Interface) verwenden, können PUT- und GET-Vorgänge an Synchronisationspunkten ausgeführt werden. In MQSeries for Compaq NSK gibt es zwei Vorgänge in Zusammenhang mit Transaktionen:

- Festschreiben (Commit): Beim Abschluss einer Transaktion werden die Änderungen an der Datenbank aufgezeichnet und übernommen. Geschützte Ressourcen werden anschließend freigegeben.
- Zurücksetzen (Backout): Bei diesem Vorgang werden alle Änderungen, die im Verlauf der aktuellen Arbeitseinheit mit Wiederherstellung vorgenommen wurden, wieder rückgängig gemacht. Anschließend kann eine neue Arbeitseinheit mit Wiederherstellung bzw. eine neue Arbeitseinheit gestartet werden.

Festschreibungen und Rücksetzvorgänge gehören zur TM/MP- bzw. TMF-Transaktionsumgebung unter Compaq NSK. In MQSeries for Compaq NSK werden die Aufrufe MQPUT, MQGET und MQPUT1 standardmäßig über Synchronisationspunkte gesteuert, d. h., sie werden an einem Synchronisationspunkt ausgeführt. Dies bedeutet, dass eine TMF-Transaktion aktiv sein *muss*, wenn der MQI-Aufruf nicht fehlschlagen soll, außer es wurde von einer Anwendung explizit die Option NOSYNCPOINT angegeben.

Eine Anwendung leitet eine TM/MP-Transaktion mit der TM/MP-Prozedur BEGINTRANSACTION ein und schreibt die Transaktion mit ENDTRANSACTION fest bzw. setzt sie mit ABORTTRANSACTION zurück.

MQSeries for Compaq NSK unterstützt auch Produkte wie beispielsweise NonStop Tuxedo, die die Verwendung von Synchronisationspunkten über TM/MP ermöglichen.

NonStop-TM/MP (Transaktionsmanager) verwenden

MQSeries for Compaq NSK V5.1 stellt transaktionale Integrität über die Funktionen für das Transaktionsmanagement von Compaq NonStop-TM/MP zur Verfügung.

Das NonStop-TM/MP-Transaktionssystem ermöglicht transaktionalen Schutz, gemeinsame Zugriffe, Nachrichtenintegrität und stellt einen Objektkatalog zur Verfügung.

TM/MP-Transaktionen können PUT- und GET-Vorgänge in MQSeries mit der Aktualisierung von ENSCRIBE oder NonStop-SQL-Datenbankdateien, die von der Anwendung erstellt wurden, koordinieren.

TM/MP-Transaktionen werden von MQSeries transparent gehandhabt. Ist beim Aufruf einer MQI-Funktion eine TM/MP-Transaktion aktiv, werden alle PUT- und GET-Vorgänge, für die die Option SYNCPOINT aktiviert ist, innerhalb dieser Transaktion ausgeführt. Dies bedeutet, dass die Warteschlangen erst aktualisiert werden, nachdem die Transaktion festgeschrieben wurde. Im Fall von Fehlern stellt TM/MP sicher, dass für alle festgeschriebenen Transaktionen die Datenbankdateien entsprechend aktualisiert werden und alle nicht festgeschriebenen Transaktionen

TM/MP verwenden

wieder zurückgesetzt werden. Beim Zurücksetzen einer Transaktion wird ein Vorimage für die Datenbanksätze verwendet, um die Änderungen, die von der abgebrochenen Transaktion vorgenommen wurden, wieder rückgängig zu machen. Änderungen werden erst nach Abschluss einer Festschreibung (COMMIT-Operation) vorgenommen.

Ist für eine Benutzeranwendung eine Transaktion aktiv und versucht diese Anwendung, einen MQI-Aufruf außerhalb der Transaktion auszugeben, so setzt MQSeries die aktuelle, übernommene Transaktion vorübergehend aus, startet eine eigene Transaktion und schreibt diese fest, und setzt anschließend die zuvor ausgesetzte Transaktion wieder fort, bevor die Steuerung an die Benutzeranwendung zurückgegeben wird. Die Aktualisierung von Warteschlangen auf Grund von PUT- und GET-Vorgängen wird sofort vorgenommen.

Mit Hilfe von TM/MP-Transaktionen werden PUT- und GET-Vorgänge für nicht permanente und permanente Nachrichten koordiniert. In MQSeries for Compaq NSK wird eine besondere Schnittstelle zu TM/MP, nämlich *OpenTMF*, für diese Koordinierung verwendet.

Begrenzung innerhalb von Transaktionen

Das Dateisystem kann den Umfang permanenter Nachrichtendaten, die innerhalb einer einzelnen Transaktion eingereicht bzw. abgerufen werden können, begrenzen, indem die Anzahl der Datensperren für die physischen Dateien, die permanente Nachrichtendaten enthalten, begrenzt werden.

Der Standardwert des Dateisystems für die maximale Anzahl von Sperren ist 5000 pro Festplattendatenträger. Sie können diesen Wert mit SCF-Befehlen ändern. Mit dem folgenden Befehl wird z. B. die Anzahl der möglichen Sperren pro Transaktion und pro Festplattendatenträger auf 10000 festgelegt:

```
ALTER DISK $DISK01, MAXLOCKSPERTCP 10000
```

Die Größe und Anzahl der Nachrichten, die in Warteschlangenüberlaufdateien gespeichert werden (da ihre Größe unterhalb des Schwellenwertes für die Verwendung von Nachrichtenüberlaufdateien liegt), ist begrenzt.

Es wird geraten, den Schwellenwert für Nachrichtenüberlaufdateien auf höchstens 200 KB festzulegen. Bei dieser Nachrichtengröße können bei dem geltenden Standardhöchstwert für Datensatzsperren etwa 100 Nachrichten innerhalb einer Transaktion gespeichert werden; für die meisten Anwendungen ist dies ausreichend. Darüber hinaus wird durch die Verwendung von Nachrichtenüberlaufdateien bei Nachrichten dieser Größe die Verwendung eine deutlich bessere Leistung erzielt.

Außerdem wird die Verwendung von TM/MP-Prüfprotokollen deutlich reduziert, wenn anstelle von Warteschlangenüberlaufdateien Nachrichtenüberlaufdateien verwendet werden.

Weitere Informationen zu den Unterschieden zwischen Nachrichten- und Warteschlangenüberlaufdateien und zu den Vorteilen von Nachrichtenüberlaufdateien finden Sie unter „Nachrichtenüberlaufdateien“ auf Seite 231.

Operationen mit Option NOSYNCPOINT für permanente Nachrichten ausführen

Für permanente Nachrichten sind intern vom Warteschlangenserver gestartete TM/MP-Transaktionen erforderlich, damit die ENSCRIBE-Dateien mit den Nachrichtendaten aktualisiert werden. Für das NSK-Dateisystem ist die Anzahl der Transaktionen, die von einem Prozess gleichzeitig gestartet werden können, auf 100 begrenzt. Ein einzelner Warteschlangenserver kann daher höchstens 100 PUT- und GET-Operationen mit der Option NOSYNCPOINT für permanente Nachrichten gleichzeitig unterstützen. Dies ist nur möglich, wenn mehrere umfangreiche Nachrichten (mit mehr als 32 KB) gleichzeitig von ein und demselben Warteschlangenserver in Warteschlangen eingereicht bzw. aus Warteschlangen abgerufen werden.

In diesem Fall wird der MQPUT- oder MQGET-Vorgang mit dem Ursachencode MQRC_SYNCPOINT_LIMIT_REACHED beendet. Sie sollten die Warteschlangen anderen Warteschlangenservern zuordnen, um die Verarbeitung auf mehrere Prozesse zu verteilen, oder die Anwendungen ändern, so dass sie andere Warteschlangen in verschiedenen Warteschlangenservern verwenden.

Operationen mit SYNCPOINT für nicht permanente Nachrichten

Da nicht permanente Nachrichten im Speicher und nicht in überprüften Festplattendateien gespeichert werden, ist kein Speicherplatz für Prüfprotokolle erforderlich. MQSeries verwendet eine interne Schnittstelle zu TM/MP, um die Verfügbarkeit nicht permanenter Nachrichten zu steuern, die an einem Synchronisationspunkt eingereicht oder abgerufen werden. Eine Operation mit der Option SYNCPOINT kann sowohl permanente als auch nicht permanente Nachrichten einschließen; MQSeries stellt sicher, dass bei Abschluss der TM/MP-Transaktion die Operationen für alle nicht permanenten Nachrichten zu demselben Zeitpunkt wie für die permanenten Nachrichten logisch festgeschrieben bzw. zurückgesetzt werden (je nachdem, ob die Transaktion erfolgreich war oder nicht).

Konfigurationsvoraussetzungen für TM/MP und MQSeries for Compaq NSK

Ihr NSK-System muss so konfiguriert werden, dass TMF- bzw. TM/MP-Prüfungen für alle Datenträger aktiviert sind, die WS-Manager oder Warteschlangen enthalten werden. Mit dem TMFCOM-Befehl `status datavols` können Sie feststellen, ob Prüfungen für die Datenträger Ihres Systems aktiviert sind oder nicht. (Für die Verwendung von TMFCOM-Befehlen müssen Sie als Benutzer SUPER.SUPER angemeldet sein.) Die TMF-Prüfprotokolle, die für die WS-Managerdatenträger konfiguriert werden, müssen so groß sein, dass sie auch bei großen Nachrichten und in Zeiten hoher Auslastung aller WS-Manager, die diese Datenträger verwenden, ausreichend sind.

Da Anwendungen, die sich nicht richtig verhalten, zu langen Transaktionen führen können, sollte das TMF-System so konfiguriert werden, dass Transaktionen mit einer langen Ausführungszeit automatisch abgebrochen werden. Die Größe der Protokolldatei und die Begrenzung der Ausführungszeit von Transaktionen sind Parameter, die von der jeweiligen Anwendung abhängen. Das Prüfprotokoll für MQSeries muss nicht so konfiguriert werden, dass ein Speicherauszug auf Band erfolgt.

Überwachung

Wenn MQSeries aktiv ist, können Sie über die TMFCOM-Schnittstelle den TMF-Status überwachen. Mit den Befehlen `status tmf` und `status datavols` können Sie den allgemeinen TMF-Status und den Status der einzelnen Datenträger abrufen.

Das Systemereignisprotokoll (EMS) sollte ebenfalls auf kritische TM/MP-Ereignisse überwacht werden, die auf potenzielle Probleme in TM/MP hinweisen, die sich nachteilig auf MQSeries oder die Anwendungen, die mit MQSeries arbeiten, auswirken. TM/MP ist eine kritische Ressource für MQSeries, deren Betrieb gewährleistet sein muss, damit MQSeries problemlos arbeiten kann.

Größe für Prüfprotokolle

Der für die TM/MP-Prüfprotokolle erforderliche Speicherplatz kann wie folgt berechnet werden:

- Speicherbereich für Prüfprotokolle ist nur für Operationen in Zusammenhang mit permanenten Nachrichten erforderlich (PUT-Vorgänge und GET-Vorgänge, bei denen die Nachrichten aus der Warteschlange gelöscht werden).
- Der Speicherbereich für das Prüfprotokoll sollte dem Umfang der gesamten Nachrichtendaten plus 1500 Bytes entsprechen.
- Für Operationen in Zusammenhang mit permanenten Nachrichten, deren Größe über dem für Nachrichtenüberlaufdateien angegebenen Schwellenwert liegen, ist ein Prüfprotokollspeicher von lediglich 4 KB pro PUT- oder GET-Vorgang erforderlich, unabhängig von der Größe der Nachricht.

Konfiguration des Ressourcenmanagers

Bei der Konfiguration der internen TM/MP-Schnittstelle muss der Umfang der Transaktionen berücksichtigt werden, der voraussichtlich unter Verwendung von MQSeries verarbeitet wird. Die MQSeries-Warteschlangenserver übernehmen für das TM/MP-Subsystem die Rolle des Ressourcenmanagers, und es gibt verschiedene Schwellenwerte und Grenzwerte im TM/MP-Subsystem, die auch die Ressourcenmanager betreffen. Die erforderliche Konfiguration hängt von der Anzahl der Warteschlangenserver, die verwendet werden, von der Verteilung dieser Warteschlangenserver auf die CPUs und von der Anzahl der SYNCPOINT-Operationen (d. h. Operationen, für die die Option SYNCPOINT angegeben wurde), die gleichzeitig aktiv sein können, ab. Mit dem TMFCOM-Befehl `ALTER BEGINTRANS` können die Werte wie folgt geändert werden:

RMOPENPERCPU Hier sollte mindestens die doppelte Anzahl Warteschlangenserver angegeben werden, die in einer einzelnen CPU aktiv sein werden. Der Standardwert von 128 ist in der Regel ausreichend.

BRANCHESPERRM Hier sollte mindestens die Anzahl von SYNCPOINT-Operationen angegeben werden, die maximal von einem einzelnen Warteschlangenserver verarbeitet werden. Der Standardwert von 128 ist in der Regel nicht ausreichend. Der Wert kann aber maximal auf 1024 erhöht werden. Ebenso besteht die Möglichkeit, Warteschlangen anderen Warteschlangenservern zuzuordnen, um so die Anzahl der SYNCPOINT-Operationen, die von einem einzelnen Warteschlangenserver ausgeführt werden, zu reduzieren.

Änderungen an diesen beiden Parametern werden erst wirksam, nachdem das TM/MP-Subsystem gestoppt und anschließend neu gestartet wurde.

Fehlerbehebung

Wenn EMS-Ereignisse oder FFST_Berichte melden, dass BEGINTRANSACTION-Befehle von TMF inaktiviert wurden, so deutet dies in der Regel daraufhin, dass das Prüfprotokoll voll ist. Dieser Fall tritt ein, wenn das Prüfprotokoll zu klein ist oder eine nicht richtig ausgeführte Anwendung zu einer langen Transaktion geführt hat, die von TMF nicht rechtzeitig beendet wurde.

In diesem Fall können Sie zu einer der folgenden Maßnahmen greifen:

- Vergrößern Sie das Prüfprotokoll.
- Ermitteln Sie die Ursache für die lange Transaktion, und beheben Sie sie.
- Konfigurieren Sie TMF so, dass lange Transaktionen bereits nach einer kurzen Zeit beendet werden.

Wenn EMS-Ereignisse und FFST-Berichte melden, dass TMF nicht aktiv ist, so deutet dies auf Fehler in der TMF-Konfiguration hin, die behoben werden müssen, bevor der WS-Manager erneut gestartet wird. Für einen problemlosen Betrieb der MQSeries-WS-Manager muss TMF richtig arbeiten. Ohne TMF gehen Nachrichten zwar nicht verloren und werden auch nicht fehlerhaft, der WS-Manager kann ohne TMF jedoch nicht arbeiten.

Kapitel 12. Wiederherstellung und Neustart

Ein Nachrichtenübertragungssystem stellt sicher, dass eingehende Nachrichten an ihre Zieladressen zugestellt werden. Darüber hinaus muss es die Möglichkeit geben, Nachrichten im System zu verfolgen und bei Ausfällen wiederherzustellen.

MQSeries for Compaq NSK stellt mit Hilfe von Compaq NonStop Transaction Manager (TM/MP) sicher, dass keine permanenten Nachrichten verloren gehen. TM/MP ermöglicht Schutz durch Transaktionen sowie die Konsistenz der Warteschlangendateien und die Möglichkeit, diese wiederherzustellen.

Darüber hinaus stellt MQSeries for Compaq NSK mit Hilfe der NonStop-Prozesspaartechnologie sicher, dass auch nicht permanente Nachrichten bei Ausfällen nicht verloren gehen. Die Warteschlangenserver, die für die Nachrichtenspeicherung zuständig sind, speichern Prüfpunktdateien für nicht permanente Nachrichten in ihrem Backup-Prozess, der in einer anderen CPU aktiv ist.

Das Prüfpunktverfahren (d. h. die Aufzeichnung von Prüfpunktdateien) für nicht permanente Nachrichten wird mit dem Befehl **altmqfls** für die einzelnen Warteschlangen aktiviert. Standardmäßig ist für Warteschlangen das Prüfpunktverfahren für nicht permanente Nachrichten aktiviert.

Das TM/MP-Subsystem verwaltet die für die aktuellen Transaktionen und die Datenbankkonsistenz erforderlichen Operationen, und zwar sowohl Benutzer- als auch die MQSeries-Operationen, und macht sie für Benutzer- und Anwendungsprogramme transparent.

Bei einer Wiederherstellung wird für den WS-Manager der Zustand wiederhergestellt, in dem er sich befand, als er gestoppt wurde. Alle nicht abgeschlossenen Transaktionen werden zurückgesetzt, wobei aus den Warteschlangen alle Nachrichten entfernt werden, die zu dem Zeitpunkt, an dem der WS-Manager gestoppt wurde, nicht festgeschrieben waren. Alle permanenten Nachrichten werden wiederhergestellt; nicht permanente Nachrichten gehen dabei jedoch verloren.

In den folgenden Abschnitten dieses Kapitels sollen die Konzepte der Wiederherstellung und des Neustarts ausführlicher beschrieben werden; anschließend werden Hinweise zur Fehlerbehebung gegeben. Auf folgende Punkte wird eingegangen:

- „Fehlertoleranz und Fehlerbehebung“
- „MQSeries-Daten sichern und zurückschreiben“ auf Seite 190

Fehlertoleranz und Fehlerbehebung

Bei einer entsprechender Konfiguration von MQSeries Version 5.1 und der Software und Hardware des Compaq NSK-Systems (wenn Sie beispielsweise alle Komponenten wie von Compaq vorgeschrieben als redundante oder gespiegelte Einheiten bzw. Prozesspaare konfiguriert haben) führt der Ausfall einer einzelnen Hardware- oder Softwarekomponente weder zu Datenverlusten, Datenduplizierungen oder fehlerhaften Daten noch zum permanenten Ausfall von Systemfunktionen, für deren Wiederherstellung ein Bedieneringriff erforderlich ist. In MQSeries for Compaq NSK V5.1 ist die Wiederherstellung nach dem einmaligen Ausfall einer Software- oder Hardwarekomponente möglich, wobei die Datenintegrität wie oben beschrieben gewahrt bleibt.

Fehlertoleranz

Wiederholte Ausfälle (wie beispielsweise Schleifen bei Wiederherstellungen nach Fehlern) ein und derselben Softwarekomponente werden abgefangen, wenn die für Ausfälle vorgegebene maximale Anzahl überschritten wird. In diesen Fällen oder bei Ausfällen mehrerer Komponenten kann das MQSeries-Produkt die Integrität der Warteschlangen nicht mehr erhalten.

Weitere Informationen zur Konfiguration eines WS-Managers im Hinblick auf Datenintegrität und Verfügbarkeit finden Sie in „Kapitel 16. Datenintegrität und Verfügbarkeit“ auf Seite 237.

MQSeries-Daten sichern und zurückschreiben

Im Normalfall werden Sie in regelmäßigen Abständen eine Sicherungskopie Ihrer WS-Managerdaten erstellen, um gegen mögliche Schäden infolge von Hardwarefehlern geschützt zu sein.

MQSeries-Daten sichern

So sichern Sie die Daten eines WS-Managers:

1. Stellen Sie sicher, dass der WS-Manager nicht aktiv ist.

Wenn der WS-Manager aktiv ist, stoppen Sie ihn mit dem Befehl **endmqm**.

Anmerkung: Wenn Sie versuchen, eine Sicherung für einen aktiven WS-Manager durchzuführen, ist diese möglicherweise nicht konsistent, weil Aktualisierungen stattfinden können, während die Dateien kopiert werden.

2. Ermitteln Sie die Datenträger und Teildatenträger, in denen die WS-Managerdaten gespeichert werden.

Informationen zu diesen Verzeichnissen finden Sie zum Beispiel in den Konfigurationsdateien. Weitere Informationen finden Sie in „Kapitel 13. Konfigurationsdateien“ auf Seite 193.

Anmerkung: Wenn Sie Mühe haben, die Namen zu erkennen, so liegt dies daran, dass diese umgewandelt wurden, um sicherzustellen, dass sie mit der Plattform, auf der MQSeries eingesetzt wird, kompatibel ist. Weitere Informationen zu Namensumwandlungen finden Sie unter „Datenträgerstruktur“ auf Seite 64.

3. Erstellen Sie Kopien aller Teildatenträger, die WS-Managerdaten und WS-Managerprotokolldateien enthalten.

Stellen Sie sicher, dass dabei keine Dateien übersehen werden.

MQSeries zurückschreiben

Um die Sicherung der Daten eines WS-Managers zurückzuschreiben, müssen Sie Folgendes tun:

1. Stellen Sie sicher, dass der WS-Manager nicht aktiv ist.
2. Ermitteln Sie die Teildatenträger, in denen die WS-Managerdaten gespeichert werden. Diese Information finden Sie in der Konfigurationsdatei.
3. Löschen Sie den Inhalt der Teildatenträger, in denen die Sicherungskopien gespeichert werden sollen.
4. Kopieren Sie die Sicherungsdaten des WS-Managers in die vorgesehenen Teildatenträger.

Überprüfen Sie die erstellte Verzeichnisstruktur, um sicherzustellen, dass alle erforderlichen Verzeichnisse vorhanden sind.

Überprüfen Sie, ob die Konfigurationsdateien von MQSeries und des WS-Managers konsistent sind, damit MQSeries in den richtigen Datenträgern nach den zurückgeschriebenen Daten sucht.

Wenn die Daten korrekt gesichert und zurückgeschrieben wurden, kann der WS-Manager gestartet werden.

Status- und Warteschlangenserver wiederherstellen und erneut starten

Bei den Status- und der Warteschlangenserverprozessen handelt es sich um Compaq NSK-Prozesspaare. Dies bedeutet, dass sie ihren Betrieb auch dann fortsetzen, wenn eine einzelne CPU oder der Prozess selbst ausfällt. Bei einem einzelnen Ausfall (z. B. bei Ausfall der CPU, die den primären Status- oder Warteschlangenserverprozess enthält, oder bei Ausfall des primären Status- oder Warteschlangenserverprozesses selbst) übernimmt der als Backup eingerichtete Status- oder Warteschlangenserverprozess die Rolle des primären Prozesses, ohne dass es zu Unterbrechungen in der Arbeit des WS-Managers kommt.

Bei Ausfall einer einzelnen Komponente sind daher keine besonderen Fehlerbehebungsmaßnahmen für den Status- oder den Warteschlangenserver erforderlich. Wenn der Backup-Server vom primären Server erneut gestartet werden musste oder Backup-Server die Rolle des primären Server übernommen hat, zeichnet der Status- bzw. Warteschlangenserver eine entsprechende Nachricht im Ausgangsterminal und in der Nachrichtenprotokolldatei auf.

Bei schwerwiegenderen Fehlern (z. B. bei Ausfällen in der Umgebung, die einen Start des redundanten oder primären Status- bzw. Warteschlangenservers verhindern) versucht der TS/MP-PATHMON-Prozess bis zu 10 Mal, den Status- oder Warteschlangenserver zu starten.

Der Status- oder Warteschlangenserver greift nur auf TM/MP-geschützte Datenbanken zu; bei Ausfällen, die sich auf die Platteneinheiten auswirken, bieten TM/MP und das DP2-Festplattensubsystem einen zuverlässigen Schutz.

Der Status- oder Warteschlangenserver kann bei Bedarf mit TS/MP-PATHCOM-Befehlen gezielt gestoppt und erneut gestartet werden, obwohl dies im Normalfall nicht erforderlich ist. Mit dem Befehl **strmqm** werden automatisch alle Status- oder Warteschlangenserverklassen gestartet, deren Namen mit MQS-STATUS oder MQS-QUEUE beginnen. Beim Herunterfahren eines WS-Managers erfolgt eine koordinierte Beendigung der aktiven und Status- und Warteschlangenserverklassen; erst nachdem alle aktiven WS-Managerverbindungen beendet sind, werden alle Status- oder Warteschlangenserver beendet, die an diesen Verbindungen beteiligt waren.

Beim Ausfall des primären Prozesses eines NonStop-Prozesspaares bricht Compaq NSK Transaktionen in bestimmten Fällen ab. Dies kann zum Fehlschlagen von MQPUT- oder MQGET-Vorgängen führen, die beim Ausfall des primären Prozesses gerade ausgeführt wurden. Weitere Informationen finden Sie unter „Kapitel 16. Datenintegrität und Verfügbarkeit“ auf Seite 237.

Status- und Warteschlangenserver wiederherstellen und erneut starten

Fehlerbehebung mit RDF

Kann das primäre System nach einem schwerwiegenden Fehler nicht mehr eingesetzt werden, sollten Sie wie folgt einen WS-Manager auf dem Ausweichsystem in Betrieb nehmen:

1. Stellen Sie sicher, dass die Datenbanken von RDF (Remote Database Duplication Facility) aktualisiert wurden.
2. Setzen Sie mit FUP die Prüfmarkierung für folgende Dateien:
 - a. Alle Dateien im Teildatenträger <WS-Manager>M.
 - b. Alle Dateien im Teildatenträger <WS-Manager>D, mit Ausnahme von QMINI, AMQRFNxx, CCSIDMEM, QMINIMEM, STABLE, UMQSINI, PATHCTL, TRACEOPT und SHUTDOWN.
 - c. Alle Dateien im Teildatenträger <WS-Manager>S.
3. Konfigurieren Sie <WS-Manager>D als Standardteildatenträger. Starten Sie PATHMON manuell, führen Sie PATHCOM aus, und laden Sie die PATHWAY-Konfiguration für den WS-Manager.
4. Geben Sie für alle Objekte den INFO-Befehl aus.
 - a. Stellen Sie sicher, dass alle Exemplare des Knotennamens als "*" angegeben werden **oder** den richtigen Namen für das Ausweichsystem haben.
 - b. Stellen Sie sicher, dass die den Serverklassen zugeordneten CPU-Nummern für das Ausweichsystem noch gültig sind.
 - c. Stellen Sie sicher, dass das angegebene Ausgangsterminal und die angegebenen Ausgabedateien für das Ausweichsystem gültig sind. Ändern Sie sie gegebenenfalls.
 - d. Stellen Sie sicher, dass der angegeben Name des alternativen TCP/IP-Prozesses für dieses System gültig ist.
 - e. Ist das Ausweichsystem nicht für den RDF-Betrieb konfiguriert, entfernen Sie aus den Serverklassendefinitionen für die Ausführungs-Controller alle eventuell vorhandenen Einstellungen der Umgebungsvariablen MQRDF.
 - f. Überprüfen Sie die PATHWAY-Konfiguration, schreiben Sie sie mit dem Befehl **shutdown2** auf die Platte zurück, und beenden Sie PATHCOM. Wurde der Name des Ausgangsterminals geändert, müssen Sie die Konfigurationsdatei (QMINI) des WS-Managers entsprechend der PATHWAY-Konfiguration ebenfalls ändern. Bei Bedarf müssen Sie in dieser Datei auch die Konfiguration der Ports für die TCP/IP-Empfangsprogramme ändern.
 - g. Starten Sie den WS-Manager mit dem Befehl **strmqm**.
 - h. Überprüfen Sie mit dem Befehl **runmqsc** die Kanalkonfiguration, und passen Sie sie gegebenenfalls an.
 - i. Wenn Sie versuchen, dieselben Kanäle wie zuvor zu starten, muss unter Umständen auch die Kanalkonfiguration in den fernen WS-Managern geändert werden, es sei denn, das Ausweichsystem kann so konfiguriert werden, dass dieselbe IP-Adresse für den Host-Namen, für TCP/IP-Kanäle oder dieselben SNAX/APC- und ICE-Ressourcennamen (z. B. Prozessnamen, LU-Name usw.) für SNA-Kanäle verwendet werden.
 - j. Sie sollten auf Fehler bei der Kanalsynchronisation oder der Reihenfolge vorbereitet sein, besonders dann, wenn bei Auftreten des Fehlers Kanäle des Primärsystems aktiv waren. Die Verwendung von RDF bietet keine Gewähr dafür, dass das Ausweichsystem aktuell, d. h. auf demselben Stand wie das primäre System ist; bei einem Komplettausfall kann es daher zu Datenverlusten kommen. Sie können die Gefahr von Datenverlusten allerdings minimieren, indem Sie sicherstellen, dass die RDF-Konfiguration in der Lage ist, die Datenbankaktualisierungen, die in Zusammenhang mit dem Nachrichtenfluss anfallen, zu handhaben.

Kapitel 13. Konfigurationsdateien

In MQSeries for Compaq NSK sind die grundlegenden Konfigurationsdaten in den so genannten *Konfigurationsdateien* enthalten. In diesem Kapitel werden diese Konfigurationsdateien beschrieben, und es wird erläutert, wie sie dazu verwendet werden können, die Arbeitsweise der WS-Manager zu ändern. Dieses Kapitel ist in folgende Abschnitte gegliedert:

- „Konfigurationsdateien - Beschreibung“
- „MQSeries-Konfigurationsdatei (MQSINI)“
- „Konfigurationsdatei der WS-Manager (QMINI)“ auf Seite 195
- „Konfigurationsdateien editieren“ auf Seite 203

Konfigurationsdateien - Beschreibung

In Konfigurationsdateien werden die für die einzelnen WS-Manager und die für MQSeries auf dem Knoten insgesamt möglichen Einstellungen definiert. Diese Dateien werden als *ini*-Dateien oder *Zeilengruppendateien* bezeichnet. Konfigurationsdateien enthalten eine oder mehrere Zeilengruppen; bei diesen Zeilengruppen handelt es sich um mehrere zusammengehörige Zeilen innerhalb der Datei, die eine Funktion beschreiben oder einen Teil des Systems definieren. So gibt es beispielsweise eigene Zeilengruppen für Protokolle, Kanäle und installierbare Services.

Konfigurationsdateien können automatisch geändert werden, und zwar mit Hilfe von Befehlen, die die Konfiguration der WS-Manager auf dem Knoten ändern, sie können aber auch manuell bearbeitet werden. Im Allgemeinen wird jedoch von einer manuellen Bearbeitung der Dateien abgeraten, wenn die WS-Manager aktiv sind.

Es gibt zwei Arten von Konfigurationsdateien:

- Die *MQSeries-Konfigurationsdatei* (MQSINI), in der die Einstellungen von MQSeries auf dem Knoten insgesamt definiert sind. In der Regel ist für jeden Knoten eine MQSeries-Konfigurationsdatei vorhanden.
- Die *Konfigurationsdateien der WS-Manager* (QMINI), in der die Einstellungen für den jeweiligen WS-Manager definiert sind. Es gibt eine Konfigurationsdatei für jeden WS-Manager auf dem Knoten.

MQSeries-Konfigurationsdatei (MQSINI)

Die MQSeries-Konfigurationsdatei MQSINI enthält Informationen, die alle WS-Manager auf einem MQSeries-Installationsknoten betreffen. Sie wird bei der Installation automatisch erstellt. Insbesondere werden mit dieser Konfigurationsdatei die Daten der einzelnen WS-Manager ermittelt. Sie befindet sich im Teildatenträger ZMQSSYS (standardmäßig in \$SYSTEM.ZMQSSYS.MQSINI). Für Systeme, auf denen die MQSeries-Konfigurationsdatei nicht unter dem Standardnamen oder nicht im Standardverzeichnis gespeichert ist, wird die Umgebungsvariable MQMACHINIFILE zur Verfügung gestellt.

MQSeries-Konfigurationsdatei

Inhalt der MQSeries-Konfigurationsdatei

Die Datei MQSINI enthält die systemweiten Standardeinstellungen, die Namen der WS-Manager und des Standard-WS-Managers sowie den Verzeichnispfad der diesen WS-Manager zugeordneten Dateien. Die Datei MQSINI enthält Zeilengruppen wie die folgenden:

AllQueueManagers

Gibt die Adressen und Datenträger der Dateien an, die die Installation insgesamt betreffen.

DefaultQueueManager

Gibt den Standard-WS-Manager der Installation an. Dieser WS-Manager verarbeitet die MQSC-Befehle, wenn kein bestimmter WS-Manager angegeben wurde. Wird ein neuer Standard-WS-Manager definiert, wird diese Zeilengruppe automatisch aktualisiert. Wurde versehentlich ein falscher Standard-WS-Manager erstellt und soll der ursprüngliche WS-Manager wiederhergestellt werden, muss diese Zeilengruppe manuell geändert werden.

QueueManager

Die Datei enthält für jeden vorhandenen WS-Manager eine Zeilengruppe dieser Art. Sie gibt den Namen des WS-Managers und den Dateipfad der ihm zugeordneten Dateien an. Die Namen dieser Dateien leiten sich vom Namen des WS-Managers ab, werden allerdings umgewandelt, wenn es sich beim Namen des WS-Managers nicht um einen gültigen Dateinamen handelt.

Ein Beispiel für die Datei MQSINI ist in Abb. 42 zu sehen.

```
#####  
** Modulname : MQSINI **  
** Typ : MQSeries-INI-Datei für das gesamte System **  
** Funktion : Definition der Konfigurationsdaten aller WS-Manager **  
** **  
#####  
** Hinweise : **  
** 1) Diese Datei enthält die Konfigurationsdaten aller WS-Manager **  
** **  
#####  
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)
```

Abbildung 42. MQSeries-Konfigurationsdatei MQSINI - Beispiel. Diese Datei wird bei der Installation mit den Datenträgern und Teildatenträgern initialisiert, die Sie angegeben haben.

Anmerkung: Da die MQSeries-Konfigurationsdatei zum Lokalisieren der Daten des WS-Managers verwendet wird, kann eine fehlende oder falsche Konfigurationsdatei zur Folge haben, dass einige oder alle MQSeries-Befehle fehlschlagen. Auch können Anwendungen nur Verbindungen mit WS-Managern herstellen, die in der MQSeries-Konfigurationsdatei definiert sind.

Konfigurationsdatei der WS-Manager (QMINI)

Die Konfigurationsdatei eines WS-Managers (QMINI) enthält alle Angaben zu einem bestimmten WS-Manager. Es gibt für jeden WS-Manager eine Konfigurationsdatei. Diese Datei wird bei Erstellung eines WS-Managers automatisch erstellt.

Diese Datei befindet sich in dem Teildatenträger, der dem WS-Manager zugeordnet ist. So lauten beispielsweise Pfad und Name der Konfigurationsdatei für einen WS-Manager mit dem Namen QMNAME wie folgt: \$VOLUME.QMNAME.QMINI.

Anmerkung: Der Name des WS-Managers darf maximal 48 Zeichen lang sein. Der Name des Teildatenträgers wird vom Namen des WS-Managers abgeleitet. Dieses Verfahren wird als *Namensumwandlung* bezeichnet und stellt sicher, dass Namen gültig und eindeutig sind.

Inhalt der Konfigurationsdatei eines WS-Managers

Die Konfigurationsdatei eines WS-Managers (QMINI) kann folgende Zeilengruppen enthalten:

Configuration

Diese Zeilengruppe definiert die globale Konfiguration des WS-Managers.

Die folgenden Einträge können geändert werden:

- DefaultQueueServerName
- DefaultStatusServerName
- HomeTerminalName
- PathmonProcName
- EMSCollectorName
- MinIdleMCALU62Responders
- MinIdleMCATCPResponders
- MinIdleMCAcallers
- MinIdleLQMAgents

Weitere Informationen zu diesen Einträgen finden Sie in „WS-Managermerkmale ändern“ auf Seite 50. Alle anderen Einträge in dieser Zeilengruppe dürfen nicht verändert werden.

DefaultProcess

Diese Zeilengruppe gibt die für die MQSeries-Prozesse verwendeten Standardwerte an. Die Einträge in dieser Zeilengruppe dürfen nicht geändert werden.

Konfigurationsdatei der WS-Manager

ECBoss

Diese Zeilengruppe definiert die Konfiguration des Managerprozesses für die Ausführungs-Controller in MQSeries. Der Eintrag `ExpectedNumECs` gibt die Anzahl der Ausführungs-Controller-Prozesse für den betreffenden WS-Manager an. Dieser Wert muss der PATHWAY-Konfiguration des WS-Managers entsprechen. Weitere Informationen finden Sie in „WS-Managermerkmale ändern“ auf Seite 50. Alle anderen Einträge in dieser Zeilengruppe dürfen nicht verändert werden.

EC Die Einträge `MCAAgentPriority` und `LQMAgentPriority` in der Zeilengruppe **EC**, die die Priorität der Agentenprozesse angeben, können geändert werden. Weitere Informationen finden Sie in „WS-Managermerkmale ändern“ auf Seite 50. Alle anderen Einträge in dieser Zeilengruppe dürfen nicht verändert werden.

Die folgenden Zeilengruppen legen die betriebsspezifischen Parameter der einzelnen MQSeries-Prozessarten fest. Die Werte dieser Parameter werden in der Regel nicht geändert. Siehe hierzu jedoch „Konfigurationsdatei für WS-Manager - Beispiel“ auf Seite 198.

MCACaller

MCATCPResponder

MCALU62Responder

MQIServer

LQMAgent

ChannelInitiator

TCPListener

Authority

Gibt das empfohlene Verfahren zum Aktivieren und Inaktivieren des OAM für einen WS-Manager an. Wird der Eintrag `MQAUTH` auf 'On' bzw. 'Off' gesetzt, kann der OAM aktiviert bzw. inaktiviert werden, ohne dass die Zeilengruppen **Service** und **ServiceComponent** hinzugefügt bzw. entfernt werden müssen.

Service

Gibt den Namen eines installierbaren Services sowie die Anzahl seiner Eingangspunkte an. Diese Zeilengruppe ist für jeden Service vorhanden. Folgende Services stehen zur Verfügung:

- Berechtigungsservice
- Namensservice

Der OAM (Object Authority Manager, Berechtigungsmanager) ist standardmäßig aktiviert, d. h., die Zeilengruppe **Service** und die zugehörige Zeilengruppe **ServiceComponent** sind standardmäßig in der Datei `QMINI` enthalten.

Sie können den OAM auf einfache Weise inaktivieren, indem Sie den Eintrag `MQAUTH` in der Zeilengruppe **Authority** auf 'Off' setzen und den WS-Manager erneut starten. Darüber hinaus haben Sie noch folgende Möglichkeiten:

1. Löschen Sie den WS-Manager mit dem Befehl `dltmqm`.
2. Setzen Sie die Umgebungsvariable `MQSNOAUT`, und erstellen Sie den WS-Manager mit dem Befehl `crtmqm` erneut.

3. Löschen Sie die Zeilengruppen für den Berechtigungsservice aus der Datei QMINI.

Die Zeilengruppe für den Namensservice muss der Datei QMINI manuell hinzugefügt werden, wenn dieser Service aktiviert werden soll.

ServiceComponent

Diese Zeilengruppen definieren die einem bestimmten Service zugeordnete Servicekomponente. Für die einzelnen Services können mehrere Zeilengruppen dieser Art vorhanden sein, die aber jeweils der entsprechenden Zeilengruppe **Service** zugeordnet sein müssen. Weitere Informationen finden Sie im Handbuch *MQSeries Programmable System Management*.

TuningParameters

Diese Zeilengruppe gibt die internen Optimierungsparameter an, die von den lokalen WS-Manageragenten verwendet werden. Diese Einstellungen sollten nicht geändert werden.

Channels

Diese Zeilengruppe enthält Informationen zu den Kanälen. Sie gibt die Anzahl der Kanäle an, die maximal definiert werden kann (**MaxChannels**), mit einem weiteren Eintrag kann die Anzahl der Kanäle, die gleichzeitig aktiv sein können, begrenzt werden (**MaxActiveChannels**). Der für **MaxActiveChannels** angegebene Wert darf nicht über dem von **MaxChannels** liegen. Die Zeilengruppe **Channels** enthält außerdem den Eintrag **ChanInitDiscInterval**, mit dem die Leistung des Kanalinitiators angepasst werden kann. Weitere Informationen zu diesen Einträgen finden Sie in „WS-Managermerkmale ändern“ auf Seite 50. Alle anderen Einträge in dieser Zeilengruppe dürfen nicht verändert werden.

Weitere Informationen zu Kanälen finden Sie im Handbuch *MQSeries Intercommunication*.

TCPConfig

Enthält die Konfigurationsparameter für das Netzprotokoll. Diese Zeilengruppe überschreibt die Standardwerte für Kanäle. Diese Zeilengruppe enthält nur Einträge für geänderte Standardwerte.

Die Werte für **TCPListenerPort** werden vom Empfangsprogramm überschreiben, wenn in der Umgebung des Empfangsprogrammprozesses der Parameter **MQLISTENPORTNUM** vorhanden ist.

Weitere Informationen finden Sie im Handbuch *MQSeries Intercommunication*.

Informationen zu Änderungen an den Einträgen **TCPPort**, **TCPNumListenerPorts** und **TCPListenerPort** finden Sie unter „WS-Managermerkmale ändern“ auf Seite 50.

Konfigurationsdatei der WS-Manager

Konfigurationsdatei für WS-Manager - Beispiel

Abb. 43 zeigt ein Beispiel für die Konfigurationsdatei eines WS-Managers (QMINI).

```
#####  
#* Modulname : QMINI *#  
#* Typ : Konfigurationsdatei für MQSeries-WS-Manager *#  
# Funktion : Konfiguration eines WS-Managers definieren *#  
#* *#  
#####  
#* Hinweise : *#  
#* 1) Diese Datei definiert die Konfiguration des WS-Managers. *#  
#* *#  
#####  
Configuration:  
 PathmonProcName=$p01p  
 DefaultStatusServerName=$p01s  
 ServerClassName=MQS-ECBOSS  
 EMSCollectorName=$0  
 HomeTerminalName=$ZTN0.#PTY001C  
 ShutdownFileName=SHUTDOWN  
 TraceOptionsFileName=TRACEOPT  
 RuntimeFileName=RUNTIME  
 StatableFileName=STABLE  
 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
```

Abbildung 43. Konfigurationsdatei für einen WS-Manager (QMINI) - Beispiel (Teile- 1 von 6)

Konfigurationsdatei der WS-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=MLQU6RES
  MCAAgentPriority=165
  LQMAgentPriority=165
  StopProcessTimer=3000
  IdleProcessTimer=3000
```

Abbildung 43. Konfigurationsdatei für einen WS-Manager (QMINI) - Beispiel (Teile- 2 von 6)

Konfigurationsdatei der WS-Manager

```
MCAaller:
  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
```

Abbildung 43. Konfigurationsdatei für einen WS-Manager (QMINI) - Beispiel (Teile- 3 von 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=MLQMAG
  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   TCPLListener:
  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
```

Abbildung 43. Konfigurationsdatei für einen WS-Manager (QMINI) - Beispiel (Teile- 4 von 6)

Konfigurationsdatei der WS-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
```

Abbildung 43. Konfigurationsdatei für einen WS-Manager (QMINI) - Beispiel (Teile- 5 von 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
```

Abbildung 43. Konfigurationsdatei für einen WS-Manager (QMINI) - Beispiel (Teile- 6 von 6)

Konfigurationsdateien editieren

Sie können die Systemstandardeinstellungen über die Standardkonfigurationsdateien ändern. Zunächst sollten Sie jedoch eine Sicherungskopie der Konfigurationsdatei erstellen, die bei Bedarf wiederhergestellt werden kann; außerdem müssen Sie sicherstellen, dass die betroffenen WS-Manager gestoppt wurden.

In folgenden Fällen kann eine Bearbeitung der Konfigurationsdateien erforderlich sein:

- Eine Konfigurationsdatei ist verloren gegangen; sie kann unter Umständen anhand der Sicherungskopie wiederhergestellt werden.
- Die Verteilung der WS-Manager auf die CPUs muss geändert werden.
- Es muss ein anderer Standard-WS-Manager angegeben werden (beispielsweise wenn der vorhandene WS-Manager versehentlich gelöscht wurde).
- Sie haben eine entsprechende Anweisung vom IBM Support Center erhalten.

Weitere Informationen finden Sie in „WS-Managermerkmale ändern“ auf Seite 50.

Änderungen in Konfigurationsdateien implementieren

Wenn Sie eine Konfigurationsdatei ändern, werden die Änderungen nicht sofort vom WS-Manager implementiert. Änderungen an der MQSeries-Konfigurationsdatei (MQSINI) werden erst wirksam, nachdem die MQSeries-WS-Manager erstellt oder gestartet werden. Änderungen an der Konfigurationsdatei eines WS-Managers (QMINI) hingegen werden nach einem Start des WS-Managers wirksam. Ist der WS-Manager aktiv, während Sie die Änderungen vornehmen, müssen Sie den WS-Manager stoppen und erneut starten, damit das System die Änderungen erkennt.

Empfehlungen für Konfigurationsdateien

Bei der Erstellung eines neuen WS-Managers sollten Sie Folgendes ausführen:

- Erstellen Sie eine Sicherungskopie der MQSeries-Konfigurationsdatei MQSINI.
- Erstellen Sie eine Sicherungskopie für die Konfigurationsdatei eines neuen WS-Managers (QMINI).

Konfigurationsdateien editieren

Kapitel 14. Fehlerbestimmung

Dieses Kapitel enthält Informationen zur Fehlerbestimmung in MQSeries for Compaq NSK. Für eine Fehlerbestimmung sollten Sie die einzelnen Symptome aufzeichnen und anschließend ihre Ursache ermitteln.

Probleme mit der Systemleistung, die auf hardwarebedingte Einschränkungen zurückzuführen sind, können nicht sofort behoben werden. Wenn Sie Fehler im MQSeries-Code vermuten, sollten Sie sich mit dem IBM Support Center in Verbindung setzen. Dieses Kapitel enthält folgende Abschnitte:

- „Erste Überprüfungen“
- „Typische Programmierfehler“ auf Seite 209
- „Die nächsten Schritte“ auf Seite 210
- „Überlegungen zum Anwendungsdesign“ auf Seite 214
- „Auswirkung der Nachrichtenlänge“ auf Seite 214
- „Fehlerprotokolle“ auf Seite 219
- „Warteschlangen für nicht zustellbare Nachrichten“ auf Seite 222
- „Konfigurationsdateien und Fehlerbestimmung“ auf Seite 222
- „MQSeries-Trace verwenden“ auf Seite 222
- „First Failure Support Technology™ (FFST)“ auf Seite 223

Erste Überprüfungen

Folgende Komponenten kommen als Fehlerursache in Frage:

- MQSeries
- Ihr Netz
- eine Anwendung
- die Software des Compaq-Systems

Die folgenden Abschnitte enthalten eine Reihe von Fragen, die beim Aufspüren von Fehlern helfen können. Gehen Sie diese Fragen durch, und notieren Sie alles, was für Ihr Problem relevant sein könnte.

Lief MQSeries bis jetzt fehlerfrei?

Ist dies nicht der Fall, wurde MQSeries unter Umständen nicht richtig installiert. Überprüfen Sie anhand des Handbuchs *MQSeries for Compaq NSK Einstieg*, ob die einzelnen Installationsschritte richtig ausgeführt wurden.

Werden Fehlernachrichten angezeigt?

MQSeries erstellt Fehlerprotokolle mit Nachrichten über den Betrieb von MQSeries selbst, über alle gestarteten WS-Manager und über fehlerhafte Daten von den verwendeten Kanälen. Überprüfen Sie die Fehlerprotokolle auf Nachrichten, die dem jeweiligen Fehler zugeordnet werden können.

Weitere Informationen zum Inhalt und zum Verzeichnispfad der Fehlerprotokolle finden Sie unter „Fehlerprotokolle“ auf Seite 219.

Werden Rückkehrcodes mit Fehlererläuterungen angezeigt?

Wenn die Anwendung einen Rückkehrcode erhält, der auf einen Fehler bei der Ausführung eines MQI-Aufrufs hinweist, schlagen Sie die Beschreibung des betreffenden Rückkehrcodes im Handbuch *MQSeries Application Programming Reference* nach.

Können Sie den Fehler reproduzieren?

Wenn Sie den Fehler reproduzieren können, sollten überprüfen:

- Wurde er durch einen Befehl oder eine entsprechende Verwaltungsanforderung verursacht?
Wird die Operation ausgeführt, wenn sie auf eine andere Weise eingegeben wird? Wird der Befehl nur dann ausgeführt, wenn Sie ihn über die Befehlszeile eingeben, überprüfen Sie, ob der Befehlsserver noch aktiv ist und ob die Warteschlangendefinition `SYSTEM.ADMIN.COMMAND.QUEUE` unverändert ist.
- Wurde er durch ein Programm verursacht?
- Gibt es Anwendungen, die immer dann auf dem System aktiv sind, wenn der Fehler auftritt? Wenn ja, überprüfen Sie die Anwendung auf Fehler.
- Ist der Datenträger mit der Datenbank des WS-Managers durch TM/MP geschützt? Sie können dies feststellen, indem Sie den `TMFCOM`-Befehl `STATUS DATAVOLS` ausgeben.
- Wurden die erforderlichen TACL-Umgebungsparameter wie unter „Anhang C. TACL-Umgebungsvariablen für MQSeries for Compaq NSK setzen“ auf Seite 329 beschrieben hinzugefügt? Geben Sie **PARAM** an der TACL-Eingabeaufforderung ein, um die momentan gesetzten Parameterwerte anzuzeigen.
- Ist das TM/MP-Prüfprotokoll groß genug? (Siehe „Kapitel 11. Transaktionale Unterstützung und transaktionale Nachrichtenübertragung“ auf Seite 183.) Geben Sie den `TMFCOM`-Befehl `INFO AUDITTRAIL` ein, um die Konfiguration des Prüfprotokolls anzuzeigen.
- Erlaubt der Dateizugriffsschutz von Compaq den Zugriff auf die Datenbankdateien im Teildatenträger mit den WS-Managerdaten? Damit auch Benutzer, die nicht zur Gruppe MQM gehören, Zugriff erhalten, muss für die Dateien zumindest der Lesezugriff möglich sein. Sind die Attribute für den Compaq-Dateizugriffsschutz so gesetzt, dass kein Zugriff möglich ist, wird auf eine `MQCONN`-Anforderung hin der Rückkehrcode 2035 (`MQRC_NOT_AUTHORIZED`) zurückgegeben.

Trat der Fehler erst auf, nachdem Änderungen vorgenommen wurden?

Berücksichtigen Sie beim Überprüfen der Änderungen, die seit dem letzten fehlerfreien Betrieb vorgenommen wurden, das MQSeries-System einschließlich der anderen Programme, mit denen es kommuniziert, sowie die Hardware und alle neuen Anwendungen. Überprüfen Sie auch, ob möglicherweise ohne Ihr Wissen eine neue Anwendung auf dem System ausgeführt wurde.

- Haben Sie Warteschlangendefinitionen geändert, hinzugefügt oder gelöscht?
- Haben Sie Kanaldefinitionen geändert oder hinzugefügt? Änderungen können an MQSeries-Kanaldefinitionen oder an zu Grunde liegenden Kommunikationsdefinitionen, die für eine Anwendung benötigt werden, vorgenommen worden sein.
- Bearbeiten Ihre Anwendungen Rückkehrcodes, die sie möglicherweise als Ergebnis von Änderungen, die Sie vorgenommenen haben, erhalten?
- Wurde die `PATHWAY`-Konfiguration für MQSeries geändert?

- Wurden die MQSeries-Installationsdateien geändert (z. B. der Dateizugriffsschutz)?
- Wurden die Compaq NSK-Datenbankdateien für MQSeries geändert (z. B. der Dateizugriffsschutz oder die TM/MP-Überprüfung)?
- Wurde die Konfigurationsdatei des WS-Managers (QMINI) geändert?

Lief die Anwendung bisher fehlerfrei?

Wenn der Fehler offenbar nur in Verbindung mit einer bestimmten Anwendung auftritt, überprüfen Sie, ob die Anwendung vorher fehlerfrei lief.

Folgendes sollten Sie überprüfen:

- Trat der Fehler erst auf, nachdem die Anwendung geändert wurde?

Wenn ja, wurde der Fehler vielleicht durch den neuen bzw. geänderten Teil der Anwendung verursacht. Stellen Sie fest, ob die Änderungen eine erkennbare Ursache für den Fehler aufweisen. Tritt der Fehler auch dann auf, wenn Sie einen früheren Stand der Anwendung verwenden?

- Wurden alle Funktionen der Anwendung vorher in vollem Umfang verwendet?

Trat der Fehler beim ersten Aufruf eines Anwendungsteils auf, der bis dahin noch nie verwendet wurde? Wenn ja, wird der Fehler unter Umständen durch diesen Teil der Anwendung verursacht. Stellen Sie fest, was die Anwendung machte, als der Fehler auftrat, und überprüfen Sie den Quellcode im betreffenden Teil des Programms auf Fehler.

Wenn ein Programm schon vorher fehlerfrei ausgeführt wurde, überprüfen Sie den Status der aktuellen Warteschlange sowie die Dateien, die verarbeitet wurden, als der Fehler auftrat. Enthalten sie ungewöhnliche Datenwerte, durch die eine nur selten verwendete Verzweigung im Programm aufgerufen wurde?

- Überprüft die Anwendung alle Rückkehrcodes?

Möglicherweise wurde Ihr MQSeries-System so geändert, dass Ihre Anwendung die Rückkehrcodes, die sie als Ergebnis auf die Änderung erhielt, nicht überprüft. Geht Ihre Anwendung zum Beispiel davon aus, dass die Warteschlangen, auf die sie zugreift, gemeinsam benutzt werden können? Wenn eine Warteschlange erneut definiert wurde, diesmal aber als exklusive Warteschlange, ist Ihre Anwendung in der Lage, Rückkehrcodes verarbeiten, die darauf hinweisen, dass ein Zugriff auf diese Warteschlange nicht mehr möglich ist?

- Kann die Anwendung auf anderen MQSeries-Systemen ausgeführt werden?

Weist das MQSeries-System, in dem der Fehler auftritt, Unterschiede zu diesen anderen MQSeries-Systemen auf, die den Fehler verursachen könnten? Wurde für die Warteschlangen zum Beispiel dieselbe Nachrichtenlänge oder -priorität definiert?

- Haben Sie die Umgebungsvariable SAVE-ENVIRONMENT ON gesetzt (PARAM SAVE-ENVIRONMENT ON)?

Wenn nicht, wird bei MQCONN-Aufrufen der MQSeries-Rückkehrcode 2058 zurückgegeben. Setzen Sie diese Umgebungsvariable wie in „Anhang C. TACL-Umgebungsvariablen für MQSeries for Compaq NSK setzen“ auf Seite 329 beschrieben.

Die Anwendung lief bisher nicht fehlerfrei

Wenn die Anwendung bisher noch nicht fehlerfrei gelaufen ist, sollten Sie sie sorgfältig auf Fehler überprüfen.

Erste Überprüfungen

Bevor Sie den Code überprüfen, sollten Sie zunächst (je nachdem, in welcher Programmiersprache der Code geschrieben wurde) die Ausgabe des Umsetzungsprogramms bzw. des Compilers und des Verbindungseditors überprüfen, um festzustellen, ob Fehler gemeldet wurden.

Wenn Ihre Anwendung vom Umsetzungsprogramm, Compiler oder Verbindungseditor nicht in die Ladebibliothek gestellt wurde, kann es nicht ausgeführt werden. Informationen zum Erstellen von Anwendungen finden Sie im Handbuch *MQSeries Application Programming Reference*.

Wenn die Dokumentation belegt, dass alle diese Schritte ohne Fehler ausgeführt wurden, sollten Sie die Codierlogik der Anwendung überprüfen. Geben die Fehler Symptome einen Hinweis auf die fehlerhafte Funktion und damit auf den fehlerhaften Code der Anwendung? Unter „Typische Programmierfehler“ auf Seite 209 finden Sie einige Beispiele für typische Fehler, die Probleme in Verbindung mit MQSeries-Anwendungen verursachen.

Betrifft der Fehler bestimmte Teile des Netzes?

Möglicherweise können Sie erkennen, ob bestimmte Teile des Netzes von dem Fehler betroffen sind (z. B. ferne Warteschlangen). Wenn die Verbindung mit dem WS-Manager einer fernen Warteschlange nicht aktiv ist, können die Nachrichten nicht an die ferne Warteschlange übertragen werden.

Überprüfen Sie, ob die Verbindung zwischen den beiden Systemen verfügbar ist und die Datenübertragungskomponente von MQSeries gestartet wurde.

Überprüfen Sie, ob Nachrichten in der Übertragungswarteschlange ankommen, sowie die lokale Warteschlangendefinition der Übertragungswarteschlange und aller fernen Warteschlangen.

Haben Sie irgendwelche das Netz betreffende Änderungen vorgenommen oder irgendwelche MQSeries-Definitionen geändert, die den Fehler verursacht haben könnten?

Überprüfen Sie, ob die PATHWAY-Serverklasse MQS-TCPLISxx für MQSeries-TCP-Empfangsprogramme gestartet wurde. Kann der Server nicht gestartet werden, überprüfen Sie mit einem SCF-Befehl (SCF; STATUS PROCESS \$ZTC0), ob eine Verbindung für den Port beendet wird (z. B. Status FIN-WAIT). Der ferne Partner muss in diesem Fall gestoppt und anschließend neu gestartet werden, damit der Port freigegeben wird. Überprüfen Sie bei Verwendung von SNA, ob das MQSeries-SNA-Empfangsprogramm für die SNAX/APC- oder ICE-PATHWAY-Umgebung konfiguriert ist (siehe „LU6.2-Responder-Prozesse“ auf Seite 384).

Stellen Sie sicher, dass für den Verbindungsnamen für Kanäle der richtige TCPIP-Prozessname, Host-Name bzw. IP-Adresse und Port verwendet wurde und dieser einer fernen Kanaldefinition entspricht.

Tritt der Fehler zu einer bestimmten Tageszeit auf?

Wenn der Fehler zu einer bestimmten Tageszeit auftritt, können Ladezeiten des Systems die Ursache sein. In der Regel ist ein System in der Mitte des Vormittags und in der Mitte des Nachmittags besonders stark durch Ladezeiten ausgelastet, so dass dies die Zeiten sind, in denen am häufigsten ladezeitabhängige Fehler auftreten. (Wenn sich Ihr MQSeries-Netz über mehrere Zeitzonen erstreckt, können sich diese Hauptladezeiten auf eine andere Zeit des Tages verschieben.)

Tritt der Fehler unregelmäßig auf?

Ein unregelmäßig auftretender Fehler kann dadurch verursacht werden, dass nicht berücksichtigt wird, dass Prozesse unabhängig voneinander ablaufen können. Zum Beispiel gibt ein Programm möglicherweise einen MQGET-Aufruf ohne Angabe einer WAIT-Option aus, bevor ein älterer Prozess beendet wurde. Ein unregelmäßiger Fehler kann auch dann auftreten, wenn Ihre Anwendung versucht, eine Nachricht aus einer Warteschlange abzurufen, obwohl der Aufruf zum Einreihen dieser Nachricht noch nicht bestätigt wurde (d. h., er wurde noch nicht festgeschrieben oder zurückgesetzt).

Hin und wieder können PATHWAY-Fehler während der Ausführung des Befehls `endmqm` auftreten. In diesem Fall werden unter Umständen beim Herunterfahren des WS-Managers die Fehler *1018* SERVER FILE (6006) oder *1018* SERVER FILE (7006) angezeigt. Diese Fehlnachrichten werden beim Beenden der Server MQS-QMGRSVR00, MQS-STATUS00 oder MQS-QUEUE00 erwartet.

Haben Sie Funktionsaktualisierungen installiert?

Wenn eine Funktionsaktualisierung für MQSeries installiert wurde, überprüfen Sie, ob der Aktualisierungsvorgang erfolgreich abgeschlossen wurde und ob dabei ein Fehler aufgetreten ist.

- Gab es für die Aktualisierung irgendwelche besonderen Anweisungen?
- Wurden Tests ausgeführt, um zu überprüfen, ob die Aktualisierung korrekt und vollständig installiert wurde?
- Tritt der Fehler auch dann noch auf, wenn die vorherige Servicestufe von MQSeries wiederhergestellt wurde?
- Wenn die Installation erfolgreich war, wenden Sie sich an das IBM Support Center, um zu überprüfen, ob es Programmkorrekturen gibt.
- Wenn für ein anderes Programm eine Programmkorrektur installiert wurde, überprüfen Sie, ob dies Auswirkungen auf die Kommunikation zwischen MQSeries und dem betreffenden Programm haben kann.
- Überprüfen Sie die Servicestufe. Öffnen Sie die Datei `ZMQSSYS.MEMOPTF` im schreibgeschützten Modus. Überprüfen Sie den Eintrag A) für `CSD HISTORY`. Geben Sie an der `TACL`-Eingabeaufforderung `VPROC $vol.ZMQSLIB.MQSRLLIB` ein; dabei steht `$vol` für den Datenträger mit der MQ-Installation. Die Ausgabe des Dienstprogramms `VPROC` sollte der Versionsangabe in `MEMOPTF` entsprechen oder kann aktueller sein, falls ein E-Fix installiert wurde. Überprüfen Sie die Datei `MEMOEFIX`, die mit dem E-Fix geliefert wird, auf aktuelle `VPROC`-Informationen.

Typische Programmierfehler

In der folgenden Liste sind die am häufigsten vorkommenden Ursachen für Fehler bei der Ausführung von MQSeries-Programmen aufgeführt. Überprüfen Sie, ob der Fehler auf Ihrem MQSeries-System möglicherweise auf eine oder mehrere der hier aufgelisteten Ursachen zurückgeführt werden kann:

- Warteschlangen werden als gemeinsame Warteschlangen behandelt, obwohl es sich in Wirklichkeit um exklusive Warteschlangen handelt.
- In einem MQI-Aufruf werden falsche Parameter übergeben.
- In einem MQI-Aufruf werden zu wenige Parameter übergeben. Dies kann bedeuten, dass MQI keine Beendigungs- oder Ursachencodes definieren kann, die von der Anwendung verarbeitet werden können.
- Rückkehrcodes von MQI-Anforderungen werden nicht überprüft.

Typische Programmierfehler

- Es werden Variablen mit falschen Längenangaben übergeben.
- Parameter werden in einer falschen Reihenfolge übergeben.
- *MsgId* (Nachrichten-ID) und *CorrelId* (Korrelations-ID) werden nicht korrekt initialisiert.
- BEGINTRANSACTION wird nicht ausgegeben, wenn MQPMO_SYNCPOINT im MQPUT-Befehl angegeben wird.

Fehler in Zusammenhang mit Befehlen

Es wird zur Vorsicht geraten, wenn Sie im beschreibenden Text für einige Befehle Sonderzeichen wie Backslash (\) oder doppelte Anführungszeichen (") verwenden. In beschreibendem Text sollte diesen Sonderzeichen ein Backslash (\) vorangestellt werden. Geben Sie also \\ oder \" ein, wenn Sie im Text \ oder " verwenden möchten.

Die nächsten Schritte

Nachdem Sie festgestellt haben, dass Ihr System nicht geändert wurde und es keine Probleme mit Ihren Anwendungsprogrammen gibt, wählen Sie jetzt die Option aus, deren Beschreibung am besten auf die Symptome des Fehlers zutrifft.

- „Haben Sie falsche Ausgaben erhalten?“
- „Haben Sie von einem PCF-Befehl keine Antwort erhalten?“ auf Seite 211
- „Betrifft der Fehler nur ferne Warteschlangen?“ auf Seite 212
- „Läuft Ihre Anwendung oder MQSeries for Compaq NSK langsam?“ auf Seite 213

Haben Sie falsche Ausgaben erhalten?

Mit „falschen Ausgaben“ ist in diesem Handbuch Folgendes gemeint:

- Die Anwendung hat nicht die erwartete Nachricht empfangen.
- Die Anwendung hat eine Nachricht mit unerwarteten oder beschädigten Informationen empfangen.
- Die Anwendung hat eine Nachricht empfangen, die sie nicht erwartet hatte, z. B. eine Nachricht, die an eine andere Anwendung gerichtet war.
- Verwenden Sie die richtige Struktur für die Anzeige der Informationen? Wurde zum Beispiel dem Nachrichtentext einer Nachricht in der DLQ (Warteschlange für nicht zustellbare Nachrichten) die Struktur MQDLH, die die Daten des DLQ-Headers enthält, vorangestellt?
- Kann die Ausgabe in einer MQSeries-Installation auf einer anderen Maschine mit derselben oder einer andere Version des Betriebssystems reproduziert werden. Ist dies nicht möglich, liegt möglicherweise ein Fehler im Betriebssystem vor, für den die Ausführung einer IPM (Interim Program Maintenance) von Compaq erforderlich ist. Wenn Sie sich an den Compaq Kundendienst.

Überprüfen Sie in allen Fällen, ob die von der Anwendung verwendeten Aliasnamen von Warteschlangen oder WS-Managern richtig angegeben wurden und ob sie den Änderungen, die im Netz vorgenommen wurden, entsprechen.

Wenn MQSeries-Fehlernachrichten (beginnen alle mit dem Präfix „AMQ“) generiert werden, sollten Sie im Fehlerprotokoll nachschauen. Weitere Informationen finden Sie unter „Fehlerprotokolle“ auf Seite 219.

Haben Sie von einem PCF-Befehl keine Antwort erhalten?

Wenn Sie einen Befehl ausgegeben, darauf aber keine Antwort erhalten haben, überprüfen Sie die Folgendes:

- Ist der Befehlsserver aktiv?

Überprüfen Sie mit Hilfe des Befehls **dspmqcvs** den Status des Befehlsservers.

Wenn die Antwort auf diesen Befehl anzeigt, dass der Befehlsserver nicht aktiv ist, starten Sie ihn mit dem Befehl **strmqcvs**. Wenn die Antwort auf den Befehl anzeigt, dass die Warteschlange SYSTEM.ADMIN.COMMAND.QUEUE nicht für MQGET-Anforderungen aktiviert ist, aktivieren Sie sie für MQGET-Anforderungen.

- Wurde eine Antwort in die Warteschlange für nicht zustellbare Nachrichten (DLQ) gestellt?

Die Header-Struktur der Warteschlange für nicht zustellbare Nachrichten enthält einen Ursachen- bzw. Rückgabecode mit einer Beschreibung des Fehlers. Weitere Informationen zur MQDLH-Struktur (DLQ-Header) finden Sie im Handbuch *MQSeries Application Programming Reference*.

Wenn sich Nachrichten in der Warteschlange für nicht zustellbare Nachrichten befinden, können Sie diese mit der zur Verfügung gestellten Beispielanwendung (AMQSBCG) und mit dem MQGET-Aufruf durchsuchen. Die Beispielanwendung durchsucht alle Nachrichten in einer angegebenen Warteschlange in einem angegebenen WS-Manager und zeigt den Nachrichtendeskriptor und die Felder mit dem Nachrichtenkontext aller Nachrichten in der angegebenen Warteschlange an.

- Wurde eine Nachricht in das Fehlerprotokoll gestellt?

Weitere Informationen finden Sie unter „Fehlerprotokolle“ auf Seite 219.

- Wurden die Warteschlangen für PUT- und GET-Operationen aktiviert?

- Ist das Warteintervall (*waitInterval*) lang genug?

Wenn Ihr MQGET-Aufruf das zulässige Zeitlimit überschritten hat, werden der Beendigungscode MQCC_FAILED und der Ursachencode MQRC_NO_MSG_AVAILABLE zurückgegeben. (Informationen zum Feld *waitInterval* sowie zu Beendigungs- und Ursachencodes des Aufrufs MQGET finden Sie im Handbuch *MQSeries Application Programming Reference*.)

- Wenn Sie ein eigenes Anwendungsprogramm verwenden, um Befehle in die Warteschlange SYSTEM.ADMIN.COMMAND.QUEUE einzureihen, müssen Sie dann eine Transaktion festschreiben?

Sofern Sie Ihre Anforderungsnachricht nicht explizit vom Synchronisationspunkt ausgeschlossen haben, müssen Sie eine Transaktion festschreiben, bevor Sie versuchen, Antwortnachrichten zu empfangen.

- Sind die Werte für die Attribute MAXDEPTH und MAXMSGL für Ihre Warteschlangen hoch genug?

- Verwenden Sie die Felder *CorrelId* und *MsgId* richtig?

Geben Sie die Werte *MsgId* und *CorrelId* in Ihrer Anwendung an, um sicherzustellen, dass Sie alle Nachrichten aus der Warteschlange empfangen.

Versuchen Sie, den Befehlsserver zu stoppen und dann erneut zu starten, wobei Sie auf alle erstellten Fehlernachrichten antworten.

Wenn das System immer noch nicht reagiert, liegt es unter Umständen am WS-Manager. Stoppen Sie den WS-Manager, und starten Sie ihn anschließend erneut. Bleibt der Fehler auch nach dem Neustart bestehen, fordern Sie Hilfe vom zuständigen IBM Support Center an.

Sind einige Ihrer Warteschlangen fehlerhaft?

Wenn Sie vermuten, dass der Fehler nur einen Teil der Warteschlangen betrifft, überprüfen Sie diejenigen lokalen Warteschlangen, die Ihrer Meinung nach fehlerhaft sind:

1. Zeigen Sie die Informationen zu jeder einzelnen Warteschlange an. Verwenden Sie dazu den MQSC-Befehl `DISPLAY QUEUE`.
2. Überprüfen Sie anhand der angezeigten Daten Folgendes:
 - Wenn der Wert für `CURDEPTH` den Wert für `MAXDEPTH` erreicht hat, bedeutet dies, dass die Warteschlange nicht verarbeitet werden kann. Überprüfen Sie, ob alle Anwendungen normal laufen.
 - Wenn der Wert für `CURDEPTH` den Wert für `MAXDEPTH` noch nicht erreicht hat, überprüfen Sie, ob die folgenden Warteschlangenattribute korrekt sind:
 - Bei Verwendung der Auslösefunktion:
 - Ist der Auslösemonitor aktiv?
 - Ist die Auslöseschwelle zu hoch? Das heißt, werden Auslöseereignisse häufig genug generiert?
 - Ist der Prozessname richtig?
 - Ist der Prozess verfügbar und betriebsbereit?
 - Kann die Warteschlange gemeinsam benutzt werden? Wenn nicht, wurde sie möglicherweise bereits von einer anderen Anwendung für eine Eingabe geöffnet.
 - Wurde die Warteschlange wie erforderlich für `GET`- und `PUT`-Aufrufe aktiviert?
 - Wenn keine Anwendungsprozesse Nachrichten aus der Warteschlange abrufen können, stellen Sie den Grund dafür fest. Es kann daran liegen, dass die Anwendungen gestartet werden müssen, eine Verbindung unterbrochen wurde oder der Aufruf `MQOPEN` aus irgendeinem Grund fehlgeschlagen ist. Überprüfen Sie die Attribute `IPPROCS` und `OPPROCS`. Diese Attribute zeigen an, ob die Warteschlange für Ein- bzw. Ausgaben geöffnet wurde. Wenn ein Attribut den Wert null hat, bedeutet dies, dass Operationen des betreffenden Typs nicht ausgeführt werden können. Beachten Sie, dass die Werte sich geändert haben können und die Warteschlange geöffnet war, aber inzwischen wieder geschlossen wurde.

Sie müssen den Status zu dem Zeitpunkt überprüfen, an dem Sie eine Nachricht einreihen oder abrufen wollen.

Wenn Sie den Fehler nicht beheben können, fordern Sie Hilfe vom zuständigen IBM Support Center an.

Betrifft der Fehler nur ferne Warteschlangen?

Wenn der Fehler nur ferne Warteschlangen betrifft, überprüfen Sie Folgendes:

- Überprüfen Sie, ob erforderliche Kanäle gestartet wurden und ausgelöst werden können und ob alle erforderlichen Initiatoren aktiv sind.
- Überprüfen Sie, ob die Programme, die Nachrichten in die ferne Warteschlange einreihen sollen, Fehler gemeldet haben.
- Wenn Sie die Auslösefunktion zum Starten des verteilten Queuing verwenden, überprüfen Sie, ob die Auslösefunktion für die Übertragungswarteschlange aktiviert ist. Überprüfen Sie außerdem, ob der Auslösemonitor aktiv ist
- Überprüfen Sie die Fehlerprotokolle auf Nachrichten, die auf Kanalfehler oder -probleme hinweisen.

- Starten Sie den Kanal gegebenenfalls manuell. Informationen hierzu finden Sie im Handbuch *MQSeries Intercommunication*.

Informationen zur Definition von Kanälen finden Sie in „Anhang M. Kommunikation einrichten“ auf Seite 383 und im Handbuch *MQSeries Intercommunication*.

Läuft Ihre Anwendung oder MQSeries for Compaq NSK langsam?

Wenn Ihre Anwendung langsam ist, so befindet sie sich unter Umständen in einer Schleife oder wartet auf eine nicht verfügbare Ressource.

Wurde die MQSeries-Trace-Funktion über den Befehl **strmqtrc** oder über die MQMC-Anzeige **Queue Manager Menu** aktiviert? Dies kann zu einer Verschlechterung der Systemleistung führen. Überprüfen Sie, ob im Teildatenträger mit den Fehlerprotokollen des WS-Managers eventuell TR-Dateien geöffnet sind. Inaktivieren Sie die Trace-Funktion über den Befehl **endmqtrc** oder über die MQMC-Anzeige **Queue Manager Menu**.

Unter Umständen liegt ein Problem mit der Betriebsleistung vor. Es besteht die Möglichkeit, dass das System an den Grenzen seiner Kapazitäten arbeitet. Dieses Problem tritt wahrscheinlich verstärkt zu Zeiten mit einer hohen Systembelastung auf, als etwa am späten Morgen und am frühen Nachmittag. (Wenn sich Ihr Netz über verschiedene Zeitzonen erstreckt, können sich die Zeiten mit hoher Systembelastung verschieben.)

Überprüfen Sie mit dem Befehl STATUS die Priorität der Anwendungs- und der WS-Managerprozesse. Bei Auftreten einer Schleife wird die Priorität eines Prozesses von NSK nach und nach auf null reduziert.

Überprüfen Sie, ob die einzelnen CPUs im NSK-System vollständig ausgelastet sind. Sind einige Prozessoren nur wenig ausgelastet, müssen Sie die NSK-Systembelastung ausgleichen. Sie können beispielsweise in anderen Prozessoren weitere Ausführungs-Controller hinzufügen, um die MQSeries-Belastung gleichmäßig zu verteilen.

Leistungsprobleme können auch durch Hardwareeinschränkungen hervorgerufen werden.

Anmerkung: Nach einer Neuinstallation von MQSeries oder nach einem Kaltstart des Compaq NSK-Systems dauert der Start der ausführbaren MQSeries-Module unter Umständen länger als beim ersten Aufruf. Dies liegt daran, dass für das Betriebssystem Compaq NSK eine „Berichtigungsphase“ ausgeführt wird, in der sichergestellt wird, dass alle externen Deklarationen aufgelöst werden.

Sollten Sie feststellen, dass die Leistungsverschlechterung nicht mit der Systemlast in Zusammenhang steht, da das Problem auch in Zeiten auftritt, in denen das System nur wenig belastet ist, liegt die Ursache wahrscheinlich an einer unzureichenden Anwendung. Dies zeigt sich dann, wenn ein Zugriff auf bestimmte Zugriffe erfolgt.

Folgende Symptome können auf ein langsames MQSeries-System hindeuten:

- Das System reagiert langsam auf die MQSeries-Befehle.

Nächste Schritte

- Bei der wiederholten Anzeige der Warteschlangenlänge zeigt sich, dass die Verarbeitung der Warteschlange für eine Anwendung, bei der eigentlich eine große Anzahl von Warteschlangenzugriffen erwartet wird, sehr langsam ist.

Kann die Systemleistung trotz allem nicht verbessert werden, liegt das Problem unter Umständen an MQSeries for Compaq NSK selbst. Wenden Sie sich in diesem Fall an das IBM Support Center.

Überlegungen zum Anwendungsdesign

Es gibt eine Reihe von Möglichkeiten, wie sich ein schlechter Programmwurf auf die Leistung auswirken kann. Dies ist nicht immer sofort erkennbar, weil das Programm scheinbar problemlos läuft, gleichzeitig aber die Leistung anderer Tasks beeinträchtigen kann. Einige Probleme, die Programme mit MQSeries-Aufrufen betreffen, werden in den folgenden Abschnitten behandelt.

Weitere Informationen zum Anwendungsdesign finden Sie im Handbuch *MQSeries Application Programming Guide*.

Auswirkung der Nachrichtenlänge

Nachrichten in MQSeries können zwar bis zu 100 MB Daten umfassen, die Datenmenge in einer Nachricht hat jedoch Auswirkungen auf die Leistung der Anwendung, von der die Nachricht verarbeitet wird. Um die Leistung Ihrer Anwendung zu optimieren, sollten Sie nur die wirklich benötigten Daten in einer Nachricht senden; so müssen zum Beispiel in einer Anforderung zur Belastung eines Kontos möglicherweise nur die Kontonummer und der Belastungsbetrag als wirklich benötigte Informationen von der Client- an die Serveranwendung übergeben werden.

Nach einer bestimmten Nachricht suchen

Der Aufruf MQGET ruft in der Regel die erste Nachricht in einer Warteschlange ab. Wenn Sie im Nachrichtendeskriptor die Nachrichten-ID und die Korrelations-ID (*MsgId* und *CorrelId*) für eine bestimmte Nachricht angeben, muss der WS-Manager die Warteschlange nach dieser Nachricht durchsuchen. Eine solche Verwendung des Aufrufs MQGET wirkt sich auf die Leistung Ihrer Anwendung aus.

Warteschlangen mit Nachrichten in unterschiedlicher Länge

Wenn die Nachrichten in einer Warteschlange unterschiedlich lang sind, kann die Anwendung die Länge einer Nachricht feststellen, indem sie den Aufruf MQGET mit dem Feld *BufferLength* gleich null verwendet; der Aufruf schlägt zwar fehl, er gibt aber die Größe der Nachrichtendaten zurück. Die Anwendung kann den Aufruf anschließend wiederholen, wobei sie die ID der Nachricht, die mit dem ersten Aufruf 'gemessen' wurde, und einen Puffer mit der passenden Größe angibt. Wenn andere Anwendungen auf dieselbe Warteschlange zugreifen, stellen Sie jedoch möglicherweise fest, dass die Leistung Ihrer Anwendung nachlässt, weil der zweite MQGET-Aufruf damit beschäftigt ist, nach einer Nachricht zu suchen, die zwischenzeitlich (zwischen dem ersten und zweiten Aufruf) von einer anderen Anwendung abgerufen wurde.

Wenn Ihre Anwendung keine Nachrichten mit fester Länge verwenden kann, können Sie das Problem beheben, indem Sie mit dem Aufruf MQINQ nach der maximalen Größe von Nachrichten in der Warteschlange suchen und diesen Wert dann in Ihrem MQGET-Aufruf verwenden. Die maximale Größe von Nachrichten in einer Warteschlange wird im Attribut *MaxMsgLength* der Warteschlange gespeichert.

Bei diesem Verfahren wird jedoch viel Speicher belegt, da für dieses Warteschlangenattribut bis zu 100 MB angegeben werden können, der in MQSeries for Compaq NSK maximal zulässige Wert.

Häufigkeit von Synchronisationspunkten

Programme, die zwischen Synchronisationspunkten zahlreiche MQPUT-Aufrufe ausgeben, ohne sie festzuschreiben, können Leistungsprobleme verursachen. Davon betroffene Warteschlangen können mit Nachrichten gefüllt werden, auf die in dieser Zeit nicht zugegriffen werden kann, die aber gleichzeitig möglicherweise von anderen Tasks dringend erwartet werden. Dies hat Auswirkungen auf den Speicher, auf die Belegung der TMF-Prüfprotokolle und auf die Prozesse bei Tasks, bei denen Nachrichten abgerufen werden.

Den Aufruf MQPUT1 verwenden

Verwenden Sie den Aufruf MQPUT1, wenn lediglich eine einzelne Nachricht in eine Warteschlange eingereiht werden soll. Verwenden Sie für mehrere Nachrichten den Aufruf MQOPEN, gefolgt von einer Serie von MQPUT-Aufrufen und einem einzelnen MQCLOSE-Aufruf.

Falsche Ausgabe

Der Begriff 'Falsche Ausgabe' kann auf unterschiedliche Weise interpretiert werden. Seine Bedeutung im Rahmen der Fehlerbestimmung in diesem Buch wird unter „Haben Sie falsche Ausgaben erhalten?“ auf Seite 210 erklärt.

In diesem Abschnitt werden zwei Arten von falschen Ausgaben behandelt:

- Nachrichten erscheinen wider Erwarten nicht in der Warteschlange.
- Nachrichten enthalten falsche Informationen oder Informationen, die beschädigt wurden.

Darüber hinaus werden weitere Fehler behandelt, die in einer Anwendung auftreten können, die verteilte Warteschlangen verwendet.

Nachrichten erscheinen nicht in der Warteschlange

Wenn Nachrichten wider Erwarten nicht in einer Warteschlange erscheinen, überprüfen Sie Folgendes:

- Wurde die Nachricht erfolgreich in die Warteschlange eingereiht?
- Wurde die Warteschlange korrekt definiert? Ist zum Beispiel MAXMSGL groß genug?
- Ist die Warteschlange für PUT-Aufrufe aktiviert?
- Ist die Warteschlange bereits voll? Dies kann bedeuten, dass eine Anwendung nicht in der Lage war, die gesuchte Nachricht in die Warteschlange einzureihen.
- Können Sie überhaupt eine Nachricht aus der Warteschlange abrufen?
- Müssen Sie einen Synchronisationspunkt festlegen?

Wenn Nachrichten zwischen Synchronisationspunkten eingereiht oder abgerufen werden, stehen sie anderen Tasks erst zur Verfügung, nachdem die Arbeitseinheit mit Wiederherstellung festgeschrieben wurde.

- Ist das Warteintervall lang genug?

Das Warteintervall kann als eine Option für den Aufruf MQGET angegeben werden. Stellen Sie sicher, dass der Aufruf lange genug auf eine Antwort wartet.

- Warten Sie auf eine bestimmte Nachricht, die durch eine Nachrichten-ID oder eine Korrelations-ID (*MsgId* oder *CorrelId*) identifiziert wurde?

Falsche Ausgabe

Überprüfen Sie, ob Sie auf eine Nachricht mit der richtigen *MsgId* bzw. *CorrelId* warten. Nach einem erfolgreichen MQGET-Aufruf enthalten diese beiden Werte die IDs der abgerufenen Nachricht, so dass Sie diese Werte möglicherweise zurücksetzen müssen, um eine andere Nachricht erfolgreich abzurufen.

Überprüfen Sie außerdem, ob Sie andere Nachrichten aus der Warteschlange abrufen können.

- Können andere Anwendungen Nachrichten aus der Warteschlange abrufen?
- Wurde die erwartete Nachricht als permanente Nachricht definiert?
Wenn nicht, ist die Nachricht nach einem erneuten Start von MQSeries verloren gegangen.
- Besitzt eine andere Anwendung einen exklusiven Zugriff auf die Warteschlange?

Wenn Sie in Zusammenhang mit der Warteschlange keinen Fehler feststellen können und MQSeries aktiv ist, überprüfen Sie wie folgt den Prozess, von dem Sie annehmen, dass er die Nachricht in die Warteschlange einreicht:

- Wurde die Anwendung gestartet?
Wenn die Anwendung durch einen Auslöser gestartet werden sollte, überprüfen Sie, ob die richtigen Auslösoptionen angegeben wurden.
- Wurde die Anwendung gestoppt?
- Ist ein Auslösemonitor aktiv?
- Wurde der Auslöseprozess korrekt definiert?
- Wurde die Anwendung ordnungsgemäß beendet?
Überprüfen Sie im Jobprotokoll, ob möglicherweise eine abnormale Beendigung stattfand.
- Hat die Anwendung ihre Änderungen festgeschrieben, oder wurden sie zurückgesetzt?

Wenn mehrere Transaktionen auf die Warteschlange zugreifen, kann es zwischen ihnen zu Konflikten kommen. Nehmen wir zum Beispiel an, dass eine Transaktion einen MQGET-Aufruf mit einer Pufferlänge gleich null ausgibt, um die Länge einer Nachricht festzustellen, und anschließend einen gezielten MQGET-Aufruf mit Angabe der *MsgId* der Nachricht ausgibt. Inzwischen wird aber von einer anderen Transaktion erfolgreich ein MQGET-Aufruf für dieselbe Nachricht ausgegeben, so dass an die erste Anwendung der Ursachencode MQRC_NO_MSG_AVAILABLE zurückgegeben wird. Anwendungen, die in einer Multiserverumgebung ausgeführt werden sollen, müssen eine solche Situation verarbeiten können.

Prüfen Sie die Möglichkeit, dass die Nachricht empfangen wurde, Ihre Anwendung sie aber aus irgendeinem Grund nicht verarbeiten konnte. Zum Beispiel kann ein Fehler in dem erwarteten Format der Nachricht die Anwendung veranlassen, die Nachricht zurückzuweisen. Wie Sie in einem solchen Fall weiter vorgehen können, ist unter „Nachrichten mit unerwarteten oder beschädigten Informationen“ beschrieben.

Nachrichten mit unerwarteten oder beschädigten Informationen

Wenn die Nachricht andere Informationen enthält, als von Ihrer Anwendung erwartet, oder die Informationen auf irgendeine Art und Weise beschädigt sind, überprüfen Sie folgende Punkte:

- Wurde Ihre Anwendung oder die Anwendung, von der die Nachricht in die Warteschlange eingereicht wurde, geändert?

Stellen Sie sicher, dass alle Änderungen auf allen Systemen, die von den Änderungen betroffen sind, gleichermaßen vorgenommen wurden.

Zum Beispiel kann sich das Format der Nachrichtendaten geändert haben, was zur Folge hat, dass beide Anwendungen erneut kompiliert werden müssen, damit sie die Änderungen berücksichtigen. Wenn eine der Anwendungen nicht erneut kompiliert wurde, werden ihre Daten von der anderen Anwendung als beschädigt betrachtet.

- Sendet eine Anwendung Nachrichten an die falsche Warteschlange?

Überprüfen Sie, ob die von Ihrer Anwendung empfangenen Nachrichten nicht möglicherweise für eine Anwendung bestimmt sind, die auf eine andere Warteschlange zugreift. Ändern Sie gegebenenfalls Ihre Sicherheitsdefinitionen, um zu verhindern, dass Nachrichten von dazu nicht berechtigten Anwendungen in die falschen Warteschlangen eingereiht werden.

Wenn Ihre Anwendung eine Aliaswarteschlange verwendet hat, überprüfen Sie, ob der Aliasname auf die richtige Warteschlange zeigt.

- Wurde die Auslöseinformation für diese Warteschlange richtig angegeben?
Überprüfen Sie, ob tatsächlich Ihre und nicht möglicherweise eine andere Anwendung gestartet werden sollte.

Wenn Sie den Fehler auch nach diesen Überprüfungen nicht beheben konnten, sollten Sie Ihre Anwendungslogik überprüfen, und zwar sowohl die des Programms zum Senden der Nachricht als auch die des Programms zum Empfangen der Nachricht.

Probleme mit falschen Ausgaben bei der Verwendung verteilter Warteschlangen

Wenn Ihre Anwendung verteilte Warteschlangen verwendet, sollten Sie zudem Folgendes überprüfen:

- Wurde MQSeries sowohl auf dem sendenden als auch auf dem empfangenden System ordnungsgemäß installiert und korrekt für verteiltes Queuing konfiguriert?
- Sind die Verbindungen zwischen den beiden Systemen verfügbar?
Überprüfen Sie, ob beide Systeme verfügbar und mit MQSeries verbunden sind. Überprüfen Sie, ob die Verbindung zwischen den beiden Systemen und die Kanäle zwischen den beiden WS-Managern aktiv sind.
- Ist auf dem sendenden System die Auslösefunktion aktiviert?
- Handelt es sich bei der Nachricht, auf die Sie warten, um eine Antwortnachricht von einem fernen System?
Überprüfen Sie, ob die Auslösefunktion auf dem fernen System aktiviert ist.
- Ist die Warteschlange bereits voll?

Dies kann bedeuten, dass eine Anwendung nicht in der Lage war, die gesuchte Nachricht in die Warteschlange einzureihen. Überprüfen Sie in diesem Fall, ob die Nachricht in die Warteschlange für nicht zustellbare Nachrichten eingereiht wurde.

Der Header der Warteschlange für nicht zustellbare Nachrichten enthält einen Ursachen- oder Rückgabecode, der Auskunft darüber gibt, warum die Nachricht nicht in die Zielwarteschlange eingereiht werden konnte. Informationen zur Struktur des Headers der Warteschlange für nicht zustellbare Nachrichten finden Sie im Handbuch *MQSeries Application Programming Reference*.

- Gibt es eine Abweichung zwischen dem sendenden und dem empfangenden WS-Manager?

Falsche Ausgabe

Es ist zum Beispiel möglich, dass der empfangende WS-Manager die Nachrichtenlänge nicht verarbeiten kann.

- Sind die Kanaldefinitionen der Sende- und Empfangskanäle kompatibel?
Zum Beispiel wird die Komponente für verteiltes Queuing gestoppt, wenn in der Folgenummernserie eine Abweichung vorliegt. Weitere Informationen zum verteilten Queuing finden Sie im Handbuch *MQSeries Intercommunication*.
- Haben Sie ein TCP/IP-Empfangsprogramm gestartet?
Erfolgt die Kommunikation zwischen MQSeries und Compaq über das Übertragungsprotokoll TCP/IP, muss ein TCP/IP-Empfangsprogrammprozess aktiv sein. Weitere Informationen finden Sie unter „TCP/IP-Empfangsprogramme konfigurieren und steuern“ auf Seite 34.
- Ist das TCP/IP-Empfangsprogramm am richtigen TCP/IP-Port empfangsbereit?
Das TCP/IP-Empfangsprogramm ist an dem Port aktiv, der im Eintrag TCPListenerPort in der Zeilengruppe TCPConfig der Konfigurationsdatei des WS-Managers (QMINI) angegeben wurde. Weitere Informationen finden Sie unter „Anzahl an TCP/IP-Ports, die vom WS-Manager überwacht werden“ auf Seite 57.
- Wurde der richtige TCP/IP-Prozess angegeben?
Wenn Sie TCP/IP als Übertragungsprotokoll verwenden: Verwendet das Compaq-System für den TCP/IP-Prozess den Standardprozessnamen (\$ztc0)? Falls nicht, müssen Sie einige der Serverklassen in Ihrer MQSeries-PATHWAY-Konfiguration ändern, damit die MQSeries-Kanäle den richtigen Prozessnamen verwenden. Weitere Informationen finden Sie unter „Serverklasse MQS-TCPLISnn für die Verwendung eines anderen TCP/IP-Prozesses und Ports rekonfigurieren“ auf Seite 58.
- Ist das MQSeries-SNA-Empfangsprogramm konfiguriert?
Falls Folgendes zutrifft:
 - Sie verwenden MQSeries-Kanäle, die SNA als Übertragungsprotokoll verwenden.
 - Der Kanaltyp unter Compaq muss von einem fernen MQSeries-System (z. B. einem RECEIVER) aktiviert werden.
 - Das ferne System hat Probleme, den Kanal zu starten.Treffen diese Bedingungen zu, ist unter Umständen das MQSeries-SNA-Empfangsprogramm in der PATHWAY-Umgebung für Ihren WS-Manager nicht aktiv. Überprüfen Sie, ob das MQSeries-SNA-Empfangsprogramm für die SNAX/APC- oder ICE-PATHWAY-Umgebung des WS-Managers konfiguriert wurde. Weitere Informationen finden Sie unter „LU6.2-Responder-Prozesse“ auf Seite 384.
- Findet eine Datenkonvertierung statt? Wenn die sendenden und empfangenden Anwendungen mit unterschiedlichen Datenformaten arbeiten, wird eine Konvertierung erforderlich. Bei der Ausgabe des Aufrufs MQGET wird eine automatische Konvertierung durchgeführt, wenn es sich bei dem erkannten Format um eines der integrierten Formate handelt.
Wird das Datenformat nicht als integriertes Format erkannt, kann ein Datenkonvertierungs-Exit verwendet werden, die eine Umwandlung mit Hilfe eigener Routinen ermöglicht. Überprüfen Sie, ob die Routine richtig geladen wird.
Weitere Informationen zur Datenkonvertierung finden Sie im Handbuch *MQSeries Application Programming Guide*.

Fehlerprotokolle

MQSeries for Compaq NSK erstellt eine Reihe von Fehlerprotokollen mit Nachrichten über den Betrieb von MQSeries selbst, über alle gestarteten WS-Manager und über fehlerhafte Daten von den verwendeten Kanälen.

Das Verzeichnis der Fehlerprotokolle hängt davon ab, ob der Name des WS-Managers bekannt ist und ob der Fehler einem Client zugeordnet ist.

- Der Name des WS-Managers ist bekannt und der WS-Manager ist verfügbar:

```
<QMVOL>.<SUBVOL>.L.MQERRLG1
```

- Der WS-Manager ist nicht verfügbar:

```
<MQSVOL>.ZMQSSYS.MQERRLG1
```

- FFST-Berichte (First Failure Support Technology) in:

```
<QMVOL>.<SUBVOL>.FDnnnnn
```

- Siehe hierzu „FFST-Daten überprüfen“ auf Seite 224.

Protokolldateien

Der Teildatenträger für Fehlerprotokolle kann bis zu drei Fehlerprotokolldateien enthalten; dabei handelt es sich um folgende:

- MQERRLG1
- MQERRLG2
- MQERRLG3

Nachdem Sie einen WS-Manager erstellt haben, werden drei Fehlerprotokolldateien erstellt, sobald sie vom WS-Manager benötigt werden. Diese Dateien erhalten die Namen MQERRLG1, MQERRLG2 und MQERRLG3 und werden in den Teildatenträger jedes von Ihnen erstellten WS-Managers gestellt.

Alle Fehler- oder Protokollnachrichten, die erstellt werden, werden in der Datei MQERRLG1 gespeichert. Sobald die Datei MQERRLG1 voll ist, wird ihr Inhalt in die Datei MQERRLG2 kopiert. Zuvor wird jedoch der Inhalt der Datei MQERRLG2 in die Datei MQERRLG3 kopiert. Der bisherige Inhalt von MQERRLG3 wird gelöscht.

Die neuesten Fehlnachrichten befinden sich daher immer in MQERRLG1, während die anderen Dateien dazu dienen, Fehlnachrichten für eine längere Zeit aufzubewahren.

Alle Nachrichten, die Kanäle betreffen, werden ebenfalls in den entsprechenden Fehlerdateien des WS-Managers gespeichert, es sei denn, der Name ihres WS-Managers ist nicht bekannt oder der WS-Manager ist nicht verfügbar. Wenn der Name des WS-Managers nicht verfügbar ist oder nicht ermittelt werden kann, werden kanalbezogene Nachrichten in das Systemfehlerprotokoll geschrieben (ZMQSSYS.MQERRLG1).

Fehlerprotokolle

Sie können den Inhalt einer Fehlerprotokolldatei anzeigen, und zwar entweder mit dem Befehl FUP COPY oder (schreibgeschützt) in Ihrem normalen Compaq NSK-Editor. (Wird das Fehlerprotokoll im Aktualisierungsmodus geöffnet, gehen unter Umständen Fehlermeldungen verloren.)

Fehler in der Startphase

Es gibt eine Reihe von besonderen Fällen, in denen Fehler auftreten, bevor die oben genannten Fehlerprotokolle vollständig eingerichtet sind. MQSeries versucht, solche Fehler in einem Fehlerprotokoll aufzuzeichnen. Das Verzeichnis dieses Protokoll ist davon abhängig, wie weit ein WS-Manager erstellt ist.

Wenn zum Beispiel auf Grund einer beschädigten Konfigurationsdatei kein Verzeichnis ermittelt werden kann, werden Fehler in einer Fehlerdatei aufgezeichnet, die bei der Installation in der Datei MQERRLG1 im Teildatenträger ZMQSSYS erstellt wird.

Weitere Information zu Konfigurationsdateien finden Sie in „Kapitel 13. Konfigurationsdateien“ auf Seite 193.

Bedienernachrichten

In MQSeries for Compaq NSK weisen Bedienernachrichten auf normale Fehler hin, die in der Regel direkt von Benutzern verursacht werden, z. B. indem sie in einem Befehl Parameter verwenden, die dort nicht zulässig sind. Diese Nachrichten werden in einem entsprechenden Fenster, sofern vorhanden, angezeigt und außerdem in eine Datei im Teildatenträger des WS-Managers geschrieben.

Fehler, die einem bestimmten WS-Manager zugeordnet werden können, werden in der Datei MQERRLG1 im Teildatenträger mit den Fehlerprotokollen des betreffenden WS-Managers aufgezeichnet. Fehler, die keinem definierten und aktiven WS-Manager zugeordnet werden können, werden in der Datei MQERRLG1 im Teildatenträger ZMQSSYS aufgezeichnet.

Nummern der Ausführungs-Controller in der Datei MQERRLG1 ermitteln

Die in der Datei MQERRLG1 aufgezeichnete Ausführungs-Controller-Nummer wird vom ECBoss vergeben, um eine Verfolgung des Ausführungs-Controller-Prozesses zu ermöglichen. Es besteht kein direkter Zusammenhang zwischen der vom ECBoss vergebenen Nummer in der Datei MQERRLG1 und der Nummer, die im Namen der PATHWAY-Serverklasse MQS-ECxx verwendet wird. So wird beispielsweise der Serverklasse MQS-EC00 nicht unbedingt die Ausführungs-Controller-Nummer 0 zugewiesen. Die für den Ausführungs-Controller vergebene Nummer wird dem Namen des Ausführungs-Controller-Prozesses in der Nachricht zugeordnet, die den Abschluss der Initialisierung meldet. Anhand des Namens des Ausführungs-Controller-Prozesses können Sie für die Fehleranalyse die betreffende PATHWAY-Serverklasse MQS-ECxx ermitteln.

Beispiel für ein Fehlerprotokoll

Dieses Beispiel zeigt einen Auszug aus einem MQSeries for Compaq NSK-Fehlerprotokoll:

```

...
02/01/01 11:41:56 AMQ8003: MQSeries-Warteschlangenmanager gestartet.
ERKLÄRUNG: Der MQSeries-Warteschlangenmanager Janet wurde gestartet.
AKTION: Keine.
-----
02/01/01 11:56:52 AMQ9002: Kanalprogramm gestartet.
ERKLÄRUNG: Das Kanalprogramm 'Janet' wurde gestartet.
AKTION: Keine.
-----
02/01/01 11:57:26 AMQ9208: Fehler beim Empfangen von Host 'camelot
(9.20.12.34)'.
ERKLÄRUNG: Beim Datenempfang von 'camelot'
(9.20.12.34)' über TCP/IP ist ein Fehler aufgetreten. Dies ist
möglicherweise auf einen Datenübertragungsfehler zurückzuführen.
AKTION: Den TCP/IP-Rückkehrcode 232 (X'E8') notieren und
den Systemadministrator benachrichtigen.
-----
02/01/01 11:57:27 AMQ9999: Kanalprogramm abnormal beendet.
ERKLÄRUNG: Das Kanalprogramm 'Janet' wurde abnormal beendet.
AKTION: Zur Fehlerbestimmung die zuvor für Kanalprogramm 'Janet'
ausgegebenen Fehlermeldungen in den Fehlerdateien überprüfen.
-----
02/01/01 14:28:57 AMQ8004: MQSeries-Warteschlangenmanager beendet.
ERKLÄRUNG: Der MQSeries-Warteschlangenmanager Janet wurde beendet.
AKTION: Keine.
-----
02/02/01 15:02:49 AMQ9002: Kanalprogramm gestartet.
ERKLÄRUNG: Das Kanalprogramm 'Janet' wurde gestartet.
AKTION: Keine.
-----
02/02/01 15:02:51 AMQ9001: Kanalprogramm normal beendet.
ERKLÄRUNG: Das Kanalprogramm 'Janet' wurde normal beendet.
AKTION: Keine.
-----
02/02/01 15:09:27 AMQ7030: Anforderung zum Stilllegen des Warteschlangenmanagers
akzeptiert. Der Warteschlangenmanager wird gestoppt, wenn keine weiteren Aufgaben
mehr für ihn anstehen.
ERKLÄRUNG: Sie haben gefordert, dass der WS-Manager beendet wird, wenn keine
Arbeit mehr für ihn vorliegt. In der Zwischenzeit wird er den Start neuer
Anwendungen ablehnen, jedoch zulassen, dass alle aktiven Anwendungen ihre
Arbeit beenden können.
AKTION: Keine.
-----
02/02/01 15:09:32 AMQ8004: MQSeries-Warteschlangenmanager beendet.
ERKLÄRUNG: Der MQSeries-Warteschlangenmanager Janet wurde beendet.
AKTION: Keine.
...

```

EMS-Ereignisse

Für jeden Fehlereintrag in die Datei MQERRLG1 wird ein EMS-Ereignis generiert. Weitere Informationen zu EMS-Ereignissen finden Sie unter „EMS-Ereignisse“ auf Seite 178.

Warteschlangen für nicht zustellbare Nachrichten

Nachrichten, die aus irgendeinem Grund nicht zugestellt werden können, werden in die Warteschlange für nicht zustellbare Nachrichten gestellt. Mit dem MQSC-Befehl `DISPLAY QUEUE` können Sie überprüfen, ob die Warteschlange Nachrichten enthält. Wenn sich Nachrichten in der Warteschlange befinden, können Sie diese mit der zur Verfügung gestellten Beispieldatenanwendung (MQSBCG0E) und dem Aufruf `MQGET` durchsuchen. Die Beispieldatenanwendung durchsucht alle Nachrichten in einer angegebenen Warteschlange in einem angegebenen WS-Manager und zeigt den Nachrichtendeskriptor und die Felder mit dem Nachrichtenkontext aller Nachrichten in der angegebenen Warteschlange an.

Sie müssen entscheiden, was mit Nachrichten, die in der Warteschlange für nicht zustellbare Nachrichten gefunden werden, weiter geschehen soll; dies ist von den Gründen abhängig, aus denen die Nachrichten in die Warteschlange eingereiht wurden.

Probleme können dann entstehen, wenn Sie nicht in jedem von Ihnen verwendeten WS-Manager eine Warteschlange für nicht zustellbare Nachrichten definiert haben. Wenn Sie diese Warteschlange erstellt haben, müssen Sie das Attribut `DEADQ` des WS-Managers mit dem Befehl `runmqsc` ändern.

Konfigurationsdateien und Fehlerbestimmung

Fehler in Konfigurationsdateien führen häufig dazu, dass WS-Manager nicht gefunden werden und Fehler des Typs „WS-Manager nicht verfügbar“ auftreten.

Sie haben mehrere Möglichkeiten, die Konfigurationsdateien zu überprüfen:

- Stellen Sie sicher, dass die Konfigurationsdateien vorhanden sind.
- Stellen Sie sicher, dass für sie die entsprechenden Berechtigungen definiert sind.
- Stellen Sie sicher, dass die MQSeries-Konfigurationsdatei auf den richtigen WS-Manager und die richtigen Verzeichnisse verweist.

MQSeries-Trace verwenden

MQSeries for Compaq NSK verwendet die folgenden Befehle für die Trace-Funktion:

- `strmqtrc` – siehe „`strmqtrc` (MQSeries-Trace starten)“ auf Seite 315
- `dspmqtrc` – siehe „`dspmqtrc` (Formatierte MQSeries-Trace-Ausgabe anzeigen)“ auf Seite 287
- `endmqtrc` – siehe „`endmqtrc` (MQSeries-Trace beenden)“ auf Seite 294

Die Trace-Einrichtung verwendet für jede Definitionseinheit, für die ein Trace durchgeführt wird, eine eigene Datei, in der die Trace-Informationen aufgezeichnet werden.

Trace-Optionen werden in der Datei `QMINI` definiert.

Anmerkung: In MQSeries for Compaq NSK kann die Trace-Funktion auch über die MQM-Anzeige **Queue Manager Menu** gesteuert werden.

Namen von Trace-Dateien

Die Namen der Trace-Dateien im Teildatenträger für Fehlerprotokolle haben das folgende Format:

TRccpppp

Dabei steht *ccpppp* für die Prozess-ID (PID) des Prozesses, der den Trace erstellt hat. Die Prozess-ID setzt sich wie folgt zusammen:

cc (die Nummer der CPU).

pppp (die Nummer des Prozesses).

Findet die Trace-Funktion eine Trace-Datei mit einer identischen Prozessnummer, die nicht gelöscht wurde, überschreibt sie das letzte Zeichen der ID mit einem Buchstaben, d. h., es sind Ausgaben von 26 Prozessen mit derselben Prozess-ID möglich. Die Trace-Datei des ersten Prozesses mit der PID 00, 0315 würde beispielsweise den Namen TR000315 erhalten. Die Trace-Datei für den zweiten Prozess mit derselben ID (00, 0315) würde den Namen TR00031A erhalten.

Anmerkung: Aufgrund dieser Einschränkung sollten Trace-Dateien auf dem System gelöscht werden, nachdem sie überprüft wurden.

Beispiel für Trace-Daten

Das folgende Beispiel ist ein Auszug aus einer Trace-Datei:

```

...
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 zxcProcessChildren
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
...

```

Abbildung 44. Trace - Beispiel

Anmerkungen:

1. In diesem Beispiel sind nicht alle Daten dargestellt. Ein tatsächlicher Trace enthält die vollständigen Funktionsnamen und Rückkehrcodes.
2. Die Rückkehrcodes werden als Werte angegeben, nicht als Literale.

First Failure Support Technology™ (FFST)

Bei FFST-Fehlern handelt es sich in der Regel um schwerwiegende Fehler, die entweder auf einen Konfigurationsfehler im System oder einen internen MQSeries-Fehler hinweisen. In den meisten Fällen kann der WS-Manager trotzdem weiterarbeiten, es kann jedoch zu kurzfristigen Betriebsunterbrechungen für einige oder alle Anwendungen kommen. FFST-Fehler werden in der Datei ZMQSSYS.MQSYSLOG aufgezeichnet.

FFST-Daten überprüfen

Die Dateien erhalten einen Namen des Formats `FDnnnnn`; dabei gilt Folgendes:
`nnnnn` Gibt die ID des Prozesses an, der den Fehler meldet.

Wenn ein Prozess einen FFST-Bericht erstellt, erstellt er gleichzeitig auch ein EMS-Ereignis.

Ein typischer FFST-Bericht ist in Abb. 45 zu sehen.

```

+-----+
| 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.$DATA0.ZMQSEXE.MQMCACAL |
| UserID              :- MQM.MANAGER |
| Process File Name   :- \HURSLEY.$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 |
|                                         |
+-----+

```

Abbildung 45. First Failure Symptom Report - Beispiel

Es gibt jedoch eine Gruppe von Fehlern, die unter Umständen behoben werden können. Wenn der FFST-Bericht Meldungen wie „out of resource“ (Ressourcenengpass) oder „out of space on device“ (Speicherengpass auf Einheit) enthält, so weist dies wahrscheinlich darauf hin, dass die Kapazität des entsprechenden Systems überschritten wurde.

Dieses Problem kann behoben werden, indem Sie den entsprechenden Grenzwert erhöhen und den WS-Manager anschließend erneut starten.

Kapitel 15. Skalierbarkeit und Systemleistung

In diesem Kapitel wird erläutert, wie Sie die Leistung und Skalierbarkeit der MQSeries-Anwendungsprogramme optimieren können. Es ist in folgende Abschnitte gegliedert:

- „Einführung“
- „Permanente Nachrichten“ auf Seite 226
- „Nicht permanente Nachrichten“ auf Seite 227
- „Warteschlangenserver und Warteschlangendateien“ auf Seite 228
- „CPU-Zuordnung“ auf Seite 233
- „FASTPATH-Bindungen für Anwendungsprogramme“ auf Seite 234

Einführung

Durch eine Optimierung der Systemleistung und der Skalierbarkeit soll die Belastung der beiden Schlüsselressourcen, nämlich der CPU(s) und des Festplatten-subsystems, reduziert werden. Anwendungen, die weniger CPU-Zeit belegen und mit weniger E/A-Zugriffen auf die Festplatte auskommen, sind schneller und flexibler (beispielsweise können sie so konfiguriert werden, dass sie durch eine maximale Ausnutzung der Hardware und der Systemsoftware immer größere Datenmengen bewältigen können).

In diesem Kapitel werden Verfahren vorgestellt, mit denen die Leistung von Anwendungsprogrammen und von MQSeries verbessert werden können. Die folgenden Abschnitte geben eine Übersicht über die allgemeinen Konzepte zur Leistungsoptimierung von Anwendungsprogrammen.

Neue Anwendungen im Hinblick auf Leistung und Skalierbarkeit erstellen

Schon beim Konzipieren neuer Anwendungen sollten Sie überlegen, wie diese neuen Anwendungen MQSeries und andere Subsysteme verwenden sollen. Welche MQSeries-Merkmale benötigt werden oder wichtig sind hängt davon ab, für welchen Zweck die einzelnen Anwendungen erstellt werden. Einige MQSeries-Merkmale, wie beispielsweise die Nachrichtenpermanenz, erfordern die Gewährleistung der Datenintegrität und der Nachrichtenzustellung, und es muss daher von einer höheren CPU-Belegung und mehr E/A-Zugriffen auf die Festplatte ausgegangen werden. Ist dies für eine bestimmte Anwendung nicht erforderlich, kann die Leistung von MQSeries durch eine entsprechende Konfiguration deutlich erhöht werden. Dies sollte bereits in einer frühen Phase der Anwendungsentwicklung entschieden werden.

Verwendung gemeinsam benutzter Ressourcen minimieren oder eliminieren

Damit der wachsende Nachrichtenverkehr aufgefangen werden kann, müssen die Kapazitäten der Hardware und Systemsoftware vollständig genutzt werden. Bei gemeinsam verwendeten Ressourcen kann es bei zunehmender Auslastung zu Engpässen kommen. Diese Engpässe entstehen entweder durch eine höhere Beanspruchung dieser gemeinsam benutzten Ressourcen (wenn beispielsweise immer mehr Benutzer dieselbe CPU verwenden) oder durch ihre intensivere Nutzung (wenn beispielsweise die einzelnen Benutzer immer mehr arbeiten).

Skalierbarkeit und Systemleistung - Einführung

Unter dem Gesichtspunkt der Leistung und der Skalierbarkeit verdienen vor allem die CPU und das Festplattensystem Aufmerksamkeit, da diese Ressourcen häufig gemeinsam benutzt werden.

MQSeries-Prozesse können auf eine beliebige Anzahl von CPUs verteilt werden. Auf einem System mit mehreren CPUs kann durch die Verteilung der MQSeries-Prozesse auf die jeweils verfügbaren CPUs eine bessere Leistung erzielt werden als wenn die Prozesse einer Standard-CPU zugeordnet werden. Ebenso kann die Nutzung des Plattensubsystems optimiert werden, indem die verschiedenen Warteschlangendateien nach Möglichkeit in unterschiedlichen Teildatenträgern untergebracht werden, für die jeweils eigene Compaq NSK-Festplattenprozesse zuständig sind.

Leistungsoptimierung (Durchsatzverbesserung) als iterativer Prozess

Eine effektive Durchsatzverbesserung wird durch ständige Leistungsmessung, Anpassung, erneute Leistungsmessungen usw. erreicht. Bei jedem Optimierungszyklus sollte immer nur eine Variable geändert werden, damit das Ergebnis mit dem Ergebnis verglichen werden kann, das durch die Änderung anderer Variablen erreicht wird. Es wird davon abgeraten, mehrere Einstellungen auf einmal zu ändern, da einige Änderungen die Gesamtleistung erheblich verbessern, andere dagegen reduzieren können. Oft beeinflusst ein einziger Engpass die Leistung eines Anwendungssystems. Wird eine Optimierung vorgenommen, führt dies in der Regel dazu, dass nun eine andere Ressource einen Engpass verursacht. Dies macht deutlich, dass eine Leistungsverbesserung nur durch sorgfältige schrittweise Anpassungen erreicht werden kann.

Nehmen wir an, eine Anwendung schreibt eine große Anzahl kurzer permanenter Nachrichten in eine Warteschlange. Hier kann es hilfreich sein, wenn Sie herausfinden, was passiert, wenn z. B. stattdessen nicht permanente Nachrichten übertragen werden oder wenn dieselben Daten in einer kleineren Anzahl großer Nachrichten untergebracht werden. Beide Änderungen führen (in der Regel) zu einer Leistungsverbesserung, aber ohne einen sorgfältigen Testzyklus für jeden dieser Fälle werden Sie nicht feststellen können, welche Änderung die bessere Lösung ist.

Permanente Nachrichten

Permanente Nachrichten bieten die höchste Sicherheit, die MQSeries hinsichtlich Zustellung und Wiederherstellbarkeit bietet. Permanente Nachrichten werden immer auf Platte gespeichert und sind daher auch nach einem Neustart des WS-Managers noch vorhanden. Nonstop-TM/MP erstellt Prüfprotokolle für Warteschlangendateien, so dass beim Abrufen bzw. Einreihen permanenter Nachrichten sowohl Lese- und Schreibzugriffe auf die Warteschlangendatei selbst als auch auf die TM/MP-Prüflistendateien erfolgen. Die Erstellung von TM/MP-Prüfprotokollen ist erforderlich, um auch bei System- oder Hardwareausfällen die transaktionale Integrität für permanente Nachrichten sicherzustellen. Die Auswirkung dieser TM/MP-Prüfprotokolle für permanente Nachrichten auf die Leistung muss bei der Konzeption von MQSeries-Anwendungen berücksichtigt werden.

Die Permanenz ist eine Eigenschaft der Nachricht, nicht der Warteschlange, in der sie gespeichert wird. Warteschlangen können daher sowohl permanente als auch nicht permanente Nachrichten enthalten; der Administrator kann jedoch angeben, ob neue Nachrichten permanent sein sollen, sofern von der Anwendung, die die

Nachrichten einreih, nicht anders vorgegeben. (Siehe die Beschreibung des Warteschlangenattributs DEFPSIST im Handbuch *MQSeries Application Programming Reference*.)

Nicht permanente Nachrichten

Anders als permanente Nachrichten werden nicht permanente Nachrichten nicht auf Platte gespeichert und sind daher nach einem Neustart des WS-Managers nicht mehr vorhanden. Je nach Konfiguration des Warteschlangenservers für die Warteschlange können jedoch für nicht permanente Nachrichten Prüfpunktinformationen in den Backup-Warteschlangenserver geschrieben werden. Die Verwendung nicht permanenter Nachrichten erfolgt vor allem unter dem Gesichtspunkt der Systemleistung. Für permanente Nachrichten muss eine gesicherte Zustellung und Wiederherstellbarkeit gewährleistet werden, daher erfolgen beim Abrufen bzw. Einreihen dieser Nachrichten sowohl Lese- und Schreibzugriffe auf die Warteschlangendateien selbst als auch auf die TM/MP-Prüflistendateien. Diese Plattenaktivität verringert den Durchsatz sowohl der Anwendungen, die permanente Nachrichten abrufen oder einreihen, als auch der Kanäle, über die permanente Nachrichten an andere WS-Manager übertragen werden.

Werden nach Möglichkeit nicht permanente Nachrichten verwendet, kann eine beträchtliche Durchsatzverbesserung erreicht werden, da diese Nachrichten nicht auf Platte geschrieben, sondern von den Nonstop-Warteschlangenservern in einem Cache im Hauptspeicher zwischengespeichert werden.

Die Permanenz ist eine Eigenschaft der Nachricht, nicht der Warteschlange, in der sie gespeichert wird. Warteschlangen können daher sowohl permanente als auch nicht permanente Nachrichten enthalten; der Administrator kann jedoch angeben, ob neue Nachrichten permanent sein sollen, sofern von der Anwendung, die die Nachrichten einreih, nicht anders vorgegeben (siehe das Warteschlangenattribut DEFPSIST).

Nicht permanente Nachrichten und Kanäle

Nachrichtenkanäle stellen durch die Erstellung von Synchronisierprotokollen am Empfänger- und Senderende sicher, dass Nachrichten nur einmal über das Netz übertragen werden. Diese Synchronisierprotokolle werden zusätzlich zu den TM/MP-Prüfprotokollen (für den WS-Manager) erstellt, und zwar wenn die Kanäle Nachrichten in die Warteschlangen einreihen bzw. aus den Warteschlangen abrufen. Nachrichtenkanäle können so konfiguriert werden, dass beim Senden bzw. Empfangen nicht permanenter Nachrichten keine Synchronisierprotokolle erstellt werden; dazu wird das Attribut NPMSPEED auf FAST gesetzt. Über das Kanalattribut NPMSPEED wird das Verhalten von Sende- und Empfangskanälen gesteuert, über die nicht permanente Nachrichten übertragen werden. Wird für einen Kanal das Attribut NPMSPEED auf NORMAL gesetzt, werden nicht permanente Nachrichten als Teil des (über die Attribute BATCHINT und BATCHSZ festgelegten) Nachrichtenstapels des Kanals verarbeitet, d. h., für diese Nachrichten muss wie für permanente Nachrichten ein Synchronisierprotokoll erstellt werden. Außerdem werden bei dieser Einstellung nicht permanente Nachrichten synchronisationspunktgesteuert abgerufen und gelesen, wodurch am Beginn und am Ende der Transaktion Zugriffe auf die TM/MP-Prüflistendatei erfolgen.

Wird NPMSPEED auf FAST gesetzt, werden nicht permanente Nachrichten nicht als Teil des aktuellen Nachrichtenstapels des Kanals verarbeitet, sondern ohne Synchronisationspunktsteuerung in die Warteschlangen eingereiht bzw. aus den Warteschlangen abgerufen. Da anders als beim synchronisationspunktgesteuerten

Nicht permanente Nachrichten

Einreihen und Abrufen von Nachrichten keine Protokolle, wie sie für die Synchronisation bei der Stapelverarbeitung im Kanal erforderlich sind, und keine TM/MP-Prüfprotokolle erstellt werden, werden durch die Einstellung NPMSPEED(FAST) zwei Ursachen für Plattenzugriffe beseitigt.

Mit der Einstellung NPMSPEED(FAST) wird eine deutliche Durchsatzverbesserung erreicht, jedoch auf Kosten der Wiederherstellbarkeit bei Ausfällen, da nicht permanente Nachrichten z. B. bei Kanal- oder Netzausfällen verloren gehen.

Die Standardeinstellung von NPMSPEED ist FAST.

Warteschlangenserver und Warteschlangendateien

Bei Warteschlangenservern handelt es sich um MQSeries-Prozesse, die das Einreihen, Abrufen und Speichern von Nachrichten steuern. Sie sind daher eine der wichtigsten MQSeries-Komponenten, die Sie sich genauer ansehen sollten. Die Konfiguration der Warteschlangenserver kann sich erheblich auf die Leistung eines intensiv genutzten MQSeries-Systems auswirken.

Warteschlangenserver sind für die physische Speicherung der Nachrichten in einer Warteschlange zuständig. Auf das Speichern permanenter und nicht permanenter Nachrichten soll hier getrennt eingegangen werden, da hier grundsätzliche Unterschiede bestehen.

Ein Warteschlangenserver kann für eine oder mehrere Warteschlangen zuständig sein. Bei ihrer Erstellung wird eine Warteschlange einem Standardwarteschlangenserver zugeordnet. Daher werden alle Warteschlangen vom Standardwarteschlangenserver gesteuert, sofern sie keinem anderen Warteschlangenserver zugeordnet werden.

Für jede von einem Warteschlangenserver verwaltete Warteschlangen sind die folgenden Dateien vorhanden:

- Eine Warteschlangendatei
- Eine Warteschlangenüberlaufdatei
- Eine Änderungsdatei

Zusätzlich kann für jede große Nachricht in der Warteschlange (die Größe wird von dem für die Warteschlange festgelegten Schwellenwert für Nachrichtenüberlauf festgelegt) eine Nachrichtenüberlaufdatei vorhanden sein. Auf diese Nachrichtenüberlaufdateien gehen wir später näher ein.

Speicherverfahren für permanente Nachrichten

Permanente Nachrichten werden immer auf Platte gespeichert. Wie sie gespeichert werden, hängt hauptsächlich von ihrer Größe ab:

Nachrichtengröße	Speicherverfahren
< 3000 Bytes (ca.)	Die gesamte Nachricht wird in der Warteschlangendatei gespeichert.
> 3000 Bytes (ca.) <= Schwellenwert für Nachrichtenüberlauf	Die ersten 3000 Bytes (ca.) werden in der Warteschlangendatei gespeichert, der Rest in der Warteschlangenüberlaufdatei
> Schwellenwert für Nachrichtenüberlauf	Die ersten 3000 Bytes (ca.) werden in der Warteschlangendatei gespeichert, der Rest in der Nachrichtenüberlaufdatei

Speicherverfahren für nicht permanente Nachrichten

Nicht permanente Nachrichten werden in Speicherpuffern des Warteschlangenserverprozesses gespeichert. Daher sind für den Warteschlangenserver beim Abrufen und Einreihen nicht permanenter Nachrichten keine E/A-Plattenzugriffe erforderlich.

Verteilung der Warteschlangenserver auf die CPUs

Bei Warteschlangenservern handelt es sich um PATHWAY-Serverklassen, daher werden sie in der PATHWAY-Konfiguration des WS-Managers definiert. Bei der Erstellung des WS-Managers wird in der PATHWAY-Konfiguration ein Standardwarteschlangenserver definiert, wobei der primäre Warteschlangenserverprozess in CPU 0 läuft. Da jedoch in CPU 0 normalerweise viele der Systemprozesse mit einer hohen Priorität laufen, sollte der Warteschlangenserver nicht ebenfalls dieser CPU zugeordnet werden. Es ist besser, wenn Sie eine Reihe von CPUs mit geringer Auslastung angeben und in jeder dieser CPUs einen Warteschlangenserver erstellen. Ein Warteschlangenserver verwaltet nur die Warteschlangen, die ihm explizit zugeordnet werden.

Warteschlangen mit dem Befehl 'altmqfls' anderen Warteschlangenservern zuordnen

Bei der Erstellung einer Warteschlange (in der Regel mit dem Befehl `runmqsc`) ordnet MQSeries diese dem Standardwarteschlangenserver zu. Alle Warteschlangen werden von diesem Standardwarteschlangenserver verwaltet, sofern Sie keine neuen Warteschlangenserver erstellen und diesen Warteschlangen zuordnen.

In einem MQSeries-System mit hoher Auslastung ist die Zuordnung aller Warteschlangen zu einem einzigen Warteschlangenserver weder effizient, noch ermöglicht sie Flexibilität. Werden neue Warteschlangenserver erstellt und diesen Warteschlangen zugeordnet, kann dadurch eine gleichmäßigere CPU-Auslastung durch die Verteilung der Warteschlangenserver auf die verfügbaren CPUs erreicht werden.

Mit dem Befehl `altmqfls` können Sie eine Warteschlange einem anderen Warteschlangenserver zuordnen. Beispiel:

```
altmqfls --qmgr QMGR --type QLOCAL --server $QS01 TEST.QUEUE
```

Weitere Informationen zum Befehl `altmqfls` finden Sie unter „altmqfls (Attribute der Warteschlangendateien ändern)“ auf Seite 260.

Cluster-Übertragungswarteschlange: SYSTEM.CLUSTER.TRANSMIT.QUEUE

MQSeries verwendet eine Cluster-Übertragungswarteschlange für alle Clusteroperationen. Gehört Ihr WS-Manager zu einem stark ausgelasteten Cluster, sollte diese Warteschlange einem dedizierten Warteschlangenserver zugeordnet werden, damit Clusteroperationen optimal und ohne größere Auswirkung auf andere Anwendungen ausgeführt werden können.

Warteschlangenserver und Warteschlangendateien

Warteschlangen mit dem Befehl 'altmqfls' verschieben

Standardmäßig werden die drei primären Warteschlangendateien (die Warteschlangendatei selbst, die Warteschlangenüberlaufdatei und die Änderungsdatei) im Teildatenträger **M** des WS-Managers gespeichert. Nachrichtenüberlaufdateien werden im Teildatenträger des Warteschlangenservers abgelegt.

Es gibt zwei Gründe, diese Warteschlangendateien in andere Datenträger zu verschieben:

- Um die E/A-Zugriffe auf die Platte gleichmäßiger zu verteilen
- Um die Begrenzung zu umgehen, die vom Betriebssystem durch die Anzahl der Dateien, die maximal pro Datenträger geöffnet werden können, vorgegeben wird

Mit dem Befehl **altmqfls** können Sie vorhandene Warteschlangen in einen anderen Teil der Platte verschieben:

```
altmqfls --qmgr QMGR --type QLOCAL --volume $DATA01 TEST.QUEUE
```

Warteschlangen müssen geschlossen werden, bevor sie mit dem Befehl **altmqfls** verschoben werden.

Nachrichtenüberlaufdateien können nicht verschoben werden. Sie können jedoch unter Angabe der Option **--msgofsubvol** im Befehl **altmqfls** Nachrichtenüberlaufdateien an einer beliebigen Stelle erstellen.

Mit dem folgenden Befehl können Sie neue Nachrichtenüberlaufdateien an einer neuen Adresse erstellen:

```
altmqfls --qmgr QMGR --type QLOCAL --msgofsubvol $DATA01.TESTMOF TEST.QUEUE
```

Weitere Informationen zum Befehl **altmqfls** finden Sie unter „altmqfls (Attribute der Warteschlangendateien ändern)“ auf Seite 260.

Warteschlangendateien partitionieren

Durch die Partitionierung von Dateien wird eine Datei auf mehrere Datenträger verteilt. Die Datei besteht also logisch aus mehreren Partitionen, die sich jeweils in einem eigenen Datenträger befinden.

Dieses Verfahren hat folgende Vorteile:

- Durch die Partitionierung werden E/A-Plattenzugriffe für eine einzelne Warteschlange auf mehrere Datenträger verteilt, und es sind daher mehrere E/A-Pfade möglich.
- Durch die Partitionierung können logische Warteschlangendateien verwendet werden, deren Größe über dem für einen Datenträger vorgegebenen maximalen Wert liegt.

Sie können eine vorhandene Warteschlangendatei mit den TACL-Standardbefehlen (das Dienstprogramm FUP) partitionieren. MQSeries stellt ein TACL-Script (PARTIT) zur Verfügung, das veranschaulicht, wie eine vorhandene Warteschlangendatei mit Hilfe von FUP partitioniert werden kann.

Warteschlangenserver und Warteschlangendateien

Nach der Partitionierung einer Warteschlangendatei versucht der Warteschlangenserver, neue Nachrichten möglichst gleichmäßig auf alle verfügbaren Partitionen zu verteilen. Dies ist ein weiterer Vorteil der Partitionierung.

Anmerkung: Partitionierte Warteschlangendateien können nicht mit dem Befehl `altmqfls` verschoben.

Nachrichtenüberlaufdateien

Nachrichtenüberlaufdateien werden für alle Nachrichten erstellt, deren Größe über dem Schwellenwert liegt, der für den Nachrichtenüberlauf definiert wurde. Bei diesen Dateien handelt es sich um unstrukturierte Dateien, für die von TM/MP keine Prüfprotokolle erstellt werden. Aus Gründen der Effizienz wird geraten, bei sehr großen Nachrichten den größten Teil der Nachricht in der Nachrichtenüberlaufdatei zu speichern (für die von TM/MP keine Prüfprotokolle erstellt werden). Bei kleineren Nachrichten ist es dagegen besser, die gesamte Nachricht in den von TM/MP protokollierten Warteschlangendateien zu speichern (die Warteschlangendatei selbst und die Warteschlangenüberlaufdateien). 200 KB wurden als empirischer Grenzwert ermittelt.

Bei der Erstellung einer Warteschlange wird der Schwellenwert für Nachrichtenüberlauf auf den Standardwert von 200 KB gesetzt. Dieser Schwellenwert kann über die Angabe der Option `--oflowsize` im Befehl `altmqfls` geändert werden.

Mit dem Befehl `altmqfls` können Sie den Schwellenwert für den Nachrichtenüberlauf wie folgt auf einen anderen Wert setzen:

```
altmqfls --qmgr QMGR --type QLOCAL --oflowsize 400000 TEST.QUEUE
```

Nachrichten zum Durchsuchen im Hauptspeicher ablegen

Der Warteschlangenserver kann die ersten n Bytes aller permanenten Nachrichten in einer Warteschlange im Hauptspeicher halten. Dies kann erhebliche Auswirkungen auf Anwendungen haben, die permanente Nachrichten in einer Warteschlange durchsuchen. Liegt die Größe der Nachrichten, die von der Anwendung durchsucht werden, unter diesem Schwellenwert, so sind beim Durchsuchen dieser permanenten Nachrichten keine E/A-Plattenzugriffe erforderlich.

Darüber hinaus können Sie beispielsweise in den ersten 100 Bytes der permanenten Nachrichten Informationen einfügen, die Aufschluss über den Inhalt der Nachricht geben. Dies gibt die Möglichkeit, bei der Suche nach bestimmten Nachrichten die jeweils ersten 100 Bytes aller Nachrichten zu durchsuchen, ohne dass hierzu E/A-Plattenzugriffe erforderlich sind. Wenn die gewünschte Nachricht gefunden wird, kann die gesamte Nachricht gelesen und gegebenenfalls abgerufen werden.

Mit der Option `--browse` im Befehl `altmqfls` können Sie den Schwellenwert bei Suchvorgängen in einer Warteschlange festlegen. Standardmäßig werden im Hauptspeicher keine Daten permanenter Nachrichten gespeichert, d. h., der Schwellenwert ist auf null gesetzt. Der maximal zulässige Wert ist 25000.

```
altmqfls --qmgr QMGR --type QLOCAL --browse 100 TEST.QUEUE
```

Warteschlangenserver und Warteschlangendateien

Weitere Warteschlangenserveroptionen

Für weitere Warteschlangenserveroptionen (z. B. ob die Warteschlange beim Start des WS-Managers von der Platte in den Cache geladen werden soll oder ob für nicht permanente Nachrichten Prüfpunktinformationen im Backup-Warteschlangenserver gespeichert werden sollen) können Sie den Parameter `--qsoptions` mit den Optionen `L`, `S` und `C` verwenden. Diese Optionen können entweder einzeln oder zusammen angegeben werden, um die Wiederherstellbarkeit und Zuverlässigkeit einer Warteschlange optimal einzustellen.

Anmerkung: Die Option `--qsoptions` kann nur jeweils einmal in der Befehlszeile angegeben werden, daher werden bei jedem Befehl immer *alle* drei Optionen (`S`, `L` und `C`) gesetzt. Bei Angabe von `--qsoptions S` beispielsweise wird der Schalter `S` gesetzt und die Schalter `L` und `C` (da sie nicht angegeben wurden) zurückgesetzt.

Laden beim Start

Diese Option wird über den Schalter `S` im Befehl `altmqfls --qsoptions` gesetzt. Bei Angabe dieser Option liest der Warteschlangenserver die Warteschlangendateien und erstellt die internen Nachrichtendatenstrukturen beim Start (in der Regel erfolgt dieser beim Start des WS-Managers), und nicht erst dann, wenn die Warteschlange zum ersten Mal geöffnet wird.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions S TEST.QUEUE
```

Durch diese Option werden zwar CPU-Auslastung und E/A-Plattenzugriffe beim ersten Öffnen einer Warteschlange gering gehalten, dies führt allerdings zu einer höheren Aktivität des Warteschlangenservers (CPU-Auslastung und E/A-Plattenzugriffe) beim Start des WS-Managers.

Sperre im Cache

Diese Option wird über den Schalter `L` im Befehl `altmqfls --qsoptions` gesetzt. Bei Angabe dieser Option sperrt der Warteschlangenserver die Datenstrukturen und Datenketten einer Warteschlange im Speicher. Die Datenstrukturen der Warteschlange werden nicht aus dem Hauptspeicher auf die Platte geladen, um Platz für weitere Warteschlangen zu schaffen. Standardmäßig werden die Datenstrukturen der Warteschlange bei Bedarf auf die Platte geladen.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions L TEST.QUEUE
```

Durch die Angabe dieser Option ist ein schnellerer Zugriff auf die Datenstrukturen der Warteschlange im Hauptspeicher möglich, allerdings möglicherweise auf Kosten anderer Warteschlangen.

Prüfpunktinformationen für nicht permanente Nachrichten

Diese Option wird über den Schalter `C` im Befehl `altmqfls --qsoptions` gesetzt. Bei Angabe dieser Option zeichnet der Warteschlangenserver Prüfpunktinformationen für nicht permanente Nachrichten mit seinem NonStop-Backup-Prozess auf. Standardmäßig werden für nicht permanente Nachrichtendaten Prüfpunktinformationen aufgezeichnet.

Warteschlangenserver und Warteschlangendateien

Diese Option sorgt für eine höhere Verfügbarkeit nicht permanenter Nachrichten, allerdings ist hierdurch eine vermehrte Kommunikation zwischen den Prozessen sowie eine größere CPU- und Speicherauslastung für den primären und sekundären Warteschlangenserverprozess erforderlich.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions C TEST.QUEUE
```

Längenzähler

Diese Option wird über den Schalter `--meascount` im Befehl `altmqfls` gesetzt. Bei Angabe dieser Option führt der Warteschlangenserver einen benutzerdefinierten Längenzähler für das Attribut `CURDEPTH` der Warteschlange. Dieser Längenzähler kann dazu verwendet werden, Daten über die Gesamtleistung eines Systems zu ermitteln. Über diese Option können Sie die Schreib- und Lesezugriffe auf Nachrichten (MQGET- und MQPUT-Vorgänge) in einer Warteschlange mit anderen systemspezifischen Variablen wie beispielsweise der CPU-Auslastung und E/A-Plattenzugriffen vergleichen.

```
altmqfls --qmgr QMGR --type QLOCAL --meascount TESTCOUNT1 TEST.QUEUE
```

Mit Hilfe dieser Option können Sie die Leistung und Skalierbarkeit eines MQSeries-Systems ermitteln und damit bessere Ergebnisse bei der Durchsatzverbesserung erzielen.

CPU-Zuordnung

Beim Start eines WS-Managers erstellt MQSeries for Compaq NSK eine Reihe von Prozessen. Einige dieser Prozesse sind für die Ausführung wichtiger Messaging-Operationen zuständig, andere wiederum unterstützen diese Operationen indirekt. Prozesse für die wichtigen Messaging-Operationen von MQSeries sind dann aktiv, wenn Anwendungen MQI-Aufrufe wie `MGGET` oder `MQPUT` ausgeben.

Folgende MQSeries- und Compaq NSK-Prozesse sind an wichtigen Messaging-Operationen beteiligt:

- Lokale WS-Manageragenten (LQMA, Local Queue Manager Agent)
- Warteschlangenserver
- Statusserver
- NSK-Plattenprozesse

Folgende MQSeries- und NSK-Prozesse sind an Operationen in Zusammenhang mit der verteilten Steuerung von Warteschlangen beteiligt:

- Nachrichtenkanalagenten (MCA, Message Channel Agent)
- Standardstatusserver
- NSK-TCPIP- oder NSK-SNA-Prozesse

Folgende MQSeries-Prozesse sind an unterstützungs- oder verwaltungsspezifischen Operationen beteiligt:

- WS-Managerserver
- Repository-Server
- Ausführungs-Controller (einschließlich des ECBoss)
- Kanalinitiatoren
- Auslösemonitore
- Empfangsprogramme

CPU-Zuordnung

In einem stark ausgelasteten MQSeries-System wird die CPU in der Regel vor allem durch Prozesse der ersten beiden Kategorien belegt (Messaging-Prozesse und Prozesse in Zusammenhang mit der verteilten Steuerung von Warteschlangen).

Um eine gleichmäßige Verteilung der CPU-Last in MQSeries zu erreichen, werden daher in der Regel die lokalen WS-Manageragenten, Warteschlangenserver und Nachrichtenkanalagenten des WS-Managers auf möglichst viele CPUs verteilt.

Die Standardkonfiguration eines neuen WS-Managers legt CPU 0 für die Prozesse fest (bei Nonstop-Prozesspaaren laufen die primären Prozesse in CPU 0, die Backup-Prozesse in CPU 1). Für eine Produktionsumgebung ist dies allerdings nicht geeignet, daher sollten Sie die PATHWAY-Konfiguration des WS-Managers ändern, damit häufig aufgerufene Prozesse auf sinnvolle Weise auf möglichst viele CPUs verteilt werden. CPU 0 enthält in der Regel eine große Anzahl von Betriebssystemprozessen mit hoher Priorität und sollte daher nicht ausgerechnet für wichtige MQSeries-Prozesse verwendet werden.

Die Verteilung der in der PATHWAY-Konfiguration des WS-Managers definierten Ausführungs-Controller auf CPUs wirkt sich wiederum auf die Ausführung der lokalen WS-Manageragenten und der Nachrichtenkanalagenten aus. Warteschlangenserver werden ebenfalls in der PATHWAY-Konfiguration des WS-Managers definiert.

Neben den MQSeries-Prozessen spielen auch die Plattenprozesse des NSK-Betriebssystems eine Rolle bei den Messaging-Operationen (insbesondere bei der Verarbeitung permanenter Nachrichten durch den Warteschlangenserver). Bei einem intensiven Einsatz der verteilten Steuerung von Warteschlangen kommt es zu einer hohen CPU-Belegung durch die entsprechenden TCPIP- oder SNA-Prozesse. Bei einer Bewertung der Gesamtleistung einer MQSeries-Installation sollten Sie die Anzahl dieser Prozesse und ihre Verteilung auf die CPUs berücksichtigen.

Informationen zur Zuordnung von Prozessen zu CPUs, zur Konfiguration der Plattenprozesse oder der TCPIP-Prozesse finden Sie in der entsprechenden Dokumentation für das Compaq NSK-System. Informationen zur Konfiguration der SNA-Prozesse finden Sie in der entsprechenden SNAX- oder ICE-Dokumentation, Informationen zur Konfiguration der TCPIP- oder SNA-Prozesse finden Sie in „Anhang M. Kommunikation einrichten“ auf Seite 383.

FASTPATH-Bindungen für Anwendungsprogramme

FASTPATH-Bindungen (Bindungen für Direktaufruf) ist ein MQI-Merkmal, das eine effizientere Ausführung von MQSeries-Anwendungen ermöglicht. Mit FASTPATH-Bindungen lässt sich der Gesamtaufwand bei allen MQI-Aufrufen, die von MQSeries-Anwendungsprogrammen ausgegeben werden, verringern. Anwendungen, die diese FASTPATH-Bindungen verwenden, werden als sichere Anwendungen bezeichnet, und zwar auf Grund der Nähe zwischen WS-Managersoftware / WS-Managerspeicher und der Anwendungssoftware des Benutzers. Fehler in sicheren Anwendungen können zu fehlerhaften MQSeries-Datenstrukturen führen und damit die Integrität des WS-Managers in Frage stellen.

Hintergrund

Wenn ein Anwendungsprogramm einen MQCONN-Aufruf ausgibt, wird von MQSeries ein besonderer Prozess erstellt bzw. wiederverwendet, der so genannte lokale WS-Manageragent (LQMA, Local Queue Manager Agent). Dieser LQMA bearbeitet alle nachfolgenden MQI-Aufrufe, die von der Anwendung, die diese Verbindungskennung verwendet, ausgegeben werden. In MQSeries for Compaq NSK kann der LQMA in derselben CPU wie die verbundene Anwendung oder in einer anderen CPU aktiv sein.

Da es sich bei dem LQMA um einen eigenen Prozess handelt, benötigt das Anwendungsprogramm keinen direkten Zugriff auf den von MQSeries verwendeten Speicher bzw. die von MQSeries verwendeten Dateien. Der LQMA kann daher nicht durch ein unerwünschtes Anwendungsprogramm geschädigt werden. Auf diese Weise sind die MQSeries-Software und -Datenstrukturen, die für die Operation kritisch sind, von der Software und den Daten der Kundenanwendung getrennt. Diese Trennung hat allerdings ihren Preis. Der MQCONN-Aufruf kann erst ausgeführt werden, wenn ein neuer LQMA-Prozess erstellt bzw. ein vorhandener LQMA-Prozess wiederverwendet wird; der größere Systemaufwand wird jedoch dadurch verursacht, dass bei jedem MQI-Aufruf, den die Anwendung ausgibt, Informationen an den LQMA übergeben werden müssen.

Diese MQI-Informationen werden über ein Verfahren zur Interprozesskommunikation (IPC, Interprocess Communication) an den LQMA übergeben. Diese IPC-Anforderungen können entweder innerhalb einer CPU (wenn sich der LQMA in derselben CPU wie die verbundene Anwendung befindet) oder zwischen verschiedenen CPUs erfolgen, was natürlich einen höheren Aufwand bedeutet.

Systemaufwand bei MQI-Aufrufen reduzieren

Anwendungsentwickler können mit Hilfe von FASTPATH-Verbindungen den Aufwand bei MQI-Aufrufen vermeiden, der sich aus der Kommunikation zwischen Anwendung und LQMA ergibt (ebenso wird auch die Erstellung des LQMA-Prozesses vermieden). Bei der Aktivierung von FASTPATH-Bindungen für eine Anwendung wird kein eigener LQMA-Prozess verwendet. Statt dessen werden die MQSeries-Komponenten, die normalerweise der LQMA enthält, in den Prozess des Benutzers geladen (z. B. in den Prozess der verbundenen Anwendung).

Für die nachfolgenden MQI-Aufrufe, die von der Anwendung ausgegeben werden, ist keine Interprozesskommunikation mit dem LQMA erforderlich, da jetzt die MQSeries-Software und die MQSeries-Datenstrukturen, die sich normalerweise im LQMA-Prozess befinden, im Anwendungsprozess enthalten sind. Die Kommunikation des WS-Managers mit anderen MQSeries-Prozessen, die eventuell stattfindet, erfolgt nach wie vor über die Interprozesskommunikation. Durch FASTPATH-Bindungen kann also nicht die gesamte Interprozesskommunikation vermieden werden, sie kann jedoch deutlich verringert werden.

FASTPATH-Bindungen für Anwendungsprogramme

FASTPATH-Bindungen aktivieren

Damit die FASTPATH-Bindung verwendet werden kann, müssen Sie mit dem MQCONNX-Aufruf unter Angabe der Option MQCNO_FASTPATH_BINDING eine Verbindung zum WS-Manager herstellen. Über den Wert (falls angegeben) der Umgebungsvariable MQCONNECTTYPE kann das Verhalten von MQCONNX gesteuert werden. Ist diese Umgebungsvariable vorhanden, muss sie auf FASTPATH gesetzt sein, damit der MQCONNX-Aufruf eine FASTPATH-Verbindung herstellen kann. Nachdem eine FASTPATH-Verbindung hergestellt wurde, verhalten sich alle anschließend ausgegebenen MQI-Aufrufe mehr oder weniger wie bei STANDARD-Verbindungen (Abweichungen sind im nächsten Abschnitt aufgeführt).

Einschränkungen bei der Verwendung von FASTPATH-Bindungen

Bei der Verwendung von STANDARD-Verbindungen sind die Anwendungssoftware und die internen MQSeries-Daten voneinander getrennt. Beim Herstellen einer FASTPATH-Verbindung durch eine sichere Anwendung ist diese Trennung nicht mehr gegeben. Fehler in sicheren Anwendungen können zu fehlerhaften MQSeries-Datenstrukturen führen und damit die Integrität des WS-Managers in Frage stellen. Dies muss berücksichtigt werden, wenn FASTPATH-Bindungen für eine Anwendung in Betracht gezogen werden.

Zusätzlich gilt für sichere Anwendungen noch Folgendes:

- Die Verbindung sicherer Anwendungen mit WS-Managern **muss** explizit beendet werden (z. B. über MQDISC).
- Sichere Anwendungen müssen gestoppt werden, bevor der Befehl **endmqm** ausgegeben wird. Sichere Anwendungen werden unter der Benutzer-ID des Administrators ausgeführt. (Dies entspricht dem Principal **mqm**.)
- Sichere Anwendungen können nur in CPUs laufen, die einen aktiven MQSeries-Repository-Server enthalten.

Kapitel 16. Datenintegrität und Verfügbarkeit

In diesem Kapitel werden die Konzepte der Datenintegrität und der Verfügbarkeit beschrieben; weiter wird erläutert, in wieweit diese wichtigen Aspekte eines Systems für die Verwaltung und die Konfiguration von MQSeries for Compaq NSK V5.1 von Bedeutung sind. In diesem Kapitel werden die verschiedenen Stufen der Datenintegrität und Verfügbarkeit vorgestellt, die in MQSeries möglich sind, und Sie erfahren, mit welchen Konfigurationsoptionen diese Stufen realisiert werden können. Dieses Kapitel ist in folgende Abschnitte gegliedert:

- „Datenintegrität“
- „Verfügbarkeit“ auf Seite 238
- „Permanente und nicht permanente Daten“ auf Seite 239
- „Datenbankkonsistenz“ auf Seite 242
- „Kritische Prozesse“ auf Seite 246
- „Cluster“ auf Seite 253
- „Konfigurationsoptionen für die Verfügbarkeit“ auf Seite 253
- „Konfigurationsoptionen für die Datenintegrität“ auf Seite 254

Sie sollten mit dem Inhalt von „Kapitel 2. MQSeries for Compaq NSK V5.1 - Architektur“ auf Seite 23 vertraut sein, um die Informationen in diesem Kapitel sinnvoll anwenden zu können.

Datenintegrität

Am verständlichsten wird das Konzept der Datenintegrität, wenn man die folgenden Anforderungen bedenkt, die vor allem bei aktiven transaktionsbasierten Anwendungen an das Speichern und Verwalten von Daten gestellt werden:

- Wird ein Datensatz geschrieben oder werden Daten aus einem Datensatz in einer Datenbank ausgelesen und werden die Daten während der Übertragung beschädigt oder dupliziert bzw. gehen sie verloren, so muss ein entsprechender Fehlerhinweis ausgegeben werden.
- Wenn Daten von mehreren Prozessen gleichzeitig verarbeitet werden sollen, müssen alle Prozesse dieselbe Sicht auf die Daten erhalten; es muss verhindert werden, dass die Daten beschädigt oder dupliziert werden bzw. verloren gehen.
- Wenn konsistente Änderungen von Daten in mehreren Datenbanken erforderlich sind, müssen diese Änderungen entweder alle vorgenommen werden oder sie dürfen überhaupt nicht erfolgen.

In MQSeries gelten die oben aufgeführten Anforderungen an die Datenintegrität bei der Datenspeicherung auch für Messaging-Operationen (z. B. MQPUT- und MQGET-Aufrufe) in Warteschlangen. Die Konsistenz bei Änderungen in mehreren Datenbanken muss nicht nur über Anwendungsdatenbanken und MQSeries-Warteschlangen hinweg, sondern auch *zwischen* Anwendungsdatenbanken und MQSeries-Warteschlangen sichergestellt werden.

Das Konzept von MQSeries for Compaq NSK V5.1 sieht vor, dass die Datenintegrität für permanente Datenoperationen auch beim Ausfall einer Hardware- oder Softwarekomponente gewahrt bleibt. In einigen Fällen kann die Datenintegrität sogar bei Ausfällen mehrerer Komponenten gewahrt bleiben. Dies soll nicht heißen, dass nicht permanente Nachrichten unzuverlässig sind. Die Architektur des Warteschlangenservers stellt verschiedene Optionen zur Verfügung, durch die nicht permanente Nachrichten ebenso zuverlässig wie permanente Nachrichten

Datenintegrität

sind, ausgenommen bei einem totalen Ausfall des Systems. Weitere Informationen zur Architektur des Warteschlangenservers finden Sie in „Kapitel 2. MQSeries for Compaq NSK V5.1 - Architektur“ auf Seite 23.

In MQSeries for Compaq NSK V5.1 wird die Datenintegrität durch eine Kombination aus verschiedenen grundlegenden Merkmalen der Software und Hardware des Compaq NSK-Systems und der MQSeries-Software selbst sichergestellt.

Es gibt verschiedene Möglichkeiten, die unterschiedlichen Stufen der Datenintegrität über Optionen der MQSeries-Konfiguration zu implementieren:

- Über die Verwendung der Nachrichtenpermanenz für Anwendungen
- Über die Speicheroptionen für permanente Nachrichten
- Über die Optimierungsoptionen für nicht permanente Nachrichten
- Über die Konfigurationsoptionen für den Warteschlangenserver
- Über die Konfiguration der Hardware, die Warteschlangendateien unterstützt
- Über die Verwendung und Konfiguration von NonStop TM/MP
- Über die Verwendung und Konfiguration von RDF (NonStop RDF, Remote Database Facility)

Auf diese Punkte soll später in diesem Kapitel einzeln eingegangen werden.

Verfügbarkeit

Allgemein versteht man unter Verfügbarkeit die Zeitspanne, in der eine Anwendung oder ein Service während der Gesamtlaufzeit betriebsbereit und verfügbar ist. *Ständige Verfügbarkeit* ist das erklärte Ziel für alle Systeme. Um die Verfügbarkeit zu ermitteln, muss sie im Verhältnis zur tatsächlichen Laufzeit eines Systems gesehen werden, denn es ist natürlich einfacher, eine hundertprozentige Verfügbarkeit über eine kurze Betriebsspanne zu erreichen.

In der Praxis wird die Verfügbarkeit eines Systems im Laufe des Betriebs immer wieder in verschiedener Weise auf die Probe gestellt:

- Ausfälle von Hardware- und Systemsoftwarekomponenten
- Ausfälle der Anwendungssoftware selbst
- Änderungen, die als vorbeugende Maßnahmen am System vorgenommen werden
- Datenverkehr oder Transaktionsauslastungen, die die System- oder Ressourcenkapazitäten überschreiten

Wie die Datenintegrität wird auch die Verfügbarkeit in MQSeries for Compaq NSK V5.1 durch eine Kombination aus verschiedenen grundlegenden Merkmalen der Software und Hardware des Compaq NSK-Systems und der MQSeries-Software selbst sichergestellt.

Hier muss auf jeden Fall angemerkt werden, dass MQSeries for Compaq NSK (zumindest in der momentanen Form) nicht die ständige Verfügbarkeit bietet, wie sie von der Compaq NSK-Systemsoftware (z. B. dem Dateisystem) zur Verfügung gestellt wird. Zwar bieten einige Komponenten des WS-Managers diesen Umfang an Verfügbarkeit, jedoch nicht der WS-Manager insgesamt.

MQSeries for Compaq NSK, V5.1 soll einen Grad an Verfügbarkeit bieten, der beim Ausfall einer Hardware- oder Softwarekomponente folgendes sicherstellt:

- Zu Ausfällen oder Unterbrechungen kommt es nur bei WS-Managerverbindungen zu Komponenten, die direkt von dem Ausfall betroffen sind (vom Ausfall einer CPU sollten beispielsweise nur die Verbindungen betroffen sein, die von LQMA-Prozessen, die in der ausgefallenen CPU aktiv waren, zur Verfügung gestellt wurden).
- Der WS-Manager bleibt auch ohne manuelle Eingriffe durch den Systemadministrator weiterhin für neue Verbindungsanforderungen verfügbar.
- Der Zugriff auf ein WS-Managerobjekt wie beispielsweise eine lokale Warteschlange muss über alle anderen Verbindungen, die nicht direkt von dem Ausfall betroffen sind, weiterhin möglich sein.

Zusätzlich zu diesen Compaq NSK-spezifischen MQSeries-Merkmalen gibt es noch einige Merkmale von MQSeries Version 5.1, die für alle Plattformen gelten und mit denen Sie die Verfügbarkeit von MQSeries for Compaq NSK verbessern können. Es gibt verschiedene Möglichkeiten, die unterschiedlichen Stufen der Verfügbarkeit über Optionen der MQSeries-Konfiguration zu implementieren:

- Über den Anwendungsentwurf
- Über die PATHWAY-Konfigurationsoptionen
- Über die Verwendung und Konfiguration der MQSeries-Standardfunktionen (einschließlich Cluster)
- Über die Hardware, die MQSeries unterstützt
- Über die Konfigurationsoptionen für nicht permanente Nachrichten
- Über die Verwendung und Konfiguration von NonStop TM/MP
- Über die Verwendung und Konfiguration von NonStop RDF

Permanente und nicht permanente Daten

In MQSeries haben *permanente Daten* folgende Merkmale:

- Änderungen an permanenten Daten sind auch nach einem Neustart des WS-Managers noch vorhanden
- Permanente Daten werden in nicht flüchtigen Datenträgern gespeichert
- Für permanente Daten ist die höchste Stufe der Datenintegrität gewährleistet, die im jeweiligen Betriebssystem möglich ist
- Um die Datenintegrität zu gewährleisten, wird bei Operationen mit permanenten Daten eine geringere Geschwindigkeit und eine höhere Ressourcenbelegung in Kauf genommen.

Die meisten verwaltungsspezifischen Operationen betreffen permanente Daten, da für die Konfigurationsdatenbanken von MQSeries die höchste Stufe der Datenintegrität erforderlich ist; damit soll verhindert werden, dass die Verfügbarkeit von MQSeries erheblich beeinträchtigt wird. Geschwindigkeit oder Ressourcenauslastung spielen bei den meisten Verwaltungs-Tasks keine große Rolle, da sie nicht häufig anfallen.

Beispiele für Verwaltungs-Tasks, die permanente Daten betreffen:

- Änderungen an den Attributen einer Warteschlange
- Änderungen an den Attributen eines Kanals
- Erstellung einer neuen Warteschlange, eines neuen Prozesses, einer neuen Namensliste oder eines neuen Kanals

Permanente und nicht permanente Daten

Beispiele für Verwaltungs-Tasks, die nicht permanente Daten betreffen:

- Starten bzw. Stoppen eines Kanals
- Abruf der Attribute eines WS-Managerobjekts

Es gibt verschiedene Möglichkeiten in MQSeries for Compaq NSK, Nachrichten zu speichern; es hängt davon ab, ob Nachrichten von der Anwendung als permanente oder nicht permanente Nachrichten eingereiht werden, und wie die Warteschlangen vom Systemadministrator konfiguriert werden.

Permanente Nachrichten

Permanente Daten werden immer auf Platte gespeichert. Als Systemadministrator können Sie für jede Warteschlange zwischen zwei Optionen wählen, die festlegen, wie permanente Nachrichten in der Warteschlange gespeichert werden sollen. Die Auswahl des Speicherverfahrens hängt von der Nachrichtengröße ab, da der Hauptgrund für die Implementierung verschiedener Speicherverfahren die Leistungsverbesserung bei Messaging-Operationen für sehr große Nachrichten ist.

Für alle permanenten Nachrichten ist in einer Prüflistendatei für die Warteschlange jeweils ein Datensatz vorhanden. Dieser Datensatz enthält die Header und wichtige Steuerinformationen zur Nachricht sowie die eigentlichen Nachrichtendaten; da die maximale Größe des Datensatzes jedoch auf 4096 Bytes begrenzt ist, ist wahrscheinlich nur ein Teil der Nachrichtendaten enthalten.

Kurze und mittlere permanente Nachrichten, die TM/MP-gesteuert sind, können am schnellsten und effizientesten gespeichert werden, wenn Nachrichtenüberlaufdaten in mehreren Datensätzen in der Warteschlangenüberlaufdatei abgelegt werden, wobei dasselbe Verfahren wie in MQSeries for Compaq NSK Version 2.2.0.1 verwendet wird. Alle Daten in diesen Nachrichten werden in TM/MP protokolliert und können daher gegebenenfalls anhand der Prüfprotokolle wiederhergestellt werden.

Für lange Nachrichten (mit mehr als 200 KB Daten) hat sich die Verwendung einer dedizierten Nachrichtenüberlaufdatei als das beste Verfahren erwiesen; für diese Nachrichtenüberlaufdatei wird kein Prüfprotokoll angelegt. Aus diesem Grunde werden die Daten, die in diese Überlaufdatei geschrieben werden, nicht auch noch im TM/MP-Prüfprotokoll aufgezeichnet, wodurch die CPU-Belegung und die E/A-Plattenzugriffe reduziert werden; außerdem können die Daten in großen Blöcken übertragen werden.

Die Unterschiede im Hinblick auf die Datenintegrität, die sich bei diesen beiden Speicherverfahren ergeben, sind in Tabelle 6 dargestellt:

Tabelle 6. Warteschlangenüberlaufdatei und Nachrichtenüberlaufdatei im Vergleich

Datenintegrität	Warteschlangenüberlaufdatei	Nachrichtenüberlaufdatei
Umfang der von TM/MP protokollierten Daten	Alle Nachrichtendaten werden protokolliert.	Es werden nur die Daten protokolliert, die in dem Datensatz der Warteschlangendatei Platz finden.
Wiederherstellbarkeit der Daten anhand des Prüfprotokolls bei mehreren Ausfällen, die zu einem kompletten Verlust des Datenträgerinhalts führen	Die gesamte Nachricht kann anhand des Prüfprotokolls wiederhergestellt werden.	Nur der Nachrichten-Header und der erste Teil der Daten können anhand des Prüfprotokolls wiederhergestellt werden.

Tabelle 6. Warteschlangenüberlaufdatei und Nachrichtenüberlaufdatei im Vergleich (Forts.)

Datenintegrität	Warteschlangenüberlaufdatei	Nachrichtenüberlaufdatei
Maximal mögliche Nachrichtengröße	Wird durch die Anzahl der Datensatzsperrern pro Datenträger und pro Transaktion begrenzt, die durch ENSCRIBE vorgegeben ist. Für eine nicht partitionierte Datei liegt der Wert bei einer standardmäßigen ENSCRIBE-Konfiguration bei 20 MB. Der tatsächliche Wert kann allerdings auf Grund begrenzter physischer Speicherkapazitäten auch darunter liegen.	Wird nur durch den tatsächlich verfügbaren Plattenspeicherplatz begrenzt bzw. durch die für MQSeries vorgegebene maximal zulässige Nachrichtengröße von 100 MB.
Kompatibilität mit RDF (Remote Database Facility)	Voll kompatibel (wird über die Umgebungsvariable MQRDF aktiviert).	Nicht mit RDF kompatibel. Da die Nachrichtendaten nicht protokolliert werden, ist es nicht möglich, Änderungen an Nachrichtendateien über RDF an das Ausweichsystem weiterzugeben.
Fehlertoleranz bei Problemen mit der Festplattenhardware	Vollständige Fehlertoleranz beim Ausfall einzelner Komponenten, wenn gespiegelte Platten vorhanden sind; da alle Daten protokolliert werden, ist bei Ausfall der primären und der gespiegelten Platte oder (falls keine gespiegelte Platte verwendet wird) bei Ausfall der einzigen Platte eine Wiederherstellung der Dateien möglich.	Vollständige Fehlertoleranz beim Ausfall einzelner Komponenten, wenn die Nachrichtenüberlaufdateien in gespiegelten Platten gespeichert werden. Bei Ausfall des gesamten Datenträgers ist keine Wiederherstellung der Nachrichtendaten mehr möglich.

Insgesamt ist also für permanente Nachrichten bei beiden Speicherverfahren ein hoher Grad an Datenintegrität sichergestellt, es bestehen jedoch Einschränkungen für Nachrichtenüberlaufdateien, da nicht alle Nachrichtendaten protokolliert werden. In der Regel betrifft diese Einschränkung nur sehr große Nachrichten, die in diesen Fällen durch die Segmentierung umgangen werden kann; dabei wird die Anwendungsnachricht in kleinere physikalische Nachrichten aufgeteilt.

Nicht permanente Nachrichten

Nicht permanente Nachrichten (NPM, Non-Persistent Messages) werden in der Regel im Hauptspeicher abgelegt. Wird bei großen Mengen an nicht permanenten Nachrichtendaten im Warteschlangenserver ein Schwellenwert erreicht, so dass es zu Engpässen im virtuellen Speicherbereich kommen könnte, sorgt der Warteschlangenserver dafür, dass die nicht permanenten Nachrichten auf Platte gespeichert werden. Bei Änderungen an der Konfiguration, die Online erfolgen, kopiert der Warteschlangenserver hin und wieder auch bei der Ausführung bestimmter

Permanente und nicht permanente Daten

Verwaltungs-Tasks permanente Nachrichten in Plattendateien. Die nicht permanenten Nachrichten, auf die Anwendungen zugreifen, befinden sich immer im Hauptspeicher.

Der Warteschlangenserver verwaltet die Nachrichtenspeicherung für alle ihm zugeordneten Warteschlangen und ermöglicht es, für jede Warteschlange den Grad an Datenintegrität festzulegen, der für nicht permanente Nachrichten gelten soll. Dabei wird über einen Konfigurationsparameter angegeben, ob der Warteschlangenserver Prüfpunktinformationen für nicht permanente Nachrichten im Backup-Prozess aufzeichnen soll, so dass nicht permanente Nachrichten genau wie permanente Nachrichten gegen Ausfälle der CPU, in der der primäre Warteschlangenserverprozess läuft, gesichert sind.

Die Auswahl dieser Option bewirkt allerdings, dass der Warteschlangenserver mehr CPU-Zeit belegt (dies gilt sowohl für die primäre CPU als auch für die Backup-CPU), bei der Aufzeichnung von Prüfpunktinformationen mehr Daten an den Backup-Warteschlangenserver überträgt und mehr Speicher im Backup-Prozess belegt, da die Nachricht noch einmal gespeichert werden muss.

Ist diese Option für eine Warteschlange nicht aktiviert und wird der primäre Warteschlangenserverprozess abnormal beendet (wenn z. B. ein Prozess oder eine CPU ausfällt), gehen alle nicht permanenten Nachrichten bei Übernahme des Betriebs durch den Backup-Prozess verloren, da sie nur im primären Prozess gespeichert wurden.

Diese Option der Fehlertoleranz für nicht permanente Nachrichten steht auf keiner anderen MQSeries-Plattform zur Verfügung. Aus Gründen der Konsistenz innerhalb der Produktserie ist diese Option daher standardmäßig aktiviert; sie kann mit dem Befehl `altmqfls` inaktiviert werden. Sie sollten sorgfältig überlegen, ob nicht permanente Nachrichten Ihren Anforderungen an die Datenintegrität entsprechen. Wenn ja, sollten Sie die Vorteile nutzen, die ihre Verwendung für die Ressourcenauslastung und Systemleistung bietet.

Datenbankkonsistenz

Die Datenbankkonsistenz muss sowohl intern durch MQSeries als auch extern bei Auswahl der Synchronisationspunktoption für Nachrichtenoperationen gewährleistet sein. Dies ist unerlässlich für die Datenintegrität und die Verfügbarkeit.

Interne Datenbankkonsistenz

Alle kritischen Datenbankdateien im WS-Manager werden von TM/MP protokolliert. Daher muss der WS-Manager Änderungen an diesen Dateien in Form von Transaktionen vornehmen. Alle Änderungen werden im TM/MP-Prüfprotokoll aufgezeichnet und (falls aktiviert) mit Hilfe von RDF in einem oder mehreren Ausweichsystemen dupliziert, um so gegen Totalausfälle geschützt zu sein. Auf diese Weise wird für interne Datenbanken das höchste Maß an Datenintegrität gewährleistet.

Durch die Verwendung von TM/MP zum Schutz interner Datenbanken ist die Integrität kritischer Datenbanken bei System- oder Softwareausfällen sichergestellt. Dies bedeutet, dass nach einem Ausfall die Prozesse oder Services, die auf diese Datenbanken zugreifen, mit hoher Wahrscheinlichkeit erfolgreich erneut gestartet werden können, was eine bessere Verfügbarkeit bedeutet.

Externe Datenbankkonsistenz

Die Koordination von Änderungen an externen Datenbanken und die Konsistenz zwischen externen Datenbanken und internen MQSeries-Datenbanken wird durch die Verwendung von TM/MP in MQSeries und in den Anwendungen ermöglicht. Messaging-Operationen können in MQSeries zwischen Synchronisationspunkten ausgeführt werden; hierfür muss für die Anwendung eine Transaktion aktiv sein, die entweder von einem anderen Prozess übernommen oder mit dem TM/MP-Service BEGINTRANSACTION gestartet wird. Diese Transaktion wird vom WS-Manager übernommen, und alle Speichervorgänge oder Änderungen an kritischen Datenbanken werden innerhalb dieser Transaktion vorgenommen. Durch TM/MP werden für alle E/A-Plattenzugriffe, die innerhalb einer Transaktion vorgenommen werden, die entsprechenden Prüfprotokolleinträge in dem Prozess erstellt, der die Aktualisierung ausführt.

Darüber hinaus stellt TM/MP die Konsistenz der Änderungen sicher, die vorgenommen, aber noch nicht festgeschrieben wurden, indem alle Datensätze in den Datenbankdateien gesperrt werden, die von den Änderungen betroffen sind. Beispielsweise kann eine Zeile, die in einer SQL-Tabelle TM/MP-gesteuert (d. h. innerhalb einer Transaktion) eingereiht wurde, erst geändert oder gelöscht werden, nachdem die Transaktion festgeschrieben wurde.

Nachdem der WS-Manager auf den Anwendungsprozess reagiert hat (und von dem MQI-Aufruf zurückkehrt), kann die Anwendung weitere Operationen innerhalb dieser Transaktion ausführen, z. B. weitere Nachrichten mit Hilfe von MQSeries einreihen oder abrufen oder Datenbankänderungen unter Verwendung von ENSCRIBE oder NonStop SQL vornehmen. Wenn die Anwendung die Änderungen an MQSeries-Warteschlangen und an anderen Datenbanken festschreiben möchte, wird der TM/MP-Service ENDTRANSACTION aufgerufen, der die Änderungen an allen Datenbanken systemweit festschreibt.

Stellt die Anwendung fest, dass während einer Transaktion ein Fehler aufgetreten ist, und wurden Änderungen an Datenbanken vorgenommen, sollte die Anwendung den Service ABORTTRANSACTION aufrufen; TM/MP setzt daraufhin alle Änderungen an den Datenbanken systemweit wieder zurück. Dadurch wird beispielsweise eine Nachricht, die bei einem MQGET-Aufruf innerhalb der Transaktion abgerufen wurde, wieder in die Warteschlange zurückgestellt, oder Daten, die in eine SQL-Datenbanktabelle eingereiht wurden, wieder entfernt. Durch eine gut durchdachte Anwendungskonzeption kann die Konsistenz auch bei Fehlern sichergestellt und so die Datenintegrität und die Verfügbarkeit der Anwendungen verbessert werden.

Bei der Ausführung von MQPUT- oder MQGET-Aufrufen in MQSeries zwischen Synchronisationspunkten wird die Warteschlangenlänge entsprechend angepasst, da davon ausgegangen wird, dass die Transaktion am Ende festgeschrieben wird. Daher beinhaltet die Warteschlangenlänge sowohl die festgeschriebenen Nachrichten als auch die nicht festgeschriebenen Nachrichten in der Warteschlange. Wird eine Transaktion explizit oder auf Grund von Fehlern abgebrochen, wird die Warteschlangenlänge aktualisiert, so dass der Wert die tatsächliche Länge angibt. Dies ist eine Verbesserung im Vergleich zu den Releases von MQSeries for Compaq NSK V2.2, in denen der WS-Manager den Ausgang einer Transaktion nicht bestimmen konnte und daher nicht in der Lage war, jederzeit die aktuelle Warteschlangenlänge anzugeben.

Das Verfahren, mit dem dies möglich ist, wird unter „OpenTMF“ auf Seite 244 beschrieben.

OpenTMF

OpenTMF ist der inoffizielle Name einer neuen internen Schnittstelle zu NonStop TM/MP, mit der MQSeries mit Unterstützung von Compaq den Ausgang einer Transaktion bestimmen kann, innerhalb der ein WS-Manager Messaging-Operationen ausführt.

Auf der Basis dieses neuen TM/MP-Merkmals wird die in D42 eingeführte heterogene Transaktionsverarbeitung, die mit NonStop TM/MP ermöglicht wird, realisiert. OpenTMF ermöglicht MQSeries die Teilnahme in allen Transaktionen, die von Anwendungen übernommen wurden. Bei Abschluss der Transaktion wird MQSeries von TM/MP über das Resultat benachrichtigt, d. h., ob die Transaktion festgeschrieben oder zurückgesetzt wurde. Auf Grund dieser Meldung von TM/MP kann MQSeries:

- jederzeit die aktuelle Warteschlangenlänge ermitteln,
- jederzeit über aktuelle interne Statusinformationen verfügen,
- die Verfügbarkeit nicht permanenter Nachrichten, die an Synchronisationspunkten verarbeitet werden, steuern,
- die Effizienz und die Antwortzeit von MQGET-Operationen, die auf die Bestätigung über den Abschluss einer Transaktion warten, verbessern.

Für den Systemadministrator zeigt sich die Verwendung von OpenTMF nur darin, dass MQSeries-Prozesse in einer Liste mit den Ressourcenmanagern enthalten sind, die mit dem TMFCOM-Befehl STATUS RESOURCEMANAGER erstellt werden kann. Alle MQSeries-Prozesse, die OpenTMF verwenden, werden in dieser Liste vom TM/MP automatisch als flüchtige (**VOLATILE**) Ressourcenmanager aufgeführt.

Für dieses neue TM/MP-Merkmal sind keine besonderen Verwaltungs-Tasks erforderlich, es wird von MQSeries automatisch verwendet und verwaltet. Sie müssen jedoch sicherstellen, dass für den TM/MP-Konfigurationsparameter RMOPENPERCPU (der die maximal zulässige Anzahl flüchtiger (**VOLATILE**) und wiederherstellbarer (**RECOVERABLE**) Ressourcenmanager pro CPU angibt) ein Wert angegeben wird, der größer ist als die Anzahl der Warteschlangen- und Statusserver, die in einem System in einer einzigen CPU laufen können. Dabei müssen die Backup-Prozesse berücksichtigt werden, da es sich bei diesen Servern um NonStop-Prozesspaare handelt. Der Standardwert von 128 ist in den meisten Installationen angemessen. Eine Beschreibung der Ressourcenmanager in der heterogenen Transaktionsverarbeitung finden Sie im Handbuch *Compaq NSK NonStop TM/MP Configuration and Planning Guide*.

NonStop Tuxedo

MQSeries kann Messaging-Operationen für OSS-Anwendungen über NonStop Tuxedo koordinieren, da dieses Produkt auf NonStop TM/MP aufbaut und für die heterogene Transaktionsverarbeitung dieselben Funktionen wie MQSeries verwendet.

Das Handbuch *Compaq NSK NonStop Tuxedo System Application Development Guide* enthält Informationen zur Verwendung der NonStop Tuxedo-Transaktionsumgebung und ihre Interaktion mit TM/MP.

Gleichzeitige Ausführung von Transaktionen in Anwendungen

Mit MQSeries for Compaq NSK können Anwendungen die Vorteile dieser Transaktionsumgebung auf eine Weise nutzen, die auf anderen Plattformen nicht möglich ist. Generell können unter Compaq NSK Prozesse mehrere Transaktionen gleichzeitig verwalten. Änderungen an protokollierten Datenbanken werden immer innerhalb der aktuellen Transaktionen ausgeführt, und die Anwendung kann jederzeit zu einer der anderen aktiven Transaktionen wechseln, bevor eine dieser Transaktionen festgeschrieben wird. Dadurch können Anwendungen mehrere MQSeries-Messaging-Operationen (die zwischen Synchronisationspunkten ausgeführt werden) gleichzeitig ausführen.

Kritische MQSeries-Datenbankdateien

Die kritischen protokollierten MQSeries-Datenbankdateien werden in Tabelle 7 aufgeführt.

Tabelle 7. Kritische protokollierte Datenbankdateien

Bezeichnung	Speicherort/Name	Funktion
Objektkatalog	Datenträger für Daten/OBJCAT und ABJCAT	Enthält die Attribute der einzelnen Warteschlangen, Prozesse und Namenslistenobjekte und den WS-Manager.
Kanaldefinitionen für andere als Client-Kanäle	Datenträger für Daten/CHDEFS	Enthält die Attribute der Kanäle, bei denen es sich nicht um Client-Kanäle handelt (die Kanaltypen SENDER, RECEIVER, SERVER, REQUESTER und CLUSTER).
Client-Kanaldefinitionen	Datenträger für Daten/CCHDEFS	Enthält die Attribute der einzelnen Client-Kanäle (Kanaltyp SVRCONN).
OAM-Datenbank	Datenträger für Daten/OAMDB	Enthält die Zugriffsberechtigungen für die einzelnen Objekte und den OAM-Principal, der auf den WS-Manager zugreifen kann.
Principal-Datenbank	Datenträger für Daten/PRIDB und PRIDBA	Enthält die Namen der berechtigten OAM-Principals und die entsprechenden Guardian-Benutzer-IDs.
Steuerdatei für Ausführungs-Controller	Datenträger für Daten/RUNTIME	Enthält Informationen, die vom ECBoss und den Ausführungs-Controllern zur Koordinierung von Start- und Wiederherstellungsvorgängen verwendet werden.
Namenslistendefinitionen	Datenträger für Daten/Lxxxxxxx	Enthalten den Inhalt der definierten Namenslistenobjekte.

Datenbankkonsistenz

Tabelle 7. Kritische protokollierte Datenbankdateien (Forts.)

Bezeichnung	Speicherort/Name	Funktion
Warteschlangendateien	Datenträger für Nachrichtendaten/Qxxxxxxx	Enthalten für jede permanente Nachricht in einer lokalen Warteschlange Header-Informationen und einen Teil der Nachrichtendaten.
Warteschlangenüberlaufdateien	Datenträger für Nachrichtendaten/Oxxxxxxx	Enthalten Daten für alle mittleren bis großen permanenten Nachrichten in einer lokalen Warteschlange.
Objektänderungsdateien	Datenträger für Daten und Nachrichtendaten/Txxxxxxx	Anhand dieser Dateien werden Änderungen an Objektattributen ermittelt.
Dateien für die Kanalsynchronisation	Datenträger mit Synchronisationsdaten /Sxxxxxxx	Enthalten Synchronisationsinformationen für ein Kanal-exemplar.

Kritische Prozesse

In Tabelle 8 sind die kritischen Prozesse des WS-Managers aufgeführt; außerdem wird in dieser Tabelle beschrieben, wie MQSeries vor Software- oder Systemausfällen geschützt wird bzw. nach einem Ausfall wiederhergestellt werden kann. In der Tabelle bedeutet 'Totalausfall' mehrere Systemausfälle bzw. den Ausfall des gesamten Systems.

Tabelle 8. Schutz kritischer Prozesse

Prozess	Schutzmechanismus	Wiederherstellung
Warteschlangenserver	NonStop-Prozesspaar. Außer bei einem Totalausfall sind jederzeit aktuelle Statusinformationen für lokale Warteschlangen und Nachrichten verfügbar.	Kann nach einem schweren Fehler anhand der protokollierten Datenbank wiederhergestellt werden. Es sind keine weiteren Maßnahmen für die Wiederherstellung erforderlich. Bei einem Totalausfall gehen nicht permanente Nachrichten verloren.
Statusserver	NonStop-Prozesspaar. Außer bei einem Totalausfall sind jederzeit aktuelle Statusinformationen für nicht lokale Warteschlangenobjekte und Informationen zum Kanalstatus verfügbar.	Kann nach einem schweren Fehler anhand der protokollierten Datenbank wiederhergestellt werden. Es sind keine weiteren Maßnahmen für die Wiederherstellung erforderlich.

Tabelle 8. Schutz kritischer Prozesse (Forts.)

Prozess	Schutzmechanismus	Wiederherstellung
<p>Lokaler WS-Manageragent (LQMA)</p>	<p>Die Verbindung wird für die Anwendung als 'abgebrochen' gekennzeichnet (2009).</p> <p>Das Speicherbereinigungsverfahren (Garbage Collection) des Ressourcenmanagers löscht die Registrierungsbereiche des Cluster-Cache, die von ausgefallenen Prozessen hinterlassen werden.</p> <p>Ausführungs-Controller, ECBoss, Warteschlangenserver und Statusserver werden über die NSK-Interprozesskommunikation sofort über Ausfälle unterrichtet und aktualisieren die Statusdaten entsprechend.</p> <p>TM/MP bricht alle aktiven Transaktionen ab, an denen die Prozesse zum Zeitpunkt des Ausfalls gerade beteiligt waren.</p>	<p>Keine Wiederherstellung möglich. Ein lokaler WS-Manageragent ist jeweils für eine Verbindung zuständig. Wird diese auf Grund eines Fehlers abgebrochen, muss von der Anwendung eine neue Verbindung hergestellt werden.</p>

Kritische Prozesse

Tabelle 8. Schutz kritischer Prozesse (Forts.)

Prozess	Schutzmechanismus	Wiederherstellung
Nachrichtenkanalagent (MCA)	<p>Der Statusserver erkennt Ausfälle sofort über die NSK-Interprozesskommunikation und kennzeichnet den Kanalstatus entsprechend.</p> <p>Der Statusserver stellt sicher, dass beim Ausfall abgehender Kanäle versucht wird, die Kanalverbindung wiederherzustellen.</p> <p>Über die Option AdoptNewMCA kann versucht werden, einen ausgefallenen Kanal wieder zu starten.</p> <p>Die Synchronisationsdaten des Kanals werden von TM/MP protokolliert und von MQSeries zur Wahrung der Kanalintegrität herangezogen. Für die seltenen Fälle, in denen ein unbestätigter Kanal nicht automatisch aufgelöst werden kann, stehen standardmäßige Verwaltungsfunktionen zur Verfügung.</p> <p>Alle anderen Schutzmechanismen entsprechen denen des LQMA-Prozesses.</p>	<p>Keine Wiederherstellung möglich. Ein Nachrichtenkanalagent ist jeweils für einen Kanal zuständig. Der Kanal wird bei einem Ausfall gestoppt und von den vorhandenen MQSeries-Funktionen automatisch erneut gestartet.</p> <p>In MQSeries for Compaq NSK V5.1 stehen darüber hinaus neue Merkmale wie Überwachungssignale für Kanäle und die Möglichkeit zur Clusterbildung zur Verfügung, die die Verfügbarkeit der Kanäle verbessert.</p>

Tabelle 8. Schutz kritischer Prozesse (Forts.)

Prozess	Schutzmechanismus	Wiederherstellung
Kanalinitiator	<p>Es besteht die Möglichkeit, mehrere Initiatoren zu konfigurieren und die Kanalinitialisierungswarteschlangen auf diese zu verteilen, um die Verfügbarkeit zu erhöhen.</p> <p>Der Kanalinitiator läuft normalerweise als PATHWAY-Serverklasse und ist so konfiguriert, dass bei einem Ausfall mehrmals ein automatischer Neustart (AUTOSTART) in vorgegebenen Abständen (Standardeinstellung sind 10 Minuten) versucht wird.</p> <p>Über die Standardoptionen der PATHWAY-Konfiguration können alternative CPUs konfiguriert werden, die bei Ausfall einer CPU einspringen.</p>	Keine Wiederherstellung möglich. Der Standardstatusserver stellt jederzeit den aktuellen Kanalstatus zur Verfügung und ist für den Kanalneustart zuständig. Der Kanalinitiator startet die Kanäle über die MQSeries-Auslösefunktionen.
Befehlsserver	<p>Der Befehlsserver führt Verwaltungsbefehle zwischen Synchronisationspunkten aus und sorgt so für Konsistenz.</p> <p>Es gelten dieselben PATHWAY-Standardmerkmale wie für den Kanalinitiator.</p>	Keine Wiederherstellung möglich. Bei einem Neustart wird eine neue Verbindung zum WS-Manager hergestellt.
ECBoss	Es gelten dieselben PATHWAY-Standard-sicherungsmerkmale wie oben beschrieben.	Bei einem Ausfall koordiniert der ECBoss (Manager für Ausführungs-Controller) mit Hilfe der protokollierten Datei RUNTIME die Wiederherstellung mit den Ausführungs-Controllern. Die Ausführungs-Controller melden sich nach einem Neustart erneut beim ECBoss an und setzen die Verarbeitung fort.
Ausführungs-Controller	Es gelten dieselben PATHWAY-Standardmerkmale wie oben beschrieben.	Bei einem Ausfall koordinieren die Ausführungs-Controller mit Hilfe der protokollierten Datei RUNTIME die Wiederherstellung mit dem ECBoss. Der Ausführungs-Controller meldet sich nach einem Neustart erneut beim ECBoss an und setzt die Verarbeitung fort.

Kritische Prozesse

Tabelle 8. Schutz kritischer Prozesse (Forts.)

Prozess	Schutzmechanismus	Wiederherstellung
TCP/IP-Empfangsprogramm	<p>Es besteht die Möglichkeit, mehrere TCP/IP-Empfangsprogrammprozesse zu konfigurieren, um eine höhere Verfügbarkeit zu erreichen, indem die Kanäle auf mehrere Ports, IP-Adressen oder TCP/IP-Serverprozesse verteilt werden.</p> <p>Es gelten dieselben PATHWAY-Standardmerkmale wie oben beschrieben.</p>	Keine Wiederherstellung möglich. Bei einem Neustart versucht das TCP/IP-Empfangsprogramm eine Verbindung zu einem für den WS-Manager definierten verfügbaren Port herzustellen.
Repository-Manager oder Repository-Cache-Manager	<p>Bei einem Ausfall des Repository-Managers oder des Repository-Cache-Managers können die aktuellen oder neuen Benutzer in derselben CPU weiterhin auf den Cache zugreifen bzw. Clusteroperationen ausführen.</p> <p>Der Repository-Manager hält immer eine auf Platte gespeicherte Kopie des Cluster-Cache in der Repository-Warteschlange bereit. Änderungen erfolgen synchron mit dem Abruf (MQGET) der Nachricht aus der Cluster-Befehlswarteschlange, die die Änderungen verursacht hat; dadurch bleibt die Konsistenz auch dann erhalten, wenn der Repository-Manager während des Speicherns auf Platte ausfällt.</p> <p>Es gelten dieselben PATHWAY-Standardmerkmale wie oben beschrieben.</p>	<p>Bei einer Wiederherstellung übernimmt ein Repository-Prozess die Rolle des Repository-Managers, falls im WS-Manager keiner vorhanden ist. Andernfalls übernimmt er die Funktion eines Repository-Cache-Servers für die CPU, in der er läuft.</p> <p>Ein Repository-Manager oder Repository-Cache-Manager versucht nach einem Ausfall erneut eine Verbindung zum Cache herzustellen, falls dieser im Hauptspeicher noch vorhanden ist. Ist dies nicht der Fall, wird der Cache, falls vorhanden, von der Platte geladen. Ist keine Plattendatei vorhanden, wird ein neuer Cache in der CPU erstellt und über die Repository-Warteschlange initialisiert.</p>

Tabelle 8. Schutz kritischer Prozesse (Forts.)

Prozess	Schutzmechanismus	Wiederherstellung
WS-Managerserver	<p>Der WS-Managerserver ruft abgelaufene Nachrichten ab und generiert Berichte über abgelaufene Nachrichten zwischen Synchronisationspunkten, so dass der Ausfall des Servers keine Inkonsistenzen verursacht.</p> <p>Es gelten dieselben PATHWAY-Standardmerkmale wie oben beschrieben.</p>	<p>Beim Neustart eines WS-Managerservers erfolgt automatisch eine Resynchronisation mit den Warteschlangenservern, da diese bei Ausführung von Verwaltungs-Tasks immer die aktuell abgelaufenen Nachrichten melden.</p>
Anwendungen (STANDARD-Bindungen)	<p>MQSeries-spezifische Ausfälle werden durch Verbindungsabbrüche oder unerwartete Fehlerrückkehr-codes bei MQI-Aufrufen festgestellt. Die Anwendung sollte den Aufruf MQDISC und anschließend erneut einen MQCONN-Aufruf ausgeben, um die Verbindung mit dem WS-Manager wiederherzustellen.</p> <p>Der WS-Manager stellt einen Ausfall des Anwendungsprozesses sofort fest. Er gibt einen impliziten MQDISC-Aufruf für die Anwendung aus, so dass alle offenen Ressourcen geschlossen und freigegeben werden.</p> <p>Alle aktiven Operationen, die zwischen Synchronisationspunkten ausgeführt werden, werden von TM/MP abgebrochen und MQSeries aktualisiert die lokalen Warteschlangen entsprechend.</p> <p>Fehler im Anwendungscode haben keine Auswirkung auf die kritischen Datenbanken des WS-Managers oder gemeinsam benutzte Ressourcen, da der einzige gemeinsam benutzte Speicherbereich im Adressraum der Anwendung schreibgeschützt ist und kein direkter Zugriff auf kritische Datenbankdateien erfolgt.</p>	<p>Keine Wiederherstellung möglich. Nach einem Ausfall stellen die Anwendungen erneut eine Verbindung zum WS-Manager her.</p>

Kritische Prozesse

Tabelle 8. Schutz kritischer Prozesse (Forts.)

Prozess	Schutzmechanismus	Wiederherstellung
Anwendungen (FASTPATH-Bindungen)	<p>MQSeries-spezifische Ausfälle werden durch unerwartete Fehlerrückkehr-codes bei MQI-Aufrufen festgestellt. Die Anwendung sollte den Aufruf MQDISC und anschließend erneut einen MQCONNX-Aufruf ausgeben, um die Verbindung mit dem WS-Manager wiederherzustellen.</p> <p>Der WS-Manager stellt einen Ausfall des Anwendungsprozesses sofort fest. Er gibt einen impliziten MQDISC-Aufruf für die Anwendung aus, so dass alle offenen Ressourcen geschlossen und freigegeben werden.</p> <p>Alle aktiven Operationen, die zwischen Synchronisationspunkten ausgeführt werden, werden von TM/MP abgebrochen und MQSeries aktualisiert die lokalen Warteschlangen entsprechend.</p> <p>Fehler im Anwendungscode können sich auf die kritischen Datenbanken des WS-Managers und die gemeinsam benutzten Ressourcen auswirken, da die Anwendung Lese- und Schreibzugriff auf das Repository-Cache und Zugriff auf die internen WS-Managerstrukturen hat, so dass der Objektkatalog beschädigt werden kann. Die Warteschlangenstrukturen und die Nachrichten selbst sind geschützt, da nur die Warteschlangenserver auf sie zugreifen.</p>	Keine Wiederherstellung möglich. Bei einem Neustart stellen die Anwendungen wieder eine Verbindung zum WS-Manager her.

Cluster

Durch die Verwendung von Clustern in MQSeries for Compaq NSK V5.1 soll die Verwaltung eines MQSeries-Netzes vereinfacht und die Verfügbarkeit und Skalierbarkeit von MQSeries als verteilter Service insgesamt verbessert werden.

WS-Manager, die zu einem Cluster gehören, können MQPUT-Aufrufe an Warteschlangen ausgeben, die im Cluster wie lokale Warteschlangen definiert sind. Die MQSeries-Clusterfunktion übernimmt die Verwaltung aller Definitionen und Kanäle, die für die Übertragung einer Nachricht an die Zielwarteschlange erforderlich sind.

Warteschlangen in Clustern können mehreren WS-Managern innerhalb des Clusters zugeordnet werden. Dadurch werden in einem Cluster mehrere Exemplare ein und derselben Warteschlange erstellt. Eine Anwendung reiht Nachrichten nur in jeweils ein Exemplar einer Warteschlange ein, das vom CWLM (Cluster Workload Manager, eine MQSeries-Komponente) vorgegeben wird. Das Exemplar wird entweder beim Öffnen der Warteschlange oder bei jedem PUT-Vorgang dynamisch ausgewählt. Der CWLM kann über den CWLM-Exit (Cluster Workload Management) feststellen, welches Exemplar einer Cluster-Warteschlange am besten geeignet ist. Er ermittelt dies anhand mehrerer Faktoren, z. B. ob der dem Exemplar zugeordnete Kanal aktiv ist, aber auch anhand der Netzpriorität oder anhand anwendungsspezifischer Überlegungen.

Daher kann mit Clustern die Verfügbarkeit des MQSeries-Netzes erhöht werden. MQSeries on Compaq NSK eignet sich als ein vollständiges Repository für Cluster, da es einen zuverlässigen und skalierbaren Betrieb ermöglicht.

Konfigurationsoptionen für die Verfügbarkeit

Hier soll eine Übersicht über die Konfigurationsoptionen gegeben werden, mit denen die Verfügbarkeit von MQSeries for Compaq NSK V5.1 und seiner Anwendungen unter Compaq NSK erhöht werden kann:

- Konfigurieren Sie PATHWAY mit alternativen CPUs für alle Serverklassen, damit Sie bei CPU-Ausfällen abgesichert sind.
- Sie haben die Möglichkeit, Cluster einzusetzen, um die Verfügbarkeit der MQSeries-Netzressourcen zu erhöhen.
- Für entsprechende Nachrichtenarten können Sie nicht permanente Nachrichten verwenden, für die das Prüfpunktverfahren aktiviert ist, und so eine hohe Leistung und gleichzeitig eine hohe Verfügbarkeit erreichen.
- Stellen Sie sicher, dass die TM/MP-Konfiguration den Anforderungen von MQSeries und der MQSeries-Anwendungen auch bei maximaler Auslastung entsprechen kann.
- Durch die Verwendung von Nachrichtenüberlaufdateien kann der für große Nachrichten anfallende Umfang an Prüfprotokollen reduziert werden.
- Sind Wiederherstellungsmaßnahmen bei Totalausfällen erforderlich, bietet sich die Verwendung von RDF für die Erstellung und Verwaltung eines Ausweichsystems für MQSeries an.
- Stellen Sie sicher, dass die CPUs über einen ausreichenden physischen Speicher verfügen, damit der Betrieb von MQSeries und der MQSeries-Anwendungen auch bei maximaler Belastung möglich ist.
- Stellen Sie sicher, dass der Auslagerungsspeicher für die CPUs, die MQSeries und die MQSeries-Anwendungen enthalten, groß genug ist.

Konfigurationsoptionen für die Datenintegrität

Hier soll eine Übersicht über die Konfigurationsoptionen gegeben werden, mit denen die Datenintegrität von MQSeries for Compaq NSK V5.1 und seiner Anwendungen unter Compaq NSK erhöht werden kann:

- Überlegen Sie sich, welcher Grad an Datenintegrität, den permanente und nicht permanente Nachrichten bieten, für die Nachrichten- bzw. Transaktionsarten in MQSeries erforderlich ist.
- Überlegen Sie sich, ob Nachrichtenüberlaufdateien das geeignete Speicherungsverfahren für sehr große Nachrichten sind.
- Überlegen Sie sich, ob das Prüfpunktverfahren für alle oder nur für einige nicht permanente Nachrichten erforderlich ist. Die verschiedenen Nachrichtenarten sollten jeweils in eigenen Warteschlangen untergebracht werden, für die dann jeweils das Prüfpunktverfahren aktiviert oder inaktiviert werden kann.
- Stellen Sie sicher, dass bei Anwendungen, die eine hohe Datenintegrität erfordern, SYNCPOINT-Operationen unter Verwendung permanenter Nachrichten eingesetzt werden.
- Sind Wiederherstellungsmaßnahmen bei Totalausfällen erforderlich, bietet sich die Verwendung von RDF für die Erstellung und Verwaltung eines Ausweichsystems für die MQSeries-Umgebung an.

Teil 2. Referenz

Kapitel 17. MQSeries-Steuerbefehle	257
Steuerbefehle - Übersicht	257
Namenskonventionen	258
Syntaxdiagramme lesen	258
altmqfls (Attribute der Warteschlangendateien ändern)	260
altmqusr (MQSeries-Benutzerinformationen ändern)	265
cleanrdf (RDF-Verwaltungsaufgaben ausführen)	267
cnvclchl (Client-Kanaldefinitionen konvertieren)	269
crtmqcvx (Datenkonvertierung)	271
crtmqm (WS-Manager erstellen)	273
dltmqm (WS-Manager löschen)	277
dspmqaout (Berechtigung anzeigen)	279
dspmqcsv (Befehlsserver anzeigen)	283
dspmqls (MQSeries-Dateiattribute anzeigen)	284
dspmqrtrc (Formatierte MQSeries-Trace-Ausgabe anzeigen)	287
dspmqusr (MQSeries-Benutzerinformationen anzeigen)	288
endmqcsv (Befehlsserver beenden)	290
endmqm (WS-Manager beenden)	292
endmqtrc (MQSeries-Trace beenden)	294
instmqm (MQSeries for Compaq NSK installieren)	295
runmqchi (Kanalinitiator ausführen)	296
runmqchl (Kanal ausführen)	297
runmqdlq (Steueroutine der DLQ ausführen)	298
runmqlsr (Empfangsprogramm ausführen)	300
runmqsc (MQSeries-Befehle ausführen)	302
runmqtrm (Auslösemonitor starten)	305
setmqaut (Berechtigung setzen/zurücksetzen)	306
strmqcsv (Befehlsserver starten)	312
strmqm (WS-Manager starten)	313
strmqtrc (MQSeries-Trace starten)	315
upgmqm (Upgrade eines WS-Managers der Version 2.2.0.1 durchführen)	318

Kapitel 17. MQSeries-Steuerbefehle

Dieses Kapitel enthält Referenzinformationen zu den in MQSeries for Compaq NSK verwendeten Steuerbefehlen.

Steuerbefehle - Übersicht

Die folgenden Steuerbefehle werden in MQSeries for Compaq NSK über TACL-Makros und kompilierte Programme unterstützt:

- `altmqfls` (Attribute der Warteschlangendateien ändern)
- `altmqusr` (MQSeries-Benutzerinformationen ändern)
- `cleanrdf` (RDF-Verwaltungsdienstprogramm)
- `cnvclchl` (Client-Kanaldefinitionen konvertieren)
- `crtmqcvx` (Datenkonvertierung)
- `crtmqm` (WS-Manager erstellen)
- `dltmqm` (WS-Manager löschen)
- `dspmqaut` (Berechtigung anzeigen)
- `dspmqcsv` (Befehlsserver anzeigen)
- `dspmqfls` (MQSeries-Dateiattribute anzeigen)
- `dspmqtrc` (formatierte Ausgabe eines MQSeries-Trace anzeigen)
- `dspmqusr` (MQSeries-Benutzerinformationen anzeigen)
- `endmqcsv` (Befehlsserver beenden)
- `endmqm` (WS-Manager beenden)
- `endmqtrc` (MQSeries-Trace beenden)
- `instmqm` (MQSeries for Compaq NSK installieren)
- `runmqchi` (Kanalinitiator ausführen)
- `runmqchl` (Kanal ausführen)
- `runmqdlq` (DLQ-Steueroutine ausführen)
- `runmqlsr` (TCP/IP-Empfangsprogramm ausführen)
- `runmqsc` (MQSeries-Befehle ausführen)
- `runmqtrm` (Auslösemonitor starten)
- `setmqaut` (Berechtigung setzen/zurücksetzen)
- `strmqcsv` (Befehlsserver starten)
- `strmqm` (WS-Manager starten)
- `strmqtrc` (MQSeries-Trace starten)
- `upgmqm` (Upgrade eines WS-Managers der Version V2.2.0.1 durchführen)

Diese Befehle sollen im Folgenden ausführlicher beschrieben werden.

Anmerkungen:

1. Optionen, die aus einem Einzelzeichen mit vorangestelltem Gedankenstrich bestehen (z. B. `-v` im Befehl `runmqsc`) müssen immer klein geschrieben werden.
2. Benutzerhinweise werden angezeigt, wenn Steuerbefehle mit `-?` oder `?` oder ohne Parameter aufgerufen werden, obwohl Parameter erwartet werden.

Namenskonventionen

Für folgende MQSeries-Objekte können Namen mit bis zu 48 Zeichen vergeben werden:

- WS-Manager
- Warteschlangen
- Prozessdefinitionen

Die maximale Länge von Kanalnamen beträgt 20 Zeichen.

Folgende Zeichen können in allen MQSeries-Namen verwendet werden:

- Großbuchstaben A - Z
- Kleinbuchstaben a - z
- Numerische Zeichen 0 - 9
- Punkt (.)
- Unterstreichung (_)
- Schrägstrich (/)
- Prozentzeichen (%)

Anmerkungen:

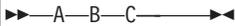
1. Schrägstrich und Prozentzeichen sind Sonderzeichen. Wenn in einem Namen eines dieser beiden Sonderzeichen steht, muss der Name immer in doppelten Anführungszeichen stehen.
2. Die Verwendung führender oder eingebetteter Leerzeichen ist nicht zulässig.
3. Landessprachliche Zeichen sind nicht zulässig.
4. Namen können in doppelten Anführungszeichen stehen; dies ist jedoch nur erforderlich, wenn ein Name Sonderzeichen enthält.

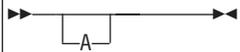
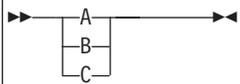
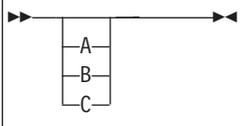
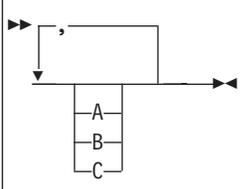
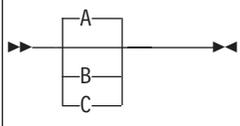
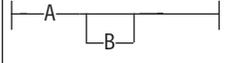
Syntaxdiagramme lesen

Dieses Kapitel enthält Syntaxdiagramme.

Jedes Syntaxdiagramm beginnt mit einem doppelten Rechtspfeil und endet mit einem Paar aus Rechtspfeil und Linkspfeil. Bei Zeilen, die mit einem einfachen Rechtspfeil beginnen, handelt es sich um Fortsetzungszeilen. Syntaxdiagramme werden von links nach rechts und von oben nach unten in Richtung der Pfeile gelesen.

Weitere Konventionen für Syntaxdiagramme:

Konvention	Erläuterung
	Sie müssen die Werte A, B und C angeben. Werte, die angegeben werden müssen, stehen in der Hauptzeile eines Syntaxdiagramms.

Konvention	Erläuterung
	<p>Sie können den Wert A angeben. Optionale Werte werden unterhalb der Hauptzeile eines Syntaxdiagramms angezeigt.</p>
	<p>Die Werte A, B und C sind alternativ. Einer der Werte muss angegeben werden.</p>
	<p>Die Werte A, B und C sind alternativ. Einer der Werte kann angegeben werden.</p>
	<p>Sie können einen der Werte A, B oder C angeben. Erforderliche Trennzeichen bei der Angabe mehrerer Werte oder bei Wiederholungen (in diesem Beispiel das Komma (,)) werden auf der Pfeillinie angezeigt.</p>
	<p>Die Werte A, B und C) sind alternativ. Einer der Werte muss angegeben werden. Wird keiner der gezeigten Werte angegeben, wird standardmäßig der Wert A (der Wert wird oberhalb Hauptzeile angezeigt) verwendet.</p>
<p data-bbox="459 1314 699 1356">Name</p> <p data-bbox="459 1398 545 1430">Name:</p> 	<p>Das Syntaxfragment Name wird getrennt vom Hauptsyntaxdiagramm aufgeführt.</p>

altmqfls (Attribute der Warteschlangendateien ändern)

Funktion

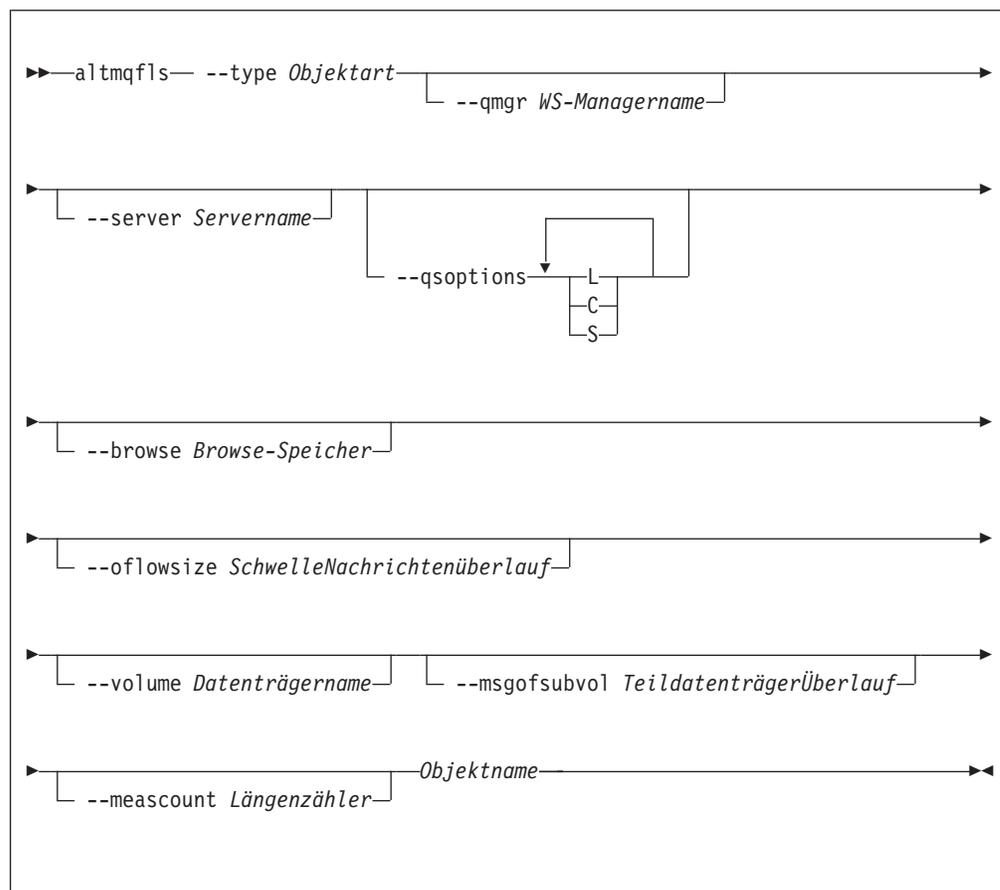
Mit dem Befehl **altmqfls** ändern Sie die Attribute von Warteschlangendateien. Mit jedem Befehl kann nur jeweils eine der folgenden Operationen ausgeführt werden:

- Die Dateien einer vordefinierten lokalen Warteschlange in einen anderen Datenträger verschieben, um den E/A-Plattenzugriff auf verschiedene Datenträger zu verteilen.
- Die Größe der Warteschlange und der ihr zugeordneten Überlaufdateien ändern. Diese Operation kann nur ausgeführt werden, wenn die Warteschlange geschlossen ist.
- Die Warteschlangenserveroptionen für eine lokale Warteschlange ändern. Mit Hilfe dieser Optionen können Sie das Speicherverfahren des Warteschlangenservers für die Warteschlange optimieren und das Prüfpunktverfahren für nicht permanente Nachrichten aktivieren bzw. inaktivieren. Darüber hinaus können Sie den einer lokalen Warteschlange zugeordneten Warteschlangenserver bzw. den einem Aliasnamen, einer fernen Warteschlange, einer Modellwarteschlange oder einem Prozess zugeordneten Befehlsserver ändern und der Warteschlange einen Längenzähler zuordnen.

Die Warteschlange kann nicht mit dem Befehl **altmqfls** erneut geladen werden, solange noch Zugriffe auf sie erfolgen.

Anmerkung: Die Benutzerschnittstelle und die Befehlszeile für den Befehl **altmqfls** haben sich im Vergleich zu Version 2.2.0.1 erheblich geändert. Es wurden neue Funktionen hinzugefügt und sinnvollere Namen für die Befehlszeilenoptionen gewählt. Außerdem sind Parametern jetzt doppelte Gedankenstriche (--) vorangestellt, damit sie den POSIX-Regeln entsprechen. Bereits vorhandene Scripts, die den Befehl **altmqfls** enthalten, müssen unter Umständen entsprechend geändert werden, da in MQSeries Version 5.1 Befehlszeilenoptionen der Version 2.2.0.1 zurückgewiesen werden.

Syntax



Erforderliche Parameter

Objektname

Gibt den Namen der permanenten lokalen Warteschlange an, deren Nachrichtendateien verschoben werden sollen. Die Warteschlange muss geschlossen sein und darf keine nicht festgeschriebenen Nachrichten enthalten.

--type *Objektart*

Gibt eine permanente Warteschlange an. Dieser Parameter muss angegeben werden; folgende Angaben sind zulässig:

ql oder **qlocal** Eine lokale Warteschlange

qa oder **qalias** Eine Aliaswarteschlange

qr oder **qremote**
Eine ferne Warteschlange

qm oder **qmodel**
Eine Modellwarteschlange

proc oder **process**
Ein Prozess

Optionale Parameter

--qmgr *WS-Managername*

gibt den Namen des WS-Managers an, dem die lokale Warteschlange zugeordnet ist. Der WS-Manager muss aktiv sein. Erfolgt keine Angabe, wird der Standard-WS-Manager verwendet.

--server *Servername*

Gibt den Namen eines Statusserver- oder Warteschlangenserverprozesses an, der für die Statusdaten des betreffenden Objekts zuständig ist. Handelt es sich bei dem Objekt um eine lokale Warteschlange oder eine Modellwarteschlange, wird hier der Name eines Warteschlangenservers angegeben. Handelt es sich um ein anderes Objekt, geben Sie den Namen des Statusservers an.

Bei der Erstellung eines Objekts wird der *Servername* auf 'DEFAULT' gesetzt, d. h. der Standardserver ist für die Warteschlange zuständig. Wenn ein anderer Server angegeben werden soll, muss die Warteschlange für Zugriffe gesperrt sein; alle nicht permanenten Nachrichten gehen bei dem Wechsel verloren.

--volume *Datenträgername*

Gibt den Namen eines Compaq NSK-Datenträgers an (z. B. \$DEV). Diese Angabe ist erforderlich, wenn Nachrichtendateien mit **altmqfls** in einen anderen Datenträger verschoben werden sollen. Dieser Parameter kann nur in Kombination mit der Objektart und dem Objektnamen angegeben werden. Er ist nicht zulässig für andere Optionen.

--qsoptions **LSC**

Diese Optionen können entweder einzeln oder zusammen angegeben werden, um die Wiederherstellbarkeit und Zuverlässigkeit einer Warteschlange optimal einzustellen.

Anmerkung: Die Option --qsoptions kann nur jeweils einmal in einer Befehlszeile angegeben werden, daher werden bei jedem Befehl immer *alle* drei Optionen (S, L und C) gesetzt. Bei Angabe von --qsoptions S beispielsweise wird der Schalter S gesetzt und die Schalter L und C (da sie nicht angegeben wurden) zurückgesetzt. Sie können mehr als eine Option angeben, es muss aber mindestens eine Option angegeben werden.

C Gibt an, ob für nicht permanente Nachrichten Prüfpunktinformationen im Backup-Warteschlangenserver aufgezeichnet werden sollen; dadurch wird eine Fehlertoleranz ermöglicht, allerdings auf Kosten der CPU-Auslastung, die sich durch die zusätzliche Aufzeichnung von Prüfpunktinformationen, die zusätzlichen IPC-Nachrichten und den zusätzlichen Speicher für die Nachrichten erhöht.

Wählen Sie diese Option, um die Wiederherstellung nicht permanenter Nachrichten nach Möglichkeit sicherzustellen. Bei der Erstellung einer Warteschlange wird dieser Parameter standardmäßig gesetzt, d. h., es werden Prüfpunktinformationen aufgezeichnet.

L Gibt an, dass der Warteschlangenserver die Datenstrukturen und Datenketten einer Warteschlange im Speicher sperren soll. In der Regel wird der einer Warteschlange zugeordnete Speicherplatz aus dem Adressbereich des Warteschlangenservers entfernt, wenn auf ihn nicht mehr zugegriffen wird. Durch die Angabe dieser Option ist ein schnellerer Zugriff auf die Datenstrukturen der Warteschlange im Hauptspeicher möglich, allerdings möglicherweise auf Kosten anderer Warteschlangen.

Bei der Erstellung einer Warteschlange wird dieser Parameter standardmäßig nicht gesetzt, d. h., es werden keine Sperren im Speicher eingerichtet.

S Gibt an, dass der Warteschlangenserver beim Start des WS-Managers die lokale Warteschlange von der Platte in den Cache lädt. In der Regel werden die Nachrichten für eine Warteschlange erst geladen, wenn eine Anwendung zum ersten Mal auf die betreffende Warteschlange verweist. Ist diese Option gesetzt, wird die Warteschlange jedoch bereits beim Start des WS-Managers geladen. Bei der Erstellung einer Warteschlange wird dieser Parameter nicht gesetzt, d. h., die Warteschlange wird nicht beim Start des WS-Managers geladen.

Durch diese Option werden CPU-Auslastung und E/A-Plattenzugriffe beim ersten Öffnen einer Warteschlange gering gehalten, dies führt allerdings zu einer erhöhten Aktivität des Warteschlangenservers (CPU-Auslastung und E/A-Plattenzugriffe) beim Start des WS-Managers.

None Gibt an, dass keine Optionen gesetzt werden.

--browse *Browse-Speicher*

Gibt die Anzahl der Datenbytes an, die maximal pro permanenter Nachricht im Cache des Warteschlangenservers (und auf Platte) gespeichert werden sollen. Bei einem Browse-Vorgang für eine permanente Nachricht liest der WS-Manager in der Regel die Daten zu einer Nachricht aus dem Plattenspeicher und gibt sie an die Anwendung zurück. Wird dieser Parameter auf einen anderen Wert als null gesetzt, wird die angegebene Anzahl an Datenbytes im Speicher behalten; diese Daten können bei einem Browse-Vorgang an die Anwendung zurückgegeben werden, ohne dass ein Zugriff auf die Platte erforderlich ist. Dadurch werden vom Warteschlangenserver mehr Speicherressourcen belegt. Der zulässige Mindestwert für diesen Parameter ist null (0) Byte, der maximale Wert 25000 Byte. Bei der Erstellung einer Warteschlange wird dieser Parameter standardmäßig auf null gesetzt.

--oflowsize *SchwelleNachrichtenüberlauf*

Gibt die Mindestgröße der Nachrichten an, die in einer Nachrichtenüberlaufdatei für die Nachrichtendaten gespeichert werden. Permanente Nachrichten, die kleiner sind, werden in der Überlaufdatei der Warteschlange gespeichert. Die Massendaten permanenter Nachrichten, die größer sind, werden in einer dedizierten Nachrichtenüberlaufdatei gespeichert. Bei der Erstellung einer Warteschlange wird dieser Parameter standardmäßig auf 200000 Bytes gesetzt.

altnmqfls

--msgsubvol *Teildatenträgerüberlauf*

Gibt den Teildatenträger in dem Datenträger an, der die Warteschlange enthält; in diesem Teildatenträger erstellt der Warteschlangenserver neue Nachrichtenüberlaufdateien. Alle Warteschlangen verwenden bei ihrer Erstellung standardmäßig den Teildatenträger mit den Nachrichtendaten des WS-Managers, dem sie zugeordnet sind.

--meascount *Längenzähler*

Gibt den Namen eines Längenzählers an, der (wenn er für eine aktive Messung verwendet wird) mit der aktuellen Warteschlangenlänge initialisiert wird und vom Warteschlangenserver, dem die Warteschlange zugeordnet ist, beim Einreihen bzw. Abrufen von Nachrichten entsprechend erhöht bzw. vermindert wird.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde beendet, aber nicht vollständig wie erwartet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

Beispiele

1. Mit dem folgenden Befehl werden die Nachrichtendateien der lokalen Warteschlange `flint.queue`, die dem WS-Manager `target.queue.mgr` zugeordnet ist, in den Datenträger `$DATA3` verschoben.

```
altnmqfls --qmgr target.queue.mgr --type ql --volume $DATA3 flint.queue
```

2. Mit dem folgenden Befehl wird der Name des Warteschlangenserverprozesses für ein Objekt geändert:

```
altnmqfls --qmgr target.queue.mgr --type ql --server $TQS2 flint.queue
```

Bei Eingabe dieses Befehls wird die lokale Warteschlange `flint.queue`, die dem WS-Manager `target.queue.mgr` zugeordnet ist, in den Warteschlangenserver mit dem Prozessnamen `$TQS2` verschoben.

Zugehörige Befehle

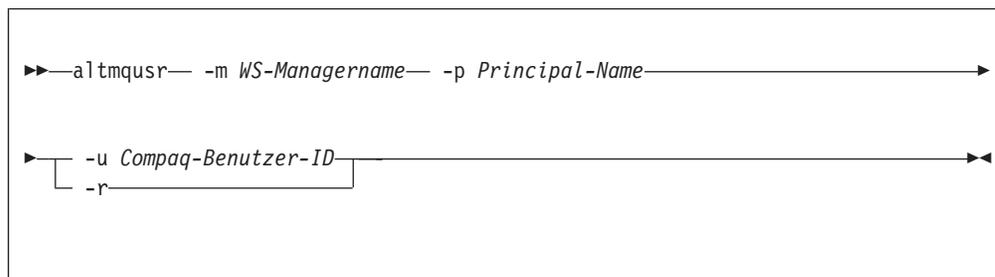
dspmqfls MQSeries-Dateien anzeigen

altmqsr (MQSeries-Benutzerinformationen ändern)

Funktion

Mit dem Befehl **altmqsr** können Sie einen Principal, der einer Compaq NSK-Benutzer-ID mit Zugriff auf MQSeries entspricht, definieren oder entfernen.

Syntax



Beschreibung

Mit diesem Befehl können Sie:

- einen Principal erstellen (d. h., einer Compaq NSK-Benutzer-ID den Zugriff auf einen WS-Manager erteilen),
- einen Principal entfernen (d. h., einer Compaq NSK-Benutzer-ID den Zugriff auf einen WS-Manager verweigern),
- die Definition eines Principals ändern.

Bei Angabe der Option *-u Compaq-Benutzer-ID* wird mit dem Befehl **altmqsr** ein Principal erstellt bzw. (falls bereits ein Principal vorhanden ist) die vorhandene Definition geändert. Für *Compaq-Benutzer-ID* kann eine Benutzer-ID mit Compaq-Verwaltungsberechtigungen oder (wenn SAFEGUARD aktiv ist) ein SAFEGUARD-Aliasname angegeben werden.

Bei Angabe der Option *-r* wird der Principal aus der Principal-Datenbank gelöscht.

Sie müssen entweder *-u Compaq-Benutzer-ID* oder *-r* angeben.

Erforderliche Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, dem der Principal zugeordnet ist.

-p *Principal-Name*

Gibt den Namen des Principals an, der erstellt, geändert oder entfernt werden soll.

-u *Compaq-Benutzer-ID*

Gibt die Compaq NSK-Benutzer-ID mit Verwaltungsberechtigungen oder den SAFEGUARD-Aliasnamen an, die bzw. der der Principal-Definition zugewiesen werden soll.

-r Gibt an, dass die Principal-Definition aus dem WS-Manager gelöscht werden soll.

altmqsr

Rückkehrcodes

- 0 Operation war erfolgreich
- 36 Ungültige Argumente übergeben
- 69 Kein Speicher verfügbar
- 71 Unerwarteter Fehler

Beispiele

Mit dem folgenden Befehl wird der Principal mquser1 der Compaq-Benutzer-ID mqtest.fred zugeordnet:

```
altmqsr -m MT02 -p mquser1 -u mqtest.fred
```

Mit dem folgenden Befehl wird der Principal mquser2 der Gruppe group.user01 zugeordnet:

```
altmqsr -m MT02 -p mquser2 -u group.user01
```

Das Ergebnis dieser Befehle kann angezeigt werden, indem Sie den Befehl **dspmqsr** wie unter „dspmqsr (MQSeries-Benutzerinformationen anzeigen)“ auf Seite 288 beschrieben eingeben.

Mit dem folgenden Befehl entfernen Sie den Principal mquser1:

```
altmqsr -m MT02 -p mquser1 -r
```

Zugehörige Befehle

dspmqsr MQSeries-Benutzerinformationen anzeigen

cleanrdf (RDF-Verwaltungsaufgaben ausführen)

Funktion

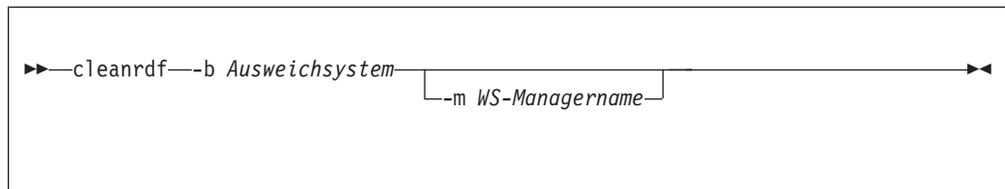
Mit dem Dienstprogramm **cleanrdf** können in einer RDF-Umgebung regelmäßig anfallende Verwaltungsaufgaben für den WS-Manager im primären System ausgeführt werden. Dieses Dienstprogramm schließt den Löschvorgang für Dateien ab, die logisch aus dem primären System und dem Ausweichsystem gelöscht wurden. Darüber hinaus kopiert dieses Dienstprogramm alle nicht protokollierten Datenbanken an die richtige Stelle im Ausweichsystem.

Das mit dem Befehl **cleanrdf** aufgerufene Dienstprogramm arbeitet den gesamten Objektkatalog und die gesamte Nachrichtendatenbank ab, daher kann die Ausführung dieses Befehls zu einer gewissen Beeinträchtigung der Systemleistung führen.

Bei den nicht protokollierten Datenbankdateien, die von diesem Dienstprogramm in das Ausweichsystem kopiert werden, handelt es sich um folgende:

- MQERRLG1
- MQSINI
- QMINI
- PATHCTL
- SHUTDOWN
- SYNCHIDX
- TRACEOPT
- UMQSINI

Syntax



Erforderliche Parameter

-b *Ausweichsystem*

Gibt den Namen des Compaq NSK-Systems an, das als RDF-Ausweichsystem für diesen WS-Manager verwendet wird. Das **Ausweichsystem** wird im Format *\Name* angegeben (das Standardformat in der Compaq NSK-Umgebung).

Optionale Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, für den der Befehl **cleanrdf** aufgerufen wird. Erfolgt keine Angabe, wird der Befehl für den Standard-WS-Manager ausgeführt.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

cleanrdf

Beispiele

Im folgenden Beispiel wird der Befehl **cleanrdf** für den WS-Manager `test.queue.mgr` ausgeführt. Der Compaq NSK-Knoten `\HAWK` wurde als RDF-Ausweichsystem für diesen WS-Manager konfiguriert.

```
cleanrdf -b \HAWK -m test.queue.mgr
```

cnvclchl (Client-Kanaldefinitionen konvertieren)

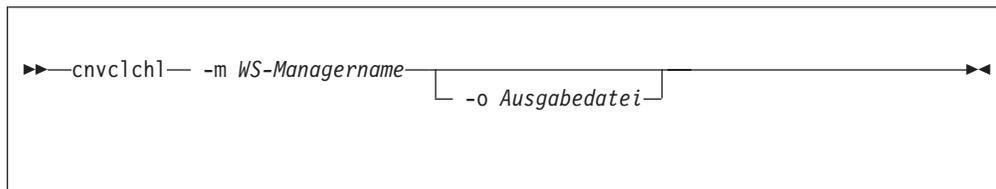
Funktion

Mit dem Befehl **cnvclchl** können Sie die Client-Kanaldefinitionsdatei, die mit MQSC für CLNTCONN-Kanäle erstellt wird, von einer strukturierten Compaq-Datei in ein unstrukturiertes Format konvertieren, das von MQSeries-Clients erkannt wird.

Anmerkung:

Definitionsdateien der Version 2.2.0.1 können nicht für Clients der Version 5.1 verwendet werden, da sich das Format der Datensätze geändert hat. Nach der Installation von MQSeries for Compaq NSK V5.1 können Sie den Befehl **cnvclchl** noch einmal ausführen, um Dateien mit den Definitionen Ihrer Clients der Version 5.1 zu erstellen.

Syntax



Erforderliche Parameter

-m *WS-Managername*

Gibt den WS-Manager an, dem die Kanaldefinitionsdatei (CCHDEF), die konvertiert werden soll, zugeordnet ist. Diese Angabe ist erforderlich.

Optionale Parameter

-o *Ausgabedatei*

Gibt die Datei an, die die konvertierten Definitionen enthalten soll. Der Standarddateiname ist AMQCLCHL.

cnvclchl

Beispiele

Mit dem folgenden Befehl wird eine strukturierte Compaq-Datei mit Client-Kanaldefinitionen für den WS-Manager MT01 in eine unstrukturierte Datei konvertiert. Die Ausgabedatei AMQCLCHL wird die Definitionen für zwei Client-Kanalverbindungen enthalten (SYSTEM.DEF.CLNTCONN und SOLARIS_TO_Compag):

```
$DATA01 SZMON 330> cnvclchl -m MV4  
MQSeries client channel table being converted  
Opening Compaq NSK v5.1 CLNTCONN table  
Opening Common v5.1 CLNTCONN table AMQCLCHL for output  
Writing Common v5.1 CLNTCONN table entry for SYSTEM.DEF.CLNTCONN  
Closing Compaq NSK v5.1 CLNTCONN table  
Closing Common v5.1 CLNTCONN table  
MQSeries client channel table conversion complete.
```

crtmqcvx (Datenkonvertierung)

Funktion

Mit dem Befehl **crtmqcvx** erstellen Sie ein Codefragment, das eine Datenkonvertierung von Datentypstrukturen ausführt. Der Befehl generiert eine C-Funktion, die in einem Exit zum Konvertieren von C-Strukturen verwendet werden kann.

Der Befehl liest eine Eingabedatei mit einer oder mehreren Strukturen, die konvertiert werden soll(en). Anschließend wird eine Ausgabedatei mit einem Codefragment oder Fragmenten zum Konvertieren dieser Strukturen erstellt.

Weitere Informationen zu diesem Befehl und zu seiner Verwendung finden Sie im Handbuch *MQSeries Application Programming Guide*.

Syntax

```
►► crtmqcvx—Quellendatei—Zieldatei—►►
```

Erforderliche Parameter

Quellendatei

Gibt die Eingabedatei mit den zu konvertierenden C-Strukturen an.

Zieldatei

Gibt die Ausgabedatei mit den Codefragmenten an, die zum Konvertieren der Strukturen generiert wurden.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde mit unerwarteten Ergebnissen beendet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

Beispiele

Das folgende Beispiel zeigt die Ergebnisse des Datenkonvertierungsbefehls bei einer Anwendung auf eine C-Quellenstruktur. Der ausgegebene Befehl lautet wie folgt:

```
crtmqcvx Quellendatei Zieldatei
```

crtmqcvx

Die Eingabedatei Quellendatei hat den folgenden Inhalt:

```
/* Dies ist eine C-Teststruktur, die mit dem Dienstprogramm */
/* crtmqcvx konvertiert werden kann. */

struct my_structure
{
    int    code;
    MQLONG value;
};
```

Die mit diesem Befehl generierte Ausgabedatei Ausgabedatei ist unten zu sehen. Sie können diese Codefragmente in Ihren Anwendungen zum Konvertieren von Datenstrukturen verwenden. Dabei sollten Sie jedoch berücksichtigen, dass das Fragment die in der MQSeries-Header-Datei **MQSVMHTH** im Teildatenträger **ZMQSLIB** zur Verfügung gestellten Makros verwendet.

```
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);
}
```

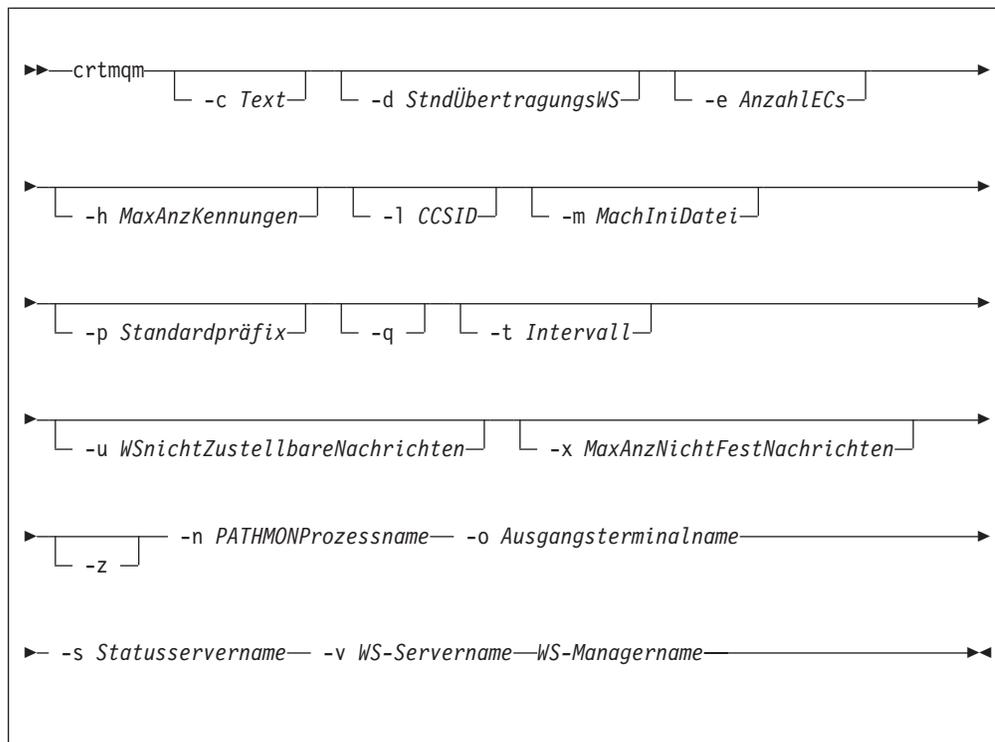
crtmqm (WS-Manager erstellen)

Funktion

Mit dem Befehl **crtmqm** erstellen Sie einen lokalen WS-Manager. Nachdem Sie einen WS-Manager erstellt haben, können Sie ihn mit dem Befehl **strmqm** starten.

Bei der Erstellung eines WS-Managers werden automatisch auch die zugehörigen System- und Standardobjekte erstellt.

Syntax



Erforderliche Parameter

- n *PATHMONProzessname*
Gibt den Namen des TS/MP-PATHMON-Prozesses für den WS-Manager an. Dieser Prozessname muss systemweit eindeutig sein.
- o *Ausgangsterminalname*
Gibt den Einheitennamen des Ausgangsterminals im Format \$DDDD.#SS an. Beispiel: \$TRM1.#A.
- s *Statusservername*
Gibt den Prozessnamen für den Standardstatusserver des WS-Managers an. Der Prozessname muss systemweit eindeutig sein.
- v *WS-Servername*
Gibt einen eindeutigen Prozessnamen für den standardmäßigen Warteschlangenserverprozess dieses WS-Managers an.

WS-Managername

Gibt den Namen des WS-Managers an, der erstellt werden soll. Der Name darf maximal 48 Zeichen lang sein. Diese Angabe muss am Ende des Befehls stehen.

Optionale Parameter

-c *Text*

Eine Beschreibung (bis zu 64 Zeichen) des WS-Managers. Standardmäßig stehen hier Leerzeichen.

Bei Verwendung von Sonderzeichen muss die Beschreibung in doppelten Anführungszeichen stehen.

-d *StndÜbertragungsWS*

Gibt den Namen der lokalen Übertragungswarteschlange an, in die ferne Nachrichten gestellt werden sollen, wenn für die Zieladresse keine Übertragungswarteschlange angegeben ist. Es gibt keinen Standardwert.

-e *AnzahlECs*

Gibt die Anzahl der Ausführungs-Controller-Prozesse (EC, Execution Controller) im WS-Manager an. Der Standardwert ist 1.

-h *MaxAnzKennungen*

Dieser Parameter wird in MQSeries for Compaq NSK ignoriert.

Gibt die maximale Anzahl von Kennungen an, die eine einzelne Anwendung gleichzeitig öffnen kann. Geben Sie einen Wert zwischen 1 und 999 999 999 an. Der Standardwert ist 256.

-l *CCSID*

Gibt die CCSID des WS-Managers an. Der Standardwert ist 819.

-m *MachIniDatei*

Dieser Wert überschreibt den Verzeichnispfad mit der Standarddatei MQSINI und den in der Umgebungsvariablen MQMACHINIFILE angegebenen Pfad.

-p *Standardpräfix*

Gibt den Datenträger für den WS-Manager an. Dieser Wert überschreibt den Eintrag QMDefaultVolume in der Datei MQSINI.

-q

Gibt an, dass dieser WS-Manager der Standard-WS-Manager werden soll. Der neue WS-Manager ersetzt den eventuell vorhandenen Standard-WS-Manager.

Wenn Sie diese Option versehentlich angeben und zum vorherigen Standard-WS-Manager zurückkehren möchten, können Sie die Zeilengruppe DefaultQueueManager in der MQSeries-Konfigurationsdatei editieren.

-t *Intervall*

Gibt das Auslösezeitintervall in Millisekunden für alle von diesem WS-Manager gesteuerten Warteschlangen an. Dieser Wert gibt die Zeit nach dem Empfang einer Nachricht mit einer Auslösefunktion an, nach deren Ablauf die Auslösefunktion ausgesetzt wird. Das heißt, wenn beim Eingang einer Nachricht in einer Warteschlange eine Auslösenachricht in die Initialisierungswarteschlange eingereicht wird, wird bei weiteren Nachrichten, die innerhalb des angegebenen Intervalls in derselben Warteschlange ankommen, keine weitere Auslösenachricht generiert.

Über das Auslösezeitintervall können Sie sicherstellen, dass Ihre Anwendung genügend Zeit zur Bearbeitung einer Auslösebedingung hat, bevor sie durch eine Alarmmeldung veranlasst wird, sich um eine weitere Auslösebedingung in derselben Warteschlange zu kümmern. Wenn Sie über alle auftretenden Auslöseereignisse informiert werden möchten, geben Sie in diesem Feld einen niedrigen Wert oder null an.

Geben Sie einen Wert im Bereich von 0 bis 999 999 999 an. Der Standardwert ist 999 999 999 Millisekunden, eine Zeit von mehr als 11 Tagen. Wenn Sie den Standardwert übernehmen, bedeutet dies nichts anderes, als dass die Auslösefunktion nach der ersten Auslösenachricht inaktiviert wird. Die Auslösefunktion kann jedoch von einer Anwendung reaktiviert werden, die für die Warteschlange einen ALTER QUEUE-Befehl ausgibt, um die Auslöseattribute zurückzusetzen.

-u *WSnichtZustellbareNachrichten*

Gibt den Namen der lokalen Warteschlange an, in die nicht zustellbare Nachrichten eingereiht werden sollen. In diese Warteschlange werden Nachrichten eingereiht, die nicht an ihre korrekte Zieladresse weitergeleitet werden können.

Standardmäßig ist keine DLQ (Dead-Letter Queue, Warteschlange für nicht zustellbare Nachrichten) vorhanden.

-x *MaxAnzNichtFestNachrichten*

Dieser Parameter wird in MQSeries for Compaq NSK ignoriert.

Gibt die maximale Anzahl nicht festgeschriebener Nachrichten an einem Synchronisationspunkt an. Die Anzahl ergibt sich aus der Summe der folgenden Nachrichten:

- alle Nachrichten, die aus Warteschlangen abgerufen werden können.
- alle Nachrichten, die in Warteschlangen eingereiht werden können.
- alle Auslösenachrichten, die in dieser Arbeitseinheit generiert werden.

Diese Begrenzung gilt nicht für Nachrichten, die außerhalb eines Synchronisationspunktes abgerufen oder eingereiht werden.

Geben Sie einen Wert zwischen 1 und 10000 an. Der Standardwert sind 1000 nicht festgeschriebene Nachrichten.

-z Diese Option unterdrückt Fehlernachrichten.

Sie wird in MQSeries normalerweise verwendet, um unerwünschte Fehlernachrichten zu unterdrücken. Da bei Angabe dieser Option Informationen verloren gehen können, wird empfohlen, sie nicht zu verwenden, wenn Befehle in einer Befehlszeile eingegeben werden.

crtmqm

Rückkehrcodes

0	WS-Manager wurde erstellt
8	WS-Manager ist bereits vorhanden
49	WS-Manager wird gestoppt
69	Kein Speicher verfügbar
70	Kein Speicher für Warteschlange verfügbar.
71	Unerwarteter Fehler
72	Falscher WS-Managername
111	Der WS-Manager wurde erstellt. Allerdings gab es einen Fehler bei der Verarbeitung der Definition für den Standard-WS-Manager in der Konfigurationsdatei des Produkts. Die Spezifikation für den Standard-WS-Manager ist möglicherweise nicht korrekt.

Beispiele

1. Mit diesem Befehl wird ein Standard-WS-Manager mit dem Namen Paint.queue.manager und der Beschreibung Paint Shop erstellt:

```
crtmqm -c "Paint Shop" -n $PANT -o $TRM1.#A -s $PNT1 -v $PQS1 Paint.queue.manager
```

2. In diesem Beispiel wird ein weitere WS-Manager mit dem Namen travel erstellt. Das Auslöseintervall wird mit 5000 Millisekunden (oder 5 Sekunden) angegeben, und die Warteschlange für nicht zustellbare Nachrichten für diesen WS-Manager heißt SYSTEM.DEAD.LETTER.QUEUE.

```
crtmqm -t 5000 -u SYSTEM.DEAD.LETTER.QUEUE -n $TRAV -o $TRM1.#A -s $TRV1 -v $TQS1 travel
```

Nach der Generierung eines Auslöseereignisses werden weitere Auslöseereignisse für fünf Sekunden inaktiviert.

Zugehörige Befehle

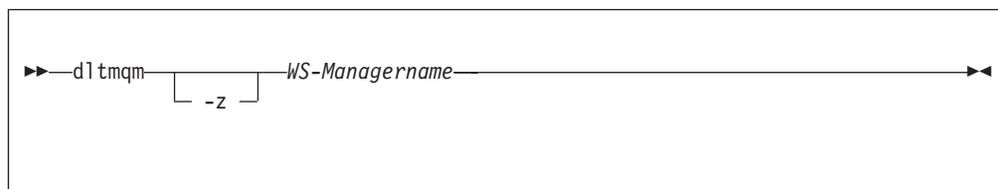
strmqm	WS-Manager starten
endmqm	WS-Manager beenden
dltmqm	WS-Manager löschen

dltmqm (WS-Manager löschen)

Funktion

Mit dem Befehl **dltmqm** löschen Sie einen angegebenen WS-Manager. Alle diesem WS-Manager zugeordneten Objekte werden ebenfalls gelöscht. Bevor Sie einen WS-Manager löschen können, müssen Sie ihn mit dem Befehl **endmqm** beenden.

Syntax



Erforderliche Parameter

WS-Managername

Gibt den Namen des zu löschenden WS-Managers an.

Optionale Parameter

-z Diese Option unterdrückt Fehlermeldungen.

Rückkehrcodes

0	WS-Manager gelöscht
5	WS-Manager aktiv
16	WS-Manager nicht vorhanden
69	Kein Speicher verfügbar
71	Unerwarteter Fehler
72	Falscher WS-Managername
112	WS-Manager wurde gelöscht. Allerdings gab es einen Fehler bei der Verarbeitung der Definition für den Standard-WS-Manager in der Konfigurationsdatei des Produkts. Die Spezifikation für den Standard-WS-Manager ist möglicherweise nicht korrekt.

Beispiele

1. Mit dem folgenden Befehl wird der WS-Manager saturn.queue.manager gelöscht:

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

2. Mit dem folgenden Befehl wird der WS-Manager travel gelöscht; außerdem werden alle durch den Befehl verursachten Nachrichten unterdrückt:

```
dltmqm -z travel
```

dltmqm

Zugehörige Befehle

crtmqm	WS-Manager erstellen
strmqm	WS-Manager starten
endmqm	WS-Manager beenden

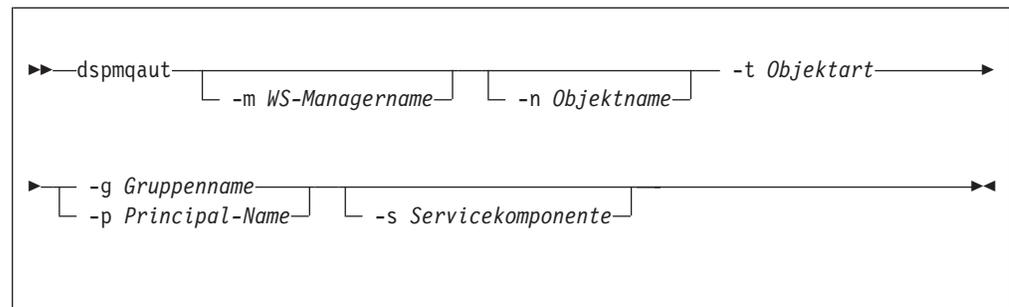
dspmqaout (Berechtigung anzeigen)

Funktion

Mit dem Befehl **dspmqaout** zeigen Sie die aktuellen Berechtigungen für ein bestimmtes Objekt an. Es kann nur eine einzige Gruppe angegeben werden.

Wenn eine Benutzer-ID zu mehreren Gruppen gehört, sehen Sie sich die Berechtigungen der einzelnen Gruppen an, um alle Berechtigungen der Benutzer-ID zu ermitteln.

Syntax



Erforderliche Parameter

-g *Gruppenname*

Gibt den Namen der Benutzergruppe an, die abgefragt werden soll. Sie können nur *einen* Namen angeben, bei dem es sich um eine vorhandene Benutzergruppe handeln muss. Sie müssen entweder -g *Gruppenname* oder -p *Principal-Name* angeben.

-p *Principal-Name*

Gibt den Namen des Principals an, für den die Berechtigungen auf das angegebene Objekt angezeigt werden sollen. Sie müssen entweder -g *Gruppenname* oder -p *Principal-Name* angeben.

-t *Objektart*

Gibt die Art des Objekts an, das abgefragt wird. Gültige Werte:

queue oder q Eine Warteschlange oder Warteschlangen, die dem Parameter für die Objektart entsprechen.

qmgr Ein WS-Managerobjekt

process oder prcs
Ein Prozess

namelist oder nl
Eine Namensliste

Optionale Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, der abgefragt werden soll.

-n *Objektname*

Gibt den Namen des Objekts an, das abgefragt wird.

Dieser Parameter muss angegeben werden, *außer* wenn es sich bei dem Objekt um den WS-Manager selbst handelt.

dspmqaout

Sie müssen den Namen des WS-Managers, der Warteschlange oder der Prozessdefinition angeben.

-s Servicekomponente

Dieser Parameter gilt nur, wenn Sie installierbare Berechtigungsservices verwenden, andernfalls wird er ignoriert.

Wenn installierbare Berechtigungsservices unterstützt werden, gibt dieser Parameter den Namen des Berechtigungsservices an, für den die Berechtigungen gelten. Dieser Parameter ist optional; wenn er nicht angegeben wird, gilt die Aktualisierung für die erste installierbare Komponente für den Service.

Zurückgegebene Parameter

Dieser Befehl gibt eine Berechtigungsliste zurück, die keinen, einen oder mehrere Berechtigungsparameter enthalten kann. Für jeden zurückgegebenen Berechtigungsparameter gilt, dass jede Benutzer-ID in der angegebenen Gruppe über die Berechtigung verfügt, die durch diesen Parameter definierte Operation auszuführen.

Tabelle 9 zeigt die Berechtigungen, die an die unterschiedlichen Objektarten erteilt werden können.

Tabelle 9. Sicherheitsberechtigungen mit dem Befehl dspmqaout

Berechtigung	Warteschlange	Prozess	WS-Manager	Namensliste
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
chgaut	✓	✓	✓	
clr	✓			
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

In der folgenden Liste werden die Berechtigungen definiert, die den einzelnen Parametern zugeordnet sind:

all	Ausführen aller Operationen für das Objekt.
alladm	Ausführen aller Verwaltungsoperationen für das Objekt.
allmqi	Ausführen aller MQI-Aufrufe für das Objekt.
altusr	Angeben einer alternativen Benutzer-ID in einem MQI-Aufruf.
Anzeigen (browse)	Abrufen einer Nachricht aus einer Warteschlange durch Ausgeben eines MQGET-Aufrufs mit der Option BROWSE.
chg	Ändern der Attribute eines angegebenen Objekts mit Hilfe des entsprechenden Befehlssatzes.
chgaut	Angeben der Berechtigungen anderer Benutzergruppen auf das Objekt unter Verwendung des Befehls setmqaut .
clr	Löschen des Inhalts einer Warteschlange (nur PCF-Befehl 'Clear queue').
connect	Verbinden der Anwendung mit dem angegebenen WS-Manager durch Ausgeben eines MQCONN-Aufrufs.
crt	Erstellen von Objekten der angegebenen Objektart mit Hilfe des entsprechenden Befehlssatzes.
dlt	Löschen des angegebenen Objekts mit Hilfe des entsprechenden Befehlssatzes.
dsp	Anzeigen der Attribute des angegebenen Objekts mit Hilfe des entsprechenden Befehlssatzes.
get	Abrufen einer Nachricht aus einer Warteschlange durch Ausgeben eines MQGET-Aufrufs.
inq	Abfragen einer angegebenen Warteschlange durch Ausgeben eines MQINQ-Aufrufs.
passall	Übergeben des gesamten Kontextes.
passid	Übergeben des Identitätskontextes.
put	Einreihen einer Nachricht in eine angegebene Warteschlange durch Ausgeben eines MQPUT-Aufrufs.
set	Setzen von Attributen für eine Warteschlange über die MQI durch Ausgeben eines MQSET-Aufrufs.
setall	Setzen des gesamten Kontextes für eine Warteschlange.
setid	Setzen des Identitätskontextes für eine Warteschlange.

Die Berechtigungen für Verwaltungsoperationen, sofern unterstützt, gelten für die folgenden Befehlssätze:

- Steuerbefehle
- mit MQSC-Befehlen
- mit PCF-Befehlen

dspmqaut

Rückkehrcodes

0	Operation war erfolgreich
36	Ungültige Argumente übergeben
40	WS-Manager ist nicht verfügbar
49	WS-Manager wird gestoppt
69	Kein Speicher verfügbar
71	Unerwarteter Fehler
72	Falscher WS-Managername
133	Unbekannter Objektname
145	Unerwarteter Objektname
146	Objektname fehlt
147	Objektart fehlt
148	Ungültige Objektart
149	Name der Definitionseinheit fehlt

Beispiele

Mit dem folgenden Befehl werden die Berechtigungen der Benutzergruppe staff für den WS-Manager saturn.queue.manager angezeigt:

```
dspmqaut -m saturn.queue.manager -t qmgr -g staff
```

Das Ergebnis sieht wie folgt aus:

```
Entity staff has the following authorizations for object :
  get
  browse
  put
  inq
  set
  connect
  altusr
  passid
  passall
  setid
```

Zugehörige Befehle

setmqaut Berechtigung setzen oder zurücksetzen

dspmqsrv (Befehlsserver anzeigen)

Funktion

Mit dem Befehl **dspmqsrv** zeigen Sie den Status des Befehlsservers für den angegebenen WS-Manager an.

Folgende Statusangaben sind möglich:

- Starten
- Aktiv
- Aktiv (ohne dass SYSTEM.ADMIN.COMMAND.QUEUE für GET-Aufrufe aktiviert ist)
- Stoppen
- Gestoppt

Syntax

```
►► dspmqsrv WS-Managername ◀◀
```

Erforderliche Parameter

WS-Managername

Gibt den Namen des lokalen WS-Managers an, für den der Status des Befehlsservers abgefragt wird.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde mit unerwarteten Ergebnissen beendet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

Beispiele

Der folgende Befehl zeigt den Status des Befehlsservers, der dem WS-Managers `venus.q.mgr` zugeordnet ist, an:

```
dspmqsrv venus.q.mgr
```

Zugehörige Befehle

- strmqsv** Befehlsserver starten
- endmqsv** Befehlsserver beenden

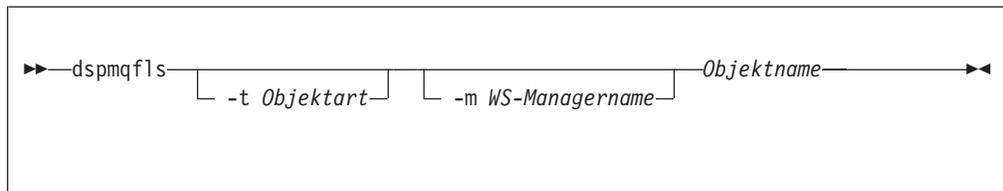
dspmqls (MQSeries-Dateiattribute anzeigen)

Funktion

Mit dem Befehl **dspmqls** zeigen Sie die tatsächlichen Dateisystemnamen für alle MQSeries-Objekte an, die einem bestimmten Kriterium entsprechen. Mit Hilfe dieses Befehls können Sie die Dateien identifizieren, die einem bestimmten MQSeries-Objekt zugeordnet sind. Dies ist bei der Sicherung bestimmter Objekte hilfreich. Weitere Informationen zur Namensumwandlung finden Sie unter „Datenträgerstruktur“ auf Seite 64.

Mit dem Befehl **dspmqls** können Sie außerdem die aktuellen Einstellungen für den Prozessnamen des Warteschlangenservers und die Optionen eines Objekts anzeigen.

Syntax



Erforderliche Parameter

Objektname

Gibt den Namen des MQSeries-Objekts an. Der Name besteht aus einer Zeichenfolge ohne Markierungen; die Angabe dieses Parameters ist erforderlich. Wenn kein Name angegeben wird, wird ein Fehler zurückgegeben.

Dieser Parameter unterstützt das Platzhalterzeichen * am Ende der Zeichenfolge.

Optionale Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, für den Dateien überprüft werden.

Wird dieser Parameter nicht angegeben, wird der Befehl für den Standard-WS-Manager ausgeführt.

-t *Objektart*

Gibt die Art des MQSeries-Objekts an. In der folgenden Liste werden die gültigen Objektarten aufgeführt. Zunächst wird die Abkürzung genannt, daneben steht der vollständiger Name der Objektart.

*** oder all** Alle Objektarten; dies ist der Standardwert.

q oder queue Eine Warteschlange oder Warteschlangen, die dem Parameter für den Objektnamen entsprechen.

ql oder qlocal Eine lokale Warteschlange

qa oder qalias Eine Aliaswarteschlange

qr oder qremote

Eine ferne Warteschlange

qm oder qmodel

Eine Modellwarteschlange

qmgr Ein WS-Managerobjekt

prcs oder process
Ein Prozess

ctlg oder catalog
Ein Objektkatalog

nl oder namelist
Eine Namensliste

Anmerkung: Der Befehl **dspmqls** zeigt die Namen aller Dateien für die Warteschlange an.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde beendet, aber nicht vollständig wie erwartet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

Beispiele

1. Mit dem folgenden Befehl werden Details zu allen Objekten angezeigt, deren Namen mit SYSTEM.ADMIN beginnen und die im Standard-WS-Manager definiert sind:

```
dspmqls SYSTEM.ADMIN*
```

2. Mit dem folgenden Befehl werden Dateinformationen für alle Prozesse angezeigt, deren Namen mit PROC beginnen und die im WS-Manager RADIUS definiert sind:

```
dspmqls -m RADIUS -t prcs PROC*
```

3. Mit dem folgenden Befehl werden Dateiangaben für MY.LOCAL.QUEUE angezeigt:

```
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:
```

- Bei QMYXLOCA handelt es sich um die Warteschlangendatei, bei OMYXLOCA um die Überlaufdatei der Warteschlange und bei TMYXLOCA um die Änderungsdatei im Datenverzeichnis des WS-Managers.

dspmqfls

4. In dem folgenden Befehl **altmqfls** sind die Warteschlangenserveroptionen S und C angegeben, d. h. die lokale Warteschlange wird beim Start des WS-Managers geladen, und für nicht permanente Nachrichten werden Prüfpunktinformationen gespeichert. Der Befehl **dspmqfls** zeigt das Ergebnis des Befehls **altmqfls** an:

```
altmqfls --qmgr MT02 --type ql --qsoptions SC MY.LOCAL.QUEUE

dspmqfls -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:
```

Zugehörige Befehle

altmqfls Datenträger für Warteschlange ändern

dspmqtrc (Formatierte MQSeries-Trace-Ausgabe anzeigen)

Funktion

Mit dem Befehl **dspmqtrc** zeigen Sie eine formatierte MQSeries-Trace-Ausgabe an. Weitere Informationen zur Verwendung der Trace-Funktion in MQSeries finden Sie unter „MQSeries-Trace verwenden“ auf Seite 222.

Syntax

```
▶▶—dspmqtrc— -t Formatschablone —Eingabedatei————▶▶
```

Erforderliche Parameter

Eingabedatei

Gibt den Namen der Datei an, die den unformatierten Trace enthält. Beispiel:
\$DATA.MQTRACE.AMQ12345.

-t *Formatschablone*

Gibt den Namen der Schablonendatei an, die beschreibt, wie der Trace angezeigt werden soll. Eine Datei mit einer Trace-Formatschablone (AMQTRC) steht im Unterverzeichnis ZMQSSMPL zur Verfügung.

Zugehörige Befehle

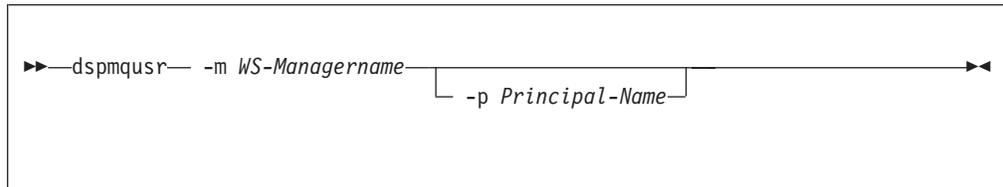
endmqtrc	MQSeries-Trace beenden
strmqtrc	MQSeries-Trace starten

dspmqsqr (MQSeries-Benutzerinformationen anzeigen)

Funktion

Mit dem Befehl **dspmqsqr** können Sie Informationen zu einem bestimmten Principal oder zu allen Principals eines WS-Managers anzeigen.

Syntax



Beschreibung

Mit diesem Befehl können Sie:

- alle Principals oder einen bestimmten Principal für einen WS-Manager anzeigen,
- die Compaq NSK-Verwaltungsgruppen und die SAFEGUARD-Gruppen mit gemeinsamem Dateizugriff anzeigen, die den Compaq NSK-Benutzer-IDs der einzelnen Principals entsprechen.

Erforderliche Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, dem der Principal zugeordnet ist.

Optionale Parameter

-p *Principal-Name*

Gibt den Namen des Principals an, der angezeigt werden soll.

Rückkehrcodes

- 0 Operation war erfolgreich
- 36 Ungültige Argumente übergeben
- 69 Kein Speicher verfügbar
- 71 Unerwarteter Fehler

Beispiele

1. Mit dem folgenden Befehl **dspmqsqr** werden Informationen zu einem neuen WS-Manager angezeigt:

```
dspmqsqr -m MT02
```

Principal	Userid	Username	Alias	GroupName	GroupType
	0.1				
NOBODY	0.0				
mqm	20.255	MQM.MANAGER	n	MQM	a

Die Principal-Datenbank enthält den Principal **mqm**, der dem Namen des Benutzers zugeordnet ist, von dem der WS-Manager erstellt wurde.

2. Im folgenden Beispiel handelt es sich um die Ausgabe für den Befehl **dspmqsqr**, nachdem mit **altmqsr** ein weiterer Principal hinzugefügt wurde:

```
dspmqsqr -m MT02
```

Principal	Userid	Username	Alias	GroupName	GroupType
	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

Der Principal **mquser1**, der dem Compaq-Benutzer ID **MQTEST.FRED** zugeordnet ist, wurde hinzugefügt. **FRED** gehört zur Gruppe **MQTEST** und zur Gruppe **MQM** unter Verwendung der **SAFEGUARD**-Aliasnamensfunktion.

Der Principal **mquser2** entspricht der Compaq-Benutzer-ID **GROUP.USER01**.

Zugehörige Befehle

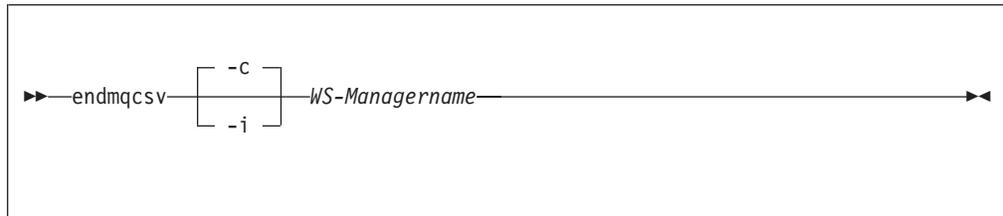
altmqsr MQSeries-Benutzerinformationen ändern

endmqcsv (Befehlsserver beenden)

Funktion

Mit dem Befehl **endmqcsv** stoppen Sie den Befehlsserver in dem angegebenen WS-Manager.

Syntax



Erforderliche Parameter

WS-Managername

Gibt den Namen des WS-Managers an, für den der Befehlsserver beendet werden soll.

Optionale Parameter

- c Gibt an, dass der Befehlsserver auf kontrollierte Weise gestoppt werden soll. Der Befehlsserver darf die Verarbeitung aller bereits gestarteten Befehlsnachrichten beenden. Es werden keine neuen Nachrichten aus der Befehlswarteschlange gelesen.
Dies ist die Standardeinstellung.
- i Gibt an, dass der Befehlsserver sofort gestoppt wird. Aktionen in Verbindung mit einer zu dem Zeitpunkt ausgeführten Befehlsnachricht werden möglicherweise nicht beendet.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde mit unerwarteten Ergebnissen beendet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

Beispiele

1. Mit dem folgenden Befehl wird der Befehlsserver im WS-Manager saturn.queue.manager gestoppt:

```
endmqcsv -c saturn.queue.manager
```

Der Befehlsserver kann die Verarbeitung der bereits gestarteten Befehle beenden, bevor er stoppt. Alle neu empfangenen Befehle verbleiben unbearbeitet in der Befehlswarteschlange, bis der Befehlsserver erneut gestartet wird.

2. Mit dem folgenden Befehl wird der Befehlsserver im WS-Manager pluto sofort gestoppt:

```
endmqcsv -i pluto
```

Zugehörige Befehle

<code>strmqcsv</code>	Befehlsserver starten
<code>dspmqs</code>	Status eines Befehlsservers anzeigen

endmqm (WS-Manager beenden)

Funktion

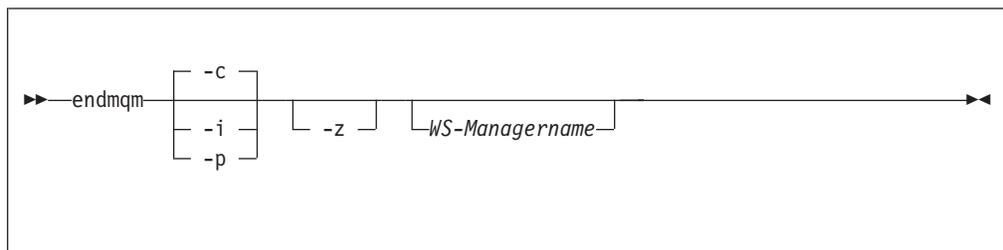
Mit dem Befehl **endmqm** beenden (stoppen) Sie einen angegebenen lokalen WS-Manager. Dieser Befehl stoppt den WS-Manager auf eine von drei Arten:

- normaler bzw. gesteuerter Abschluss
- sofortiger Abschluss
- erzwungener Abschluss

Die Attribute des WS-Managers und die ihm zugeordneten Objekte sind nicht betroffen. Sie können den WS-Manager mit dem Befehl **strmqm** (WS-Manager starten) erneut starten.

Um einen WS-Manager zu löschen, müssen Sie ihn zunächst stoppen, bevor Sie dann den Befehl **dltmqm** (WS-Manager löschen) verwenden.

Syntax



Optionale Parameter

WS-Managername

Gibt den Namen des WS-Managers an, der gestoppt werden soll. Wird kein Name angegeben, wird der Standard-WS-Manager gestoppt.

- c Kontrollierter (bzw. gesteuerter) Abschluss. Der WS-Manager stoppt erst, nachdem alle Anwendungen ihre Verbindungen getrennt haben. Alle zu dem Zeitpunkt verarbeiteten MQI-Aufrufe werden beendet. Dies ist die Standardeinstellung.
- i Sofortiger Abschluss. Der WS-Manager stoppt, nachdem er alle zu dem Zeitpunkt verarbeiteten MQI-Aufrufe beendet hat. Alle MQI-Anforderungen, die nach Ausgabe des Befehls eingehen, schlagen fehl. Nicht beendete Arbeitseinheiten werden zurückgesetzt, wenn der WS-Manager das nächste Mal gestartet wird.
- p Erzwungener Abschluss.

Verwenden Sie diese Abschlussart nur in Ausnahmesituationen. Zum Beispiel, wenn ein WS-Manager nach Ausgabe eines normalen **endmqm**-Befehls nicht gestoppt wird.

Der WS-Manager stoppt, ohne darauf zu warten, dass Anwendungen ihre Verbindung beenden oder MQI-Aufrufe beendet werden. Dies kann unvorhersehbare Folgen für MQI-Anwendungen haben. Alle Prozesse des WS-Managers, der nicht gestoppt werden konnte, werden 30 Sekunden nach Ausgabe des Befehls beendet.

- z Unterdrückt Fehlermeldungen für diesen Befehl.

Rückkehrcodes

- 0 WS-Manager wurde beendet
- 16 WS-Manager nicht vorhanden
- 36 Ungültige Argumente
- 40 WS-Manager ist nicht verfügbar
- 69 Kein Speicher verfügbar
- 71 Unerwarteter Fehler
- 72 Falscher WS-Managername

Beispiele

Das folgende Beispiel zeigt Befehle, mit denen der angegebene WS-Manager beendet (gestoppt) wird.

1. Mit diesem Befehl wird ein gesteuerter Abschluss des Standard-WS-Managers ausgeführt. Alle Anwendungen, mit denen eine Verbindung besteht, können ihre Verbindungen trennen.

```
endmqm
```

2. Dieser Befehl beendet den WS-Manager `saturn.queue.manager` sofort. Alle aktuellen MQI-Aufrufe werden beendet, neue Aufrufe sind jedoch nicht zulässig.

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

Zugehörige Befehle

- | | |
|---------------------|----------------------|
| <code>crtmqm</code> | WS-Manager erstellen |
| <code>strmqm</code> | WS-Manager starten |
| <code>dltmqm</code> | WS-Manager löschen |

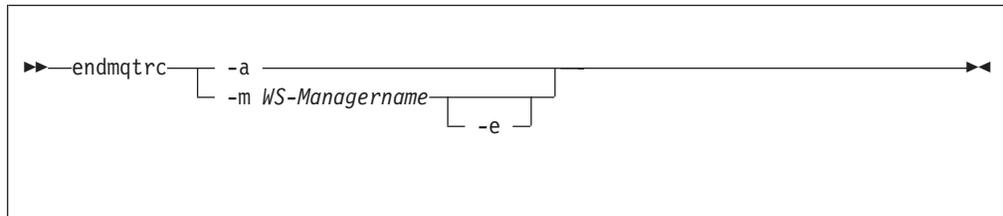
endmqtrc (MQSeries-Trace beenden)

Funktion

Mit dem Befehl **endmqtrc** beenden Sie den Trace für den angegebenen Server.

Weitere Informationen zur Verwendung der Trace-Funktion in MQSeries finden Sie unter „MQSeries-Trace verwenden“ auf Seite 222.

Syntax



Erforderliche Parameter

-m *WS-Managername*

Dies ist der Name des WS-Managers, für den der Trace beendet werden soll.

Es ist möglich, einen WS-Manager und die Option **-e** in demselben Befehl anzugeben.

-a Bei Angabe dieser Option werden alle Trace-Vorgänge beendet.

Diese Option *muss* allein angegeben werden.

Optionale Parameter

-e Bei Angabe dieser Option wird im WS-Manager der Trace, der in der Anfangsphase durchgeführt wird, beendet.

Rückkehrcodes

AMQ5611 Diese Nachricht wird ausgegeben, wenn im Befehl ungültige Argumente angegeben wurden.

Beispiele

Mit dem folgenden Befehl wird der Daten-Trace für den WS-Manager QM1 beendet:

```
endmqtrc -m QM1
```

Zugehörige Befehle

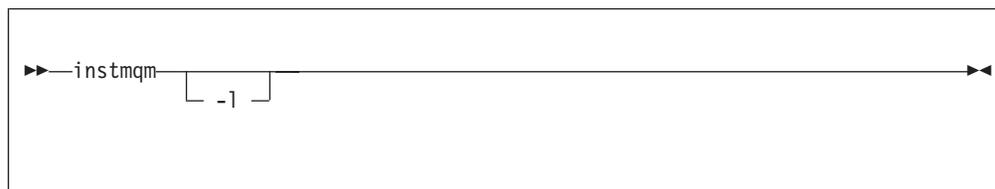
dspmqtrc Formatierte Trace-Ausgabe anzeigen
strmqtrc MQSeries-Trace starten

instmqm (MQSeries for Compaq NSK installieren)

Funktion

Mit dem Befehl **instmqm** können Sie MQSeries for Compaq NSK installieren oder Lizenzinformationen aktualisieren.

Syntax



Optionale Parameter

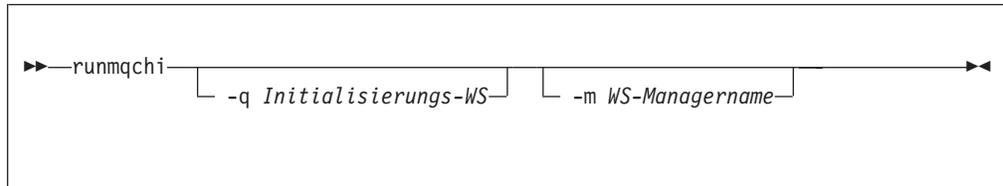
- l Gibt an, dass **instmqm** zum Aktualisieren der Lizenzinformationen aufgerufen werden soll.

runmqchi (Kanalinitiator ausführen)

Funktion

Mit dem Befehl **runmqchi** führen Sie einen Kanalinitiatorprozess aus. Weitere Informationen zu diesem Befehl finden Sie im Handbuch *MQSeries Intercommunication*.

Syntax



Optionale Parameter

-q *Initialisierungs-WS*

Gibt den Namen der Initialisierungswarteschlange an, die von diesem Kanalinitiator verarbeitet werden soll. Erfolgt keine Angabe, wird die Warteschlange SYSTEM.CHANNEL.INITQ verwendet.

-m *WS-Managername*

Gibt den Namen des WS-Managers an, in dem sich die Initialisierungswarteschlange befindet. Wird kein Name angegeben, wird der Standard-WS-Manager verwendet.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde mit unerwarteten Ergebnissen beendet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

Wenn Fehler auftreten, die zu einem Rückkehrcode 10 oder 20 führen, überprüfen Sie die Protokolldatei des WS-Managers, dem der Kanal zugeordnet ist, auf Fehlnachrichten. Überprüfen Sie außerdem das Systemfehlerprotokoll, denn dort werden Fehler aufgezeichnet, die auftreten, bevor der Kanal dem WS-Manager zugeordnet wird. Weitere Informationen zu Fehlerprotokollen finden Sie unter „Fehlerprotokolle“ auf Seite 219.

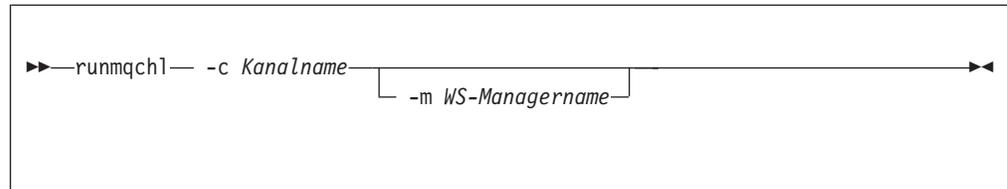
runmqchl (Kanal ausführen)

Funktion

Mit dem Befehl **runmqchl** starten Sie einen Senderkanal (SDR), einen Requester-Kanal (RQSTR) oder einen vollständig qualifizierten Serverkanal.

Der Kanal wird asynchron ausgeführt. Geben Sie zum Stoppen des Kanals den MQSC-Befehl STOP CHANNEL ein.

Syntax



Erforderliche Parameter

-c *Kanalname*

Gibt den Namen des Kanals an, der gestartet werden soll.

Optionale Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, dem dieser Kanal zugeordnet ist. Wird kein Name angegeben, wird der Standard-WS-Manager verwendet.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde mit unerwarteten Ergebnissen beendet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

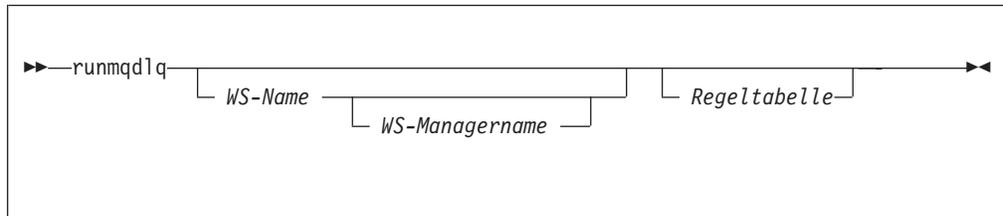
Wenn die Rückkehrcodes 10 oder 20 generiert werden, überprüfen Sie die Fehlerprotokolle des zugeordneten WS-Managers auf Fehlnachrichten. Überprüfen Sie außerdem die MQSeries-Systemfehlerprotokolle (in ZMQSSYS), denn dort werden Fehler aufgezeichnet, die auftreten, bevor der Kanal dem WS-Manager zugeordnet wird.

runmqdlq (Steerroutine der DLQ ausführen)

Funktion

Mit dem Befehl **runmqdlq** starten Sie die Steerroutine der Warteschlange für nicht zustellbare Nachrichten (DLQ, Dead-Letter Queue); dabei handelt es sich um ein Dienstprogramm, das die Nachrichten in einer Warteschlange für nicht zustellbare Nachrichten verarbeitet.

Syntax



Optionale Parameter

WS-Name

Gibt den Namen der Warteschlange für nicht zustellbare Nachrichten an, die verarbeitet werden soll.

Wenn Sie hier eine Warteschlange angeben, überschreibt dieser Wert den in einer Regeltabelle eventuell für INPUTQ angegebenen Wert. Ist weder im Befehl **runmqdlq** noch in der Regeltabelle ein Name angegeben, wird die DLQ verwendet, die dem im Parameter *WS-Managername* angegebenen WS-Manager zugeordnet ist.

WS-Managername

Gibt den Namen des WS-Managers an, dem die Warteschlange, die verarbeitet werden soll, zugeordnet ist.

Wenn Sie hier einen WS-Manager angeben, überschreibt dieser Wert den in einer Regeltabelle eventuell für INPUTQM angegebenen Wert. Ist weder im Befehl **runmqdlq** noch in der Regeltabelle ein Name angegeben, wird davon ausgegangen, dass die Warteschlange dem Standard-WS-Manager zugeordnet ist.

Regeltabelle

Gibt den Namen der Datei mit der Regeltabelle an, die mindestens eine Regel enthalten muss.

Standardmäßig wird im Befehl **runmqdlq** die Eingabe aus der Standardeingabedatei (IN) übernommen. Nach der Ausführung wird ein Bericht mit dem Ergebnis und einer Zusammenfassung erstellt und an die Standardausgabedatei (OUT) gesendet. Ebenso können Sie die angegebene Warteschlange durch Umleiten der Eingabe aus einer Datei mit einer Regeltabelle verknüpfen.

Wird für die Eingabe im Befehl **runmqdlq** keine Regeltabelle angegeben, müssen Regeln und Aktionen interaktiv angegeben werden.

In diesem Fall verhält sich die Steuerroutine der DLQ wie folgt:

- Sie erwartet Eingaben über die Tastatur.
- Sie beginnt erst mit der Verarbeitung der angegebenen Warteschlange, wenn sie das Dateiendezeichen ((Strg-Y)) empfängt.

Die MQSC-Regeln für Kommentarzeilen und das Verknüpfen von Zeilen gelten auch für die Eingabeparameter der Steuerroutine der Warteschlange für nicht zustellbare Nachrichten.

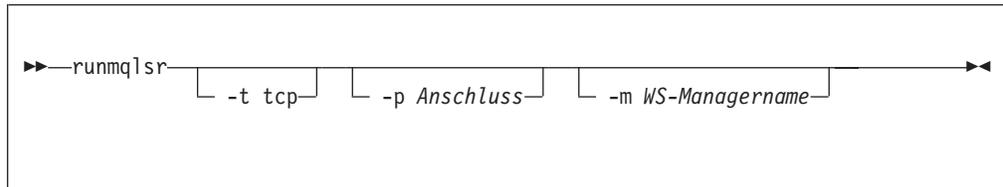
Weitere Informationen zu Regeltabellen und zu deren Erstellung finden Sie unter „Regeltabelle für die DLQ-Steuerroutine“ auf Seite 164.

runmqtsr (Empfangsprogramm ausführen)

Funktion

Mit dem Befehl **runmqtsr** wird ein TCP/IP-Empfangsprogrammprozess ausgeführt.

Syntax



Beschreibung

Wird der Befehl **runmqtsr** an einer TACL-Eingabeaufforderung eingegeben, gibt er die Steuerung erst nach Beendigung des Empfangsprogramms an die TACL-Eingabeaufforderung zurück. Das bedeutet, dass vor Abschluss des Befehls **runmqtsr** kein anderer Befehl an dem Terminal eingegeben werden kann.

Die TACL-Eingabeaufforderung steht erst wieder zur Verfügung, wenn ein Fehler auftritt oder das Empfangsprogramm stoppt. Wird das Terminal (TACL) vor Beendigung des Befehls **runmqtsr** gestoppt, kann das Empfangsprogramm weder auf das Ausgangsterminal noch auf die Ausgabedatei zugreifen. Vor dem Aufruf von **runmqtsr** müssen zunächst alle Umgebungsvariablen wie beispielsweise MQEMSEVENTS definiert werden.

Daher wird geraten, die Empfangsprogramme über die PATHWAY-Umgebung des WS-Managers zu starten bzw. zu stoppen, da so mehr Steuerungsmöglichkeiten gegeben sind.

Optionale Parameter

-p *Port*

Gibt die Port-Nummer für TCP/IP an. Wird kein Wert angegeben, wird die Port-Nummer übernommen, die im Eintrag TCPListenerPort in der Zeilengruppe TCPConfig der Datei QMINI angegeben ist. Der Standardwert ist 1414. Sind in der Datei QMINI mehrere Ports für Empfangsprogramme definiert, wird der nächste verfügbare Port verwendet.

Ist die Umgebungsvariable MPORTNUMBER in der TACL-Umgebung oder eine PATHWAY-Serverklassendefinition für das Programm angegeben, übernimmt **runmqtsr** diesen und nicht den in der Datei QMINI angegebenen Port.

Ist keiner der in der Datei QMINI angegebenen Ports frei oder ist der im Befehl **runmqtsr** angegebene Port nicht verfügbar, schlägt der Befehl **runmqtsr** fehl.

-m *WS-Managername*

Gibt den Namen des WS-Managers an. Wird kein Name angegeben, gilt der Befehl für den Standard-WS-Manager.

-t tcp

Gibt an, dass TCP/IP als Übertragungsprotokoll verwendet wird.

Ist DEFINE =TCPIP^PROCESS^NAME in der TACL-Umgebung oder eine PATHWAY-Serverklassendefinition für das Programm angegeben, verwendet **runmqtsr** den Guardian-TCP/IP-Serverprozess und nicht das Standardprotokoll.

Ist die Umgebungsvariable MQPORTNUMBER oder die PATHWAY-Serverklassendefinition für das Programm angegeben, übernimmt **runmqtsr** diesen und nicht den in der Datei QMINI angegebenen Port.

Dies ist die einzige zulässige Angabe (und der Standardwert) in MQSeries for Compaq NSK.

Rückkehrcodes

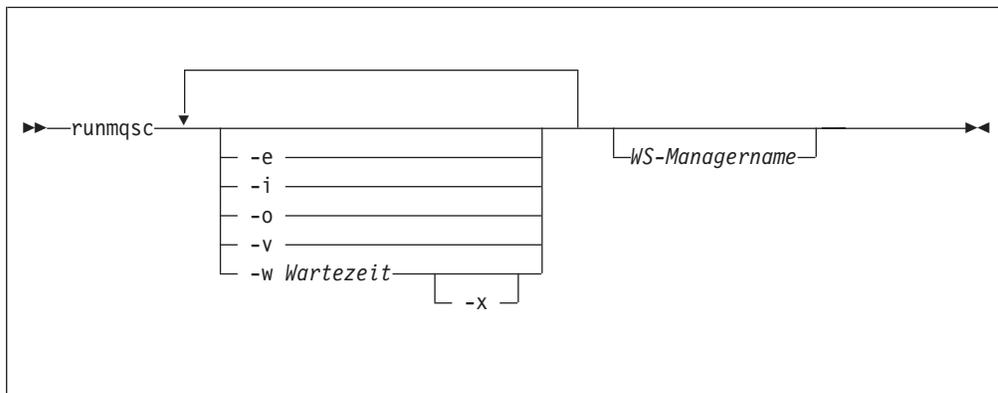
- 0 Befehl wurde normal beendet
- 10 Befehl wurde mit unerwarteten Ergebnissen beendet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

runmqsc (MQSeries-Befehle ausführen)

Funktion

Mit dem Befehl **runmqsc** geben Sie MQSC-Befehle an einen WS-Manager aus. Mit Hilfe von MQSC-Befehlen können Sie Verwaltungs-Tasks ausführen, z. B. ein lokales Warteschlangenobjekt definieren, ändern oder löschen. MQSC-Befehle und ihre Syntax werden im Handbuch *MQSeries MQSC - Befehle* beschrieben.

Syntax



Beschreibung

Sie können den Befehl **runmqsc** in drei Modi aufrufen:

Prüfmodus

MQSC-Befehle werden geprüft, aber nicht ausgeführt. Es wird ein Bericht generiert, der für jeden einzelnen Befehl angibt, ob er erfolgreich ausgeführt werden kann oder nicht. Dieser Modus ist nur in einem lokalen WS-Manager möglich.

Direkter Modus

MQSC-Befehle werden direkt an einen lokalen WS-Manager gesendet.

Indirekter Modus

MQSC-Befehle werden in einem fernen WS-Manager ausgeführt. Diese Befehle werden in die Befehlswarteschlange in einem fernen WS-Manager eingereicht und in der Reihenfolge ausgeführt, in der sie eingereicht wurden. Berichte des Befehls werden an den lokalen WS-Manager zurückgegeben.

Der Befehl **runmqsc** erhält seine Eingabe aus der Standardeingabedatei (IN). Nach der Ausführung wird ein Bericht mit dem Ergebnis und einer Zusammenfassung erstellt und an die Standardausgabedatei (OUT) gesendet.

Wird die Eingabe von der Standardeingabedatei an die Tastatur umgeleitet, können Sie MQSC-Befehle interaktiv eingeben.

Indem Sie als Eingabequelle eine Datei angeben, können Sie häufig verwendete Befehlsfolgen in der Datei eingeben und so leichter ausführen. Sie können auch den Ausgabebericht in eine Datei umleiten.

Anmerkung: Diesen Befehl können Sie nur ausführen, wenn Ihre Benutzer-ID zur Benutzergruppe **MQM** gehört.

Optionale Parameter

- e Verhindert, dass Quelltext für die MQSC-Befehle in einen Bericht kopiert wird. Dies ist hilfreich, wenn Sie Befehle interaktiv eingeben.
- i Gibt den Namen der Eingabedatei an.
- o Gibt den Namen der Ausgabedatei an.
- v Gibt an, dass der Prüfmodus verwendet wird, d. h., die angegebenen Befehle werden geprüft, die Aktionen jedoch nicht ausgeführt. Dieser Modus steht nur lokal zur Verfügung. Die Optionen -w und -x werden ignoriert, wenn sie gleichzeitig angegeben werden.

-w *Wartezeit*

Gibt an, dass der indirekte Modus verwendet wird, d. h., die MQSC-Befehle werden in einem anderen WS-Manager ausgeführt. Dazu müssen Sie den erforderlichen Kanal und die erforderlichen Übertragungswarteschlangen entsprechend konfiguriert sein.

Wartezeit

Gibt die Zeit in Sekunden an, die **runmqsc** auf Antworten wartet. Alle nach Ablauf dieser Zeit eingehenden Antworten werden gelöscht, die MQSC-Befehle jedoch weiter ausgeführt. Geben Sie eine Zeitspanne zwischen 1 und 999.999 Sekunden an.

Jeder Befehl wird als PCF-Escape-Befehl an die Befehlswarteschlange (SYSTEM.ADMIN.COMMAND.QUEUE) im Ziel-WS-Manager gesendet.

Die Antworten werden in der Warteschlange SYSTEM.MQSC.REPLY.QUEUE empfangen, und die Ausgabe wird dem Bericht hinzugefügt. Diese kann entweder als lokale Warteschlange oder als Modellwarteschlange definiert werden.

Operationen im indirekten Modus werden über den Standard-WS-Manager ausgeführt.

Diese Option wird ignoriert, wenn die Option -v angegeben ist.

- x Gibt an, dass der Ziel-WS-Manager aktiv ist. Diese Option kann nur im indirekten Modus angegeben werden. Die Option -w muss ebenfalls angegeben werden. Im indirekten Modus werden die MQSC-Befehle in einem Format geschrieben, das mit dem der MQSeries for MVS/ESA-Befehlswarteschlange kompatibel ist.

WS-Managername

Gibt den Namen des Ziel-WS-Managers an, in dem die MQSC-Befehle ausgeführt werden sollen. Wird kein Name angegeben, werden die MQSC-Befehle im Standard-WS-Manager ausgeführt.

Rückkehrcodes

- 00 MQSC-Befehlsdatei wurde erfolgreich ausgeführt.
- 10 MQSC-Befehlsdatei wurde mit Fehlern ausgeführt - Fehlerursachen sind im Bericht beschrieben.
- 20 Fehler - MQSC-Befehlsdatei nicht ausgeführt.

runmqsc

Beispiele

1. Geben Sie an der TACL-Eingabeaufforderung den folgenden Befehl ein:

```
runmqsc
```

Jetzt können Sie MQSC-Befehle direkt eingeben. Da kein WS-Manager angegeben wurde, werden die MQSC-Befehle im Standard-WS-Manager ausgeführt.

2. Das folgende Beispiel zeigt, wie MQSC-Befehle nur überprüft werden:

```
runmqsc -i $SYSTEM.CONFIG.MQSCIN -v BANK
```

Mit diesem Befehl wird die MQSC-Befehlsdatei \$SYSTEM.CONFIG.MQSCIN überprüft. Der Name des WS-Managers lautet BANK. Die Ausgabe wird im aktuellen Fenster angezeigt.

3. Mit diesem Befehl wird eine MQSC-Befehlsdatei für den WS-Manager BANK ausgeführt:

```
runmqsc -i MQSCFILE -o $TEST.MQ.MQSCOUT BANK
```

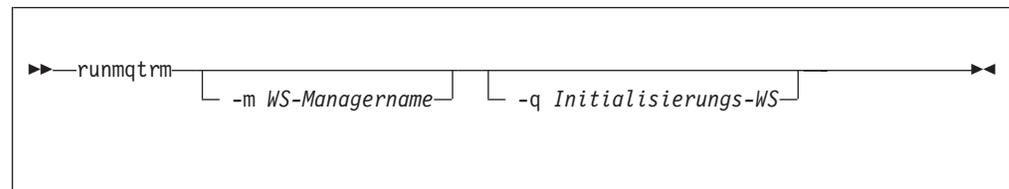
In diesem Beispiel wird die Ausgabe an die Datei \$TEST.MQ.MQSCOUT übertragen. Bei der Eingabedatei handelt es sich um MQSCFILE im aktuellen Teildatenträger.

runmqtrm (Auslösemonitor starten)

Funktion

Mit dem Befehl **runmqtrm** rufen Sie einen Auslösemonitor auf. Weitere Informationen zur Verwendung von Auslösemonitoren finden Sie im Handbuch *MQSeries Application Programming Guide*.

Syntax



Optionale Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, in dem der Auslösemonitor ausgeführt werden soll. Wird dieser Parameter nicht angegeben, wird der Auslösemonitor für den Standard-WS-Manager ausgeführt.

-q *Initialisierungs-WS*

Gibt den Namen der zu verarbeitenden Initialisierungswarteschlange an. Wird dieser Parameter nicht angegeben, wird SYSTEM.DEFAULT.INITIATION.QUEUE verwendet.

Rückkehrcodes

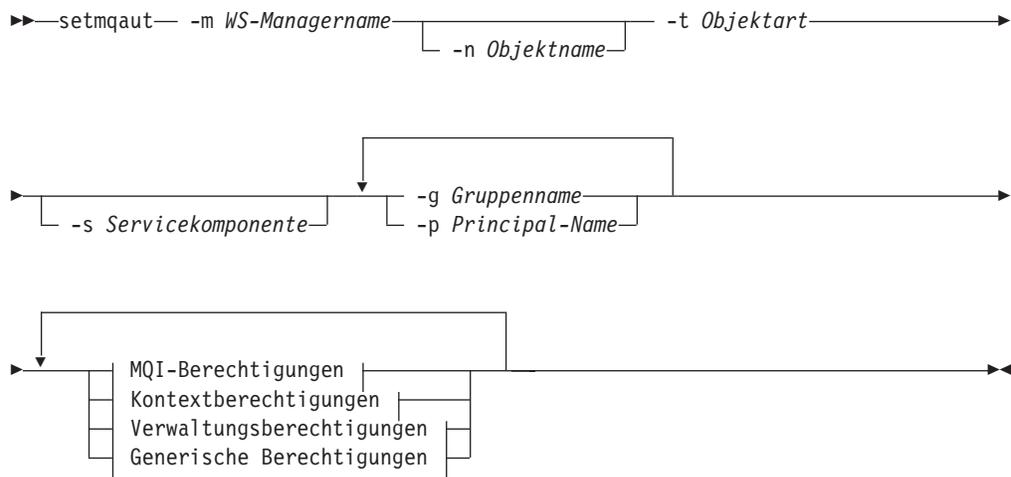
- 10 Auslösemonitor wurde durch einen Fehler unterbrochen.
- 20 Fehler—Auslösemonitor nicht ausgeführt.

setmqaut (Berechtigung setzen/zurücksetzen)

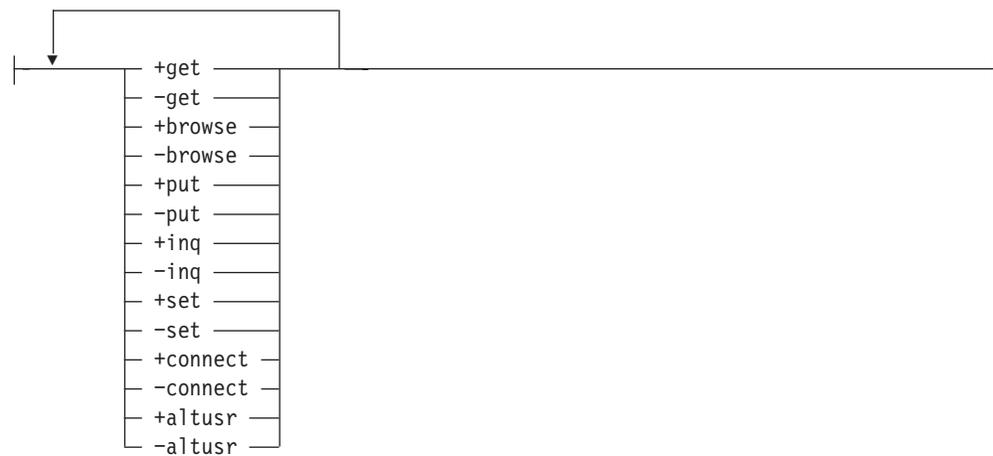
Funktion

Mit dem Befehl **setmqaut** ändern Sie die Berechtigungen für ein Objekt oder eine Objektklasse. Berechtigungen können beliebig vielen Principals oder Gruppen erteilt bzw. ihnen entzogen werden.

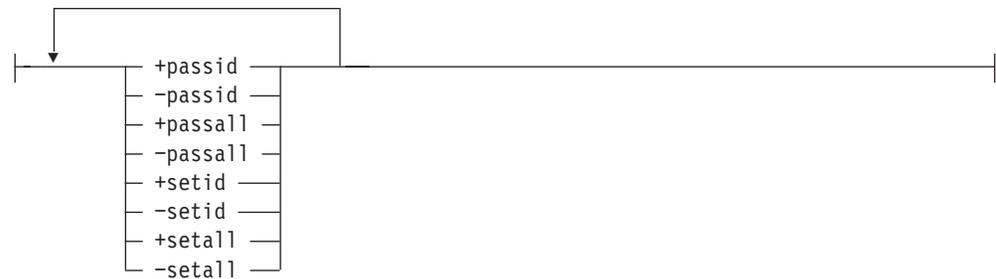
Syntax



MQI-Berechtigungen:



Kontextberechtigungen:



Verwaltungsberechtigungen:



Generische Berechtigungen:



Beschreibung

Mit diesem Befehl können Sie sowohl eine Berechtigung *setzen*, d. h. einer Benutzergruppe die Berechtigung zum Ausführen einer Operation erteilen, als auch eine Berechtigung *zurücksetzen*, d. h. die Berechtigung zum Ausführen einer Operation entziehen. Sie müssen die Benutzergruppen, für die die Berechtigungen gelten, sowie den WS-Manager und Art und Name des Objekts angeben. Sie können beliebig viele Gruppen angeben.

Die zu erteilenden Berechtigungen können in folgende Kategorien eingeteilt werden:

- Berechtigungen für das Ausgeben von MQI-Aufrufen
- Berechtigungen für MQI-Kontext

setmqaut

- Berechtigungen für das Ausgeben von Befehlen für Verwaltungs-Tasks
- generische Berechtigungen

Jede Berechtigung, die geändert werden soll, wird in einer Berechtigungsliste als Teil des Befehls angegeben. Jeder Eintrag in der Liste besteht aus einer Zeichenfolge mit einem Pluszeichen ('+') oder einem Minuszeichen ('-') als Präfix. Wenn Sie zum Beispiel den Eintrag '+put' in die Liste aufnehmen, erteilen Sie die Berechtigung zum Ausgeben von MQPUT-Aufrufen für eine Warteschlange. Wenn Sie dagegen den Eintrag '-put' in die Berechtigungsliste aufnehmen, entziehen Sie die Berechtigung zum Ausgeben von MQPUT-Aufrufen.

Berechtigungen können in jeder beliebigen Reihenfolge angegeben werden, solange sie sich nicht widersprechen. Wenn Sie zum Beispiel gleichzeitig allmqi und set angeben, führt dies zu einem Widerspruch.

Sie können in einem einzigen Befehl so viele Gruppen oder Berechtigungen angeben wie erforderlich.

Wenn eine Benutzer-ID Mitglied mehrerer Gruppen ist, entsprechen die gültigen Berechtigungen der Zusammenfassung der Berechtigungen aller Gruppen, denen die Benutzer-ID angehört.

Erforderliche Parameter

-g *Gruppenname*

Gibt den Namen der Benutzergruppe an, deren Berechtigungen geändert werden sollen. Sie können mehrere Gruppennamen angeben, aber jedem Namen muss die Option -g vorangestellt werden.

Sie müssen mindestens einen Principal oder eine Gruppe angeben.

-m *WS-Managername*

Gibt den Namen des WS-Managers des Objekts an, für das die Berechtigungen geändert werden sollen. Der Name darf maximal 48 Zeichen lang sein.

-p *Principal-Name*

Gibt den Namen des Principals an, für den die Berechtigungen geändert werden sollen. Sie können mehrere Principals angeben, aber jedem Namen muss die Option -p vorangestellt werden.

Sie müssen mindestens einen Principal oder eine Gruppe angeben.

-t *Objektart*

Gibt die Art des Objekts an, für das die Berechtigungen geändert werden sollen.

Gültige Werte:

- **q** oder **queue**
- **procs** oder **process**
- **qmgr**
- **nl** oder **namelist**

Optionale Parameter

-n *Objektname*

Gibt den Namen des Objekts an, für das die Berechtigungen geändert werden sollen.

Dieser Parameter ist erforderlich, *außer* wenn es sich bei dem Objekt um den WS-Manager selbst handelt. Sie müssen den Namen des WS-Managers, der Warteschlange oder der Prozessdefinition angeben, dürfen aber keinen generischen Namen verwenden.

-s Servicekomponente

Dieser Parameter gilt nur, wenn Sie installierbare Berechtigungsservices verwenden, andernfalls wird er ignoriert.

Wenn installierbare Berechtigungsservices unterstützt werden, gibt dieser Parameter den Namen des Berechtigungsservices an, für den die Berechtigungen gelten. Dieser Parameter ist optional; wenn er nicht angegeben wird, gilt die Aktualisierung für die erste installierbare Komponente für den Service.

Berechtigungen

Gibt die Berechtigungen an, die erteilt oder entzogen werden sollen. Vor jedem Eintrag in der Liste steht entweder ein Pluszeichen ('+'), d. h., die Berechtigung wird erteilt, oder ein Minuszeichen ('-'), d. h., die Berechtigung wird entzogen. Geben Sie in der Liste zum Beispiel '+put' ein, um die Berechtigung zum Ausgeben eines MQPUT-Aufrufs über die MQI zu erteilen. Geben Sie '-put' ein, um die Berechtigung zum Ausgeben eines MQPUT-Aufrufs zu entziehen.

Tabelle 10 zeigt die Berechtigungen, die für die unterschiedlichen Objektarten erteilt werden können.

Tabelle 10. Berechtigungen für verschiedene Objektarten erteilen

Berechtigung	Warteschlange	Prozess	WS-Manager	Namensliste
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
clr	✓			
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
put	✓			
inq	✓	✓	✓	✓
get	✓			
passall	✓			
passid	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

Berechtigungen für MQI-Aufrufe

- altusr** Verwenden einer alternativen Benutzer-ID in einer Nachricht.
Weitere Informationen zu alternativen Benutzer-IDs finden Sie im Handbuch *MQSeries Application Programming Guide*.
- Anzeigen (browse)**
Abrufen einer Nachricht aus einer Warteschlange durch Ausgeben eines MQGET-Aufrufs mit der Option BROWSE.
- connect**
Verbinden der Anwendung mit dem angegebenen WS-Manager durch Ausgeben eines MQCONN-Aufrufs.
- get** Abrufen einer Nachricht aus einer Warteschlange durch Ausgeben eines MQGET-Aufrufs.
- inq** Abfragen einer angegebenen Warteschlange durch Ausgeben eines MQINQ-Aufrufs.
- put** Einreihen einer Nachricht in eine angegebene Warteschlange durch Ausgeben eines MQPUT-Aufrufs.
- set** Setzen von Attributen für eine Warteschlange über die MQI durch Ausgeben eines MQSET-Aufrufs.

Anmerkung: Wenn Sie eine Warteschlange für mehrere Optionen öffnen, müssen Sie über die Berechtigung für jede einzelne verfügen.

Berechtigungen für Kontext

- passall**
Übergeben des gesamten Kontextes an die angegebene Warteschlange. Alle Kontextfelder werden aus der ursprünglichen Anforderung kopiert.
- passid** Übergeben des Identitätskontextes an die angegebene Warteschlange. Der Identitätskontext ist derselbe wie der in der Anforderung.
- setall** Setzen des gesamten Kontextes für die angegebene Warteschlange. Dies wird von bestimmten Systemdienstprogrammen verwendet.
- setid** Setzen des Identitätskontextes für die angegebene Warteschlange. Dies wird von bestimmten Systemdienstprogrammen verwendet.

Berechtigungen für Befehle

- chg** Ändern der Attribute des angegebenen Objekts.
- clr** Inhalt der angegebenen Warteschlange löschen (nur PCF-Befehl 'Clear queue').
- crt** Erstellen von Objekten der angegebenen Objektart.
- dlt** Löschen des angegebenen Objekts.
- dsp** Anzeigen der Attribute des angegebenen Objekts.

Berechtigungen für generische Operationen

- all** Verwenden aller auf das Objekt anwendbaren Operationen.
- alladm**
Ausführen aller auf das Objekt anwendbaren Verwaltungsoperationen.
- allmqi** Verwenden aller auf das Objekt anwendbaren MQI-Aufrufe.

Rückkehrcodes

0	Operation war erfolgreich
36	Ungültige Argumente übergeben
40	WS-Manager ist nicht verfügbar
49	WS-Manager wird gestoppt
69	Kein Speicher verfügbar
71	Unerwarteter Fehler
72	Falscher WS-Managername
133	Unbekannter Objektname
145	Unerwarteter Objektname
146	Objektname fehlt
147	Objektart fehlt
148	Ungültige Objektart
149	Name der Definitionseinheit fehlt
150	Berechtigungsspezifikation fehlt
151	Ungültige Berechtigungsspezifikation

Beispiele

1. Dieses Beispiel zeigt einen Befehl, in dem als Objekt, für das Berechtigungen erteilt werden, die Warteschlange `orange.queue` im WS-Manager `saturn.queue.manager` angegeben ist.

```
setmqaut -m saturn.queue.manager -n orange.queue -t queue -g tango +inq +alladm
```

Die Berechtigungen werden an die Benutzergruppe `tango` erteilt, wobei die zugehörige Berechtigungsliste die folgenden Festlegungen für die Benutzergruppe `tango` enthält:

- Sie erhält die Berechtigung zum Ausgeben von MQINQ-Aufrufen.
 - Sie erhält die Berechtigung zum Ausführen aller Verwaltungsoperationen für das Objekt.
2. In diesem Befehl enthält die Berechtigungsliste folgende Festlegungen für die Benutzergruppe `foxy`:
 - Sie erhält keine Berechtigung zum Ausgeben von Aufrufen über die MQI für die angegebene Warteschlange.
 - Sie erhält Berechtigung zum Ausführen aller Verwaltungsoperationen für die angegebene Warteschlange.

```
setmqaut -m saturn.queue.manager -n orange.queue -t queue -g foxy -allmqi +alladm
```

Zugehörige Befehle

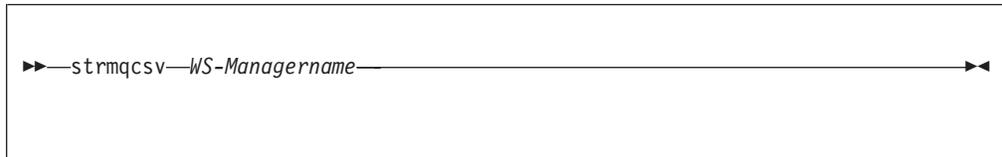
`dspmqaut` Berechtigung anzeigen

strmqcsv (Befehlsserver starten)

Funktion

Mit dem Befehl **strmqcsv** starten Sie den Befehlsserver für den angegebenen WS-Manager. Dadurch ist es MQSeries möglich, Befehle an die Befehlswarteschlange zu senden.

Syntax



Erforderliche Parameter

WS-Managername

Gibt den Namen des WS-Managers an, für den der Befehlsserver gestartet werden soll.

Rückkehrcodes

- 0 Befehl wurde normal beendet
- 10 Befehl wurde mit unerwarteten Ergebnissen beendet.
- 20 Bei der Verarbeitung ist ein Fehler aufgetreten.

Beispiele

Der folgende Befehl startet einen Befehlsserver für den WS-Manager Earth:

```
strmqcsv earth
```

Zugehörige Befehle

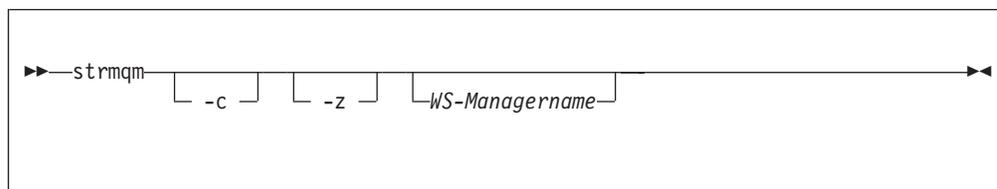
- endmqcsv** Befehlsserver beenden
- dspmqcsv** Status eines Befehlsservers anzeigen

strmqm (WS-Manager starten)

Funktion

Mit dem Befehl **strmqm** starten Sie einen lokalen WS-Manager. Der Befehl **strmqm** kehrt erst zur Befehlszeile zurück, nachdem der WS-Manager für Prozessverbindungen oder Anforderungen zur Verfügung steht.

Syntax



Optionale Parameter

-c Startet den WS-Manager, definiert die Standard- und Systemobjekte erneut und stoppt dann den WS-Manager. (Die Standard- und Systemobjekte für einen WS-Manager werden bei dessen Erstellung mit dem Befehl **crtmqm** erstellt.) Alle vorhandenen System- und Standardobjekte, die dem WS-Manager zugeordnet sind, werden ersetzt, wenn Sie diese Option angeben.

WS-Managername

Gibt den Namen eines lokalen WS-Managers an, der gestartet werden soll. Wird kein Name angegeben, wird der Standard-WS-Manager gestartet.

-z Diese Option unterdrückt Fehlernachrichten.

Sie wird in MQSeries verwendet, um unerwünschte Fehlernachrichten zu unterdrücken. Da bei Angabe dieser Option Informationen verloren gehen können, sollte sie nicht beim Eingeben von Befehlen in einer Befehlszeile verwendet werden.

Rückkehrcodes

- 0 WS-Manager gestartet
- 3 WS-Manager wird erstellt
- 5 WS-Manager aktiv
- 16 WS-Manager nicht vorhanden
- 49 WS-Manager wird gestoppt
- 69 Kein Speicher verfügbar
- 71 Unerwarteter Fehler
- 72 Falscher WS-Managername

Beispiele

Mit dem folgenden Befehl wird der WS-Manager account gestartet:

```
strmqm account
```

strmqm

Zugehörige Befehle

crtmqm	WS-Manager erstellen
dltmqm	WS-Manager löschen
endmqm	WS-Manager beenden

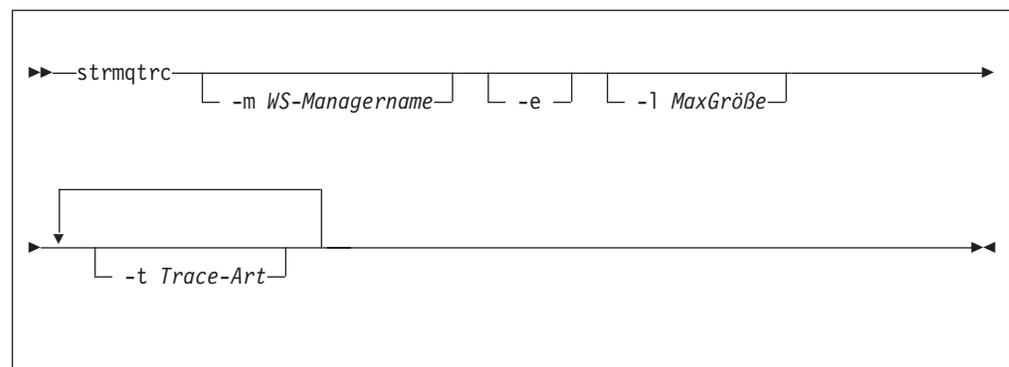
strmqtrc (MQSeries-Trace starten)

Funktion

Mit dem Befehl **strmqtrc** aktivieren Sie die Trace-Einrichtung. Dieser Befehl kann unabhängig davon ausgeführt werden, ob die Trace-Einrichtung aktiviert ist oder nicht. Wenn die Trace-Einrichtung bereits aktiviert ist, werden die aktuellen Trace-Optionen durch diejenigen überschrieben, die mit dem zuletzt aufgerufenen Befehl angegeben wurden.

Weitere Informationen zur Verwendung der Trace-Funktion in MQSeries finden Sie unter „MQSeries-Trace verwenden“ auf Seite 222.

Syntax



Optionale Parameter

-m *WS-Managername*

Gibt den Namen des WS-Managers an, für den der Trace durchgeführt werden soll. Wird kein Name angegeben, wird der Standard-WS-Manager verwendet.

Der angegebene WS-Manager muss nicht aktiv sein, er muss nicht einmal vorhanden sein. Daher ist es möglich, für die Erstellung oder den Start eines WS-Managers einen Trace durchzuführen.

Es ist möglich, einen WS-Manager und die Option -e in demselben Befehl anzugeben. Sind für eine Definitionseinheit, für die ein Trace durchgeführt wird, mehrere Trace-Optionen angegeben, umfasst der Trace alle angegebenen Trace-Optionen.

- e Wenn diese Option angegeben wird, wird ein Trace in der Startphase durchgeführt. Das heißt, es werden Trace-Informationen geschrieben, bevor die Prozesse wissen, welcher MQSeries-Komponente sie zugeordnet sind. Jeder Prozess, der einer Komponente eines WS-Managers zugeordnet ist, führt während seiner eigenen Startphase einen Trace durch, wenn diese Option angegeben wird. Ist sie nicht angegeben, gilt die Standardeinstellung, dass kein Trace in der Startphase durchgeführt wird.

-l *MaxGröße*

Der Wert *MaxGröße* gibt die maximale Größe einer Trace-Datei (AMQnnnn.TRC) in Millionen Bytes an. Wenn Sie zum Beispiel *MaxGröße* = 1 angeben, ist die Größe der Trace-Datei auf 1 Million Bytes begrenzt.

Wenn eine Trace-Datei die angegebene maximale Größe erreicht, wird sie von AQnnnn.TRC in AMQnnnn.TRS umbenannt und dann eine neue Datei AMQnnnn.TRC gestartet. Alle Trace-Dateien werden erneut gestartet, wenn die maximale Größe erreicht wird. Wenn eine ältere Version der Datei AMQnnnn.TRS vorhanden ist, wird sie gelöscht.

-t Trace-Art

Definiert, für welche Punkte während der Verarbeitung ein Trace durchgeführt wird. Eine oder mehrere der folgenden Optionen können angegeben werden:

all	Für jeden Trace-Punkt im System werden Daten ausgegeben. Dies ist die Standardeinstellung, wenn die Option -t nicht angegeben wird.
api	Für Trace-Punkte, die der MQI und Hauptkomponenten des WS-Managers zugeordnet sind, werden Daten ausgegeben.
comms	Für Trace-Punkte, die Datenflüssen in Kommunikationsnetzen zugeordnet sind, werden Daten ausgegeben.
csflows	Für Trace-Punkte, die Verarbeitungsabläufen in allgemeinen Services zugeordnet sind, werden Daten ausgegeben.
lqmfloWS	Für Trace-Punkte, die Verarbeitungsabläufen im lokalen WS-Manager zugeordnet sind, werden Daten ausgegeben.
remotefloWS	Für Trace-Punkte, die Verarbeitungsabläufen in der DFV-Komponente zugeordnet sind, werden Daten ausgegeben.
otherfloWS	Für Trace-Punkte, die Verarbeitungsabläufen in anderen Komponenten zugeordnet sind, werden Daten ausgegeben.
csdata	Für Trace-Punkte, die internen Datenpuffern in allgemeinen Services zugeordnet sind, werden Daten ausgegeben.
lqmdata	Für Trace-Punkte, die internen Datenpuffern im lokalen WS-Manager zugeordnet sind, werden Daten ausgegeben.
remotedata	Für Trace-Punkte, die internen Datenpuffern in der DFV-Komponente zugeordnet sind, werden Daten ausgegeben.
otherdata	Für Trace-Punkte, die internen Datenpuffern in anderen Komponenten zugeordnet sind, werden Daten ausgegeben.
versiondata	Für Trace-Punkte, die der aktiven Version von MQSeries zugeordnet sind, werden Daten ausgegeben.
commentary	Für Trace-Punkte, die Kommentaren in den MQSeries-Komponenten zugeordnet sind, werden Daten ausgegeben.

Wenn diese Option nicht angegeben wird, sind alle Trace-Punkte aktiviert, und ein vollständiger Trace wird durchgeführt.

Anmerkung: Wenn mehrere Trace-Arten angegeben werden, *muss* jede über eine eigene Option -t verfügen. Es können beliebig viele -t-Optionen angegeben werden, jedoch muss jeder eine gültige Trace-Art zugeordnet sein.

Es ist zulässig, für mehrere -t-Optionen dieselbe Trace-Art anzugeben.

Rückkehrcodes

AMQ7024

Diese Nachricht wird ausgegeben, wenn im Befehl ungültige Argumente angegeben wurden.

AMQ8304

Es ist bereits die maximale Anzahl von neun Traces aktiv.

Beispiele

Dieser Befehl aktiviert den Trace für Daten von allgemeinen Services und vom lokalen WS-Manager für den WS-Manager QM1.

```
strmqtrc -m QM1 -t csdata -t lqmdata
```

Zugehörige Befehle

dspmqtrc	Formatierte Trace-Ausgabe anzeigen
endmqtrc	MQSeries-Trace beenden

upgmqm (Upgrade eines WS-Managers der Version 2.2.0.1 durchführen)

Funktion

Mit diesem Befehl wird ein WS-Manager der Version 2.2.0.1 für den Einsatz in MQSeries for Compaq NSK V5.1 aufgerüstet. Das über den Befehl **upgmqm** aufgerufene Dienstprogramm sendet Statusnachrichten an das Terminal, an dem es aufgerufen wurde. Bei Abschluss des Dienstprogramms kann der WS-Manager in MQSeries for Compaq NSK V5.1 eingesetzt werden. Die neuen WS-Managerattribute in Version 5.1 werden auf ihre Standardwerte gesetzt. Sie können sie auf die übliche Weise ändern.

Da sich der neue Statusserver der Version 5.1 in seiner Funktionsweise vom MQSS-Server der Version 2.2.0.1 unterscheidet, werden bei dem Upgrade-Vorgang alle vorhandenen MQS-Status nm -Serverklassen aus der bestehenden PATHWAY-Konfiguration gelöscht und durch zwei Standardserverklassen (MQS-Status00 und MQS-Queue00) ersetzt. Daher müssen Sie nach dem Upgrade des WS-Managers alle den vorherigen MQSS-Serverprozessen zugeordneten Objekte je nachdem den neuen Statusserverprozessen oder Warteschlangenserverprozessen zuordnen. (In Version 5.1 werden den Warteschlangenservern nur lokale Warteschlangen zugeordnet; alle anderen Objekte werden dem Statusserver zugeordnet.) Darüber hinaus müssen Sie unter Umständen alle bereits vorhandenen Scripts ändern, in denen auf die PATHWAY-Statusserverklassen verwiesen wird.

Wenn Sie die Bereinigungsarbeiten nicht bereits während des Upgrades durchführen, können Sie die folgenden Dateien zu einem späteren Zeitpunkt löschen:

Tabelle 11. Dateien, die nach einem Upgrade gelöscht werden können - Beispiele

Pfad	Dateien	Beispiel
Teildatenträger wird über die Option -p im Befehl upgmqm angegeben	Alle Dateien	\$VOL.scratch
Teildatenträger 'S' des WS-Managers	Z*	\$VOL.myv2201S

Bei diesen Beispielen wird davon ausgegangen, dass Ihr WS-Manager sich in \$VOL.myv2201S befindet.

Anmerkung: Bei WS-Managern der Version 2.2.0.1, die nicht mit dem Befehl **upgmqm** aufgerüstet werden, haben die Steuerbefehle keine Wirkung. Dies gilt auch für **dltmqm**; dieser Befehl schlägt bei dem Versuch, die Principal-Datenbank zu öffnen, mit einer -Nachricht fehl. WS-Manager der Version 2.2.0.1, die in Version 5.1 nicht benötigt werden, müssen vor einem Code-Upgrade oder einem Upgrade über den Befehl **upgmqm** entfernt werden.

Syntax

```

▶—upgmqm—-m WS-Managername—-v StandardWSServer—-p TeildatenträgerPfad—▶
▶--s Standardstatusserver—▶

```

Erforderliche Parameter

- m *WS-Managername*
Gibt den Namen des WS-Managers an, für den das Dienstprogramm **upgmqm** ausgeführt werden soll.
- v *StandardWSServer*
Gibt einen eindeutigen Prozessnamen für den Standardwarteschlangenserver des WS-Managers an.
- p *TeildatenträgerPfad*
Gibt den Teildatenträgerpfad (\$VOL.SUBVOL) an, in dem das Upgrade-Dienstprogramm die Arbeitsdateien ablegen kann. Dieser Teildatenträger **muss** sich in demselben Datenträger wie der WS-Manager befinden. Geben Sie hier nur den Teildatenträger, nicht den Datenträger an. Der Befehl **upgmqm** prüft zuerst, ob der Teildatenträger nicht bereits vorhanden ist, bevor er ihn als gültige Angabe akzeptiert.
- s *Standardstatusserver*
Gibt einen eindeutigen Prozessnamen für den Standardstatusserver des WS-Managers an.

Beispiele

Mit dem folgenden Befehl wird ein WS-Manager der Version 2.2.0.1 (Myv2201qm) aufgerüstet; der Name des Standardwarteschlangenservers ist \$MYQS, der Name des Standardstatusservers ist \$MYSS, die Arbeitsdateien werden im Teildatenträger \$VOL.scratch abgelegt (dabei steht \$VOL für den Datenträger, in dem sich der WS-Manager befindet):

```
upgmqm -m Myv2201qm -p scratch -s $MYSS -v $MYQS
```

Teil 3. Anhänge und Schlussteil

Anhang A. MQSeries for Compaq NSK - Übersicht

Programm und Teilenummer

- 5724-A39 MQSeries for Compaq NSK Version 5 Release 1 Teilenummer 0791006
-

Hardwarevoraussetzungen

Mindest-Hardwarevoraussetzungen:

- Eine von Guardian D45 oder höher (D4x), G06 oder höher (G0x) unterstützte Compaq NSK-Maschine.
- Hardware, die die benutzerdefinierten Netzübertragungsprotokolle unterstützt.

Darüber hinaus wird empfohlen, eine oder mehrere gespiegelte Datenträger für Daten mit dem für die TMF-Prüfprotokolle und die MQSeries-Datenbank erforderlichen Speicherplatz zu verwenden.

Softwarevoraussetzungen

Mindest-Softwarevoraussetzungen:

- Compaq NSK Guardian-Betriebssysteme D45 oder höher (D4x; Hardware der K-Serie) und G06 oder höher (G0x; Hardware der S-Serie), einschließlich TM/MP (TMF), ENSCRIBE und EMS.
- TS/MP (PATHWAY) entsprechend dem Betriebssystem.
- SCF für die Konfiguration und Steuerung von TCP- und SNA-Netzübertragungen.

Für SNA-Konnektivität:

- SNAX/APC und SNAX/XF oder SNAX/APN entsprechend dem Betriebssystem.

ODER

- Insession ICE Version 3.2 oder höher.

Für TCP/IP-Konnektivität:

- TCP/IP entsprechend dem Betriebssystem.

Für die Verwendung der OSS-basierten Teile von MQSeries (MQI-Bindungen, OSS-Anwendungen, Java-Bindungen) muss die Version des OSS-Produkts mit dem Betriebssystem kompatibel sein.

Die Transaktionsprotokollfunktion wird durch Compaq TM/MP (TMF) zur Verfügung gestellt.

Sicherheit

MQSeries for Compaq NSK verwendet die Sicherheitseinrichtungen des NSK-Dateisystems, das es ermöglicht, Benutzern (USER) und Gruppen (GROUP) die Berechtigung für Lese-, Schreib-, Ausführungs- und Löschvorgänge pro Datei zu erteilen. SAFEGUARD ist für den Betrieb von MQSeries for Compaq NSK nicht erforderlich, das Produkt ist jedoch mit der SAFEGUARD-Umgebung kompatibel.

Alle MQSeries-Ressourcen sind einer einzigen Benutzer-ID in der Gruppe **MQM** zugeordnet. Für die Ausführung von MQSeries-Verwaltungsaufgaben über die SCOBOL-Menüs oder den Befehl **runmqsc** müssen Sie unter einer Benutzer-ID angemeldet sein, die entweder zur Gruppe **MQM** gehört oder mit ihr verknüpft ist.

Verwaltungsfunktionen

Die MQSeries-Verwaltung erfolgt über:

- Die Funktion MQM (Message Queue Management), wobei die SCOBOL-Anzeigen für Requester-Konfiguration in einer PATHWAY-Umgebung verwendet werden.
- Die Befehlszeilenschnittstelle **runmqsc**.
- Das Dienstprogramm SCF, das die Konfigurations- und Steuerungsfunktionen zur Verfügung stellt, die für die Verwaltung der TCP/IP- und SNA-Umgebungen für die Compaq-Netzprotokolloptionen erforderlich sind.
- ICE-Dienstprogramme (im Lieferumfang enthalten) für die Steuerung der ICE-LU6.2-Schnittstelle.
- MQSeries Explorer (gehört nicht zum Lieferumfang von MQSeries for Compaq NSK).
- Ein beliebiges Produkt oder Dienstprogramm, das PCF-Standardbefehle für die Fernverwaltung verwendet.

Kompatibilität

Die MQI-Schnittstelle für MQSeries for Compaq NSK V5.1 ist mit vorhandenen Anwendungen kompatibel, die unter MQSeries for Tandem NonStop Kernel V2.2.0.1 laufen, wenn die Programmkorrektur PTF U473441 verwendet wird.

Unterstützte Compiler

In MQSeries for Compaq NSK V5.1 wird CRE (Common Runtime Environment) für die Verknüpfung aller Objekte verwendet. Für die Benutzer von MQI-Versionen älter als Version 2.2.0.1 ergeben sich daraus die folgenden Einschränkungen:

1. COBOL- und C-Objektcode aus Versionen, die älter als D45 sind, müssen mit dem Compiler der Version D45 (oder höher) erneut kompiliert werden, um die CRE-Verknüpfungen zu integrieren.
2. TAL-Objektcode aus Versionen, die älter als D45 sind, müssen mit einem Compiler der Version D45 (oder höher) erneut kompiliert werden, damit das TAL-Programm den im Handbuch *Common Run-time Environment Programmer's Guide* angegebenen Programmiervorgaben entspricht. Weitere Informationen zu diesen Programmiervorgaben finden Sie im Handbuch *TAL Programmer's Guide*.
3. Für Objektcode, der mit nativen Compilern in D45 erstellt wurde, werden eigene Bindungen zur Verfügung gestellt.
4. C-Programme müssen das Wide Memory-Modell (32-Bit-Integer) verwenden.

5. COBOL-Programme müssen den CRE-Anforderungen entsprechen.
6. In TAL-Programmen müssen alle Integer als 32-Bit-Integer an die MQI-Funktionen übergeben werden (oder mit dem Makro \$INT32() in 32-Bit-Integer umgesetzt werden).

Die MQSeries-Programme selbst werden mit den nativen Tools für Guardian NSK kompiliert und verbunden. Native Anwendungen stellen in der Regel eine direkte Verbindung zur SRL (Shared Resource Library) des WS-Managers her, vorausgesetzt diese Anwendungen verwenden keine private SRL. Da Anwendungen höchstens eine private SRL verwenden können, muss die Anwendung in letzterem Fall entweder eine Verbindung zur statischen MQI-Bindungsbibliothek herstellen, oder es muss eine neue private SRL erstellt werden, die den Code aus der privaten SRL der Anwendung und die SRL von MQSeries enthält.

Lizenzverwaltung

Sie müssen den Systemtyp angeben, um das Optimierungsniveau für das Programm zu definieren. Dieser Parameter kann bei der Installation oder nachträglich bei einem Lizenz-Upgrade eingegeben werden. Beim Start wird dieser Wert mit der Konfiguration der physischen Compaq-Maschine verglichen. Sind Lizenzregistrierung und Optimierungsniveau nicht ausreichend, wird eine entsprechende Warnung ausgegeben.

Sprachenauswahl

Eine Nachrichtentextdatei, die im Lieferumfang enthalten ist, ist mit dem 7-Bit-Zeichensatz codiert, der auch vom Compaq NSK-Betriebssystem verwendet wird. Die Landessprache kann bei der Installation von MQSeries for Compaq NSK angegeben werden. Als Standardsprache ist für Nachrichten amerikanisches Englisch angegeben.

CCSID-Auswahl

MQSeries for Compaq NSK ermöglicht die Angabe der CCSID bei der Erstellung des WS-Managers; diese kann später geändert werden. Standardwert für die CCSID des WS-Managers ist 819. MQSeries for Compaq NSK unterstützt die Zeichensatzkonvertierung in die für den WS-Manager konfigurierte CCSID. Informationen darüber, welche CCSIDs (einschließlich der, die das Euro-Zeichen unterstützen) für WS-Manager in MQSeries for Compaq NSK angegeben werden können, finden Sie im Handbuch *MQSeries Application Programming Reference*.

Anhang B. Systemstandardwerte

Bei der Erstellung eines WS-Managers mit dem Steuerbefehl **crtmqm** werden die Systemobjekte und die Standardobjekte automatisch erstellt.

- Bei den Systemobjekten handelt es sich um die MQSeries-Objekte, die für den Betrieb eines WS-Managers bzw. Kanals erforderlich sind.
- Über die Standardobjekte sind alle Attribute eines Objekts definiert. Bei Erstellung eines Objekts wie beispielsweise eine lokale Warteschlange werden für alle Attribute, für die Sie keinen Wert angeben, die Werte aus dem Standardobjekt übernommen.

Tabelle 12. System- und Standardobjekte für Warteschlangen

Objektname	Beschreibung
SYSTEM.DEFAULT.ALIAS.QUEUE	Standardaliaswarteschlange
SYSTEM.DEFAULT.LOCAL.QUEUE	Lokale Standardwarteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standardmodellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Ferne Standardwarteschlange
SYSTEM.DEAD.LETTER.QUEUE	Beispielwarteschlange für nicht zustellbare Nachrichten
SYSTEM.DEFAULT.INITIATION.QUEUE	Standardinitialisierungwarteschlange
SYSTEM.CICS.INITIATION.QUEUE	Standardmäßige CICS®-Initialisierungwarteschlange
SYSTEM.ADMIN.COMMAND.QUEUE	Warteschlange für Verwaltungsbefehle. Wird für ferne MQSC- und PCF-Befehle verwendet.
SYSTEM.MQSC.REPLY.QUEUE	MQSC-Warteschlange für zu beantwortende Nachrichten. Hierbei handelt es sich um eine Modellwarteschlange, die eine temporäre dynamische Warteschlange für Antworten auf ferne MQSC-Befehle erstellt.
SYSTEM.ADMIN.QMGR.EVENT	Ereigniswarteschlange für WS-Managerereignisse
SYSTEM.ADMIN.PERFM.EVENT	Ereigniswarteschlange für leistungsspezifische Ereignisse
SYSTEM.ADMIN.CHANNEL.EVENT	Ereigniswarteschlange für Kanalereignisse
SYSTEM.CHANNEL.INITQ	Kanalstartwarteschlange
SYSTEM.CHANNEL.SYNCQ	Die Warteschlange, die die Synchronisationsdaten für Kanäle enthält. (Dieses Objekt wird erstellt, aber nicht in MQSeries for Compaq NSK verwendet. Die Synchronisationsdaten für Kanäle werden in TM/MP-geschützten Datenbanken gespeichert.)
SYSTEM.CLUSTER.COMMAND.QUEUE	Die Warteschlange, die für die Übertragung von Nachrichten an den Repository-WS-Manager verwendet wird.
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Die Warteschlange, in der alle Repository-Daten gespeichert werden.

Systemstandardwerte

Tabelle 12. System- und Standardobjekte für Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Die Übertragungswarteschlange für alle an Cluster gerichteten Nachrichten.

Tabelle 13. System- und Standardobjekte für Kanäle

Objektname	Beschreibung
SYSTEM.DEF.SENDER	Standardsenderkanal
SYSTEM.DEF.SERVER	Standardserverkanal
SYSTEM.DEF.RECEIVER	Standardempfängerkanal
SYSTEM.DEF.REQUESTER	Standard-Requester-Kanal
SYSTEM.DEF.SVRCONN	Standardkanal für Serververbindungen
SYSTEM.DEF.CLNTCONN	Standardkanal für Client-Verbindungen
SYSTEM.AUTO.RECEIVER	Dynamischer Empfängerkanal
SYSTEM.AUTO.SVRCONN	Dynamischer Kanal für Serververbindungen
SYSTEM.DEF.CLUSRCVR	Standardempfängerkanal für Cluster, über den Standardwerte für alle Attribute zur Verfügung gestellt werden, für die bei Erstellung eines CLUSRCVR-Kanals in einem WS-Manager innerhalb eines Clusters keine Werte angegeben wurden.
SYSTEM.DEF.CLUSSDR	Standardsenderkanal für Cluster, über den Standardwerte für all Attribute zur Verfügung gestellt werden, für die bei Erstellung eines CLUSSDR-Kanals in einem WS-Managers innerhalb eines Clusters keine Werte angegeben wurden.

Tabelle 14. System- und Standardobjekte für Namenslisten

Objektname	Beschreibung
SYSTEM.DEFAULT.NAMELIST	Standardnamensliste

Tabelle 15. System- und Standardobjekte für Prozesse

Objektname	Beschreibung
SYSTEM.DEFAULT.PROCESS	Standardprozessdefinition

Anhang C. TACL-Umgebungsvariablen für MQSeries for Compaq NSK setzen

MQSeries erstellt und verwendet eine Reihe von Compaq NSK-Umgebungsvariablen (PARAMs). Beim setzen dieser Umgebungsvariablen sollten Sie Folgendes beachten:

- Die Umgebungsvariable MQDEFAULTPREFIX muss in der Umgebung aller Programme vorhanden sein. Die TS/MP-Konfiguration (PATHWAY-Konfiguration), die automatisch mit dem Befehl **crtmqm** eingerichtet wird, stellt sicher, dass in diesen Umgebungsvariablen für alle WS-Managerserverprozesse die richtigen Werte gesetzt werden. Die Benutzer der MQSeries-Anwendungen und der MQSeries-Steuerbefehle müssen sicherstellen, dass diese Variablen auch in den TACL-Umgebungen und TS/MP-Konfigurationen, die verwendet werden, angegeben werden.
- Es wird geraten, die Anweisungen für die Variablen in den TACLSTM-Dateien so einzufügen, dass diese Variablen bei der Anmeldung richtig erstellt werden und alle Programme, die in der TACL-Umgebung ausgeführt werden, die korrekten Werte übernehmen. Die folgende Umgebungsvariable sollte außerdem geändert werden, damit die ausführbaren MQSeries-Module ermittelt werden können:

```
#SET #PMSEARCH $SYSTEM.ZMQSEX [ #PMSEARCH ]
```

MQCONNECTTYPE

Über diese Umgebungsvariable kann (falls vorhanden) für Anwendungen die Option, FASTPATH-Verbindungen zu verwenden, inaktiviert werden. Wird für diese Umgebungsvariable der Wert STANDARD angegeben, können Anwendungen nur STANDARD-Verbindungen verwenden, auch wenn sie FASTPATH-Verbindungen anfordern. Die Angabe eines anderen Wertes als FASTPATH oder STANDARD hat dieselbe Wirkung wie überhaupt keine Angabe für diese Variable, d. h., Anwendungen können FASTPATH-Verbindungen anfordern und verwenden.

MQDEFAULTPREFIX

Gibt den Namen des Datenträgers an, der den installierten Teildatenträger ZMQSSYS enthält. Diese Umgebungsvariable muss in allen Umgebungen richtig definiert werden.

Beispiel:

```
PARAM MQDEFAULTPREFIX $data00
```

TACL-Umgebungsvariablen

MQEMSEVENTS

Über diese Umgebungsvariable werden MQSeries-EMS-Ereignisse aktiviert. Um beispielsweise alle EMS-Ereignisse für MQSeries zu aktivieren, müssen Sie die Umgebungsvariable MQEMSEVENTS wie folgt definieren):

```
PARAM MQEMSEVENTS 127
```

MQMACHINIFILE

Gibt an, wo sich die Datei MQSINI für die Installation befindet. Der Standardwert ist *MQDEFAULTPREFIX.ZMQSSYS.MQSINI*. Diese Umgebungsvariable muss nur gesetzt werden, wenn ein anderes als das Standardverzeichnis verwendet werden soll.

Beispiel:

```
PARAM MQMACHINIFILE $data00.altinst.mqsini
```

MQRDF

Wird diese Umgebungsvariable auf ON gesetzt, ändert MQSeries das Verhalten bei Löschvorgängen dahingehend, dass RDF sich auch auf protokollierte Dateien auswirkt. Wird diese Umgebungsvariable nicht definiert oder auf einen anderen Wert als ON gesetzt, arbeitet die MQSeries-Löschfunktion normal. Falls diese Umgebungsvariable verwendet wird, muss sie in der TACL-Umgebung aller Benutzer, die Verwaltungsprogramme ausführen, und in der Umgebung aller im WS-Manager konfigurierten TS/MP-Serverklassen gesetzt werden.

MQRDFFUPPROCESSNAME

Diese Umgebungsvariable wird nur vom Dienstprogramm **cleanrdf** ausgewertet. Sie gibt einen Guardian-Prozessnamen an, der dem vom Dienstprogramm **cleanrdf** erstellten FUP-Serverprozess zugeordnet wird. Wird diese Umgebungsvariable nicht definiert, wird der FUP-Serverprozessnamen vom Betriebssystem vergeben.

MQRDFFUPPROGNAME

Diese Umgebungsvariable wird nur vom Dienstprogramm **cleanrdf** ausgewertet. Sie gibt den vollständigen Namen der ausführbaren FUP-Datei an, die vom Dienstprogramm verwendet wird. Der Standardwert ist `<default system name>.$SYSTEM.SYS00.FUP`.

MQSNOAUT

Wird diese Umgebungsvariable bei der Ausführung von **crtmqm** (WS-Manager erstellen) auf 1 gesetzt, wird der neue WS-Manager mit inaktiviertem Objektberechtigungsmanager (OAM) erstellt.

Beispiel:

```
PARAM MQSNOAUT 1
```

MQLISTENPORTNUM

Wird diese Umgebungsvariable gesetzt, ermittelt das TCP/IP-Empfangsprogramm anhand ihres Wertes, an welchem Port er empfangsbereit sein soll. Wird kein Wert angegeben, wird der Port anhand der Zeilengruppe mit den TCP/IP-Informationen in der Datei QMINI ermittelt.

SAVE-ENVIRONMENT ON

Diese Umgebungsvariable ist bei Ausführung von Anwendungsprogrammen erforderlich, um sicherzustellen, dass CRE (Common Runtime Environment) die Umgebungsvariablen von der Umgebung an das Anwendungsprogramm übergibt.

Beispiel:

```
PARAM SAVE-ENVIRONMENT ON
```

Wird diese Umgebungsvariable nicht gesetzt, wird der Rückkehrcode 2058 zurückgegeben, der auf einen fehlerhaften WS-Managernamen hinweist.

Diese Umgebungsvariable ist nur für TAL-oder COBOL-Anwendungen erforderlich, die nicht als native Programme laufen.

Optimierungsparameter für Warteschlangenserver

Wenn die folgenden Umgebungsvariablen in der TACL-Umgebung eines Warteschlangenservers gesetzt werden, überschreiben sie bei verschiedenen Systemverwaltungsvorgängen die integrierten Standardwerte des Warteschlangenservers:

MQQSHKEEPINT

Wird diese Umgebungsvariable gesetzt, kann ein numerischer Wert in Sekunden angegeben werden, der das für den Warteschlangenserver standardmäßig auf 60 Sekunden angesetzte Intervall für die Systemverwaltung überschreibt. Dieses Intervall gibt an, wie oft der Warteschlangenserver die Warteschlangen auf abgelaufene Nachrichten und die Speicherauslastung überprüft, um einen optimalen Betrieb zu gewährleisten.

MQQSSIGTIMEOUT

Wird diese Umgebungsvariable gesetzt, kann ein numerischer Wert in Sekunden angegeben werden, der das Standardzeitlimit (60 Sekunden) für die Zustellung eines IPC-Signals an eine Anwendung, die einen MQGET-Aufruf unter Angabe der Option MQGMO_SET_SIGNAL ausgegeben hat, überschreibt. Kann ein Server das Signal nicht innerhalb der angegebenen Zeit zustellen (nachdem die Bedingungen für die Generierung dieses Signals erfüllt sind), wird dies vom Warteschlangenserver protokolliert, der das Signal anschließend abbricht.

TACL-Umgebungsvariablen

MQQSMAXBATCHEXPIRE

Wird diese Umgebungsvariable gesetzt, kann ein numerischer Wert angegeben werden, der die standardmäßig maximal zulässige Anzahl an permanenten Nachrichten (100), die von einem Warteschlangenserver innerhalb einer einzelnen Transaktion bei Bereinigungsvorgängen gelöscht werden können, überschreibt. Wenn eine permanente Nachricht abläuft, muss sie aus den Warteschlangendatenbanken entfernt werden; dafür ist eine interne TM/MP-Transaktion erforderlich. Mit dieser Umgebungsvariable kann die maximal Anzahl an Nachrichten festgelegt werden, die innerhalb einer TM/MP-Transaktion entfernt werden können.

MQQSMAXMSGSEXPIRE

Wird diese Umgebungsvariable gesetzt, kann ein numerischer Wert angegeben werden, der die standardmäßig maximal zulässige Anzahl an Nachrichten (300), die innerhalb einer Systemverwaltungssitzung vom Warteschlangenserver gelöscht werden können, überschreibt.

Anhang D. Befehlssätze - Übersicht

Tabelle 16 bis Tabelle 21 auf Seite 336 enthalten eine Gegenüberstellung der Funktionen der verschiedenen verwaltungsspezifischen Befehlssätze.

Anmerkung: Es sind nur die MQSC-Befehle aufgeführt, die für MQSeries for Compaq NSK gelten.

Tabelle 16. Befehle für die Verwaltung des WS-Managers

PCF	MQSC-Befehl	Steuerbefehl
Change Queue Manager	ALTER QMGR	–
(Create queue manager)*	–	crtmqm
(Delete queue manager)*	–	dltmqm
Inquire Queue Manager	DISPLAY QMGR	–
(Stop queue manager)*	–	endmqm
Ping Queue Manager	PING QMGR	–
(Start queue manager)*	–	strmqm
Anmerkung: * Nicht als PCF-Befehle verfügbar.		

Tabelle 17. Befehle für die Verwaltung des Befehlsservers

Beschreibung	Steuerbefehl
Befehlsserver anzeigen	dspmqcsv
Befehlsserver stoppen	endmqcsv
Befehlsserver starten	strmqcsv
Anmerkung: Anstelle der Steuerbefehle können Sie auch PATHCOM-Befehle verwenden (siehe „TS/MP-Verwaltung (PATHWAY-Verwaltung)“ auf Seite 34). Für die Befehle in dieser Tabelle gibt es keine entsprechenden MQSC- oder PCF-Befehle.	

Befehlssätze - Übersicht

Tabelle 18. Befehle für die Warteschlangenverwaltung

PCF	MQSC-Befehl
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
Anmerkung: Für diese Funktionen gibt es keine Steuerbefehle.	

Tabelle 19. Befehle für die Prozessverwaltung

PCF	MQSC-Befehl
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
Anmerkung: Für diese Funktionen gibt es keine Steuerbefehle.	

Tabelle 20. Befehle für die Kanalverwaltung

PCF	MQSC-Befehl	Steuerbefehl
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	–	runmqlsr
Stop Channel	STOP CHANNEL	–
<p>Anmerkung: Starten Sie in MQSeries for Compaq NSK die TCP/IP-Kanalempfangsprogramme mit TS/MP oder mit dem Steuerbefehl runmqlsr. Weitere Informationen finden Sie unter „TCP/IP-Empfangsprogramme konfigurieren und steuern“ auf Seite 34 und „runmqlsr (Empfangsprogramm ausführen)“ auf Seite 300.</p>		

Befehlssätze - Übersicht

Table 21. Weitere Steuerbefehle

Beschreibung	Steuerbefehl
Datenträger für Warteschlange ändern, Optionen des Warteschlangenservers für das Speichern von Nachrichten	altnqfls
MQSeries-Principals hinzufügen, löschen oder ändern	altnqusr
RDF-Dienstprogramm für die Systemverwaltung	cleanrdf
Tabelle mit Client-Kanaldefinitionen konvertieren	cvclchl
MQSeries-Konvertierungs-Exit erstellen	crtmqcvx
Berechtigung anzeigen	dspmqaut
Von Objekten verwendete Dateien anzeigen; die Optionen des Warteschlangenservers für das Speichern von Nachrichten, die für ein Objekt konfiguriert wurden	dspmqfls
Formatierte MQSeries-Trace-Ausgabe anzeigen	dspmqtrc
MQSeries-Principals anzeigen	dspmqusr
MQSeries-Trace beenden	endmqtrc
MQSeries for Compaq NSK installieren	instmqm
Steuerroutine der Warteschlange für nicht zustellbare Nachrichten ausführen	runmqdlq
MQSC-Befehle ausführen	runmqsc
Auslösemonitor ausführen	runmqtrm
Berechtigung setzen oder zurücksetzen	setmqaut
MQSeries-Trace starten	strmqtrc
WS-Manager der Version 2.2.0.1 aufrüsten	upgmqm
<p>Anmerkung: Anstelle des Steuerbefehls runmqtrm können Sie auch PATHCOM-Befehle verwenden (siehe „TS/MP-Verwaltung (PATHWAY-Verwaltung)“ auf Seite 34). Für die Befehle in dieser Tabelle gibt es keine entsprechenden MQSC- oder PCF-Befehle.</p>	

Anhang E. WS-Manager manuell stoppen und entfernen

Falls es nicht möglich sein sollte, WS-Manager auf normalem Wege zu stoppen und zu beenden, können Sie die etwas drastischeren Verfahren anwenden, die hier beschrieben werden.

WS-Manager manuell stoppen

Normalerweise sollte es möglich sein, einen WS-Manager selbst bei Auftreten von internen Fehlern mit dem Befehl **endmqm** zu beenden. Sollte jedoch der besondere Fall eintreten, dass dieses Verfahren versagt, können Sie den WS-Manager wie folgt manuell stoppen:

1. Ermitteln Sie die Prozess-IDs des WS-Managers, die noch aktiv sind.
2. Bei Ausführung von FUP LISTOPENS für die Datei TRACEOPT im Teildatenträger mit den Daten des WS-Managers werden CPU und PIN der Prozesse zurückgegeben, die dem WS-Manager zugeordnet sind.
3. Beenden Sie die noch aktiven WS-Managerprozesse. Verwenden Sie hierzu den Befehl **STOP** unter Angabe der unter Schritt 2 ermittelten Prozess-IDs.

Die Prozesse sollten in der folgenden Reihenfolge beendet werden:

- a. MQECBOSS – ECBoss
- b. MQEC – Ausführungs-Controller
- c. Alle übrigen Prozesse, die noch aktiv sind

Anmerkung: Bei der manuellen Beendigung eines WS-Managers werden unter Umständen FFST-Berichte und FD-Dateien erstellt. Dies sollte nicht als Fehler im WS-Manager angesehen werden.

Der WS-Manager sollte auch bei Anwendung des gerade beschriebenen Verfahrens anschließend wieder normal hochfahren.

Soll der WS-Manager nach der manuellen Beendigung gelöscht werden, geben Sie den Befehl **dltmqm** ein. Sollte es aus irgendeinem Grund nicht möglich sein, den WS-Manager mit diesem Befehl zu löschen, können Sie ihn wie unter „WS-Manager manuell entfernen“ beschrieben manuell löschen.

WS-Manager manuell entfernen

So entfernen Sie einen WS-Manager manuell:

1. Stellen Sie sicher, dass für den WS-Manager, der entfernt werden sollte, keine Prozesse mehr aktiv sind.
2. Ändern Sie die Datei MQSINI, indem Sie die Zeilengruppe für den WS-Manager entfernen und gegebenenfalls die Zeilengruppe für den Standard-WS-Manager ändern. Notieren Sie sich das Verzeichnis mit den Dateien des WS-Managerdateien, bevor Sie die Zeilengruppe löschen.
3. Löschen Sie mit dem Befehl FUP PURGE alle Dateien aus allen Teildatenträgern des WS-Managers. Beispiel: FUP PURGE \$VOL.QMSVOL*.*.

WS-Manager manuell entfernen

Anhang F. MQSeries und Compaq NonStop Server for Java

MQSeries for Compaq NSK ist mit Compaq NonStop Server for Java Version 1.5 und höher kompatibel. Das Produkt unterstützt alle MQSeries-Java-Klassen. Eine ausführliche Beschreibung dieser Klassen finden Sie im Handbuch *MQSeries Using Java*.

Es gibt zwei Möglichkeiten für den Einsatz von MQSeries mit dem NonStop Server for Java:

- MQSeries kann als Servlet im Kontext von Compaq iTP WebServer ausgeführt werden
- MQSeries kann als Java-Anwendung direkt in der Befehlszeile aufgerufen werden

Informationen zur Verwendung von Servlets finden Sie im Handbuch *Compaq NonStop Server for Java 1.5 Reference*.

Für den Zugriff auf MQSeries von Java (über Servlets oder Anwendungen) muss eine benutzerangepasste Version der NonStop-JVM erstellt werden, die mit den folgenden MQSeries-Produktbibliotheken verbunden ist:

- der Produkt-SRL MQSRLLIB vom Guardian-Teildatenträger ZMQSEXE
- dem Java-Bindungsarchiv libMQSESSION.a von /opt/mqm/lib

In der Compaq-Java-Dokumentation wird beschrieben, wie eine erneute Verbindung zu JVM hergestellt wird. Das MQSeries-Verzeichnis mit Beispielen (/opt/mqm/samp) enthält zwei Make-Beispieldateien

MakeJVM.smp

Diese Beispieldatei veranschaulicht, wie die NonStop-JVM wiederhergestellt werden muss, damit ein Zugriff auf MQSeries möglich ist. Sie können diese Make-Datei verwenden, indem Sie in ihr das Verzeichnis angeben, das Ihre MQSeries-Produktbibliotheken enthält.

MakeJava.smp

Mit dieser Beispieldatei kann eine Java-Anwendung erstellt werden.

Transaktionale Überlegungen

Java bietet keinen direkten Zugriff auf die TM/MP-Schnittstelle. MQSeries unterstützt Transaktionen in Java über den Java Transaction Service (JTS) *Current Class*. In der folgenden Tabelle wird die *Current*-Schnittstelle beschrieben

Tabelle 22. Java-Schnittstelle

Funktion	Beschreibung
begin()	Startet eine neue Transaktion und ordnet Sie dem Thread zu, von dem sie aufgerufen wurde.
commit(boolean)	Schreibt die dem aufrufenden Thread zugeordnete Transaktion fest.
get_control()	Gibt ein Objekt der Art Control zurück, das die Transaktion darstellt, die dem aufrufenden Thread zugeordnet ist.
get_status()	Gibt den Status der dem aufrufenden Thread zugeordneten Transaktion zurück.
get_transaction_name()	Gibt den (beschreibenden) Namen der Transaktion zurück, die dem aufrufenden Thread zugeordnet ist.
resume(ControlRef)	Stellt die Zuordnung einer Transaktion mit dem aufrufenden Thread her bzw. wieder her.
rollback()	Setzt die dem aufrufenden Thread zugeordnete Transaktion zurück.
suspend()	Setzt die Zuordnung zwischen aufrufendem Thread und einem Transaktionskontext vorübergehend aus.

Weitere Informationen zu JTS finden Sie in der Dokumentation für Compaq NSK NonStop Java oder für JDK.

Anhang G. In MQSeries for Compaq NSK unterstützte MQSC-Befehle

In diesem Anhang sind die in MQSeries for Compaq NSK unterstützten Befehle aufgeführt. Informationen zur Syntax dieser Befehle finden Sie im Handbuch *MQSeries MQSC - Befehle*.

Tabelle 23. In MQSeries for Compaq NSK unterstützte MQSC-Befehle

Befehl	Beschreibung	Wie in MQSeries MQSC - Befehle beschrieben
ALTER CHANNEL	Ändert Kanalattribute.	Ja, jedoch mit den unter „CONNNAME“ auf Seite 344 aufgeführten Ausnahmen.
ALTER NAMELIST	Ändert eine Namensliste.	Ja
ALTER PROCESS	Ändert Prozessattribute.	Ja
ALTER QALIAS	Ändert die Attribute einer Aliaswarteschlange.	Ja
ALTER QLOCAL	Ändert die Attribute einer lokalen Warteschlange.	Ja, jedoch mit den unter „HARDENBO und NOHARDENBO“ auf Seite 344 aufgeführten Ausnahmen.
ALTER QMGR	Ändert die Attribute des WS-Managers.	Ja, jedoch mit den unter „MAXUMSGS und MAXHANDS“ auf Seite 344 aufgeführten Ausnahmen.
ALTER QMODEL	Ändert die Attribute einer Modellwarteschlange.	Ja, jedoch mit den unter „HARDENBO und NOHARDENBO“ auf Seite 344 aufgeführten Ausnahmen.
ALTER QREMOTE	Ändert die Attribute einer lokalen Definition eines fernen WS-Managers, eines WS-Manageraliasnamens oder einer Aliaswarteschlange für zu beantwortende Nachrichten.	Ja
CLEAR QLOCAL	Löscht Nachrichten aus einer lokalen Warteschlange.	Ja
DEFINE CHANNEL	Erstellt eine Kanaldefinition.	Ja, jedoch mit den unter „CONNNAME“ auf Seite 344 aufgeführten Ausnahmen.
DEFINE NAMELIST	Definiert eine Namensliste.	Ja
DEFINE PROCESS	Erstellt eine Prozessdefinition.	Ja
DEFINE QALIAS	Erstellt die Definition einer Aliaswarteschlange.	Ja

MQSeries-Befehle

Table 23. In MQSeries for Compaq NSK unterstützte MQSC-Befehle (Forts.)

Befehl	Beschreibung	Wie in MQSeries MQSC - Befehle beschrieben
DEFINE QLOCAL	Erstellt die Definition einer lokalen Warteschlange.	Ja, jedoch mit den unter „HARDENBO und NOHARDENBO“ auf Seite 344 aufgeführten Ausnahmen.
DEFINE QMODEL	Erstellt die Definition einer Modellwarteschlange.	Ja, jedoch mit den unter „HARDENBO und NOHARDENBO“ auf Seite 344 aufgeführten Ausnahmen.
DEFINE QREMOTE	Erstellt eine lokale Definition eines fernen WS-Managers, einen WS-Manageraliasnamens oder eine Aliaswarteschlange für zu beantwortende Nachrichten.	Ja
DELETE CHANNEL	Löscht eine Kanaldefinition.	Ja
DELETE NAMELIST	Löscht eine Namensliste.	Ja
DELETE PROCESS	Löscht eine Prozessdefinition.	Ja
DELETE QALIAS	Löscht die Definition einer Aliaswarteschlange.	Ja
DELETE QLOCAL	Löscht die Definition einer lokalen Warteschlange.	Ja
DELETE QMODEL	Löscht die Definition einer Modellwarteschlange.	Ja
DELETE QREMOTE	Löscht die lokale Definition einer fernen Warteschlange.	Ja
DISPLAY CHANNEL	Zeigt eine Kanaldefinition an.	Ja, jedoch mit den unter „CONNNAME“ auf Seite 344 aufgeführten Ausnahmen.
DISPLAY CHSTATUS	Zeigt den Status eines oder mehrerer Kanäle an.	Ja, jedoch mit den unter „Kanalstatusinformationen (DISPLAY CHSTATUS)“ auf Seite 344 und „CONNNAME“ auf Seite 344 aufgeführten Ausnahmen.
DISPLAY CLUSQMGR	Zeigt den Status eines oder mehrerer Kanäle an.	Ja, jedoch mit den unter „CONNNAME“ auf Seite 344 aufgeführten Ausnahmen.
DISPLAY NAMELIST	Zeigt eine Namensliste an.	Ja
DISPLAY PROCESS	Zeigt eine Prozessdefinition an.	Ja
DISPLAY QMGR	Zeigt die Attribute eines WS-Managers an.	Ja, jedoch mit den unter „MAXUMSGS und MAXHANDS“ auf Seite 344 aufgeführten Ausnahmen.

Tabelle 23. In MQSeries for Compaq NSK unterstützte MQSC-Befehle (Forts.)

Befehl	Beschreibung	Wie in MQSeries MQSC - Befehle beschrieben
DISPLAY QUEUE	Zeigt die Attribute einer Warteschlange an.	Ja, jedoch mit den unter „HARDENBO und NOHARDENBO“ auf Seite 344 aufgeführten Ausnahmen.
PING CHANNEL	Wird zum Test eines Kanals verwendet.	Ja
PING QMGR	Überprüft, ob der WS-Manager auf Befehle reagiert.	Ja
REFRESH CLUSTER	Löscht alle lokal gespeicherten Cluster-Daten und erzwingt deren Wiederherstellung.	Ja
RESET CHANNEL	Setzt die Nachrichtenfolge-nummer für einen Kanal zurück.	Ja
RESET CLUSTER	Führt bestimmte Operationen für Cluster aus.	Ja
RESOLVE CHANNEL	Löst unbestätigte Nachrichten in Sender- oder Serverkanälen auf.	Ja
RESUME QMGR	Meldet den anderen WS-Managern im Cluster, dass der lokale WS-Manager wieder für die Verarbeitung zur Verfügung steht und Nachrichten an ihn geschickt werden können.	Ja
START CHANNEL	Startet einen Kanal.	Ja
STOP CHANNEL	Stoppt einen Kanal.	Ja
SUSPEND QMGR	Meldet den anderen WS-Managern im Cluster, dass der lokale WS-Manager nicht zur Verfügung steht und daher keine Nachrichten an ihn geschickt werden können.	Ja

Falls Sie ein Script mit MQSC-Befehlen erstellen, dürfen die einzelnen Zeilen nicht mehr als 72 Zeichen enthalten.

MQSC-Attribute

Dieser Abschnitt enthält Informationen zu MQSC-Attributen, die speziell für MQSeries for Compaq NSK gelten.

Kanalstatusinformationen (DISPLAY CHSTATUS)

Der Befehl DISPLAY CHSTATUS ist wie im Handbuch *MQSeries MQSC - Befehle* beschrieben implementiert, allerdings wird der Kanalstatus nur am Beginn bzw. am Ende der Stapelverarbeitung aktualisiert. Die Kanalstatusinformationen werden nicht bei jeder Nachrichtenübertragung aktualisiert, da sich dies unter Umständen nachteilig auf den Kanalbetrieb auswirken könnte. Dies bedeutet, dass die allgemeinen Statusdaten in aktuellen und gespeicherten Sätze identisch sind.

MAXUMSGS und MAXHANDS

Die WS-Managerobjektattribute MAXUMSGS und MAXHANDS werden ignoriert. Dies betrifft die folgenden Befehle:

```
ALTER QMGR  
DISPLAY QMGR
```

HARDENBO und NOHARDENBO

In MQSeries for Compaq NSK werden die Attribute HARDENBO und NOHARDENBO für lokale Warteschlangen und Modellwarteschlangen ignoriert. Der Wert für *Backoutcount* (Anzahl der Rücksetzungsvorgänge) wird für permanente Nachrichten immer, für nicht permanente Nachrichten nie gespeichert. Dies betrifft die folgenden Befehle:

```
ALTER QLOCAL  
ALTER QMODEL  
DEFINE QLOCAL  
DEFINE QMODEL  
DISPLAY QUEUE
```

CONNAME

Das Attribut CONNAME für TCP-Kanäle kann optional ein zusätzliches Feld am Anfang haben, der den Namen eines bestimmten Guardian-TCP/IP-Serverprozesses angibt, der für den Kanal verwendet werden soll. Dies betrifft die folgenden Befehle:

```
ALTER CHANNEL  
DEFINE CHANNEL  
DISPLAY CHANNEL  
DISPLAY CHSTATUS  
DISPLAY CLUSQMGR
```

USERDATA für ausgelöste Programme

Wenn über das Attribut USERDATA im MQSC-Befehl DEFINE PROCESS oder ALTER PROCESS an den Auslösemonitor Daten übergeben werden, bei denen es sich um eine Zeichenfolge, die Leerzeichen enthält, handelt, müssen die Daten in doppelten Anführungszeichen stehen. Soll beispielsweise über das Attribut USERDATA die Zeichenfolge -o \$DISK.VOLUME.PROGRAM -d 1 an den Auslösemonitor übergeben werden, muss sie im MQSC-Befehl wie folgt in doppelte Anführungszeichen gesetzt werden:

```
'" -o $DISK.VOLUME.PROGRAM -d 1"'
```

Wird die Prozessdefinition über einen MQSC-Befehl angezeigt, hat sie folgendes Format:

```
"-o $DISK.VOLUME.PROGRAM -d 1"
```

Namen von Benutzer-Exits als Objektattribute verwenden

Werden Exit-Namen in Objektattributen angegeben, erhalten sie das für MQSeries for Compaq NSK spezifische Format.

Anhang H. Anwendungsprogrammierung - Referenz

Die folgenden Abschnitte enthalten Informationen zu Neuerungen in MQSeries for Compaq NSK; diese sollten zusammen mit dem Handbuch *MQSeries Application Programming Reference* verwendet werden.

Strukturdatentypen

In diesem Abschnitt werden die Änderungen an Datentypen beschrieben.

Strukturdatentyp	Unterstützt in V2.x	Unterstützt in V5.1	Funktionsweise wie in <i>MQSeries Application Programming Reference</i> beschrieben
MQBO (Begin Options)	Nein	Nein	
MQCH (CICS Bridge Header)	Nein	Ja	Ja
MQCNO (Connect Options)	Nein	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQCNO (Connect Options)“ auf Seite 348.
MQDH (Distribution Header)	Nein	Ja	Ja
MQDLH (Dead Letter Header)	Ja	Ja	Ja
MQGMO (Get Message Options)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQGMO (Get Message Options)“ auf Seite 349.
MQIH (IMS Bridge Header)	Ja	Ja	Ja
MQMD (Message Descriptor)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQMD (Message Descriptor)“ auf Seite 350.
MQMDE (Message Descriptor Extension)	Nein	Ja	Ja
MQOD (Object Descriptor)	Ja	Ja	Ja
MQOR (Object Record)	Nein	Ja	Ja
MQPMO (Put Message Options)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQPMO (Put Message Options)“ auf Seite 350.

Strukturdatentypen

Strukturdatentyp	Unterstützt in V2.x	Unterstützt in V5.1	Funktionsweise wie in <i>MQSeries Application Programming Reference</i> beschrieben
MQPMR (Put Message Record)	Nein	Ja	Ja
MQRMH (Message Reference Header)	Nein	Ja	Ja
MQRR (Response Record)	Nein	Ja	Ja
MQTM (Trigger Message)	Ja	Ja	Ja
MQTMC2 (Trigger Message Character Format)	Ja	Ja	Ja
MQWIH (Workload Information Header)	Nein	Ja	Ja
MQXQH (Transmission Queue Header)	Ja	Ja	Ja

In diesem Abschnitt werden die folgenden MQSeries-Strukturdatentypen beschrieben:

MQCNO (Connect Options)

Für die Datenstruktur MQCNO gelten die Angaben im Handbuch *MQSeries Application Programming Reference*, mit den folgenden Zusätzen:

- Die einzelnen Ausführungen werden als Prozess bezeichnet.
- MQCNO_FASTPATH_BINDING kann nur in einem Prozess verwendet werden, für den eine einzige Verbindung zu einem WS-Manager besteht.
- Bei MQCNO_FASTPATH_BINDING muss die Anwendung unter der Benutzer-ID ausgeführt werden, die zur Administratorbenutzergruppe **MQM** gehört, von der der WS-Manager erstellt wurde.
- Der Guardian-Parameter MQCONNECTTYPE kann in Verbindung mit dem im Feld **Options** angegebenen Bindungstyp verwendet werden, um so die Bindung anzugeben, die verwendet werden soll. Wenn dieser Parameter verwendet wird, muss entweder **FASTPATH** oder **STANDARD** angegeben werden; andere Werte werden ignoriert. Bei der Angabe muss die Groß-/Kleinschreibung beachtet werden.
- MQSeries for Compaq NSK unterstützt MCNO_VERSION_2 und MQCNO_VERSION_1, allerdings werden die Felder *ClientConnOffset* und *ClientConnPtr* ignoriert.

MQGMO (Get Message Options)

Bei der Struktur MQGMO handelt es sich um einen Ein-/Ausgabeparameter des Aufrufs MQGET. Folgendes sollten Sie in MQSeries for Compaq NSK in Zusammenhang mit den Optionen MQGMO_SET_SIGNAL, MQGMO_WAIT, MQGMO_SYNCPOINT und MQGMO_NO_SYNCPOINT beachten:

- Soll die Anwendung, solange sie auf das Eintreffen einer Nachricht wartet, mit der Ausführung anderer Aufgaben fortfahren, sollten Sie die Signaloption MQGMO_SET_SIGNAL anstelle von MQGMO_WAIT verwenden. Die Signaloption ist jedoch umgebungsspezifisch und sollte nicht für Anwendungen verwendet werden, die zwischen verschiedenen Umgebungen portiert werden.
- Stehen für eine Nachricht mit einer Kombination von Warte- (MGWAIT) und Signaloptionen (MQGMO_SET_SIGNAL) mehrere MQGET-Aufrufe an, wird jeder Aufruf mit gleicher Priorität behandelt. MQGMO_SET_SIGNAL und MQGMO_WAIT dürfen nicht zusammen angegeben werden. Ebenso darf für MQGMO_WAIT keine Warteschlangenkennung angegeben werden, für die ein Signal aussteht.
- Gibt eine Anwendung einen MQGET-Aufruf mit der Option MQGMO_SET_SIGNAL und einem Warteintervall (**WaitInterval**) gleich null an, wird die Option MQGMO_SET_SIGNAL ignoriert; die Wirkung ist dieselbe wie bei der Angabe von MQGMO_NO_WAIT in einem MQGET-Aufruf.

Dies bedeutet, dass eine Anwendung unter Umständen den Ursachencode MQRC_NO_MSG_AVAILABLE erhält, wenn in einem MQGET-Aufruf mit der Option MQGMO_SET_SIGNAL das Feld **WaitInterval** den Wert Null enthalten kann. Anwendungen empfangen ein IPC-Signal nur in den folgenden Fällen:

- Die Anwendung erhält auf einen MQGET-Aufruf hin den Ursachencode MQRC_SIGNAL_REQUEST_ACCEPTED (gibt an, dass ein Signal *gesendet* wurde).
- Die Anwendung konnte die Systemnachricht 'file_open' verarbeiten und das IPC-Signal empfangen, bevor das Zeitlimit des Warteschlangenservers für die Signalzustellung abgelaufen ist. Dieses Zeitlimit liegt für den Warteschlangenserver standardmäßig bei 60 Sekunden; es kann jedoch über die Umgebungsvariable MQQSSIGTIMEOUT in der Umgebung des Warteschlangenservers geändert werden.

Wenn es nicht möglich war, vor Ablauf des Zeitlimits den Prozess zu öffnen und die IPC-Nachricht zu senden, protokolliert der WS-Manager diesen fehlgeschlagenen Versuch, die IPC-Nachricht an eine Anwendung zu übertragen. Der WS-Manager wird anschließend keine Zustellung mehr versuchen. Anwendungen sollten auf dieses Verhalten eingestellt sein und nicht unbegrenzt auf ein IPC-Signal warten.

- MQGMO_SYNCPOINT_IF_PERSISTENT wird jetzt unterstützt.
- Wird weder MQGMO_SYNCPOINT noch MQGMO_NO_SYNCPOINT gesetzt, wird in MQSeries for Compaq NSK standardmäßig die Option MQGMO_SYNCPOINT übernommen.
- MQSeries for Compaq NSK unterstützt das Feld *MsgToken* nicht.

Strukturdatentypen

MQMD (Message Descriptor)

Die MQMD-Struktur enthält die Steuerinformationen für eine Nachricht. Folgendes sollten Sie beachten:

- Die *BackoutCount*-Funktion wie im Handbuch *MQSeries Application Programming Reference* beschrieben. Dabei handelt es sich um einen Zähler, der angibt, wie oft die Nachricht bereits von einem MQGET-Aufruf innerhalb einer UOW (Unit of Work, Arbeitseinheit) zurückgegeben und anschließend zurückgesetzt wurde. Dies soll der Anwendung bei der Erkennung von Verarbeitungsfehlern helfen, die vom Inhalt der Nachricht abhängig sind. In Version 2.2.0.1 wurde dieser Wert geschätzt.
- In MQSeries for Compaq NSK erfolgt das Löschen einer Nachricht (und bei Bedarf die Generierung eines Berichts) nicht während eines MQGET-Aufrufs, sondern unter Steuerung des Warteschlangenservers in regelmäßigen Intervallen, die über den WS-Manager eingestellt werden.
- Falls der WS-Manager das Feld **UserIdentifier** bei einem MQPUT- oder MQPUT1-Aufruf setzt, gibt es den Namen des MQSeries-Principals an, der in der Principal-Datenbank des WS-Managers gefunden wurde und der tatsächlichen Benutzer-ID der Anwendung entspricht.

MQPMO (Put Message Options)

Bei der Struktur MQPMO handelt es sich um einen Ein-/Ausgabeparameter der Aufrufe MQPUT und MQPUT1. Folgendes sollten Sie in Zusammenhang mit der Option MQPMO_NO_SYNCPOINT in MQSeries for Compaq NSK beachten:

- Wird weder MQPMO_SYNCPOINT noch MQPMO_NO_SYNCPOINT gesetzt, wird in MQSeries for Compaq NSK standardmäßig die Option MQPMO_SYNCPOINT übernommen.

MQI-Aufrufe

In diesem Abschnitt werden Änderungen an den folgenden MQI-Aufrufen beschrieben:

MQI-Aufruf - Beschreibung	Unterstützt in V2.x	Unterstützt in V5.1	Funktionsweise wie in <i>MQSeries Application Programming Reference</i> beschrieben
MQBACK (Back Out Changes)	Gibt Fehler zurück ^{1 auf Seite 351}	Gibt Fehler zurück ^{1 auf Seite 351}	Ja
MQBEGIN (Begin Unit of Work)	Nein	Gibt Fehler zurück ^{1 auf Seite 351}	Ja
MQCLOSE (Close Object)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQCLOSE (Close Object)“ auf Seite 352.
MQCMIT (Commit Changes)	Nein	Gibt Fehler zurück ^{1 auf Seite 351}	Ja
MQCONN (Connect Queue Manager)	Ja	Ja	Ja
MQCONNX (Connect Queue Manager (Extended))	Nein	Ja	Ja

MQI-Aufruf - Beschreibung	Unterstützt in V2.x	Unterstützt in V5.1	Funktionsweise wie in <i>MQSeries Application Programming Reference</i> beschrieben
MQDISC (Disconnect Queue Manager)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQDISC (Disconnect Queue Manager)“ auf Seite 352.
MQGET (Get Message)	Ja	Ja	Ja
MQINQ (Inquire About Object Attributes)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQINQ (Inquire About Object Attributes)“ auf Seite 352.
MQOPEN (Open Object)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQOPEN (Open Object)“ auf Seite 352.
MQPUT (Put Message)	Ja	Ja	Ja
MQPUT1 (Put One Message)	Ja	Ja	Ja
MQSET (Set Object Attributes)	Ja	Ja	Ja, allerdings sind zusätzliche Hinweise zu beachten. Weitere Informationen finden Sie unter „MQSET (Set Object Attributes)“ auf Seite 353.
MQSYNC (Synchronize Statistics Updates)	Gibt Fehler zurück ²	Gibt Fehler zurück ²	Ja

Anmerkungen:

1. Der MQI-Aufruf kann von der Anwendung ausgegeben werden, er gibt allerdings immer den Beendigungscode MQCC_FAILED und den Ursachencode MQRC_ENVIRONMENT_ERROR zurück.
2. Bei diesem Aufruf wird für *CompCode* immer MQCC_OK und als Ursachencode MQRC_NONE zurückgegeben.

MQCLOSE (Close Object)

Mit dem Aufruf MQCLOSE, der das Gegenteil des Aufrufs MQOPEN darstellt, wird der Zugriff auf ein Objekt beendet.

Steht in MQSeries for Compaq NSK für die Warteschlangenkennung, die geschlossen werden soll, eine MQGET-Anforderung mit der Option MQGMO_SET_SIGNAL an, so wird die Anforderung gelöscht. Signalanforderungen für dieselbe Warteschlange, die jedoch unter einer anderen Kennung erfolgen (*Hobj*), sind nicht davon betroffen (außer es handelt sich um eine dynamische Warteschlange, die gelöscht werden soll; in diesem Fall werden die Anforderungen ebenfalls gelöscht).

Für eine FASTPATH-Anwendung, die eine dynamische Warteschlange öffnet oder schließt, startet und beendet MQSeries unter Umständen eine TM/MP-Transaktion, um die protokollierten Datenbanken zu aktualisieren. Öffnet die Anwendung eine TM/MP-Änderungsdatei (da sie mehrere Transaktionen starten kann), so wird nicht auf den Abschluss von ENDTRANSACTION gewartet, und die Anwendung erhält einen Beendigungscode für die von MQSeries eingeleitete Transaktion. Anwendungen sollten überprüfen, ob dies der Fall ist, und sicherstellen, dass die für Beendigungen zuständige Logik die Beendigungsvorgänge der von MQSeries eingeleiteten Prozedur ENDTRANSACTION handhaben kann.

MQDISC (Disconnect Queue Manager)

Der Aufruf MQDISC, der das Gegenteil von MQCONN bewirkt, unterbricht die Verbindung zwischen dem MQSeries-WS-Manager und dem Anwendungsprogramm.

Hinweis 3 im Handbuch *MQSeries Application Programming Reference* ist falsch. In MQSeries for Compaq NSK kommt es nicht zu einem impliziten Synchronisationspunkt, wenn eine vom WS-Manager koordinierte Arbeitseinheit bei einem MQDISC-Aufruf aktiv ist. Dies ist deswegen nicht der Fall, weil ein NSK-WS-Manager keine Arbeitseinheit koordinieren kann. Die Koordination erfolgt durch das TM/MP-Subsystem.

MQINQ (Inquire About Object Attributes)

Der Aufruf MQPUT gibt einen Bereich ganzer Zahlen (Integer) sowie eine Gruppe von Zeichenfolgen zurück, die die Attribute eines Objekts enthalten.

MQOPEN (Open Object)

Der Aufruf MQOPEN richtet den Zugriff auf ein Objekt ein. In MQSeries for Compaq NSK wird das Attribut **MaxHandles** des WS-Managers ignoriert.

Für eine FASTPATH-Anwendung, die eine dynamische Warteschlange öffnet oder schließt, startet und beendet MQSeries unter Umständen eine TM/MP-Transaktion, um die protokollierten Datenbanken zu aktualisieren. Öffnet die Anwendung eine TM/MP-Änderungsdatei (da sie mehrere Transaktionen starten kann), so wird nicht auf den Abschluss von ENDTRANSACTION gewartet, und die Anwendung erhält einen Beendigungscode für die von MQSeries eingeleitete Transaktion. Anwendungen sollten überprüfen, ob dies der Fall ist, und sicherstellen, dass die für Beendigungen zuständige Logik die Beendigungsvorgänge der von MQSeries eingeleiteten Prozedur ENDTRANSACTION handhaben kann.

MQSET (Set Object Attributes)

Der Aufruf MQSET ändert die Attribute eines über eine Kennung angegebenen Objektes. Bei diesem Objekt muss es sich um eine Warteschlange handeln. In MQSeries for Compaq NSK wird der Selektor MQIA_DIST_LISTS unterstützt.

Für eine FASTPATH-Anwendung, die die Attribute eines Objekts mit MQSET ändert, startet und beendet MQSeries eine TM/MP-Transaktion, um die protokollierten Datenbanken zu aktualisieren. Öffnet die Anwendung eine TM/MP-Änderungsdatei (da sie mehrere Transaktionen starten kann), so wird nicht auf den Abschluss von ENDTRANSACTION gewartet, und die Anwendung erhält einen Beendigungscode für die von MQSeries eingeleitete Transaktion. Anwendungen sollten überprüfen, ob dies der Fall ist, und sicherstellen, dass die für Beendigungen zuständige Logik die Beendigungsvorgänge der von MQSeries eingeleiteten Prozedur ENDTRANSACTION handhaben kann.

Attribute von MQSeries-Objekten

Alle Objektattribute in MQSeries for Compaq NSK werden im Handbuch *MQSeries Application Programming Reference* beschrieben, ausgenommen der nachfolgend aufgeführten Attribute.

Attribute für alle Warteschlangen

Alle Warteschlangenattribute in MQSeries for Compaq NSK werden im Handbuch *MQSeries Application Programming Reference* beschrieben, ausgenommen der nachfolgend aufgeführten Attribute.

Die Attribute **AlterationDate** und **AlterationTime** werden nur bei Änderungen an Objektattributen im Zusammenhang mit Verwaltungs-Tasks aktualisiert. Die Attribute **CurrentQDepth**, **OpenInputCount** und **OpenOutputCount** können nur dynamisch geändert werden. Die Attribute **QDepthHighCount**, **QDepthLowEvent**, **QDepthMaxEvent** und **QServiceIntervalEvent** können sowohl dynamisch als auch durch Verwaltungs-Tasks geändert werden, allerdings wirken sich nur die von Verwaltungs-Tasks verursachten Änderungen (wie sie z. B. mit MQSC-Befehlen oder über MQSET ausgeführt werden) auf die Attribute **AlterationDate** und **AlterationTimes** aus.

Attribute der lokalen Warteschlangen und der Modellwarteschlangen

In MQSeries for Compaq NSK gilt Folgendes:

- Das Attribut *Archive* wird ignoriert.
- Das Attribut *HardenGetBackout* wird ignoriert, da die Anzahl der Rücksetzungsvorgänge nicht auf Platte gespeichert wird. Es gibt keine Möglichkeit, Nachrichten zu archivieren.

Attribute für alle Warteschlangen

- Für permanente Nachrichten wird das Attribut *BackoutCount* immer gespeichert, für nicht permanente Nachrichten dagegen nie. Wenn jedoch für die lokale Warteschlange die Option `-qoptions C` gesetzt ist, werden für das Attribut *BackoutCount* Prüfpunktinformationen im Backup-Warteschlangenserver aufgezeichnet. Nachrichten, für die auf diese Weise Prüfpunktinformationen aufgezeichnet werden, können nach einem Ausfall des Warteschlangenservers wiederhergestellt werden. Aus Gründen der Kompatibilität mit anderen MQSeries-Plattformen kann das Attribut mit dem Aufruf `MQINQ` unter Angabe des Selektors `MQIA_HARDEN_GET_BACKOUT` abgefragt werden.

Attribute der WS-Manager

- Der Wert für *MaxMsgLength* ist 100 MB.
- Der Wert für *CommandLevel* ist `MQCMDL_LEVEL_510`.
- Der Wert für *SyncPoint* ist `MQSP_AVAILABLE`.
- *CodedCharSetId* gibt die CCSID an, die bei der Erstellung des WS-Managerexemplars angegeben wurde.
- Das Attribut *MaxHandles* wird ignoriert. Es ist in MQSeries for Compaq NSK nicht möglich, eine maximal zulässige Anzahl offener Kennungen anzugeben. Dieser Wert hängt von den Systemressourcenvorgaben ab.
- Das Attribut *MaxUncommittedMsgs* wird ignoriert. Es ist nicht möglich, eine innerhalb einer Arbeitseinheit maximal zulässige Anzahl von Nachrichten anzugeben. Dieser Wert hängt von den Ressourcenvorgaben ab.
- Die CCSID kann geändert werden.

Datenkonvertierung

Eine Beschreibung des Verfahrens, mit dem alle Exits in MQSeries for Compaq NSK V5.1 unterstützt werden finden Sie in „Anhang L. Benutzer-Exits“ auf Seite 373. Dieses Verfahren unterscheidet sich von dem früherer Versionen, um eine konsistentere und portablere Implementierung von Exits zu ermöglichen.

Anhang I. Anwendungen erstellen und ausführen

Die im Teildatenträger ZMQSSMPL enthaltenen Beispielprogramme sowie die Beispiel-Scripts für die Kompilier- und Bindevorgänge veranschaulichen die wichtigsten Funktionen der MQI-Schnittstelle in MQSeries for Compaq NSK und zeigen, wie eine Anwendung kompiliert und gebunden wird.

Anwendungen schreiben

Dieser Abschnitt enthält aktuelle Informationen zu einigen geringfügigen Unterschieden zwischen der im Handbuch *MQSeries Application Programming Guide* beschriebenen MQI-Standardschnittstelle der Version 5.1 und der MQI-Schnittstelle für MQSeries for Compaq NSK. Das Handbuch *MQSeries Application Programming Guide* sollte im Hinblick auf MQSeries for Compaq NSK V5.1 mit den Informationen in den folgenden Abschnitten aktualisiert werden.

Warteintervall in MQGET und Kanalattribute DISCINT und HBINT verwenden

Bei einem MQGET-Aufruf mit der Option MQGMO_NO_SYNCPOINT wird eine Transaktion von MQSeries nur dann gestartet und beendet, wenn eine permanente Nachricht zur Verfügung steht, die den Abrufkriterien entspricht. Bei Operationen mit der Option NO_SYNCPOINT muss die Anwendung den für das Warteintervall (**WaitInterval**) angegebenen Wert nicht berücksichtigen.

Bei einem MQGET-Aufruf mit der Option MQGMO_SYNCPOINT oder MQGMO_SYNCPOINT_IF_PERSISTENT wird die TM/MP-Transaktion von der Benutzeranwendung gesteuert, von der die Transaktion mit BEGINTRANSACTION eingeleitet wurde. Das Warteintervall sollte den für das TMF-Zeitlimit für automatischen Abbruch angegebenen Wert nicht überschreiten und im Idealfall kurz sein, um zu verhindern, dass ein großer Teil des TM/MP-Prüfprotokolls gesperrt wird (in der Regel sollte der Wert unter einer Minute liegen). Bei Angabe eines zeitlich unbegrenzten Warteintervalls für eine Warteschlange, auf die nur selten ein Zugriff erfolgt oder die über Nacht nicht aktiv ist, wird unter Umständen das Zeitlimit für den automatischen Abbruch überschritten und der Rückkehrcode MQRC_UOW_CANCELLED (2297) an den MQGET-Aufruf zurückgegeben, wenn schließlich eine Nachricht in der Warteschlange zur Verfügung steht. Bei Angabe eines großen oder sogar unbegrenzten Warteintervalls werden TMF-Prüfprotokolle unter Umständen gesperrt; wird hiergegen nichts unternommen, kann dies unter Umständen dazu führen, dass das TM/MP-Subsystem Transaktionen im ganzen System inaktiviert.

Ähnlich steuern auch die Werte der Attribute DISCINT und HBINT für Senderkanäle die Länge einer TM/MP-Transaktion. Momentan inaktive Kanäle haben die Möglichkeit, Transaktionen zyklisch auszuführen; dadurch können lange Abbruchintervalle (d. h., bevor der Kanal beendet wird) angegeben werden, ohne dass sich dies nachteilig auf TM/MP-Prüfprotokolle auswirkt.

Anwendungen schreiben

Mit dem Parameter MQTRANSACTIONLIFE kann die Aktualisierung einer TM/MP-Transaktion, für die das Kanalattribut DISCINT und Überwachungssignale auf null gesetzt ist, gesteuert werden. Dies ist hilfreich, wenn eine längere oder kürzere TM/MP-Transaktionsdauer erwünscht ist oder der Umfang der TM/MP-Aktivitäten des inaktiven Kanals geändert werden soll. Bei Angabe eines höheren Wertes verringert sich der Umfang, bei einem niedrigeren Wert erhöht er sich.

Fügen Sie für jede MQSeries-PATHWAY-Serverklasse MQS-ECxx Folgendes hinzu:

```
PARAM MQTRANSACTIONLIFE <Zahl>
```

Dabei steht <Zahl> für eine Zahl wie beispielsweise 100.

Der Wert dieses Parameters überschreibt das Standardaktualisierungsintervall für TM/MP-Transaktionen, das bei 10 Sekunden liegt. Ein Kanal mit einem unbegrenzten Abbruchintervall, für den dieser Parameter nicht angegeben ist, wird während der Wartezeit auf eine Nachricht alle 10 Sekunden aktualisiert.

Arbeitseinheiten (UOW) verwalten

Die Transaktionsverwaltung wird von Compaq TM/MP, und nicht von MQSeries selbst gesteuert.

Dies bewirkt Folgendes:

- Die standardmäßige SYNCPOINT-Option für MQPUT- und MQGET-Aufrufe ist SYNCPOINT, und nicht NO_SYNCPOINT.
- Bei Verwendung der Standardoption (SYNCPOINT) für MQPUT-, MQGET- und MQPUT1-Aufrufe muss für die Anwendung eine aktive TM/MP-Transaktion bestehen, die der Arbeitseinheit (UOW, Unit of Work) entspricht, die festgeschrieben werden soll. Eine Anwendung leitet eine TM/MP-Transaktion durch Aufruf der Funktion BEGINTRANSACTION() ein. Alle MQPUT-, MQPUT1- und MQGET-Aufrufe, die von der Anwendung ausgeführt werden, während diese Transaktion aktiv ist, erfolgen innerhalb derselben Arbeitseinheit (Transaktion). Alle anderen Datenbankoperationen der Anwendung werden ebenfalls innerhalb dieser Arbeitseinheit ausgeführt. Die maximale Anzahl und Größe der Nachrichten, die innerhalb einer TM/MP-Transaktion geschrieben bzw. gelöscht werden können, hängt von den Systemvorgaben ab. Bei Abschluss einer Arbeitseinheit wird die TM/MP-Transaktion mit der Funktion ENDTRANSACTION() beendet, d. h. die Arbeitseinheit wird festgeschrieben. Bei Auftreten eines Fehlers kann die Anwendung die TM/MP-Transaktion mit der Funktion ABORTTRANSACTION() abbrechen, d. h. die Arbeitseinheit wird zurückgesetzt. Die Standardfunktionen MQCMIT(), MQBACK() und MQBEGIN() von Version 5 werden daher in diesem Produkt nicht unterstützt. Bei ihrem Aufruf wird ein Fehler zurückgegeben.

Anwendungen schreiben

- Gibt eine Anwendung einen MQPUT-, MQGET- oder MQPUT1-Aufruf mit der Option NO_SYNCPOINT aus, wird von MQSeries selbst eine TM/MP-Transaktion gestartet, die entsprechende Operation ausgeführt und die Transaktion festgeschrieben, bevor die Steuerung wieder an die Anwendung zurückgegeben wird. Jeder Vorgang wird daher innerhalb einer eigenen Arbeitseinheit ausgeführt und kann nach Abschluss von der Anwendung nicht mehr mit TM/MP zurückgesetzt werden.
- MQSeries muss jedoch eine eigene TM/MP-Transaktion (falls eine Operation mit der Option NO_SYNCPOINT angefordert wurde) nur für permanente Nachrichten starten, da für diese die Aktualisierung einer durch TM/MP geschützten Warteschlangendatei erforderlich ist.
- TM/MP-Transaktionen sind nur für die MQI-Aufrufe MQGET, MQPUT und MQPUT1 erforderlich.
- Mit dem Ursachencode MQRC_SYNCPOINT_LIMIT_REACHED meldet MQSeries for Compaq NSK einer Anwendung, dass die vom System vorgegebene maximale Anzahl an E/A-Operationen, die innerhalb einer TM/MP-Transaktion erfolgen können, erreicht wurde. Wurde von der Anwendung die Option SYNCPOINT angegeben, sollte die Transaktion von dieser abgebrochen, d. h. die Arbeitseinheit zurückgesetzt werden und mit einer geringeren Anzahl an E/A-Operationen während der Arbeitseinheit wiederholt werden.
- Der Ursachencode MQRC_UOW_CANCELED informiert die Anwendung darüber, dass die Arbeitseinheit (d. h. die TM/MP-Transaktion) abgebrochen wurde, und zwar entweder vom System selbst (TM/MP gibt einen systemweiten Schwellenwert für die Ressourcenbelegung vor, der zum Abbruch führt), durch Benutzerintervention oder vom eigentlichen Initiator der Transaktion.
- Der Ursachencode MQRC_BACKED_OUT informiert die Anwendung darüber, dass MQSeries die Arbeitseinheit auf Grund eines Fehlers oder wegen Ausfalls des primären Warteschlangenservers abgebrochen hat. Die Anwendung sollte in diesem Fall die Prozedur ABORTTRANSACTION aufrufen (falls die Option SYNCPOINT für die Operation angegeben wurde) und die Transaktion wiederholen.
- Der Ursachencode MQRC_SYNCPOINT_NOT_AVAILABLE informiert die Anwendung darüber, dass MQSeries eine TM/MP-Transaktion, die für die Ausführung einer Operation erforderlich war, nicht starten oder nicht verwenden konnte. Dies deutet in der Regel auf ein TM/MP-internes Problem hin; Sie finden zusätzliche Informationen unter Umständen im Fehlerprotokoll oder in den vom WS-Manager generierten FFST-Berichten.

Anwendungen schreiben

Allgemeine Überlegungen bei der Erstellung von Anwendungen

Folgendes sollten Sie beachten:

- Die MQI-Bibliothek (die in den Anwendungsprozess eingebunden wird) öffnet weder \$RECEIVE noch \$TMP (TM/MP-Transaktionspseudodatei) selbst, daher sollten Sie eventuell Ihre Anwendung für die Verwendung dieser Merkmale codieren.
- Die MQI-Bibliothek verwendet für die Kommunikation mit dem WS-Manager in der Anfangsphase den Aufruf `SERVERCLASS_SEND_()`. Für die Dauer der Verbindung sind mindestens zwei Prozessdateien (Prozess `LINKMON` und ein lokaler WS-Manageragent) und eine kleine Anzahl weiterer Plattendateien (weniger als 10) geöffnet. Prozessverbindungen werden auch mit allen Warteschlangenservern hergestellt, die auf einen `MQOPEN`-Aufruf hin für Eingaben, Ausgaben oder Browse-Vorgänge geöffnet werden.
- Bei Aufruf von `MQCONN` sollten Sie zunächst sicherstellen, dass keine `PATHSEND-E/A`-Vorgänge, für die die Option `NOWAIT` angegeben ist, anstehen. `MQCONN` selbst führt `PATHSEND-E/A`-Vorgänge mit der Option `NOWAIT` aus, was zu Konflikten mit ausstehenden `E/A`-Vorgängen für die Anwendung und daher zu Fehlern führen könnte.
- Für `FASTPATH`-Anwendungen gelten spezielle Bedingungen, wenn sie auch noch mehrere `TM/MP`-Transaktionen einleiten können (siehe die Beschreibung von `MQOPEN`, `MQCLOSE` und `MQSET` oben).

XA-Schnittstelle

Für die Koordination von Arbeitseinheiten steht keine XA-Schnittstelle zur Verfügung. Die Koordination wird vollständig von `TM/MP` übernommen.

MQGMO_BROWSE_* mit MQGMO_LOCK

`MQGMO_BROWSE_*` mit `MQGMO_LOCK` wird jetzt unterstützt (siehe *MQSeries Application Programming Reference*).

Ausgelöste Anwendungen

Ausgelöste `MQSeries`-Anwendungen in der Compaq NSK-Umgebung empfangen Benutzerdaten über Umgebungsvariablen, die in dem aktiven `TACL`-Prozess gesetzt werden. Dies liegt daran, dass die Länge der an den `Compaq-C`-Prozess übergebenen Argumentenliste begrenzt ist.

Damit ein Zugriff auf diese Informationen möglich ist, sollten die Anwendungen Code wie den folgenden enthalten (siehe das Beispiel `amqsinqa` für ausführlichere Informationen):

```

MQTMC2 *trig;                /* trigger message structure */
MQTMC2 trigdata;            /* trigger message structure */
char    *applId;
char    *envData;
char    *usrData;
char    *qmName;

/*****
/*
/* Set the program argument into the trigger message
/*
/*
*****/
trig = (MQTMC2*)argv[1];    /* -> trigger message */

/* get the environment variables and load the rest of the 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;

```

Unterstützte Programmiersprachen und Umgebungen

MQSeries for Compaq NSK V5.1 unterstützt die in Tabelle 24 auf Seite 360 aufgeführten Programmiersprachen und Umgebungen. In dieser Tabelle ist außerdem angegeben, ob Anwendungen FASTPATH- oder STANDARD-Bindungen verwenden können.

Anwendungen schreiben

Tabelle 24. Unterstützte Programmiersprachen und Umgebungen - Überblick

Sprache	Läuft unter Guardian? ¹	Läuft unter OSS?	Kann STANDARD-Bindung verwenden?	Kann FASTPATH-Bindung verwenden? ¹
C (nativ)	Ja	Ja	Ja	Ja
C (nicht nativ)	Ja	Nein	Ja	Nein
COBOL (nativ)	Ja	Ja	Ja	Ja
COBOL (nicht nativ)	Ja	Nein	Ja	Nein
C++ (nativ)	Ja	Ja	Ja	Ja
TAL (nicht nativ)	Ja	Nein	Ja	Nein
NonStop Java ²	Nein	Ja	Ja	Nein

Anmerkungen:

1. Die Guardian-Umgebung und OSS-Anwendungen, die FASTPATH-Bindungen verwenden, können keine Threads verwenden. Nur unter OSS mit STANDARD-Bindungen und unter Java können Threads verwendet werden. Weitere Informationen zur Verwendung von Threads in Ihrer Anwendung finden Sie unter „Überlegungen bei der Erstellung von Anwendungen mit Threads“.
2. NonStop Java-Anwendungen verwenden JTS (Java Transaction Services) für Transaktionen.
3. Eine native Anwendung, die FASTPATH-Bindungen verwendet, kann MQI-Aufrufe nur über die Bibliothek für gemeinsam benutzte Ressourcen (Shared Resource Library) von MQSeries for Compaq NSK auflösen. Eine native Anwendung, die STANDARD-Bindungen verwendet, kann MQI-Aufrufe entweder über die Shared Resource Library von MQSeries for Compaq NSK oder über eine statische native MQI-Bibliothek auflösen. Weitere Informationen zu FASTPATH- und STANDARD-Bindungen finden Sie unter „FASTPATH- und STANDARD-Bindungen“ auf Seite 361.

Überlegungen bei der Erstellung von Anwendungen mit Threads

Guardian-Anwendungen unterstützen keine Threads. Sie können ein eigenes kooperatives Thread-Verfahren verwenden, aber die Regeln, die für die Verwendung der MQI-Schnittstelle in der Guardian-Umgebung gelten, müssen von dem Prozess, der sie verwendet, eingehalten werden.

In der OSS-Umgebung steht ein Emulationspaket, das auf POSIX-Threads beruht, zur Verfügung. Die Emulation implementiert ein kooperatives Scheduling-Verfahren, bei dem ein Thread die Ausführungssteuerung zurückgeben muss, bevor der Code in einem anderen Thread ausgeführt werden kann. Anwendungen können mit diesem Thread-Paket die Verarbeitung in Threads einteilen, es gelten jedoch die folgenden Einschränkungen:

- Die MQI-Schnittstelle unterstützt kein kooperatives Scheduling zwischen Threads. Wird die MQI-Schnittstelle von einem Anwendungs-Thread aufgerufen, kann die Ausführungssteuerung an keinen anderen Thread übergeben werden, unabhängig davon, wie lange dies dauern kann. Gibt beispielsweise ein Thread einen MQGET-Aufruf mit der Option 'wait indefinitely' aus, können weitere Threads des Anwendungsprozesses erst nach Rückkehr dieses MQGET-Aufrufs ausgeführt werden.

- FASTPATH-Anwendungen in einer OSS-Umgebung können keine Threads verwenden.
- Der WS-Manager unterstützt keine Multi-Threads für lokale WS-Manageragenten ((LQMA- oder MQLQMAG-Prozesse) oder für Nachrichtenkanalagenten (MCAs).

Anwendungen kompilieren und binden

Die MQI-Schnittstelle in MQSeries for Compaq NSK wird über das Wide Memory-Modell von Compaq (der Datentyp `int` hat 4 Bytes) und CRE (Common Runtime Environment) implementiert. Anwendungen können nur richtig arbeiten, wenn sie mit dieser Umgebung kompatibel sind. Die richtigen Optionen für die einzelnen Compiler, die diese Kompatibilität sicherstellen, finden Sie in den Beispieldateien zum Erstellen von Anwendungen.

Besonders bei TAL- und COBOL-Anwendungen müssen die für die Kompatibilität mit CRE erforderlichen Regeln beachtet werden; diese Regeln sind in den Compaq-Handbüchern für CRE aufgeführt.

Für eine erfolgreiche Ausführung der Funktion `MQGMO_SET_SIGNAL` in `MQGET` müssen Sie das Attribut `HIGHREQUESTERS` im Objektcode für COBOL- und TAL-Anwendungen auf `ON` setzen.

Für die Installation stehen für MQSeries for Compaq NSK drei Versionen der MQI-Bibliothek in `ZMQSLIB` zur Verfügung. Stellen Sie sicher, dass Sie die richtige Bibliothek verwenden (siehe Tabelle 25).

Tabelle 25. Richtige Version der MQI-Bibliothek

Programmiersprache	Nicht nativ	Nativ/statisch	Nativ/dynamisch
TAL	MQMLIB	Nicht zutreffend	Nicht zutreffend
COBOL	MQMLIB	MQMLIBN	MQSRLLIB
C	MQMLIB	MQMLIBN	MQSRLLIB
C++	Nicht zutreffend	MQMLIBN	MQSRLLIB

FASTPATH- und STANDARD-Bindungen

MQSeries for Compaq NSK V5.1 unterstützt sowohl FASTPATH- als auch STANDARD-Bindungen. Aus Tabelle 25 wird ersichtlich, welche Bindung von den einzelnen Programmiersprachen und Umgebungen unterstützt wird.

STANDARD-Bindungen

Bei der Verwendung von STANDARD-Bindungen in einer Anwendung sollten Sie folgendes beachten:

- Sowohl nicht native als auch native Anwendungen können STANDARD-Bindungen verwenden.
- Native Anwendungen, die STANDARD-Bindungen verwenden, haben folgende Möglichkeiten für die Auflösung von MQI-Aufrufen:
 - Die Shared Resource Library von MQSeries for Compaq NSK.
 - Eine statische native MQI-Bibliothek. Dies ermöglicht die Unterstützung von Anwendungen, die bereits eine Shared Resource Library verwenden.

Anwendungen kompilieren

FASTPATH-Bindungen

Bei der Verwendung von FASTPATH-Bindungen in einer Anwendung sollten Sie folgendes beachten:

- Nur native Anwendungen können FASTPATH-Bindungen verwenden.
- Eine native Anwendung, die FASTPATH-Bindungen verwendet, kann MQI-Aufrufe nur über die Bibliothek für gemeinsam benutzte Ressourcen (Shared Resource Library) von MQSeries for Compaq NSK auflösen.
- FASTPATH-Anwendungen unter OSS können keine Threads verwenden.
- FASTPATH-Anwendungen müssen unter der Benutzer-ID in der Administratorbenutzergruppe **MQM** in Compaq NSK laufen, unter der der WS-Manager erstellt wurde.

Anwendungen ausführen

Damit eine Verbindung zu einem WS-Manager hergestellt werden kann, muss die Umgebung eines Anwendungsprogramms richtig definiert werden:

- Die Umgebungsvariable MQDEFAULTPREFIX muss in der Umgebung aller Anwendungen gesetzt werden.
- Wenn Sie die Datei MQSINI in einem anderen als dem Standardverzeichnis unterbringen, kann eine Anwendung keine Verbindung zum WS-Manager herstellen, wenn die Umgebungsvariable MQMACHINIFILE nicht richtig gesetzt ist.
- In den Umgebungen von TAL- und COBOL-Anwendungen muss die Umgebungsvariable SAVE-ENVIRONMENT auf ON gesetzt sein, andernfalls können sie keine Verbindung zum WS-Manager herstellen.

Eine Anwendung kann entweder als Lowpin- oder als Highpin-Prozess ausgeführt werden. Ausführbare MQSeries-Module werden so konfiguriert, dass sie als Highpin-Prozesse ausgeführt werden.

MQSeries-Anwendungen werden sowohl in Guardian- als auch in OSS-Umgebungen unterstützt.

Eine MQSeries-Anwendung kann unter PATHWAY, über TAACL oder als Kindprozess eines anderen Prozesses ausgeführt werden. Anwendungen können sogar der PATHWAY-Konfiguration des WS-Managers hinzugefügt werden, vorausgesetzt sie verhalten sich beim Beenden des WS-Managers korrekt.

Anhang J. Die MQSeries-Verwaltungsschnittstelle

MQSeries for Compaq NSK V5.1 unterstützt die MQSeries-Verwaltungsschnittstelle (MQSeries Administration Interface, MQAI).

Bei dieser Schnittstelle handelt es sich um die Programmierschnittstelle zu MQSeries, die die Programmiersprache C verwendet. Über diese Schnittstelle werden Verwaltungs-Tasks für den MQSeries-WS-Manager unter Verwendung von Datenbehältern (Data Bags) ausgeführt. Diese Data Bags ermöglichen eine einfachere Handhabung von Objektmerkmalen bzw. -parametern als dies mit der anderen Verwaltungsschnittstelle, den PCF-Befehlen (Programmable Command Format) möglich ist.

Die Verwaltungsschnittstelle MQAI erlaubt eine einfachere Bearbeitung von PCF-Befehlen als die Aufrufe MQGET und MQPUT. Die Schnittstelle kann für Folgendes verwendet werden:

- Implementieren von Anwendungen, die sich selbst verwalten, und von Verwaltungs-Tools.
- Einfachere Verwendung von PCF-Nachrichten. Die Schnittstelle MQAI erleichtert die Verwaltung von MQSeries. Sie müssen keine eigenen PCF-Nachrichten erstellen, wodurch Probleme im Zusammenhang mit komplexen Datenstrukturen vermieden werden.
- Einfachere Handhabung von Fehlerbedingungen. Es ist schwierig, Rückkehrcode für MQSeries-Befehle (MQSC-Befehle) zu erhalten, die Schnittstelle MQAI erleichtert jedoch dem Programm die Bearbeitung von Fehlerbedingungen.

Anmerkung: MQSeries for Compaq NSK V5.1 stellt nur C-Header-Dateien zur Verfügung. Es werden keine VB-Header-Dateien (Visual Basic) bereitgestellt.

Weitere Informationen zur Verwaltungsschnittstelle finden Sie im Handbuch *MQSeries Administration Interface Programming Guide and Reference*

Anhang K. MQSeries for Compaq NSK - Beispielprogramme

Die folgenden C- und COBOL-Beispielprogramme stehen in MQSeries for Compaq NSK V5.1 zur Verfügung:

Beschreibung	C-Quellcode	Ausführbare C-Datei	COBOL85-Quellcode	Ausführbare COBOL85-Datei
Liest für jede Nachricht in einer Warteschlange den Nachrichten-deskriptor und Kontext und gibt ihn aus.	amqsbcg0	amqsbcg	Kein Beispiel	Kein Beispiel
Meldet eine Nachricht in einer Nachrichtenwarteschlange an die Warteschlange für Antwortnachrichten zurück.	amqsecha	amqsech	amq0ech0	amq0ech
Schreibt Nachrichten von einer Warteschlange in die Standardausgabe (stdout), wobei die Nachrichten in der Warteschlange verbleiben (Browse-Vorgang).	amqsgbr0	amqsgbr	amq0gbr0	amq0gbr
Entfernt Nachrichten aus der angegebenen Warteschlange und schreibt sie in die Standardausgabe (stdout).	amqsget0	amqsget	amq0get0	amq0get
Liest die ausgelöste Warteschlange und gibt Informationen über die Warteschlange zurück.	amqsinqa	amqsinq	Kein Beispiel	Kein Beispiel
Verwendet eine gemeinsam benutzte Eingabewarteschlange.	Kein Beispiel	Kein Beispiel	amq0inq0	amq0inq
Kopiert die Standardeingabe (stdin) in die Nachricht und reiht die Nachricht in die angegebene Warteschlange ein.	amqsput0	amqsput	amq0put0	amq0put
Reiht eine Anforderungsnachricht in die angegebene Warteschlange ein und zeigt die Antworten an.	amqsreq0	amqsreq	amq0req0	amq0req
(Auslösefunktion) Verhindert PUT-Vorgänge für die angegebene Warteschlange und gibt als Antwort eine Ergebnismnachricht zurück.	amqsseta	amqsset	amq0set0	amq0set
Auslösemonitor	amqstrg0	amqstrg	Kein Beispiel	Kein Beispiel
Beispielentwurf für einen Datenkonvertierungs-Exit.	amqsvfcn	Kein Beispiel	Kein Beispiel	Kein Beispiel
Beispielentwurf für einen Kanal-Exit.	amqsvchn	Kein Beispiel	Kein Beispiel	Kein Beispiel
Beispielentwurf für einen Cluster-Workload-Exit.	amqswlm0	Kein Beispiel	Kein Beispiel	Kein Beispiel
Beispielentwurf für MQLOADEXIT.	amqslxp0	Kein Beispiel	Kein Beispiel	Kein Beispiel

Beispielprogramme

Die folgenden TAL-Beispielprogramme stehen in MQSeries for Compaq NSK V5.1 zur Verfügung:

Beschreibung	TAL-Quelle	Ausführbare TAL-Datei
Liest n Nachrichten aus einer Warteschlange aus.	zmqreadt	zmqread
Schreibt n Nachrichten mit einer Länge von n in eine Warteschlange.	zmqwritt	zmqwrit

C-Beispielprogramm erstellen

Nicht native Anwendungen (unter Verwendung der nicht nativen statischen Bibliothek MQMLIB) verwenden

Der Teildatenträger ZMQSSMPL enthält die folgenden TACL-Makrodateien, die für die Erstellung von nicht nativen C-Beispielanwendungen verwendet werden können:

CSAMP

Verwendung: *CSAMP Name_der_Quellcodedatei*

Mit diesem Basismakro wird eine C-Quelldatei unter Verwendung der Include-Dateien im Teildatenträger ZMQSLIB kompiliert. Beispielsweise wird die Beispielanwendung AMQSBCG0 unter Verwendung von CSAMP AMQSBCG0 kompiliert. Bei einer erfolgreichen Kompilierung erstellt das Makro eine Objektdatei, wobei das letzte Zeichen des Dateinamens durch den Buchstaben O ersetzt wird (Beispiel: AMQSBCGO).

BSAMP

Verwendung: *BSAMP Name_der_EXE-Datei*

Mit diesem Basismakro wird eine Objektdatei in die Benutzerbibliothek MQMLIB im Teildatenträger ZMQSLIB eingebunden. Beispielsweise wird die kompilierte Beispielobjektdatei AMQSBCG0 unter Verwendung von BSAMP AMQSBCG gebunden. Das Makro erstellt eine ausführbare Datei, deren Name das Format *Name_der_ausführbaren_DateiE* hat (Beispiel: AMQSBCGE).

COMPALL

Verwendung: COMPALL

Mit diesem TACL-Makro werden die einzelnen Beispielquellcodedateien unter Verwendung des Makros CSAMP kompiliert.

BINDALL

Verwendung: BINDALL

Mit diesem TACL-Makro werden die einzelnen Beispielobjektdateien unter Verwendung des Makros BSAMP zu ausführbaren Dateien gebunden.

BUILDC

Verwendung: BUILDC

Mit diesem TACL-Makro werden alle C-Beispieldateien unter Verwendung der Makros COMPALL und BINDALL kompiliert und gebunden.

Native Anwendungen (unter Verwendung der nativen statischen Bibliothek MQMLIBN)

Bei einer nativen Installation werden die folgenden TACL-Makrodateien für die Erstellung von MQI-Beispielanwendungen verwendet:

NMCALL	Verwendung: NMCALL Mit diesem Makro werden alle Beispiele im nativen Modus unter Verwendung von NMCSAMP kompiliert.
NMCSAMP	Verwendung: NMCSAMP <i>Name_der_Quellcodedatei</i> Mit diesem Basismakro wird eine C-Quellendatei unter Verwendung der Include-Dateien im Teildatenträger ZMQSLIB kompiliert. Beispielsweise wird die Beispielanwendung AMQSBCG0 unter Verwendung von NMCSAMP AMQSBCG0 kompiliert. Bei einer erfolgreichen Kompilierung erstellt das Makro eine Objektdatei, wobei das letzte Zeichen des Dateinamens durch den Buchstaben O ersetzt wird (Beispiel: AMQSBCGO).
NMLDSAMP	Verwendung: NMLDSAMP <i>Name_der_EXE-Datei</i> Mit diesem Basismakro wird eine Objektdatei mit der statischen nativen MQI-Bibliothek MQMLIBN im Teildatenträger ZMQSLIB verbunden.
NMLDALL	Verwendung: NMLDALL Mit diesem TACL-Makro werden die einzelnen Beispielobjektdateien unter Verwendung des Makros NMLDSAMP zu ausführbaren Dateien gebunden.
NMBUILD	Verwendung: NMBUILD Mit diesem TACL-Makro werden alle nativen C-Beispieldateien unter Verwendung der Makros NMCALL und NMLDALL kompiliert und gebunden.

Native Anwendungen (unter Verwendung der SRL MQSRLLIB)

NMLDSSMP	Verwendung: NMLDSSMP <i>Name_der_EXE-Datei</i> Mit diesem Basismakro wird eine Objektdatei mit der nativen MQSeries-SRL MQSRLLIB im Teildatenträger ZMQSLIB verbunden.
NMLDSALL	Verwendung: NMLDSALL Mit diesem TACL-Makro werden die einzelnen Beispielobjektdateien unter Verwendung des Makros NMLDSSMP zu ausführbaren Dateien verknüpft.
NMBULDSC	Verwendung: NMBULDSC Mit diesem TACL-Makro werden alle nativen C-Beispieldateien unter Verwendung der Makros NMCALL und NMLDSALL kompiliert und verknüpft.

C-Beispiele erstellen

NMLDPSRL

Verwendung: NMLDPSRL *Name_der_EXE-Datei*

Mit diesem Basismakro wird eine Objektdatei mit der privaten MQSeries-SRL im Teildatenträger ZMQSLIB verbunden.

NMCPSSL

Verwendung: NMCPSSL *Name_der_Quellcodedatei*

Mit diesem Makro wird der Benutzercode für die Einbindung in die MQSeries-PSRL kompiliert.

NMLDUSRL

Verwendung: NMLDUSRL *Objekteingabedatei*; dabei steht *Objekteingabedatei* für eine Datei, die eine Liste der Objekte enthält, die verbunden werden sollen.

Mit diesem Basismakro wird der Benutzercode in eine Bibliothek eingebunden, die erneut verbunden werden kann.

Anmerkung: Nicht native Anwendungen können mit nativen WS-Managern verbunden werden und umgekehrt. Alle Kombinationen aus nativen und nicht nativen Operationen sind zulässig und werden unterstützt.

C++-Beispielprogramme erstellen

Native Anwendungen (unter Verwendung der nativen statischen Bibliothek MQMLIBN)

NMCPALL

Verwendung: NMCPALL

Mit diesem Makro werden alle nativen Beispielanwendungen unter Verwendung von NMCCPP kompiliert.

NMCCPP

Verwendung: NMCCPP *Name_der_Quellcodedatei*

Mit diesem Basismakro wird eine C++-Quelldatei unter Verwendung der Include-Dateien im Teildatenträger ZMQSLIB kompiliert. Beispielsweise wird die Beispielanwendung IMQSGETP unter Verwendung von NMCCPP IMQSGETP kompiliert. Bei einer erfolgreichen Kompilierung erstellt das Makro eine Objektdatei, wobei das letzte Zeichen des Dateinamens durch den Buchstaben O ersetzt wird (Beispiel: IMQSGETO).

NMLDCPP

Verwendung: NMLDCPP *Name_der_EXE-Datei*

Mit diesem Basismakro wird eine Objektdatei mit der statischen nativen MQI-Bibliothek MQMLIBN im Teildatenträger ZMQSLIB verbunden.

NMLDCPPA

Verwendung: NMLDCPPA

Mit diesem TACL-Makro werden die einzelnen Beispielobjektdateien unter Verwendung des Makros NMLDSAMP zu ausführbaren Dateien gebunden.

NMBLDCPP

Verwendung: NMBUILD

Mit diesem TACL-Makro werden alle nativen C++-Beispieldateien unter Verwendung der Makros NMCPPALL und NMLDCPPA kompiliert und gebunden.

Native Anwendungen (unter Verwendung der SRL MQSRLLIB)**NMLDCPPS**Verwendung: NMLDCPPS *Name_der_EXE-Datei*

Mit diesem Basismakro wird eine Objektdatei mit der nativen MQSeries-SRL MQSRLLIB im Teildatenträger ZMQSLIB verbunden.

NMLDCPSA

Verwendung: NMLDCPSA

Mit diesem TACL-Makro werden die einzelnen Beispielobjektdateien unter Verwendung des Makros NMLDCPPS zu ausführbaren Dateien gebunden.

NMBLDSCP

Verwendung: NMBLDSCP

Mit diesem TACL-Makro werden alle nativen C++-Beispieldateien unter Verwendung der Makros NMCPPALL und NMLDCPSA kompiliert und gebunden.

COBOL-Beispielprogramme erstellen**Nicht native Anwendungen (unter Verwendung der nicht nativen statischen Bibliothek MQMLIB)**

Der Teildatenträger ZMQSSMPL enthält die folgenden Dateien, die für die Erstellung von COBOL-Beispielanwendungen verwendet werden können:

COBSAMPVerwendung: COBSAMP *Name_der_Quellcodedatei*

Mit diesem Basismakro wird eine COBOL-Quellendatei unter Verwendung der Definitionsdateien im Teildatenträger ZMQSLIB kompiliert. Beispielsweise wird das Programm AMQ0GBR0 unter Verwendung von COBSAMP AMQ0GBR0 kompiliert. Bei einer erfolgreichen Kompilierung erstellt das Makro eine Objektdatei, wobei das letzte Zeichen des Dateinamens durch den Buchstaben O ersetzt wird (Beispiel: AMQ0GBRO).

BCOBSAMPVerwendung: BCOBSAMP *Name_der_EXE-Datei*

Mit diesem Basismakro wird ein Objekt in die Benutzerbibliotheken im Teildatenträger ZMQSLIB eingebunden. Beispielsweise wird die kompilierte Beispielanwendung AMQ0GBRO unter Verwendung von BCOBSAMP AMQ0GBR gebunden. Dieses Makro erstellt eine ausführbare Datei mit dem Namen *Name_der_EXE-Datei* AMQ0GBR.

CCBSMPLS

Verwendung: CCBSMPLS

Mit diesem TACL-Makro werden die einzelnen COBOL-Beispielquellcodedateien kompiliert.

COBOL-Beispiele erstellen

BCBSMPLS	Verwendung: BIND /IN BCBSMPLS/ Mit dieser Bindeeingabedatei werden die einzelnen COBOL-Beispielobjektdateien zu ausführbaren Dateien gebunden.
BUILDCOB	Verwendung: BUILDCOB Mit diesem TACL-Makro werden alle nativen COBOL-Beispieldateien unter Verwendung der Makros CCBSMPLS und BCBSMPLS kompiliert und gebunden.

Native Anwendungen (unter Verwendung der nativen statischen Bibliothek MQMLIBN)

NMCOBSMP	Verwendung: NMCOBSMP <i>Name_der_Quellcodedatei</i> Mit diesem Makro werden COBOL-Anwendungen im nativen Modus 'NMCOBOL' unter Verwendung der nativen MQSeries-Bibliothek MQMLIBN im Teildatenträger ZMQSLIB kompiliert. Beispielsweise wird das Programm AMQ0GBR0 unter Verwendung von NMCOBSMP AMQ0GBR0 kompiliert. Bei einer erfolgreichen Kompilierung erstellt das Makro eine Objektdatei, wobei das letzte Zeichen des Dateinamens durch den Buchstaben O ersetzt wird (Beispiel: AMQ0GBRO).
NMLDCOB	Verwendung: NMLDCOB <i>Name_der_EXE-Datei</i> Mit diesem Makro wird ein Objekt mit der nativen MQSeries-Bibliothek MQMLIBN im Teildatenträger ZMQSLIB verbunden. Beispielsweise wird die kompilierte Beispielanwendung AMQ0GBR0 unter Verwendung von NMLDCOB AMQ0GBR0 gebunden. Dieses Makro erstellt eine ausführbare Datei mit dem Namen <i>Name_der_EXE-Datei</i> AMQ0GBR.
NMCOBALL	Verwendung: NMCOBALL Mit diesem TACL-Makro werden die einzelnen COBOL-Beispielquellcodedateien unter Verwendung von NMCOBSMP kompiliert.
NMLDACOB	Verwendung: NMLDACOB Mit dieser Bindeeingabedatei werden die NMLDCOB-Beispielobjektdateien zu ausführbaren Dateien gebunden.
NMBLDCOB	Verwendung: NMBLDCOB Mit diesem TACL-Makro werden alle nativen COBOL-Beispieldateien unter Verwendung der Makros NMCOBALL und NMLDACOB kompiliert und gebunden.

Native Anwendungen (unter Verwendung der SRL MQSRLLIB)

NMCBSSMP

Verwendung: NMCBSSMP *Name_der_Quellcodedatei*

Mit diesem Makro werden COBOL-Anwendungen im nativen Modus 'NMCOBOL' unter Verwendung der MQSeries-SRL MQSRLLIB im Teildatenträger ZMQSLIB kompiliert. Beispielsweise wird das Programm AMQ0GBR0 unter Verwendung von NMCBSSMP AMQ0GBR0 kompiliert. Bei einer erfolgreichen Kompilierung erstellt das Makro eine Objektdatei, wobei das letzte Zeichen des Dateinamens durch den Buchstaben O ersetzt wird (Beispiel: AMQ0GBRO).

NMLDSCOB

Verwendung: NMLDSCOB *Name_der_EXE-Datei*

Mit diesem Makro wird ein Objekt mit der MQSeries-SRL MQSRLLIB im Teildatenträger ZMQSLIB verbunden. Beispielsweise wird die kompilierte Beispielanwendung AMQ0GBRO unter Verwendung von NMLDSCOB AMQ0GBR gebunden. Dieses Makro erstellt eine ausführbare Datei mit dem Namen *Name_der_EXE-Datei* AMQ0GBR.

NMCBSALL

Verwendung: NMCBSALL

Mit diesem TACL-Makro werden die einzelnen COBOL-Beispielquellcodedateien unter Verwendung von NMCBSSMP kompiliert.

NMLDSCOB

Verwendung: NMLDSCOB

Mit dieser Bindeeingabedatei werden die einzelnen NMLDSCOB-Beispielobjektdateien zu ausführbaren Dateien gebunden.

NMBLDSCB

Verwendung: NMBLDSCB

Mit diesem TACL-Makro werden alle nativen COBOL-Beispieldateien unter Verwendung der Makros NMCBSALL und NMLDSCOB kompiliert und gebunden.

TAL-Beispielprogramm erstellen

Nicht native Anwendungen (unter Verwendung der nicht nativen statischen Bibliothek MQMLIB)

Der Teildatenträger ZMQSSMPL enthält die folgenden Dateien, die für die Erstellung von TAL-Beispielanwendungen verwendet werden können:

TALSAMP	Verwendung: TALSAMP <i>Name_der_Quellcodedatei</i> . Mit diesem Basismakro wird eine TAL-Quelldatei unter Verwendung der Definitionsdateien im Teildatenträger ZMQSLIB kompiliert. Beispielsweise wird das Programm ZMQWRITT unter Verwendung von TALSAMP ZMQWRITT kompiliert. Bei einer erfolgreichen Kompilierung erstellt das Makro eine Objektdatei, wobei das letzte Zeichen des Dateinamens durch den Buchstaben O ersetzt wird (Beispiel: ZMQWRITO).
BTALSAMP	Verwendung: BTALSAMP <i>Name_der_EXE-Datei</i> Mit diesem Basismakro wird ein Objekt in die Benutzerbibliotheken im Teildatenträger ZMQSLIB eingebunden. Beispielsweise wird die kompilierte Beispielanwendung ZMQWRITO unter Verwendung von BTALSAMP ZMQWRIT gebunden.
CTLSMPLS	Verwendung: CTLSMPLS Mit diesem TACL-Makro werden die einzelnen TAL-Beispielquellcodedateien kompiliert.
BTLMSPLS	Verwendung: BIND /IN BTLMSPLS/ Mit dieser Bindeeingabedatei werden die TAL-Beispielobjektdateien zu ausführbaren Dateien gebunden.
BUILDTAL	Verwendung: BUILDTAL Mit diesem TACL-Makro werden alle TAL-Beispieldateien unter Verwendung der Makros CTLSMPLS und BTLMSPLS kompiliert und gebunden.

Beispielprogramme unter OSS erstellen (nur nativer Modus)

Das Verzeichnis /opt/mqm/samp enthält die Beispielprogramme für MQSeries sowie die Make-Datei MQMAKE.SMP. Diese Make-Datei enthält alle Ziele, die für die Erstellung aller C-, C++- und NMCOBOL-Beispiele erforderlich ist.

Das Verzeichnis /opt/mqm/inc enthält alle Copylibs und Header-Dateien, die für die Erstellung von Programmen unter OSS erforderlich sind.

Anmerkung: Die MQSeries-SRL-Datei und die native MQI-Bibliothek (MQMLIBN) sind nur im Guardian-Dateisystem vorhanden. Sie müssen gegebenenfalls Ihre Erstellungsscripts ändern, um auf diese Datei und diese Bibliothek zu verweisen, z. B. /G/system/zmqslib/mqsrllib.

Anhang L. Benutzer-Exits

MQSeries for Compaq NSK V5.1 unterstützt Exit-Programme für Kanäle, die Datenkonvertierung und für CLWL (Cluster Workload Management). Darüber hinaus wird noch ein Ladeprogramm-Exit von Compaq NSK unterstützt. Informationen zu Kanal-Exits finden Sie im Handbuch *MQSeries Intercommunication*. Informationen zu Datenkonvertierungs-Exits finden Sie in den Handbüchern *MQSeries Application Programming Guide* und *MQSeries Application Programming Reference*. Weitere Informationen zu Cluster Workload Management-Exits finden Sie im Handbuch *Cluster-Unterstützung in MQSeries*.

Die Informationen in diesem Anhang beziehen sich auf die Verwendung von Exit-Programmen in MQSeries for Compaq NSK.

Unterstützte Benutzer-Exits

In Tabelle 26 sind die Merkmale der verschiedenen Benutzer-Exits, die in MQSeries for Compaq NSK unterstützt werden, aufgeführt.

Tabelle 26. In MQSeries for Compaq NSK unterstützte Benutzer-Exits

Benutzer-Exit	Max. Länge des Exit-Namens	Max. Länge der Exit-Daten	Wo aktiviert?	Verkettet?
Kanalnachrichten-Exit	32	32	DEFINE CHANNEL	Ja
Kanalsende-Exit	32	32	DEFINE CHANNEL	Ja
Kanalempfangs-Exit	32	32	DEFINE CHANNEL	Ja
Kanalsicherheits-Exit	32	32	DEFINE CHANNEL	
Kanal-Exit für Nachrichtenwiederholung	32	32	DEFINE CHANNEL	
Exit für automatische Kanaldefinition	32	32	ALTER QMGR	
Cluster Workload Management-Exit	32	32	ALTER QMGR	
Datenkonvertierungs-Exit	8	Nicht zutreffend	Name mit unbekanntem Format	
MQ_LOAD_ENTRY_POINT_EXIT	Fest zugewiesener Name	Nicht zutreffend	Wird jedesmal aufgerufen, wenn einer der oben aufgeführten Exits erforderlich ist bzw. aktiviert wird.	

Unterstützte Exit-Programme

Nur bei MQ_LOAD_ENTRY_POINT_EXIT handelt es sich um einen Compaq NSK-spezifischen Exit.

Format für Exit-Namen

Exit-Namen (mit Ausnahme von MQ_LOAD_ENTRY_POINT_EXIT) können aus einer alphanumerischen Zeichenfolge mit bis zu 32 Zeichen bestehen. Exits, die ein zugeordnetes Datenfeld unterstützen, unterstützen Daten mit einer Länge von bis zu 32 Zeichen.

MQ_LOAD_ENTRY_POINT_EXIT (Benutzer-Exits laden)

Alle Benutzer-Exit-Programme müssen in die private MQSeries-SRL oder in die statische Bibliothek eingebunden werden. Die Exit-Programme müssen mindestens eine externe Funktion (Symbol) enthalten, die bei Bedarf von MQSeries aufgerufen werden kann.

Vor der Aktivierung eines MQSeries-Exits müssen Sie das Programm MQ_LOAD_ENTRY_POINT_EXIT installieren, um die Exit-Namen Eingangspunktadressen zuzuordnen. Dieses Programm muss in die MQSeries-SRL oder in die statische Bibliothek eingebunden werden und wird von MQSeries jedes Mal aufgerufen, wenn einer der anderen Benutzer-Exits aktiviert wird.

Der Name des Programms MQ_LOAD_ENTRY_POINT_EXIT ist fest vergeben, d. h. der Name der externen Funktion (MQ_LOAD_ENTRY_POINT) darf nicht geändert werden.

Das Exit-Programm MQ_LOAD_ENTRY_POINT_EXIT kann einen Exit-Namen einer beliebigen Eingangspunktadresse oder mehrere Exit-Namen ein und derselben Eingangspunktadresse zuordnen.

MQSeries stellt die Stub-Funktion MQ_LOAD_ENTRY_POINT_EXIT zur Verfügung, die beim Aufruf immer MQXCC_FAILED (Exit-Name nicht gefunden) zurückgibt. Sie müssen vor der Aktivierung Ihrer Benutzer-Exits zunächst diesen Stub-Exit durch Ihren eigenen ersetzen.

MQ_LOAD_ENTRY_POINT_EXIT (*Exit-Parameter*)

Parameter:

Exit-Parameter (PMQLXP) – input/output
LoadExit Parameter Block

Verwendungshinweise: Die Funktion, die vom Programm MQ_LOAD_ENTRY_POINT_EXIT ausgeführt wird, wird von dem Ersteller des Exits definiert.

In Abb. 46 auf Seite 377 ist ein Beispiel für MQ_LOAD_ENTRY_POINT_EXIT zu sehen; dieses Programm ordnet drei Exit-Namen Eingangspunktadressen zu.

MQLXP (Parameterstruktur von MQ_LOAD_ENTRY_POINT_EXIT)

Die MQLXP-Struktur beschreibt die Informationen, die an den Lade-Exit übergeben werden.

Diese Struktur wird nur für Compaq NSK unterstützt.

Felder

StrucId (MQCHAR4)

Struktur-ID.

Wert: MQLXP_STRUC_ID.

Gibt die ID für die Parameterstruktur des Lade-Exits an.

Für die Programmiersprache C wird auch die Konstante MQLXP_STRUC_ID_ARRAY definiert. Sie hat denselben Wert wie MQLXP_STRUC_ID, nur handelt es sich hier nicht um eine Zeichenfolge, sondern um einen Zeichenbereich.

Dies ist ein Eingabefeld für den Exit.

Version (MQLONG)

Versionsnummer der Struktur.

Wert: MQLXP_VERSION_1

Parameterstruktur des Lade-Exits, Version 1.

Die folgenden Konstanten geben die Versionsnummer der aktuellen Version an: MQLXP_CURRENT_VERSION

Parameterstruktur des Lade-Exits, aktuelle Version.

Dies ist ein Eingabefeld für den Exit.

QMgrName (MQCHAR48)

Name des lokalen WS-Managers.

Hierbei handelt es sich um den Namen des lokalen WS-Managers, der den Lade-Exit aufgerufen hat. Der Name wird mit Leerzeichen auf die Gesamtlänge des Feldes aufgefüllt.

Dies ist ein Eingabefeld für den Exit.

EntryPointName (MQCHAR32)

Name des angeforderten Eingangspunktes.

Dabei handelt es sich um den Namen des Eingangspunktes, den der Lade-Exit zum Auflösen einer aufrufbaren Adresse benötigt. Der Name wird mit Leerzeichen auf die Gesamtlänge des Feldes aufgefüllt.

Dies ist ein Eingabefeld für den Exit.

EntryAddress (PMQFUNC)

Zurückgegebene aufrufbare Adresse.

Dabei handelt es sich um die Adresse des angeforderten Eingangspunktes.

Dies ist ein Ausgangsfeld für den Exit.

Unterstützte Exit-Programme

ExitResponse (MQLONG)

Antwort vom Exit.

Wird vom Exit gesetzt, um anzugeben, ob der Name des Eingangspunktes erfolgreich in eine aufrufbare Adresse aufgelöst werden konnte. Hier wird einer der folgenden Werte angegeben:

MQXCC_OK
Success.

Gibt an, dass der Exit den in den Exit-Parametern übergebenen Namen des Eingangspunktes erfolgreich in eine aufrufbare Adresse auflösen konnte. Die aufrufbare Adresse wird im Feld **EntryAddress** der MQLXP-Struktur zurückgegeben.

MQXCC_FAILED
Failed.

Gibt an, dass der Exit den in den Exit-Parametern übergebenen Namen des Eingangspunktes nicht in eine aufrufbare Adresse auflösen konnte.

Alle im Feld **ExitResponse** angegebenen Werte, bei denen es sich nicht um MQXCC_OK oder MQXCC_FAILED handelt, werden wie MQXCC_FAILED interpretiert.

Dies ist eine Ausgangsfeld für den Exit.

MQ_LOAD_ENTRY_POINT_EXIT - Beispiel

In Abb. 46 auf Seite 377 sehen Sie ein Beispiel für ein ausführbares MQ_LOAD_ENTRY_POINT_EXIT-Programm, das drei Exit-Namen (zwei Kanal-Exits und einen Datenkonvertierungs-Exit) Eingangspunktadressen zuordnet. Der Quellcode für dieses Beispiel für MQ_LOAD_ENTRY_POINT_EXIT steht im Teildatenträger mit den Beispielen (AMQSLXP0) zur Verfügung.

```

/*****/
/*
/* 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
/*****/

```

Abbildung 46. MQLOADEXIT - Beispiel

Unterstützte Exit-Programme

Mit diesem Beispiel für MQ_LOAD_ENTRY_POINT_EXIT und den installierten Kanal- und Datenkonvertierungs-Exits können Sie Ihren Kanalempfangs- und Kanalsende-Exit mit dem folgenden MQSC-Befehl aktivieren:

```
ALTER CHANNEL(CHAN) CHLTYPE(SDR) SENDEXIT(MY_CHANNEL_SEND_EXIT)
ALTER CHANNEL(CHAN) CHLTYPE(SDR) RCVEXIT(MY_CHANNEL_RCV_EXIT)
```

Der Datenkonvertierungs-Exit wird von MQSeries bei einem MQGET-Aufruf mit aktivierter Konvertierungsoption (MQGMO_CONVERT) und dem Nachrichtenformatnamen MY_FORMAT aufgerufen.

Benutzer-Exits installieren

Alle von Ihnen erstellten Benutzer-Exits müssen in der privaten SRL von MQSeries installiert werden. Darüber hinaus müssen einige Exit-Arten in allen statischen MQSeries-Bibliotheken, die von Anwendungsprogrammen verwendet werden, installiert werden.

Benutzer-Exit	Wird installiert in
Kanalnachrichten-Exit	Private SRL
Kanalsende-Exit	Private SRL
Kanalempfangs-Exit	Private SRL
Kanalsicherheits-Exit	Private SRL
Kanal-Exit für Nachrichtenwiederholung	Private SRL
Exit für automatische Kanaldefinition	Private SRL
Cluster Workload Management-Exit	Private SRL
Datenkonvertierungs-Exit	Private SRL (außerdem in der nativen statischen Bibliothek und/oder in der nicht nativen statischen Bibliothek, falls diese von den Anwendungen verwendet werden)
MQ_LOAD_ENTRY_POINT_EXIT	Private SRL (außerdem in der nativen statischen Bibliothek und/oder in der nicht nativen statischen Bibliothek, falls diese von den Anwendungen verwendet werden)

Exit in der privaten SRL von MQSeries installieren

Um ein Exit-Programm in der privaten SRL von MQSeries zu installieren, müssen Sie eine neue Version dieser privaten MQSeries-SRL, die den neuen Exit enthält, erstellen:

1. Kompilieren Sie die Exit-Funktion. Beispiel:

```
NMCP SRL AMQSVFCN
```

Unterstützte Exit-Programme

2. Kompilieren Sie die MQLOAD-Eingangspunktfunktion. Beispiel:

```
NMCPSRL AMQSLXP0
```

3. Verbinden Sie den Exit und die Eingangspunktobjekte zu einer erneut verbindbaren (relinkable) Bibliothek für die SRL. Beispiel:

```
NMLDUSRL OBJECTS EXITS
```

4. Erstellen Sie eine neue Version der privaten MQSeries-SRL mit dem Namen NEWMQSRL, indem Sie dieses Datenkonvertierungsobjekt mit der erneut verbindbaren Version der privaten MQSeries-SRL verbinden. Beispiel:

```
MAKEPSRL EXITS $VOL.ZMQSLIB NEWMQSRL
```

5. Stoppen Sie alle WS-Manager und Anwendungen, die die aktuelle private SRL von MQSeries verwenden.
6. Verbinden Sie alle MQSeries-Anwendungen erneut mit der neuen PSRL. Beispiel:

```
NMLDEXES $VOL.ZMQSLIB.NEWMQSRL $VOL.ZMQSEXE
```

7. Kompilieren Sie die GET-Anwendung. Beispiel:

```
NMCSAMP AMQSGET0
```

8. Verbinden Sie alle Anwendungen erneut mit der neuen PSRL. Beispiel:

```
AMQSGET NMLDPSRL AMQSGET
```

9. Starten Sie MQSeries und alle MQI-Anwendungen erneut.

Die Schritte 6 und 8 sind schnell ausgeführt, können jedoch ausgelassen werden, wenn die neue MQSeries-PSRL in den Teildatenträger ZMQSLIB gestellt und MQSRLLIB genannt wird. Diese Schritte können wiederholt werden, um eine andere MQSeries-PSRL zu verbinden.

Exit in der nativen statischen MQSeries-Bibliothek installieren

Ein Exit kann mit der gewünschten Anwendung und der MQI-Bibliothek unter Verwendung des TACL-Makros NMLDEXIT verbunden werden. Beispiel:

```
NMLDEXIT Objektdatei Exit-Objektdatei
```

Unterstützte Exit-Programme

Exit in der nicht nativen statischen MQSeries-Bibliothek installieren

Ein Exit kann in die gewünschte ausführbare Datei (oder Bibliothek) unter Verwendung des TACL-Makros BEXITE gebunden werden.

Anmerkung: Bei diesem Verfahren wird die ausführbare Zielfdatei verändert; daher wird empfohlen, vor Verwendung des Makros eine Sicherungskopie der ausführbaren Zielfdatei bzw. der Zielbibliothek zu erstellen.

Exit-Funktionen müssen nach dem Kompilieren direkt in die ausführbare Zielfdatei bzw. die Zielbibliothek gebunden werden, damit sie von MQSeries aufgerufen werden können. Hierzu wird das TACL-Makro BEXITE verwendet. Beispiel:

```
BEXITE
Ausführbare_Zielfdatei_ODER_Zielbibliothek Quellen_Exit-Datei_ODER_Quellenbibliothek
```

So binden Sie beispielsweise den Beispieldatenkonvertierungs-Exit in das Beispiel MQSGETA ein:

1. Kompilieren Sie die Exit-Funktion. Beispiel:

```
CSAMP AMQSVFCN
```

2. Kompilieren Sie die MQLOAD-Eingangspunktfunktion. Beispiel:

```
CSAMP AMQSLXP0
```

3. Kompilieren Sie die GET-Anwendung. Beispiel:

```
CSAMP AMQSGET0
```

4. Verbinden Sie die GET-Anwendung. Beispiel:

```
BSAMP AMQSGET
```

5. Verbinden Sie die Exit-Funktion mit der GET-Anwendung. Beispiel:

```
BEXITE AMQSGET AMQSVFCO
```

6. Verbinden Sie die Eingangspunktfunktion mit der GET-Anwendung. Beispiel:

```
BEXITE AMQSGET AMQSLXP0
```

Unterstützte Exit-Programme

Soll dieser Datenkonvertierungs-Exit allen Anwendungen zur Verfügung stehen, können Sie auch mit den folgenden Schritten eine Benutzerbibliothek und eine Anwendung mit dem eingebundenen Exit erstellen:

1. Kompilieren Sie die Exit-Funktion. Beispiel:

```
CSAMP AMQSVFCN
```

2. Kompilieren Sie die MQLOAD-Eingangspunktfunktion. Beispiel:

```
CSAMP AMQSLXP0
```

3. Kompilieren Sie die GET-Anwendung. Beispiel:

```
CSAMP AMQSGET0
```

4. Verbinden Sie die Exit-Funktion mit der Benutzerbibliothek. Beispiel:

```
BEXITE ZMQSLIB.MQMLIB AMQSVFCO
```

5. Verbinden Sie die Exit-Funktion mit der Benutzerbibliothek. Beispiel:

```
BEXITE ZMQSLIB.MQMLIB AMQSLXP0
```

6. Verbinden Sie die GET-Anwendung mit der geänderten Bibliothek. Beispiel:

```
BSAMP AMQSGET
```

Unterstützte Exit-Programme

Anhang M. Kommunikation einrichten

In diesem Anhang wird erläutert, wie das Kommunikationssystem in MQSeries for Compaq NSK unter Verwendung der SNA- und TCP/IP-Übertragungsprotokolle eingerichtet wird. Dieser Anhang enthält die folgenden Beispiele:

- „SNAX-Kommunikationskonfiguration - Beispiel“ auf Seite 390
- „ICE-Kommunikationskonfiguration - Beispiel“ auf Seite 396
- „TCP/IP-Kommunikationskonfiguration - Beispiel“ auf Seite 399

SNA-Kanäle

Die folgenden Kanalattribute sind für SNA-Kanäle in MQSeries for Compaq NSK V5.1 erforderlich:

CONNNAME

Der Wert von CONNNAME hängt davon ab, ob SNAX oder ICE als Übertragungsprotokoll verwendet wird:

Bei Verwendung von SNAX:

CONNNAME('\$PPPP.LOCALLU.REMOTELU')

Für Sender-, Requester und vollständig qualifizierte Serverkanäle; dabei gilt Folgendes:

\$PPPP Prozessname des SNAX/APC-Prozesses.
LOCALLU Name der lokalen LU.
REMOTELU Name der Partner-LU auf dem fernen System.

Beispiel:

```
CONNNAME('$BP01.IYAHT080.IYCNVM03')
```

Bei Verwendung von ICE:

CONNNAME('\$PPPP.#OPEN.LOCALLU.REMOTELU')

Für Sender-, Requester und vollständig qualifizierte Serverkanäle; dabei gilt Folgendes:

\$PPPP Prozessname des ICE-Prozesses.
#OPEN Name des geöffneten ICE-Prozesses.
LOCALLU Name der lokalen LU.
REMOTELU Name der Partner-LU auf dem fernen System.

Beispiel:

```
CONNNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+)
```

MODENAME

SNA-Modusname. Beispiel: MODENAME(LU62PS).

Kommunikation einrichten

TPNAME('LOCALTP[.REMOTETP]')

TP-Name (Transaktionsprozess).

- LOCALTP** Name der Serverklasse (in der Regel MQSeries) in der für die SNA-Kommunikation verwendeten PATHWAY-Konfiguration. Bei SNAX muss die Serverklasse sich in derselben PATHWAY-Umgebung wie der SNAX-Dispatcher und die Serverklassen des APC-Prozesses befinden. Bei Verwendung von ICE muss sie sich in der im Befehl NOF - ADD TP deklarierten PATHWAY-Umgebung befinden.
- REMOTETP** Name des Transaktionsprozesses auf dem fernen System. Diese Angabe ist optional. Erfolgt keine Angabe und kann der Kanal einen Datenaustausch einleiten (d. h., wenn es sich um einen Sender-, Requester- oder um einen vollständig qualifizierten Serverkanal handelt), wird der Name LOCALTP übernommen.

Für LOCALTP und REMOTETP können Angaben mit bis zu 16 Zeichen erfolgen.

LU6.2-Responder-Prozesse

In MQSeries for Compaq NSK V5.1 wird ein SNA-Empfangsprogrammprozess benötigt, der für ankommende Verbindungsanforderungen von fernen WS-Managerkanälen empfangsbereit ist.

Die MQSeries-SNA-Empfangsprogramme müssen vom SNA-Produkt zugeteilt werden, wenn eine ankommende Verbindungsanforderung von einem fernen Kanal eintrifft.

Bei Verwendung des SNAX-APC-Dispatchers kann SNAX ankommende Verbindungsanforderungen von Transaktionsprogrammen auf der Partnerseite akzeptieren. Hierzu ist für den SNAX-Dispatcher Folgendes erforderlich:

- Er darf nicht in derselben PATHWAY-Umgebung wie der WS-Manager laufen.
- Der APC-Prozess muss als Serverklasse in derselben PATHWAY-Umgebung ausgeführt werden.

Bei Verwendung von Insession ICE ist Folgendes erforderlich:

- In einer PATHWAY-Umgebung muss ein Transaktionsprogramm (Serverklasse) definiert werden.
- In NOF muss ein Transaktionsprogramm hinzugefügt werden, das auf diese Serverklasse verweist.

Es gibt keinen eigenen ICE-Dispatcher; die ankommenden Verbindungsanforderungen werden vom ICE-Prozess selbst bearbeitet.

SNAX - TP-Zuteilung

Das Compaq-SNAX-SNA-Produkt unterstützt die Aktivierung von APPC-Transaktionsprogrammen, wenn eine APPC-Verbindungsanforderung von einem Transaktionsprogramm auf dem Partnersystem eintrifft. Über eine lokale Transaktionsprogramm-Serverklasse teilt ein SNAX-Dispatcher diese Anforderungen dem zugeordneten SNAX-Prozess \$APC zu.

Für den SNAX-Dispatcher für MQSeries gelten folgende Voraussetzungen:

- Der SNAX-Dispatcher muss in derselben PATHWAY-Umgebung wie der zugeordnete \$APC-Prozess ausgeführt werden.

- Jeder ankommende TPNAME muss als eine Serverklasse (in der Regel MQSeries) in derselben PATHWAY-Umgebung wie der Dispatcher und der \$APC-Prozess definiert sein.
- Der Programmname der Serverklasse lautet **runmqlsr**; das Programm befindet sich im Teildatenträger mit den ausführbaren MQSeries-Modulen (in der Regel ZMQSEXE).
- Der Dispatcher-Prozess startet die Serverklasse und übergibt alle wichtigen Informationen (\$APC-Prozess, Name der LU, Name des Transaktionsprozesses) an diese Serverklasse (TP) in einer IPC-Anforderung DISPATCH-TP.

ICE - TP-Zuteilung

ICE Version 3.2 implementiert den SNA-Anschlussmanager auf ähnliche Weise wie SNAX, da es sich bei dem Transaktionsprozess um eine PATHWAY-Serverklasse handelt. Der ICE-Prozess akzeptiert eine Verbindungsanforderung und übernimmt selbst die Funktion eines Dispatchers. Dieser ICE-Prozess muss jedoch nicht in derselben PATHWAY-Umgebung wie der Transaktionsprozess (Serverklasse) ausgeführt werden. In diesem Fall wird für die Umgebung Folgendes vorausgesetzt:

- Ein ICE-Prozess muss aktiv sein.
- In NOF muss ein Zuteilungstransaktionsprozess hinzugefügt werden. Beispiele:

```
ADD TP <TP-Name>, PROCESS <$Prozess>, SERVERCLASS <Serverklassenname>
```

Bei *Serverklassenname* handelt es sich in der Regel um MQSeries.

- Jeder ankommende TPNAME muss als eine Serverklasse (in der Regel MQSERIES) im PATHWAY<\$Prozess> definiert sein.
- Der Programmname der Serverklasse lautet RUNMQLSR; das Programm befindet sich im Teildatenträger mit den ausführbaren MQSeries-Modulen (in der Regel ZMQSEXE).
- Der ICE-Prozess startet die Serverklasse und übergibt alle wichtigen Informationen (\$ICE-Prozess, Name der Anwendung, Name des Transaktionsprozesses) an diese Serverklasse (TP) in einer IPC-Anforderung DISPATCH-TP.

Beispielkonfiguration für SNA-Umgebung

Im folgenden einige Beispiele für die Konfiguration Ihrer SNA-Umgebung.

SNAX APC verwenden: Bei Verwendung von SNAX APC gilt Folgendes:

- Es muss eine PATHWAY-Umgebung erstellt werden, die ausschließlich von diesem Empfangsprogramm verwendet wird.
- Eine APC-Prozessserverklasse muss in dieser PATHWAY-Umgebung ausgeführt werden.

Geben Sie an einer TACL-Eingabeaufforderung den folgenden Befehl ein:

```
TACL> Pathmon /name $PMAP, nowait, out $vhs, cpu 3/4
TACL> Pathcom $PMAP
= 0 LU62SCFG
```

Kommunikation einrichten

Dabei steht LU62SCFG für eine Edit-Datei mit dem folgenden Inhalt:

```
[ 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
```

Abbildung 47. MQSeries-SNAX-Konfigurationsdatei - Beispiel (Teile- 1 von 3)

```
[Configure the 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
```

Abbildung 47. MQSeries-SNAX-Konfigurationsdatei - Beispiel (Teile- 2 von 3)

```
[Add MQSeries SNAX Listener]
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 *

[Configure the DISPATCHER]
SET TERM FILE $s.#displg
SET TERM INITIAL SNAXAPC-DISPATCHER
SET TERM TYPE CONVERSATIONAL
SET TERM TCP SNAXAPC-TCP
ADD TERM SNAXAPCSVR01 [First 10 chars are the SNAX/APC server name]

start server MQSERIES
start server SNAXAPCSVR
start term SNAXAPCSVR01
```

Abbildung 47. MQSeries-SNAX-Konfigurationsdatei - Beispiel (Teile- 3 von 3)

Anmerkung: Die Serverklasse des Empfangsprogramms entspricht der Serverklasse MQS-TCPLIS00 in der PATHWAY-Umgebung des WS-Managers, allerdings enthält sie noch einen zusätzlichen Startparameter (SET SERVER STARTUP "-t LU62").

Insession ICE verwenden

Bei Verwendung von Insession ICE muss eine PATHWAY-Umgebung erstellt werden, die ausschließlich von diesem Empfangsprogramm verwendet wird. Der ICE-Prozess wird nicht von dieser PATHWAY-Umgebung ausgeführt.

1. Fügen Sie in NOF den Transaktionsprozess wie folgt hinzu:

```
ADD TP <TP-Name>, PROCESS <Prozess>, SERVERCLASS
<Server> [, <Option> ...]
```

Dabei gilt:

Prozess	Name des PATHMON-Prozesses, der den Transaktionsprozess verwaltet.
Server	Name der PATHMON-Serverklasse, zu der der Transaktionsprozess gehört.

Kommunikation einrichten

Option

Hier kann Folgendes angegeben werden:

- [ATTACHTIMER n] -- Gibt an (in Hundertstelsekunden), wie lange ICE nach der Zuteilung eines neuen TP-Threads auf eine Verbindungsanforderung (ATTACH) wartet. Der Standardwert ist 6000 (60 Sekunden).
- [MAXDISPATCHTHREADS n] -- Gibt die maximale Anzahl an TPs an, die gleichzeitig zugeteilt werden können. DEFAULT = Die Anzahl der TPs, die gleichzeitig zugeteilt werden können, ist nicht begrenzt.
- [TIMEOUT n] -- Gibt an, wie ICE auf eine Verbindungsanforderung (ATTACH) reagiert, wenn MAXDISPATCHTHREADS für einen Transaktionsprozess den Höchstwert erreicht; TIMEOUT -1 = Die Verbindungsanforderung (ATTACH) wird unbegrenzt eingereicht; TIMEOUT >0 = Die Verbindungsanforderung (ATTACH) wird für n-hundertstel Sekunden eingereicht; DEFAULT: 0 = Die Verbindungsanforderung (ATTACH) wird sofort zurückgewiesen.

2. Die Serverklasse muss auf jeden Fall der PATHWAY-Umgebung hinzugefügt werden. Geben Sie an der TACL-Eingabeaufforderung den folgenden Befehl ein:

```
TACL> Pathmon /name $PMAP, nowait, out $vhs, cpu 3/4
TACL> Pathcom $PMAP
= 0 LU62ICFG
```

Dabei steht LU62ICFG für eine Edit-Datei mit dem folgenden Inhalt:

```
[
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
```

Abbildung 48. MQSeries-SNA-Konfigurationsdatei für ICE - Beispiel (Teile- 1 von 2)

```
[Add MQSeries ICE Listener]
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
```

Abbildung 48. MQSeries-SNA-Konfigurationsdatei für ICE - Beispiel (Teile- 2 von 2)

Anmerkung: Die Serverklasse des Empfangsprogramms entspricht der Serverklasse MQS-TCPLIS00 in der PATHWAY-Umgebung des WS-Managers, allerdings enthält sie noch einen zusätzlichen Startparameter (SET SERVER STARTUP "-t LU62").

TCP/IP-Kanäle

In MQSeries for Compaq NSK haben Sie die Möglichkeit, mehrere TCP/IP-Prozesse innerhalb einer MQSeries-WS-Managerumgebung zu verwenden. Sie können also TCP/IP-Prozesse, die in einem WS-Manager verwendet werden, auswählen, indem Sie den erforderlichen TCP/IP-Prozess einem bestimmten Kanal zuordnen. Für abgehende Kanäle (Sende-, Server- und Requester-Kanäle) kann der Name des erforderlichen TCP/IP-Prozesses im Feld CONNAME der Kanaldefinition angegeben werden.

Bei Verwendung des Steuerbefehls **runmqsc** wird Folgendes eingegeben:

```
alter channel ... conname ('$ZTC1.123.456.789.012(1415)')
alter channel ... conname ('$ZTC1.dnshostname(1415)')
```

Bei Verwendung der MQMC-Anzeigen wird Folgendes angegeben:

```
TCPIP/SNA Process: $ZTC1
```

Bei Verwendung von PCF-Befehlen wird Folgendes eingegeben:

```
strncpy( pPCFString->String, ('$ZTC1.123.456.789.012(1415)', len );
```

Kommunikation einrichten

Soll für die DNS-Auflösung nicht der Standard-Resolver verwendet werden, muss in allen PATHWAY-Serverklassen des Typs ECnn Folgendes hinzugefügt werden:

```
DEFINE =TCPIP^RESOLVER^NAME, FILE filename
```

Dabei steht `filename` für den Verzeichnispfad der Resolver-Datei.

Soll eine Host-Datei verwendet werden, muss in allen PATHWAY-Serverklassen des Typs ECnn Folgendes hinzugefügt werden:

```
DEFINE =TCPIP^HOST^FILE, FILE filename
```

Dabei steht `filename` für den Verzeichnispfad der Host-Datei.

Ankommende Kanäle ermitteln den TCP/IP-Prozess, der verwendet werden soll, anhand von Umgebungsvariablen. Die TCP/IP-Empfangsprogramme übergeben mit Hilfe ihrer Agenten diesen Prozessnamen den ihnen zugeordneten Ausführungs-Controllern und TCP/IP-Responder-Prozessen. So legen Sie den TCP/IP-Prozess für einen ankommenden Kanal fest:

Bei Verwendung von TACL:

```
ADD DEFINE =TCPIP^PROCESS^NAME, FILE processname
```

Dabei steht `processname` für den Namen des TCP/IP-Prozesses.

Bei Verwendung von PATHWAY für MQS-TCPLIS nn -Serverklassen (dabei steht nn für die Nummer der Empfangsprogramm-Serverklasse:

```
DEFINE =TCPIP^PROCESS^NAME, FILE \HAWK.$ZTC1  
PARAM MQLISTENPORTNUM "1415"
```

Informationen zur Verwendung eines anderen als des standardmäßigen TCP/IP-Prozesses für eine Übertragung über TCP/IP finden Sie unter „Serverklasse MQS-TCPLIS nn für die Verwendung eines anderen TCP/IP-Prozesses und Ports rekonfigurieren“ auf Seite 58. Informationen über die TCP/IP-Ports, an denen ein WS-Manager empfangsbereit ist, finden Sie unter „Anzahl an TCP/IP-Ports, die vom WS-Manager überwacht werden“ auf Seite 57.

Kommunikationskonfiguration - Beispiele

Dieser Abschnitt enthält einige Beispiele für Kommunikationskonfigurationen für SNA (SNAX und ICE) und TCP/IP.

SNAX-Kommunikationskonfiguration - Beispiel

Dieser Abschnitt enthält folgende Beispiele:

- SCF-Beispielkonfigurationsdatei für SNA-Verbindung
- Einige SYSGEN-Beispielparameter für die Unterstützung dieser Verbindung

- Eine SCF-Beispielkonfigurationsdatei für die SNA-Prozessdefinition
- Einige MQSC-Beispielkanaldefinitionen

SCF-Konfigurationsdatei für SNA-Verbindung

Hier eine SCF-Beispielkonfigurationsdatei:

```
==
== 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

==
== 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
```

Kommunikation einrichten

```
== MVS Non-CICS
ADD LU #PU2LU5, TYPE(14,21), PUNAME #PU2, SNANAME IYCNVM03
== finnr100 (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
```

SYSGEN-Parameter

Im Folgenden sehen Sie CONFTEXT-Dateieinträge für eine SYSGEN zur Unterstützung der SNA- und Token-Ring-Verbindungen:

```
!*****
!                               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
```

Kommunikation einrichten

```
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;
```

SNAX/APC-Prozesskonfiguration

Mit den folgenden Definitionen wird für die SNA-Verbindung ein APC-Beispielprozess (Prozessname \$BP01) über SCF erstellt.

Anmerkung: Der APC-Prozess \$BP01 ist als Serverklassenprozess definiert, der in derselben PATHWAY-Umgebung wie der SNAX-APC-Dispatcher aufgeführt wird.

```
==
== SCF Configuration file for SNAX/APC Lus
==

ALLOW ERRORS

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, SNAME 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, SNAME GBIBMIYA.IYABT0F0, &
PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES
```

Kommunikation einrichten

```
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, &
    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 *

```

Für MQSeries-Anwendungen muss die Umgebungsvariable Maxapplio angegeben werden, die das maximale Datenvolumen, das bei der Kommunikation zwischen MQSeries und den Übertragungsserverprozessen ausgetauscht werden kann, auf 32000 festsetzt; dieser Wert liegt über dem Standardwert.

Kanaldefinitionen

Hier einige MQSeries-Beispielkanaldefinitionen, die die SNAX-Konfiguration unterstützen:

- Ein Senderkanal zu MQSeries unter MVS/ESA (für die Kommunikation zwischen den Systemen wird nicht die CICS-Kommunikation verwendet):

```

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)

```

Kommunikation einrichten

- Ein Empfängerkanal von MQSeries zu MVS/ESA:

```
DEFINE CHANNEL(VM03.MT01.SDR0.0002) CHLTYPE(RCVR) +
  TRPTYPE(LU62) REPLACE DESCR('Receiver channel from VM03NCM') +
  SEQWRAP(9999999) +
  MAXMSGL(2048)
```

- Ein Serverkanal zu MQSeries unter MVS/ESA, der selbst einen Datenaustausch einleiten oder von einem fernen Requester-Kanal gestartet werden kann:

```
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)
```

Dabei steht MQSeries für den Namen des Transaktionsprozesses (TPNAME), für den der MVS™-WS-Manager empfangsbereit ist.

ICE-Kommunikationskonfiguration - Beispiel

Die ICE-Konfiguration für MQSeries erfolgt in zwei Schritten:

1. Der ICE-Prozess selbst muss konfiguriert werden.
2. Informationen zur Verbindung (in diesem Beispiel \$ICE01) und zu SNA müssen als Eingabe an den ICE-Prozess übergeben werden.

ICE-Prozess konfigurieren

Im Folgenden sehen Sie ein Beispiel für eine ICE-Prozesskonfiguration. Sie befindet sich standardmäßig in einer Datei mit dem Namen **GOICE**:

```
?tacl macro
clear all
param backupcpu 1
param cinittimer 120
param collector $0
param config icectl
param idblk 05d
param idnum 312FF
param cname IYHR00C
param datapages 64
param dynamicrlu yes
param genesis $gen
param maxrcv 32000
param loglevel info
param netname GBIBMIYA
param password xxxxxxxxxxxxxxxxxxxxxx
param retrys1 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/
```

Anmerkungen:

1. Die Kennwortumgebungsvariable wurde durch xxxxxxxxxxxxxxxxxxxxxx ersetzt.
2. Für MQSeries-Anwendungen muss die Umgebungsvariable maxrcv angegeben werden, die das maximale Datenvolumen, das bei der Kommunikation zwischen MQSeries und den Übertragungsserverprozessen ausgetauscht werden kann, auf 32000 festsetzt; dieser Wert liegt über dem Standardwert.

Verbindungs- und APC-Informationen definieren

Nach dem Start des ICE-Prozesses mit dieser Konfiguration werden die folgenden Informationen über NOF** (Node Operator Facility) als Eingabe an den ICE-Prozess übergeben. In diesem Beispiel wird die Verbindung \$ICE01 für den Token-Ring-Port \$CHAMB.#ICE definiert:

```

==
== ICE definitions for PU IYHR00C.
== Local LU for this PU is IYHT0C0.
==

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
==

ADD PU IYHR00C, 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 IYHT0C0
ADD APPL IYHT0C0, ALIAS IYHT0C0, PROTOCOL CPIC, &
      OPENNAME #IYHT0C

==
== Add Mode LU62PS
==

DELETE MODE LU62PS
ADD MODE LU62PS, MAXSESS 8, MINCONWIN 4, MINCONLOS 3

==
== Add Partner LU Objects
==

== spinach (HP)

ABORT RLU IYHT0F0

```

Kommunikation einrichten

```
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
DELETE RLU IYAHT080
ADD RLU IYAHT080, MODE LU62PS, PARSESS YES

==
== START UP ICE LINE $ICE01 AND SUB DEVICE
==

START LINE $ICE01, SUB ALL
```

Anmerkung: Damit diese Konfiguration verwendet werden kann, muss der Port #ICE für die Token-Ring-Verbindung definiert werden.

Hierfür können beispielsweise die folgenden beiden Befehle in SCF eingegeben werden:

```
add port $chamb.#ice, type tr8025, address %H08
start port $chamb.#ice
```

Dabei steht **\$chamb** für einen Token-Ring-Controller; der SAP des Ports ist %08.

Kanaldefinitionen für ICE

Hier einige MQSeries-Kanaldefinitionen, die diese ICE-Konfiguration unterstützen:

- Ein Senderkanal zu MQSeries unter MVS/ESA (für die Kommunikation zwischen den Systemen wird nicht die CICS-Kommunikation verwendet):

```
DEFINE CHANNEL(MT01.VM03.SDR.CE) CHLTYPE(SDR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.SDR.CE') +
  CONNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+
  MODENAME('LU62PS') TPNAME(MQSERIES)
```

- Ein Empfängerkanal von MQSeries zu MVS/ESA:

```
DEFINE CHANNEL(VM03.MT01.SDR.CE) CHLTYPE(RCVR) +
  TRPTYPE(LU62) REPLACE DESCR('Receiver channel from VM03NCM') +
  SEQWRAP(9999999) +
  MAXMSGL(2048) +
  TPNAME(VM03NCMSDRRCVR)
```

- Ein Serverkanal zu MQSeries unter MVS/ESA, der selbst einen Datenaustausch einleiten oder von einem fernen Requester-Kanal gestartet werden kann:

```
DEFINE CHANNEL(MT01.VM03.RQSV.CE) CHLTYPE(SVR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.RQSV.CE') +
  CONNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+
  MODENAME('LU62PS') TPNAME(MQSERIES) +
```

Dabei steht MQSeries für den Namen des Transaktionsprozesses (TPNAME), für den der MVS-WS-Manager empfängsbereit ist.

TCP/IP-Kommunikationskonfiguration - Beispiel

In diesem Beispiel wird die Kommunikation mit einem fernen MQSeries-System über TCP/IP eingerichtet.

Zeilengruppe TCPConfig in QMINI

Die Datei QMINI muss die entsprechende TCPConfig-Zeilengruppe enthalten. Beispiel:

```
TCPConfig:
  TCPPort=1414
  TCPNumListenerPorts=1
  TCPListenerPort=1996
  TCPKeepAlive=1
```

Kommunikation einrichten

Der Wert für TCPPort gibt den standardmäßigen ausgehenden Port an, der für Kanäle verwendet wird, in denen im Feld CONNAME kein Port angegeben ist. TCPListenerPort gibt den Standard-Port an, der verwendet wird, wenn der Befehl **runmqtsr** in der Befehlszeile ohne die Option **-p** eingegeben wird.

TCP/IP-Senderkanal definieren

Es muss ein TCP/IP-Senderkanal definiert werden. In diesem Beispiel ist MH01 der WS-Manager, der sich auf dem Host SPINACH befindet:

```
DEFINE CHANNEL(MT01_MH01_SDR_0001) CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  SEQWRAP(9999999) MAXMSGL(4194304) +
  XMITQ('MH01_TQ_SDR_0001') +
  CONNAME('SPINACH.HURSLEY.IBM.COM(2000)')
```

Dieser Kanal wird versuchen, eine Verbindung zum TCP/IP-Port mit der Nummer 2000 auf dem Host SPINACH herzustellen.

Im folgenden Beispiel handelt es sich um eine TCP/IP-Senderkanaldefinition für einen WS-Manager namens MH01, der sich auf dem Host SPINACH befindet; es wird der *standardmäßige* abgehende TCP/IP-Port verwendet:

```
DEFINE CHANNEL(MT01_MH01_SDR_0001) CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  SEQWRAP(9999999) MAXMSGL(4194304) +
  XMITQ('MH01_TQ_SDR_0001') +
  CONNAME('SPINACH.HURSLEY.IBM.COM')
```

Im Feld CONNAME ist keine Port-Nummer angegeben. Daher wird der im Eintrag TCPPort in der Datei QMINI angegebene Wert (1414) verwendet.

TCP/IP-Empfängerkanal definieren

Hier ein Beispiel für einen TCP/IP-Empfängerkanal:

```
DEFINE CHANNEL(MH01_MT01_SDR_0001) CHLTYPE(RCVR) +
  TRPTYPE(TCP)
```

Für einen TCP/IP-Empfängerkanal muss kein Wert im Feld CONNAME angegeben werden; es muss jedoch ein TCP/IP-Empfangsprogramm aktiv sein. Es gibt zwei Möglichkeiten, ein TCP/IP-Empfangsprogramm zu starten. Entweder:

1. Wechseln Sie in die PATHWAY-Umgebung des WS-Managers unter Verwendung von PATHCOM, und geben Sie den folgenden Befehl ein:

```
start server mqs-tcplis00
```

ODER

Kommunikation einrichten

Geben Sie an der TACL-Eingabeaufforderung den folgenden Befehl ein:

```
runmqtsr -m QMgrName
```

Ein TCP/IP-Empfangsprogramm wird gestartet, das an dem in der Datei QMINI festgelegten Port (in unserem Beispiel Port 1996) empfangsbereit ist.

Anmerkung: Diese Port-Nummer kann über die Option `-p Port` im Befehl `runmqtsr` überschrieben werden.

TCP/IP-Senderkanal auf dem fernen System definieren

Die folgende Senderkanaldefinition auf dem fernen System für eine Verbindung mit diesem Empfängerkanal ist denkbar:

```
DEFINE CHANNEL(MH01_MT01_SDR_0001) CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  XMITQ('MT01_TQ_SDR_0001') +  
  CONNAME('Compaq.ISC.UK.IBM.COM(1996)')
```

QMINI für die Unterstützung mehrerer TCP/IP-Empfangsprogramme konfigurieren

Soll ein WS-Manager mehrere TCP/IP-Empfangsprogramme unterstützen, müssen Sie für jedes zusätzliche Empfangsprogramm auf Basis der Serverklasse MQS-TCPLIS00 eine eigene PATHWAY-Serverklasse definieren.

Darüber hinaus muss für jedes TCP/IP-Empfangsprogramm in der Zeilengruppe TCPConfig der Datei QMINI ein eigener TCPListenerPort-Eintrag vorhanden sein.

Beispiel:

```
TCPCConfig:  
  TCPPort=1414  
  TCPNumListenerPorts=3  
  TCPListenerPort=1996  
  TCPListenerPort=1997  
  TCPListenerPort=1998  
  TCPKeepAlive=1
```

TCPCNumListenerPorts muss der Anzahl der TCPListenerPort-Einträge entsprechen (in unserem Beispiel drei). Diese Datei QMINI unterstützt drei TCP/IP-Empfangsprogramme, die an den Ports 1996, 1997 und 1998 empfangsbereit sind. Typischerweise sollten die Serverklassen, die diese drei Ports unterstützen, die Namen MQS-TCPLIS00, MQS-TCPLIS01 und MQS-TCPLIS02 erhalten.

Weitere Informationen zum Hinzufügen von Serverklassen finden Sie unter „TS/MP-Verwaltung (PATHWAY-Verwaltung)“ auf Seite 34.

Anhang N. MQSeries-Clients

Bei einem MQSeries-Client handelt es sich um ein MQSeries-System, das keinen WS-Manager enthält. Der MQSeries-Client-Code leitet MQI-Anrufe von Anwendungen, die auf dem Client-System ausgeführt werden, an einen WS-Manager in einem MQSeries-Serversystem weiter, mit dem es verbunden ist.

Dieser Anhang enthält Informationen zu MQSeries-Clients, die speziell für MQSeries for Compaq NSK V5.1 gelten. Diese sollten zusammen mit dem Handbuch *MQSeries Clients* verwendet werden.

Client-Unterstützung

MQSeries for Compaq NSK kann als MQSeries-Serversystem für alle MQSeries-Clients eingesetzt werden, die eine Serververbindung über die Protokolle TCP/IP oder SNA-LU6.2 herstellen können. Es gibt jedoch keine MQSeries for Compaq NSK-Clients.

Wenn ein MQSeries-Client eine Verbindung zu einem WS-Manager in MQSeries for Compaq NSK herstellt, gilt Folgendes:

- Für alle MQGET-, MQPUT- oder MQPUT1-Aufrufe, in denen die Option MQ*_SYNCPOINT aktiviert ist, wird eine Compaq-Transaktion gestartet, falls der Verbindungskennung noch keine Transaktion zugeordnet ist.
- Für alle MQGET-, MQPUT- oder MQPUT1-Aufrufe, in denen weder die Option MQ*_SYNCPOINT noch die Option MQ*_NO_SYNCPOINT angegeben ist, wird eine Compaq-Transaktion gestartet, falls der Verbindungskennung noch keine Transaktion zugeordnet ist.
- Mit dem MQCMIT-Aufruf wird eine Compaq-Transaktion festgeschrieben, wenn der Verbindungskennung eine Transaktion zugeordnet ist. Mit dem Aufruf MQBACK wird eine Compaq-Transaktion abgebrochen, wenn dieser Verbindungskennung eine Transaktion zugeordnet ist.

In allen Fällen werden bei einem Abbruch der Compaq-Prozedur BEGINTRANSACTION der Compaq-Code (*CompCode*) mit der Meldung MQCC_FAILED und der Ursachencode (*Reason*) MQRC_SYNCPOINT_NOT_AVAILABLE an den Aufrufer zurückgegeben.

Sicherheitsüberlegungen

MQSeries for Compaq NSK unterstützt die Verwendung von Kanalsicherheits-Exits, die wie folgt für eine Überprüfung der Clients sorgen:

- Nachdem eine Verbindung zwischen dem MQSeries-Client und dem Server hergestellt wurde, ruft der Client vor einer Rückkehr von dem Aufruf MQCONN den Sicherheits-Exit auf dem Server auf.
- Der Serversicherheits-Exit kann Informationen an den Sicherheits-Exit des Clients zurückgeben.

Dieser Dialog ermöglicht beispielsweise den Austausch vertraulicher Daten zwischen Server und Client. Ist auf dem Client kein Sicherheits-Exit definiert, werden die Werte der lokalen Umgebungsvariablen MQ_USER_ID und MQ_PASSWORD über Kanalattribute an den Server übergeben. Diese Attribute stehen dem Serversicherheits-Exit zur Überprüfung zur Verfügung.

MQSeries-Clients

Anhang O. Programmierbare Systemverwaltung

MQSeries for Compaq NSK unterstützt die folgenden Systemverwaltungsfunktionen von MQSeries:

- Instrumentierungsereignisse
- PCF-Befehle (Programmable Command Format)
- installierbare Services

Dieser Anhang gibt eine Übersicht über diese Funktionen in MQSeries for Compaq NSK. Eine ausführliche Beschreibung finden Sie im Handbuch *MQSeries Programmable System Management*.

Instrumentierungsereignisse

MQSeries for Compaq NSK unterstützt die standardmäßigen MQSeries-Instrumentierungsereignisse, die zur Generierung einer Ereignisnachricht in einer Ereigniswarteschlange führen.

Ereignisse werden aktiviert bzw. inaktiviert, indem für die Warteschlangen- und WS-Managerattribute die entsprechenden Werte wie folgt angegeben werden:

- Über MQSC-Befehle (siehe das Handbuch *MQSeries MQSC - Befehle*)
- Über PCF-Befehle (siehe das Handbuch *MQSeries Programmable System Management*)
- Über MQM (Message Queue Management) (siehe „Kapitel 4. WS-Manager verwalten“ auf Seite 47)

In MQSeries for Compaq NSK unterstützte Ereignisarten

In MQSeries for Compaq NSK werden die folgenden Ereignisarten unterstützt:

Tabelle 27. In MQSeries for Compaq NSK unterstützte Ereignisarten

Ereignisart	Ereignisname
Berechtigungsereignisse	Not Authorized (type 1)
Kanalereignisse	Channel Activated Channel Conversion Error Channel Not Activated Channel Started Channel Stopped
Inhibit-Ereignisse	Get Inhibited Put Inhibited
Lokale Ereignisse	Alias Base Queue Type Error Queue Type Error Unknown Alias Base Queue Unknown Object Name
Leistungsereignisse	Queue Depth High Queue Depth Low Queue Full Queue Service Interval High Queue Service Interval OK

Ereignisse

Tabelle 27. In MQSeries for Compaq NSK unterstützte Ereignisarten (Forts.)

Ereignisart	Ereignisname
Ferne Ereignisse	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
Start- und Stoppereignisse	Queue Manager Active Queue Manager Not Active

Format der Ereignisnachrichten

MQSeries for Compaq NSK unterstützt das Standardformat für MQSeries-Ereignisnachrichten. Die Nachricht besteht in diesem Fall aus zwei Teilen, dem *Nachrichtendeskriptor* (MQMD) und den *Nachrichtendaten*. Die Nachrichtendaten setzen sich aus einem Ereignis-Header und einigen ereignisspezifischen Daten zusammen.

Eine Übersicht über die MQMD-Struktur einer Ereignisnachricht ist in „MQMD (Message Descriptor)“ auf Seite 350 zu sehen. Eine Übersicht über die Header-Struktur von Ereignissen (MQCFH) ist in Tabelle 29 auf Seite 407 zu sehen.

Tabelle 28. MQMD-Struktur einer Ereignisnachricht

Parameter	Art	Werte
<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	Codierung des WS-Managers, der das Ereignis generiert.
<i>CodedCharSetId</i>	MQLONG	ID des codierten Zeichensatzes (CCSID) des WS-Managers, der das Ereignis generiert.
<i>Format</i>	MQCHAR8	MQFMT_EVENT
<i>Priority</i>	MQLONG	Standardpriorität einer Ereigniswarteschlange, wenn es sich um eine lokale Warteschlange oder um ihre lokale Definition in dem WS-Manager, der das Ereignis generiert, handelt.
<i>Persistence</i>	MQLONG	Standardwert für Permanenz einer Ereigniswarteschlange, wenn es sich um eine lokale Warteschlange oder um ihre lokale Definition in dem WS-Manager, der das Ereignis generiert, handelt.
<i>MsgId</i>	MQBYTE24	Dieser Wert wird vom WS-Manager generiert und ist eindeutig.
<i>CorrelId</i>	MQBYTE24	MQCI_NONE

Tabelle 28. MQMD-Struktur einer Ereignisnachricht (Forts.)

Parameter	Art	Werte
<i>BackoutCount</i>	MLONG	Der Wert ist immer 0.
<i>ReplyToQ</i>	MQCHAR48	Ist immer leer.
<i>ReplyToQMgr</i>	MQCHAR48	Der Name des WS-Managers im Ausgangssystem.
<i>UserIdentifier</i>	MQCHAR12	Ist immer leer.
<i>AccountingToken</i>	MQBYTE32	MQACT_NONE
<i>ApplIdentityData</i>	MQCHAR32	Ist immer leer.
<i>PutApplType</i>	MLONG	Die Art der Anwendung, die die Nachricht einreicht.
<i>PutApplName</i>	MQCHAR28	Der Name der Anwendung, die die Nachricht einreicht.
<i>PutDate</i>	MQCHAR8	Das Datum, an dem die Nachricht eingereicht wurde; wird vom WS-Manager generiert.
<i>PutTime</i>	MQCHAR8	Der Zeitpunkt, zu dem die Nachricht eingereicht wurde; wird vom WS-Manager generiert.
<i>ApplOriginData</i>	MQCHAR4	Ist immer leer.

Tabelle 29. Ereignis-Header-Struktur (MQCFH)

Parameter	Art	Werte
<i>Type</i>	MLONG	MQCFT_EVENT
<i>StrucLength</i>	MLONG	MQCFH_STRUC_LENGTH
<i>Version</i>	MLONG	MQCFH_VERSION_1
<i>Command</i>	MLONG	MQCMD_Q_MGR_EVENT MQCMD_PERFM_EVENT MQCMD_CHANNEL_EVENT
<i>MsgSeqNumber</i>	MLONG	Der Wert ist immer 1.
<i>Control</i>	MLONG	MQCFC_LAST
<i>CompCode</i>	MLONG	MQCC_OK MQCC_WARNING
<i>Reason</i>	MLONG	Ursachencode, der das Ereignis angibt.
<i>ParameterCount</i>	MLONG	Die Anzahl der Parameterstrukturen, die auf die MQCFH-Struktur folgen.

PCF-Befehle (Programmable Command Format)

MQSeries for Compaq NSK unterstützt die standardmäßigen PCF-Befehle (Programmable Command Format) (siehe das Handbuch *MQSeries Programmable System Management*). PCF-Nachrichten setzen sich aus zwei Teilen zusammen, dem *Nachrichtendeskriptor* (MQMD) und den *Nachrichtendaten*. Die *Nachrichtendaten* wiederum setzen sich aus einem PCF-Header (MQCFH) und einigen PCF-Parametern (die über die Strukturen MQCFIN, MQCFIL, MQCFST und MQCFSL definiert werden) zusammen.

PCF-Befehle

Eine Übersicht über den PCF-Nachrichtendeskriptor (MQMD) finden Sie in Tabelle 30. Eine Übersicht über die PCF-Header-Struktur (MQCFH) finden Sie in Tabelle 31. Eine Übersicht über die PCF-Parameterstrukturen finden Sie in Tabelle 32 auf Seite 409 bis Tabelle 35 auf Seite 410.

PCF-Nachrichtendeskriptor

In MQSeries for Compaq NSK wird der standardmäßige PCF-Nachrichtendeskriptor verwendet. Dieser enthält daher die folgenden Felder:

Tabelle 30. PCF-Nachrichtendeskriptor

Feld	Werte
<i>Report</i>	Ein beliebiger gültiger Wert
<i>MsgType</i>	MQMT_REQUEST
<i>Expiry</i>	Ein beliebiger gültiger Wert
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	Codierung, die für die Nachrichtendaten verwendet wird; bei Bedarf wird eine Konvertierung durchgeführt.
<i>CodedCharSetId</i>	Die CCSID, die für die Nachrichtendaten verwendet wird; bei Bedarf wird eine Konvertierung durchgeführt.
<i>Format</i>	MQFMT_ADMIN MQFMT_PCF (für Benutzerdaten)
<i>Priority</i>	Ein beliebiger gültiger Wert
<i>Persistence</i>	Ein beliebiger gültiger Wert
<i>MsgId</i>	Ein beliebiger gültiger Wert (einschließlich MQMI_NONE)
<i>CorrelId</i>	Ein beliebiger gültiger Wert (einschließlich MQMI_NONE)
<i>ReplyToQ</i>	Name der Warteschlange
<i>ReplyToQMgr</i>	Name des WS-Managers
Nachrichtenkontextfelder	Ein beliebiger gültiger Wert (einschließlich MQPMO_DEFAULT_CONTEXT)

PCF-Header (MQCFH)

In MQSeries for Compaq NSK wird der PCF-Standard-Header verwendet. Dieser enthält daher die folgenden Felder:

Tabelle 31. PCF-Header

Feld	Art	Werte
<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	Eine gültige Befehls-ID
<i>MsgSeqNumber</i>	MLONG	Die Folgenummer der Nachricht.
<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	Ursachencode, der den Beendigungscode ergänzt.
<i>ParameterCount</i>	MLONG	Gibt die Anzahl der Parameterstrukturen an.

PCF-Zeichenfolgeparameter (MQCFST)

In MQSeries for Compaq NSK wird die Standardstruktur für PCF-Zeichenfolgeparameter (MQCFST) verwendet. Diese Struktur enthält daher die folgenden Felder:

Tabelle 32. PCF-Zeichenfolgeparameter

Feld	Art	Wert
<i>Type</i>	MQLONG	MQCFT_STRING
<i>StrucLength</i>	MQLONG	Länge der MQCFST-Struktur in Byte.
<i>Parameter</i>	MQLONG	ID des Parameters.
<i>CodedCharSetId</i>	MQLONG	Die ID des codierten Zeichensatzes (CCSID).
<i>StringLength</i>	MQLONG	Länge der Daten (in Byte) im Feld <i>String</i> .
<i>String</i>	MQCHAR \times <i>StringLength</i>	Zeichenfolgewart.

PCF-Parameter für Integerliste (MQCFIL)

In MQSeries for Compaq NSK wird die Standardstruktur für den PCF-Integerlistenparameter (MQCFIL) verwendet. Diese Struktur enthält daher die folgenden Felder:

Tabelle 33. PCF-Integerliste

Feld	Art	Wert
<i>Type</i>	MQLONG	MQCFT_INTEGER_LIST
<i>StrucLength</i>	MQLONG	Länge der MQCFIL-Struktur in Byte.
<i>Parameter</i>	MQLONG	ID des Parameters.
<i>Count</i>	MQLONG	Anzahl der Elemente im Bereich <i>Values</i> .
<i>Values</i>	MQLONG \times <i>Count</i>	Parameterwerte.

PCF-Integer (MQCFIN)

In MQSeries for Compaq NSK wird die standardmäßige PCF-Integerstruktur (MQCFIN) verwendet. Diese Struktur enthält daher die folgenden Felder:

Tabelle 34. PCF-Integer

Feld	Art	Wert
<i>Type</i>	MQLONG	MQCFT_INTEGER
<i>StrucLength</i>	MQLONG	MQCFIN_STRUC_LENGTH
<i>Parameter</i>	MQLONG	ID des Parameters.
<i>Value</i>	MQLONG	Parameterwert.

PCF-Befehle

PCF-Zeichenfolgeliste (MQCFSL)

In MQSeries for Compaq NSK wird die Standardstruktur für die PCF-Zeichenfolgeliste (MQCFSL) verwendet. Diese Struktur enthält daher die folgenden Felder:

Table 35. PCF-Zeichenfolgeliste

Feld	Art	Wert
<i>Type</i>	MQLONG	MQCFT_STRING_LIST
<i>StrucLength</i>	MQLONG	Länge der MQCFSL-Struktur in Byte.
<i>Parameter</i>	MQLONG	ID des Parameters.
<i>CodedCharSetId</i>	MQLONG	CCSID der Daten im Feld <i>Strings</i> .
<i>Count</i>	MQLONG	Anzahl der Zeichenfolgen im Feld <i>Strings</i> .
<i>StringLength</i>	MQLONG	Länge (in Byte) der einzelnen Zeichenfolgen im Feld <i>Strings</i> .
<i>Strings</i>	MQCHAR \times <i>StringLength</i> \times <i>Count</i>	Zeichenfolgewerte für den im Feld <i>Parameter</i> angegebenen Parameter.

In MQSeries for Compaq NSK unterstützte PCF-Befehle

In MQSeries for Compaq NSK werden die folgenden MQSeries-PCF-Befehle unterstützt. Eine ausführliche Beschreibung dieser Befehle finden Sie im Handbuch *MQSeries Programmable System Management*.

Table 36. In MQSeries for Compaq NSK unterstützte PCF-Befehle

Befehl	Befehls-ID
Change Channel	MQCMD_CHANGE_CHANNEL
Change Namelist	MQCMD_CHANGE_NAMELIST
Change Process	MQCMD_CHANGE_PROCESS
Change Queue	MQCMD_CHANGE_Q
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

Tabelle 36. In MQSeries for Compaq NSK unterstützte PCF-Befehle (Forts.)

Befehl	Befehls-ID
Inquire Channel Status	MQCMD_INQUIRE_CHANNEL_STATUS
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
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

Anmerkung: Die Befehle Start Channel Initiator und Start Channel Listener werden in MQSeries for Compaq NSK nicht unterstützt.

Antworten auf PCF-Befehle

In MQSeries for Compaq NSK generiert der Befehlsserver Antwortnachrichten für jeden PCF-Befehl. Es gibt drei Antwortarten:

- Die Meldung, dass die Ausführung erfolgreich verlaufen ist
- Fehlermeldung
- Die Rückgabe von Daten

Weitere Informationen finden Sie im Handbuch *MQSeries Programmable System Management*.

Installierbare Services

MQSeries for Compaq NSK unterstützt den Berechtigungs- und den Namensservice.

Schnittstelle für Berechtigungsservice

Der Berechtigungsservice ermöglicht WS-Managern den Aufruf von Berechtigungsfunktionen. Beispielsweise kann ein WS-Manager über den Berechtigungsservice feststellen, ob eine bestimmte Benutzer-ID berechtigt ist, eine Warteschlange zu öffnen.

In MQSeries for Compaq NSK wird ein Berechtigungsservice zur Verfügung gestellt. Bei dieser Komponente handelt es sich um den Objektberechtigungsmanager (OAM, Object Authority Manager). Standardmäßig ist der OAM aktiviert und wird über die Steuerbefehle **dspmqa** (display authority) und **setmqa** (set authority) gesteuert.

Sie können den Objektberechtigungsmanager entweder durch einen eigenen Berechtigungsservice ergänzen oder ersetzen; weitere Informationen hierzu finden Sie im Handbuch *MQSeries Programmable System Management*.

Schnittstelle für Namensservice

Der Namensservice unterstützt den WS-Manager bei der Auflösung des WS-Managers, dem eine Warteschlange zugeordnet ist.

MQSeries for Compaq NSK unterstützt die Standardschnittstelle für den Namensservice (siehe *MQSeries Programmable System Management*).

Anhang P. In MQSeries for Compaq NSK verwendete EMS-Ereignisschablone

Die EMS-Schablonendatei (SMQSTMPL) enthält den Quellcode für die Definitionen der EMS-Ereignisse in MQSeries. Über diese Definitionen wird vorgegeben, wie die Informationen in den EMS-Ereignisnachrichten angezeigt werden; darüber hinaus geben sie Typ und Bedeutung der in den einzelnen EMS-Ereignisnachrichten enthaltenen Daten an.

Folgende Ereignisarten werden generiert:

ZMQS-VAL-EVT-ERROR

FFST-Fehler (Fehler in Zusammenhang mit einer Systemressource, einer Software- oder einer Hardwarekomponente).

ZMQS-VAL-EVT-ERR

MQSeries-Fehler; es wird ein Verweis auf ein FFST-Ereignis und auf Daten, die auf Platte gespeichert sind, gegeben.

ZMQS-VAL-EVT-MSG

Eine MQSeries-Nachricht, wie beispielsweise die Meldung, dass ein WS-Manager oder ein Kanal gestartet wurde. Allen diesen Ereignissen entspricht jeweils eine MQSeries-Protokollnachricht und alle enthalten dieselben Informationen und denselben Text. Die variablen Daten in den einzelnen Nachrichten sind in eigenen Token innerhalb der Ereignisnachricht enthalten. Weitere Informationen zu den MQSeries-Nachrichten finden Sie im Handbuch *MQSeries Messages*.

ZMQS-VAL-EVT-QMGR

Ein WS-Managerereignis für Berechtigungs-, Inhibit-, Start- und Stoppereignisse sowie für lokale und ferne Ereignisse. Diese EMS-Ereignisse haben denselben Informationsgehalt wie die entsprechenden PCF-Ereignisnachrichten, die im Handbuch *MQSeries Programmable System Management* beschrieben werden. Individuelle Token in den Ereignisnachrichten enthalten die jeweils variablen Daten.

ZMQS-VAL-EVT-PERF

Ein Leistungsereignis, das den standardmäßigen MQSeries-Leistungsereignissen entspricht. Diese Ereignisse liefern statistische Daten zu den Warteschlangen in einem WS-Manager. Die variablen Daten in Leistungsereignissen sind in den eigenen Token der einzelnen Ereignisnachrichten enthalten.

ZMQS-VAL-EVT-CHNL

Ein Kanalereignis, das den standardmäßigen MQSeries-Kanalereignissen entspricht. Kanalereignisse melden Änderungen am Kanalstatus oder Probleme mit der Kommunikation zwischen WS-Managern. Wie bei den anderen Ereignisnachrichtenarten sind die variablen Daten in Kanalereignissen jeweils in eigenen Token in der Ereignisnachricht enthalten.

EMS-Ereignisschablone

Hier ein Auszug aus den Definitionen der EMS-Schablonen:

```
VERSION: "IBM.MQS - 10JAN97"  
SSID: ZMQS-VAL-SSID  
SSNAME: "MQSeries", "MQS"
```

==

== This is an EMS FFST message

==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-ERROR  
OVERRIDE ZEMS-TKN-EMPHASIS ZSPI-VAL-TRUE  
"MQSeries FFST from component COMP_<1>"  
"<*CR> Error Code : <2>"  
"<*CR> Severity : <3>"  
"<*CR> Module Name : <4>"  
"<*CR> Probe ID : <5>"  
"<*CR> Error Text :"  
"<*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
```

==

== This is an EMS Display Message Event

==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-MSG  
"MQSeries message: <1>"  
"  
"<*CR> EXPLANATION :"  
" <2>"  
"  
"<*CR> ACTION :"  
" <3>"  
1: ZMQS-TKN-ERROR-TEXT  
2: ZMQS-TKN-ERROR-TEXT-2  
3: ZMQS-TKN-ERROR-TEXT-3
```

==

== This is an EMS Report Error Event

==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-ERR  
OVERRIDE ZEMS-TKN-EMPHASIS ZSPI-VAL-TRUE  
"MQSeries Error"  
"<*CR> Error Code : <1>"  
"<*CR> Function : <2>"  
"<*CR> Probe ID : <3>"  
"<*CR> FFST File : <4>"  
1: ZMQS-TKN-ERROR-CODE  
2: ZMQS-TKN-MODULE-NAME  
3: ZMQS-TKN-PROBE-ID  
4: ZMQS-TKN-FILE-NAME
```

==

== This is an EMS copy of PCF Queue Manager event message
== for authority, inhibit, local, remote, start_and_stop events

==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-QMGR  
"MQSeries QMgr Event from <1>"  
"<*CR> Reason : <2>"  
"<*IF 3><*CR> Reason Qualifier : <4>"  
"<*IF 5><*CR> User ID : <6>"
```

EMS-Ereignisschablone

```
"<*IF 13><*CR> Object QMgr : <14>                <*ENDIF>"
"<*IF 9><*CR> Options : <10>                      <*ENDIF>"
"<*IF 11><*CR> Command : <12>                     <*ENDIF>"
"<*IF 15><*CR> Queue Name : <16>                  <*ENDIF>"
"<*IF 17><*CR> Queue Type : <18>                 <*ENDIF>"
"<*IF 19><*CR> Base Queue Name : <20>           <*ENDIF>"
"<*IF 21><*CR> XMit Queue Name : <22>           <*ENDIF>"
"<*IF 30><*CR> Application Type : <31>          <*ENDIF>"
"<*IF 32><*CR> Application Name : <33>         <*ENDIF>"
```

- 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

==

== This is an EMS copy of PCF Performance event message

==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-PERF
"MQSeries Performance Event from <1>           "
"<*CR> Reason : <2>                             "
"<*CR> Queue Name : <3>                          "
"<*CR> Time Since Last Reset : <4>               "
"<*CR> Highest Queue Depth : <5>                 "
"<*CR> # Of Messages Enqueued : <6>              "
"<*CR> # Of Messages Dequeued : <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
```

==

== This is an EMS copy of PCF Channel event message

==

```
MSG: ZEMS-TKN-EVENTNUMBER, ZMQS-VAL-EVT-CHNL
"MQSeries Channel Event from <1>               "
"<*CR> Reason : <2>                             "
"<*CR> Channel Name : <3>                        "
"<*CR> XMit Queue Name : <5>                    "
"<*CR> Connection Name : <7>                    "
"<*CR> Reason Qualifier : <9>                    "
"<*CR> Format : <11>                             "
"<*CR> Return Code : <13>                       "
```

EMS-Ereignisschablone

```
"<*CR> Auxiliary rc 1 : <15>          "  
"<*CR> Auxiliary rc 2 : <17>          "  
"<*CR> CCSID 1 : <19>                "  
"<*CR> Auxiliary string 1 : <21>     "  
"<*CR> CCSID 2 : <23>                "  
"<*CR> Auxiliary string 2 : <25>     "  
"<*CR> CCSID 3 : <27>                "  
"<*CR> Auxiliary string 3 : <29>     "  
1: ZMQS-TKN-QMGR  
2: ZMQS-TKN-REASON  
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
```

Anhang Q. Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Dienstleistungen können auch andere, ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Veröffentlichung ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France zu richten. Anfragen an obige Adresse müssen auf Englisch formuliert werden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekannt gegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Hinweise in dieser Veröffentlichung auf WWW-Sites anderer Unternehmen als IBM dienen nur der Information; es kann kein Anspruch auf diese WWW-Sites abgeleitet werden. Die in diesen WWW-Sites aufgeführten Komponenten gehören nicht zum Lieferumfang dieses IBM Produkts; die Verwendung dieser WWW-Sites erfolgt auf Risiko des Kunden.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

Bemerkungen

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der IBM Kundenvereinbarung oder einer äquivalenten Vereinbarung.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen hinsichtlich des Leistungsspektrums von Produkten anderer Hersteller als IBM sind an den jeweiligen Hersteller des Produkts zu richten.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen.

Marken

Folgende Namen sind in gewissen Ländern Marken der IBM Corporation:

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 und NonStop sind Marken der Compaq Computer Corporation.

Intel ist in gewissen Ländern eine eingetragene Marke der Intel Corporation.

Microsoft, Windows und Windows NT sind Marken der Microsoft Corporation.

Java sowie alle Java-basierten Marken und Logos sind in gewissen Ländern Marken oder eingetragene Marken der Sun Microsystems Inc.

UNIX ist in gewissen Ländern eine eingetragene Marke und wird ausschließlich von der X/Open Company Limited lizenziert.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

Literaturverzeichnis

In diesem Abschnitt wird die für alle aktuellen MQSeries-Produkte verfügbare Dokumentation beschrieben.

Plattformübergreifende MQSeries-Veröffentlichungen

Der Großteil dieser Veröffentlichungen (auch als Handbücher zur MQSeries-„Produktfamilie“ bezeichnet), bezieht sich auf alle MQSeries-Produkte. Hierzu gehören die folgenden Produkte:

- MQSeries for AIX Version 5.2
- MQSeries for AS/400, V5.2
- MQSeries for AT&T GIS UNIX, V2.2
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for Compaq Tru64 UNIX Version 5.1
- MQSeries for HP-UX Version 5.2
- MQSeries for Linux Version 5.2
- MQSeries for OS/2 Warp Version 5.1
- MQSeries for OS/390, V5.2
- MQSeries for SINIX and DC/OSx Version 2.2
- MQSeries for Sun Solaris, V5.2
- MQSeries for Sun Solaris, Intel Platform Edition Version 5.1
- MQSeries for Compaq NSK V5.1
- MQSeries for VSE/ESA, V2.1.1
- MQSeries for Windows, V2.0
- MQSeries for Windows, V2.1
- MQSeries for Windows NT and Windows 2000, V5.2

Folgende plattformübergreifende MQSeries-Veröffentlichungen sind verfügbar:

- *MQSeries Brochure* (G511-1908)
- *An Introduction to Messaging and Queuing* (GC33-0805)
- *MQSeries Intercommunication* (SC33-1872)
- *Cluster-Unterstützung in MQSeries* (SC12-2640)
- *MQSeries Clients* (GC33-1632)
- *MQSeries System Administration* (SC33-1873)
- *MQSeries MQSC - Befehle* (SC12-2645)
- *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 Application Programming Reference* (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)

Plattformspezifische MQSeries-Veröffentlichungen

Informationen zu den einzelnen MQSeries-Produkten finden Sie nicht nur in den Handbüchern zur MQSeries-Produktfamilie, sondern auch mindestens in einer der betriebssystem-spezifischen Veröffentlichungen.

MQSeries for AIX Version 5.2

MQSeries for AIX Einstieg, GC12-2578

MQSeries for AS/400, V5.2

MQSeries for AS/400 Quick Beginnings (GC34-5557)

MQSeries for AS/400 System Administration (SC34-5558)

MQSeries for AS/400 Application Programming Reference (ILE RPG) (SC34-5559)

MQSeries for AT&T GIS UNIX, V2.2

MQSeries for AT&T GIS UNIX System Management Guide (SC33-1642)

MQSeries for Compaq OpenVMS Alpha V5.1

MQSeries for Compaq OpenVMS Alpha Einstieg (GC12-2997)

MQSeries for Compaq OpenVMS Alpha Systemverwaltung (SC12-2998)

MQSeries for Compaq NSK V5.1

MQSeries for Compaq NSK Einstieg (GC12-2995)

MQSeries for Compaq NSK Systemverwaltung (SC12-2996)

MQSeries for Compaq Tru64 UNIX Version 5.1

Bibliographie

MQSeries for Compaq Tru64 UNIX Einstieg (GC12-2798)

MQSeries for HP-UX Version 5.2

MQSeries for HP-UX Einstieg, (GC12-2579)

MQSeries for Linux Version 5.2

MQSeries for Linux Einstieg (GC12-2779)

MQSeries for OS/2 Warp Version 5.1

MQSeries for OS/2 Warp Einstieg, (GC12-2513)

MQSeries for OS/390, V5.2

MQSeries for OS/390 Concepts and Planning Guide (GC34-5650)

MQSeries for OS/390 System Setup Guide (SC34-5651)

MQSeries for 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 Link for R/3 Benutzerhandbuch (GC12-2533)

MQSeries for SINIX and DC/OSx Version 2.2

MQSeries for SINIX and DC/OSx System Management Guide (GC33-1768)

MQSeries for Sun Solaris, V5.2

MQSeries for Sun Solaris Einstieg, GC12-2580

MQSeries for Sun Solaris, Intel Platform Edition Version 5.1

MQSeries for Sun Solaris, Intel Platform Edition Einstieg (GC12-2927)

MQSeries for VSE/ESA, V2.1.1

MQSeries for VSE/ESA Licensed Program Specifications (GC34-5365)

MQSeries for VSE/ESA System Management Guide (GC34-5364)

MQSeries for Windows, V2.0

MQSeries for Windows Benutzerhandbuch (GC12-2418)

MQSeries for Windows, V2.1

MQSeries for Windows Benutzerhandbuch (GC12-2515)

MQSeries for Windows NT and Windows 2000, V5.2

MQSeries for Windows NT and Windows 2000 Einstieg (GC12-2642)

MQSeries for Windows NT Using the Component Object Model Interface (SC34-5387)

MQSeries LotusScript Extension (SC12-2639)

Softcopy-Bücher

Der Großteil der MQSeries-Bücher ist sowohl als Hard- als auch als Softcopy erhältlich.

HTML-Format

Relevante MQSeries-Veröffentlichungen im HTML-Format sind im Lieferumfang der folgenden MQSeries-Produkte enthalten:

- MQSeries for AIX Version 5.2
- MQSeries for AS/400, V5.2
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for Compaq Tru64 UNIX Version 5.1
- MQSeries for HP-UX Version 5.2
- MQSeries for Linux Version 5.2
- MQSeries for OS/2 Warp Version 5.1
- MQSeries for OS/390, V5.2
- MQSeries for Sun Solaris, V5.2
- MQSeries for Sun Solaris, Intel Platform Edition Version 5.1
- MQSeries for Windows NT and Windows 2000, V5.2 (kompiliertes HTML-Format)
- MQSeries Link for R/3 V1.2

Die MQSeries-Bücher stehen außerdem im HTML-Format auf der Website der MQSeries-Produktfamilie zur Verfügung:

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

Bücher im PDF-Format (Portable Document Format)

PDF-Dateien können mit dem Adobe Acrobat Reader angezeigt und gedruckt werden.

Falls Sie den Adobe Acrobat Reader selbst benötigen oder aktuelle Informationen zu den Betriebssystemen, die den Acrobat Reader unterstützen, besuchen Sie die Website der Adobe Systems Inc. unter:

<http://www.adobe.com/>

PDF-Versionen relevanter MQSeries-Bücher sind im Lieferumfang der folgenden MQSeries-Produkte enthalten:

- MQSeries for AIX Version 5.2
- MQSeries for AS/400, V5.2
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for Compaq Tru64 UNIX Version 5.1
- MQSeries for HP-UX Version 5.2
- MQSeries for Linux Version 5.2
- MQSeries for OS/2 Warp Version 5.1
- MQSeries for OS/390, V5.2
- MQSeries for Sun Solaris, V5.2
- MQSeries for Sun Solaris, Intel Platform Edition Version 5.1
- MQSeries for Windows NT and Windows 2000, V5.2
- MQSeries Link for R/3 V1.2

PDF-Versionen der aktuellen MQSeries-Bücher stehen außerdem auf der Website der MQSeries-Produktfamilie zur Verfügung:

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

BookManager-Format

Die MQSeries-Bibliothek steht im IBM BookManager-Format in einer Reihe von CKITs mit Online-Bibliotheken zur Verfügung, einschließlich der CKITs *Transaction Processing and Data* (SK2T-0730). Diese Softcopy-Bücher im IBM BookManager-Format können mit folgenden IBM Lizenzprogrammen angezeigt werden:

BookManager READ/2
 BookManager READ/6000
 BookManager READ/DOS
 BookManager READ/MVS
 BookManager READ/VM
 BookManager READ für Windows

Bücher im PostScript-Format

Die MQSeries-Bibliothek im PostScript-Format (.PS) ist im Lieferumfang einer Reihe von MQSeries-Produkten der Version 2 enthalten. Bücher in diesem Format können mit einer entsprechenden Anzeigefunktion angezeigt bzw. mit einem PostScript-Drucker gedruckt werden.

Format der Online-Hilfe von Windows

Das *MQSeries for Windows Benutzerhandbuch* wird zusammen mit MQSeries for Windows Version 2.0 und MQSeries for Windows Version 2.1 im Format der Online-Hilfe von Windows geliefert.

Im Internet verfügbare Informationen zu MQSeries

Die Website der MQSeries-Produktfamilie finden Sie unter:

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

Über die Links auf dieser Website können Sie:

- aktuelle Informationen zur MQSeries-Produktfamilie abrufen
- auf MQSeries-Bücher im HTML- und PDF-Format zugreifen
- MQSeries SupportPacs herunterladen

Referenzliteratur

- *SNAX/APC Planning and Configuration Manual* (Compaq-Teilenummer 098289)
 SNAX/APC stellt LU6.2-Unterstützung für die Compaq-spezifische Implementierung von SNA zur Verfügung. Dieses Handbuch beschreibt die Installation und Konfiguration von SNAX/APC.
- *SCF Reference Manual for SNAX/APC*, (Compaq-Teilenummer 064525)
 SNAX/APC stellt LU6.2-Unterstützung für die Compaq-spezifische Implementierung von SNA zur Verfügung. Dieses Handbuch erläutert die interaktive SCF-Schnittstelle (Subsystem Control Facility), über die Bediener und Netzmanager SNAX/APC konfigurieren und steuern können.
- *Pathway System Management Guide*, (Compaq-Teilenummer 096881)
 Dieses Handbuch enthält Hinweise zur Konfiguration und Steuerung von PATHWAY-Transaktionsverarbeitungssystemen.
- *Introduction to NonStop Transaction Manager/MP (TM/MP)*, (Compaq-Teilenummer 085812)
 Dieses Handbuch erläutert die Verwendung des TMF-Subsystems, um unternehmensspezifische Transaktionen zu schützen und die Integrität Ihrer Datenbanken sicherzustellen.
- *Introduction to Compaq Networking and Data Communications*, (Compaq-Teilenummer 093148)
 Dieses Handbuch enthält eine Übersicht über die Netzwerk- und Datenübertragungskonzepte, Tasks, Produkte und Handbücher von Compaq.
- *Intersystem Communications Environment (ICE) Installation Guide*, (Version 3 Release 2 oder höher)

Referenzliteratur

Dieses Handbuch erläutert die Installation von ICE und die Konfigurierung der ICE-Startparameter. (ICE bietet LU6.2-Unterstützung für die Insessions-spezifische Implementierung von SNA.)

- *Intersystem Communications Environment (ICE) Administrator's Guide*, (Version 3 Release 2 oder höher)

Dieses Handbuch erläutert die Konfiguration und den Betrieb von ICE sowie der ICE-Schnittstellen und ICE-Dienstprogramme.

Glossar der Begriffe und Abkürzungen

In diesem Glossar werden die MQSeries-spezifischen Begriffe und Abkürzungen erläutert, die im vorliegenden Handbuch verwendet werden. Bei Begriffen, die hier nicht aufgeführt sind, können Sie entweder im Index nachsehen oder im *IBM Dictionary of Computing*, New York: McGraw-Hill (1994) nachschlagen.

In diesem Glossar werden Begriffe und Definitionen aus dem *American National Dictionary for Information Systems*, ANSI X3.172-1990 (Copyright 1990 by the American National Standards Institute (ANSI)) aufgenommen. Dieses Wörterbuch kann über das American National Standards Institute, 11 West 42 Street, New York, New York 10036, USA, bezogen werden. Definitionen aus diesem Wörterbuch sind mit (A) gekennzeichnet.

A

Advanced Program-to-Program Communication (APPC) . Ein grundlegendes Merkmal der LU 6.2-Architektur und ihrer vielfältigen Implementierung in Produkten.

Alert. Eine Nachricht, die zur Identifizierung eines Problems bzw. eines anstehenden Problems an das zentrale Alert-Verarbeitungssystem für Verwaltungsservices gesendet wird.

Aliaswarteschlangenobjekt. Ein MQSeries-Objekt, bei dessen Namen es sich um den Aliasnamen einer Basiswarteschlange handelt, die im lokalen WS-Manager definiert ist. Bei Verwendung einer Aliaswarteschlange durch eine Anwendung bzw. einen WS-Manager wird der Aliasname aufgelöst und die angeforderte Operation für die entsprechende Basiswarteschlange ausgeführt.

Alternative Benutzer-ID. Eine Sicherheitseinrichtung, bei der die Berechtigung einer Benutzer-ID von einer anderen Benutzer-ID übernommen werden kann (z. B. zum Öffnen von MQSeries-Objekten).

Anforderungsnachricht. Eine Nachrichtenart, mit der eine Antwort von einem anderen Programm angefordert wird.

Anstehendes Ereignis. Ein außerplanmäßiges Ereignis, das bei Verbindungsanforderungen von einem CICS-Adapter ausgegeben wird.

Antwortnachricht. Eine Nachrichtenart, die als Antwort auf Anforderungsnachrichten verwendet wird.

Anwendungswarteschlange. Warteschlangen, die von Anwendungen verwendet werden.

Anzeigeursor. Beim Message-Queuing ein Anzeiger, der beim Durchsuchen einer Warteschlange nach der nächsten Nachricht verwendet wird.

Anzeigen (browse). Beim Message-Queuing der Abruf des Inhalts einer Nachricht durch einen MQGET-Aufruf, ohne dass die Nachricht aus der Warteschlange entfernt wird. Siehe auch *GET (Abrufen)*.

APAR. Siehe *Authorized Program Analysis Report*.

APPC. Siehe *Advanced Program to Program Communication*.

Arbeitseinheit. Eine wiederherstellbare Operationsfolge, die von einer Anwendung zwischen zwei konsistenten Zuständen ausgeführt wird. Eine Einheit beginnt mit dem Start einer Transaktion oder im Anschluss an einen vom Benutzer angeforderten Synchronisationspunkt. Sie endet entweder bei einem vom Benutzer angeforderten Synchronisationspunkt oder beim Abschluss einer Transaktion. Vgl. *Arbeitseinheit mit Wiederherstellung*.

Arbeitseinheit mit Wiederherstellung. Eine wiederherstellbare Operationsfolge in einem einzelnen Ressourcenmanager. Vgl. *Arbeitseinheit*.

Asynchrone Nachrichtenübertragung. Ein Kommunikationsverfahren zwischen verschiedenen Programmen, bei dem von den einzelnen Programmen Nachrichten in Nachrichtenwarteschlangen eingereicht werden. Bei der asynchronen Nachrichtenübertragung können die sendenden Programme mit der Verarbeitung momentan anstehender Vorgänge fortfahren, ohne erst auf eine Antwort auf die abgeschickte Nachricht warten zu müssen. Vgl. *Synchrone Nachrichtenübertragung*.

Attribut. Eines der verschiedenen Merkmale, die die Eigenschaften eines MQSeries-Objekts festlegen.

Auflösungspfad. Die Warteschlangen, die geöffnet werden, wenn eine Anwendung als Eingabe in einem MQOPEN-Aufruf eine Aliaswarteschlange bzw. eine ferne Warteschlange angibt.

Ausführungs-Controller. Ein Hilfssteuerprozess im WS-Manager, der für eine Reihe von Agenten zuständig ist.

Ausgabeparameter. Ein Parameter eines MQI-Aufrufs, über den der WS-Manager Informationen über die erfolgreiche bzw. nicht erfolgreiche Ausführung des Aufrufs zurückgibt.

Auslöseereignis. Ein Ereignis (z. B. eine Nachricht, die in einer Warteschlange empfangen wird), auf Grund dessen ein WS-Manager eine Auslösenachricht in einer Initialisierungswarteschlange generiert.

Auslösefunktion (Triggering). In MQSeries eine Einrichtung, mit der ein WS-Manager eine Anwendung automatisch starten kann, sobald die hierfür vorgegebenen Bedingungen in einer Warteschlange erfüllt sind.

Auslösemonitor. Eine ständig aktive Anwendung, die für eine oder mehrere Initialisierungswarteschlangen zuständig ist. Die in eine Initialisierungswarteschlange eingehenden Auslösenachrichten werden vom Auslösemonitor abgerufen. Aufgrund der Informationen in der Auslösenachricht startet der Auslösemonitor einen Prozess für die Warteschlange, in der das Auslöseereignis aufgetreten ist.

Auslösemonitorschnittstelle (Trigger Monitor Interface, TMI). Eine MQSeries-Schnittstelle, mit der benutzerdefinierte oder kommerzielle Auslösemonitorprogramme kompatibel sein müssen. Bestandteil von MQSeries Framework.

Auslösenachricht. Eine Nachricht mit Informationen über das Programm, das von einem Auslösemonitor gestartet werden soll.

Authorized Program Analysis Report (APAR). Ein Bericht zu einem Problem, dessen Ursache auf einen vermutlichen Fehler in einem aktuellen, noch nicht geänderten Programm-Release zurückgeführt wird.

B

Basic Mapping Support (BMS, Anzeigeformatierungsunterstützung). Eine Schnittstelle zwischen CICS und Anwendungsprogrammen, die eine Formatierung der Ein- und Ausgabeanzeigedaten vornimmt und mehrseitige Ausgabenachrichten ungeachtet der von den verschiedenen Terminals verwendeten Steuerzeichen weiterleitet.

Beendigungscode. Ein Rückkehrcode, der das Resultat eines MQI-Aufrufs angibt.

Befehl. In MQSeries eine Anweisung, die von einem WS-Manager ausgeführt werden kann.

Befehlsbehälter (Command Bag). In Zusammenhang mit der Verwaltungsschnittstelle MQAI ein Datenbehälter (Data Bag), der für die MQSeries-Verwaltung erstellt wird, der jedoch nicht über die Optionen zum Ändern der Reihenfolge der Datenelemente und zum Erstellen von Listen verfügt.

Befehlsprozessor. Die MQSeries-Komponente, von der Befehle verarbeitet werden.

Befehlsserver. Die MQSeries-Komponente, die Befehle aus der Eingabewarteschlange für Systembefehle liest, überprüft und gültige Befehle an den Befehlsprozessor weiterleitet.

Behälter (Bag). Siehe *Datenbehälter (Data Bag)*.

Benutzerbehälter (User Bag). In Zusammenhang mit der Verwaltungsschnittstelle MQAI eine besondere Art von Datenbehälter (Data Bag), die vom Benutzer erstellt wird.

Berechtigungsdatei. Eine Datei mit Sicherheitsdefinitionen für ein Objekt, eine Objektklasse oder alle Objektklassen.

Berechtigungsprüfungen. Sicherheitsprüfungen, die durchgeführt werden, wenn ein Benutzer Verwaltungsbefehle für ein Objekt (z. B. zum Öffnen einer Warteschlange oder zum Aufbau einer Verbindung zu einem WS-Manager) absetzt.

Berechtigungsprüfungen. Sicherheitsprüfungen, die bei dem Versuch, ein MQSeries-Objekt zu öffnen, durchgeführt werden.

Berichtsnachricht. Eine Nachrichtenart, die Informationen zu einer anderen Nachricht enthält. Eine Berichtsnachricht gibt z. B. an, dass eine Nachricht übertragen wurde, bei der Zieladresse eingegangen ist oder aus bestimmten Gründen nicht verarbeitet werden konnte.

BMS. Siehe *Basic Mapping Support*.

C

CCF. Channel Control Function (siehe *Kanalsteuerfunktion*).

CCSID. Coded Character Set Identifier (siehe *ID für den codierten Zeichensatz*).

CDF. Channel Definition File (siehe *Kanaldefinitionsdatei*).

CICS. Siehe *Customer Information Control System*.

Client. Eine Laufzeitkomponente, die lokalen Benutzeranwendungen den Zugriff auf Services zur Steuerung von Warteschlangen ermöglicht, die auf einem Server zur Verfügung stehen. Die von den Anwendungen verwendeten Warteschlangen befinden sich auf dem Server. Sie auch *MQI-Client*.

Client-Anwendung. Eine Anwendung, die auf Workstations zur Ausführung kommt und mit einem Client verbunden ist, über den sie Zugriff auf Services zur Steuerung von Warteschlangen erhält, die auf einem Server zur Verfügung stehen.

Cluster. Ein Verbund von WS-Managern, zwischen denen logische Beziehungen bestehen.

Common Run-Time Environment (CRE). Eine Gruppe von Services, die System- und Anwendungsprogrammierern die Erstellung von Programmen unter Verwendung verschiedener Programmiersprachen ermöglichen. Diese gemeinsam benutzten Laufzeit-services können von C-, COBOL85-, FORTRAN-, Pascal- und TAL-Programmen verwendet werden.

CRE. Common Run-Time Environment.

Customer Information Control System (CICS).

Ein IBM Transaktionsverwaltungssystem, das über benutzerdefinierte Anwendungsprogramme gleichzeitige Zugriffe auf Datendateien online ermöglicht. CICS bietet darüber hinaus Tools für die Erstellung, Verwendung und Verwaltung von Datenbanken.

D

Datagramm. Die einfachste Nachricht, die von MQSeries unterstützt wird. Für diese Nachrichtenart wird keine Antwort erwartet.

Datenbehälter (Data Bag). In Zusammenhang mit der Verwaltungsschnittstelle MQAI ein Behälter, der die Bearbeitung von Objekteigenschaften bzw. Objektparametern erleichtert.

Datenelement. In Zusammenhang mit der Verwaltungsschnittstelle MQAI ein Element in einem Datenbehälter (Data Bag). Dabei kann es sich um ein benutzer- oder systemspezifisches Integer oder eine benutzerspezifische oder systemspezifische Zeichenfolge handeln.

Datenumsetzungsschnittstelle (DCI). Die MQSeries-Schnittstelle, mit der vom Benutzer bzw. Lieferanten geschriebene Programme zur Umsetzung von Anwendungsdaten zwischen verschiedenen Maschinenverschlüsselungen und CCSIDs kompatibel sein müssen. Bestandteil von MQSeries Framework.

DCE. Siehe *Distributed Computing Environment*.

DCI. Data Conversion Interface (siehe *Datenumsetzungsschnittstelle*).

Dienstprogramm. In MQSeries eine Reihe von Programmen, die dem Systembediener bzw. dem Systemadministrator Funktionen zur Verfügung stellen, die jene der MQSeries-Befehle ergänzen. Mit einigen dieser Dienstprogramme werden mehrere Funktionen aufgerufen.

Distributed Computing Environment (DCE). Eine Middleware, die grundlegende Services zur Verfügung stellt, die die Entwicklung verteilter Anwendungen erleichtert. DCE wird von der OSF (Open Software Foundation) definiert.

DLQ (Warteschlange für nicht zustellbare Nachrichten). Eine Warteschlange, in die von WS-Managern bzw. Anwendungen Nachrichten eingereiht werden, die nicht an die eigentliche Zieladresse übertragen werden können.

Dynamische Warteschlange. Eine lokale Warteschlange, die beim Öffnen eines Modell-WS-Objekts durch ein Programm erstellt wird. Siehe auch *Permanente dynamische Warteschlange* und *Temporäre dynamische Warteschlange*.

E

ECBoss. Der Manager der Ausführungs-Controller (Execution Controller Boss) ist der Hauptsteuerprozess im WS-Manager.

Ein-/Ausgabeparameter. Ein Parameter eines MQI-Aufrufs, in dem beim Absetzen des Aufrufs Informationen angegeben werden können, die vom WS-Manager je nach Ergebnis des Aufrufs (erfolgreich bzw. nicht erfolgreich) geändert werden.

Eingabeparameter. Ein Parameter eines MQI-Aufrufs, in dem beim Absetzen des Aufrufs Informationen angegeben werden können.

Einphasige Festschreibung. Ein Verfahren, bei dem ein Programm Aktualisierungen in einer Warteschlange festschreiben kann, ohne dass diese Aktualisierungen mit den Änderungen synchronisiert werden, die vom Programm an Ressourcen vorgenommen wurden, die von einem anderen Ressourcenmanager gesteuert werden. Vgl. *Zweiphasige Festschreibung*.

Einphasige Rücksetzung. Ein Verfahren, bei dem ein aktiver Vorgang nicht beendet werden darf und bei dem alle Änderungen, die im Verlauf des Vorgangs vorgenommen wurden, zurückgesetzt werden.

Empfängerkanal. Beim Message-Queuing ein Kanal, der auf einen Senderkanal reagiert sowie Nachrichten aus einer Kommunikationsverbindung abrufen und in eine lokale Warteschlange einreicht.

Empfangsprogramm. In MQSeries-Umgebungen mit verteilter Steuerung von Warteschlangen ein Programm, das die Informationen zu ankommenden Netzverbindungen überwacht.

EMS. Event Monitoring System.

Ereignis. Siehe *Kanalereignis*, *Instrumentierungsereignis*, *Leistungsereignis* und *WS-Managerereignis*.

Ereignisdaten. Der Teil der Nachrichtendaten in einer Ereignisnachricht, der Informationen über das Ereignis enthält, z. B. den Namen des WS-Managers sowie der Anwendung, durch die das Ereignis ausgelöst wurde. Siehe auch *Ereignis-Header*.

Ereignis-Header. Der Teil der Nachrichtendaten in einer Ereignisnachricht, der Aufschluss über die Ereignisart des Ursachencodes für das betreffende Ereignis gibt.

Ereignisnachricht. Enthält Informationen (z. B. die Ereigniskategorie, den Namen der Anwendung, von dem das Ereignis ausgelöst wurde, und Statistikdaten für den WS-Manager), die Aufschluss über den Ursprung eines Instrumentierungsereignisses in einem Netz aus MQSeries-Systemen geben.

Ereigniswarteschlange. Die Warteschlange, in die der WS-Manager bei der Entdeckung eines Ereignisses eine entsprechende Ereignisnachricht stellt. Für jede Ereigniskategorie (WS-Manager-, Leistungs- oder Kanalereignis) ist eine eigene Ereigniswarteschlange vorhanden.

Erzwungener Abschluss. In MQSeries der Systemabschluss eines WS-Managers, ohne eine Verbindungsunterbrechung von Anwendungen bzw. den Abschluss von MQI-Aufrufen abzuwarten. Vgl. *Sofortiger Abschluss* und *Gesteuerter Abschluss*.

F

Ferner WS-Manager. In Bezug auf ein Programm handelt es sich bei einem fernen WS-Manager um einen WS-Manager, zu dem das Programm keine Verbindung hat.

Fernes Warteschlangenobjekt. Siehe *Lokale Definition einer fernen Warteschlange*.

Ferne Warteschlange. Eine Warteschlange, die einem fernen WS-Manager zugeordnet ist. Programme können Nachrichten in ferne Warteschlangen einreihen, jedoch nicht aus ihnen abrufen. Vgl. *Lokale Warteschlange*.

Festschreiben. Bezeichnet den Abschluss einer Transaktion, bei dem die Änderungen an der Datenbank aufgezeichnet und übernommen werden. Geschützte Ressourcen werden anschließend freigegeben.

FFST. First Failure Support Technology.

FIFO. Siehe *First In /First Out*.

First Failure Support Technology (FFST). Dieses Verfahren wird von MQSeries auf UNIX-Systemen, MQSeries for OS/2 Warp, MQSeries for Windows NT and Windows 2000 und MQSeries for AS/400 zur Ermittlung und Meldung von Softwareproblemen eingesetzt.

First In/First Out (FIFO). Ein Abrufverfahren im Zusammenhang mit Warteschlangen, bei dem immer zunächst der Eintrag abgerufen wird, der am längsten in der Warteschlange vorhanden ist.

Framework. In MQSeries eine Reihe von Programmierschnittstellen, die Kunden bzw. Lieferanten die Erstellung von Programmen ermöglicht, die bestimmte

Funktionen in MQSeries-Produkten erweitern oder ersetzen. Dabei handelt es sich um folgende Schnittstellen:

- MQSeries-Datenumsetzungsschnittstelle (DCI)
- MQSeries-Nachrichtenkanalschnittstelle (MCI)
- MQSeries-Namensserviceschnittstelle (NSI)
- MQSeries-Schnittstelle für Sicherheitsaktivierung (SEI)
- MQSeries-Auslösemonitorschnittstelle (TMI)

G

Gesteuerter Abschluss. In MQSeries der Systemabschluss eines WS-Managers, bei dem die Verbindungen aller aktiven Anwendungen ordnungsgemäß beendet werden. Vgl. *Sofortiger Abschluss* und *Erzwungener Abschluss*.

Gesteuerter Systemabschluss. Siehe *Gesteuerter Abschluss*.

GET. Beim Message-Queuing der Abruf einer Nachricht aus einer Warteschlange mit einem MQGET-Aufruf. Siehe auch *Anzeigen (browse)*.

I

ICE. Bei ICE (Intersystem Communications Environment) handelt es sich um eine Compaq-Softwareproduktfamilie, die den Zugriff auf eine Reihe von Anwendungen auf Compaq-Computern ermöglicht.

ID des codierten Zeichensatzes (CCSID). Der Name eines codierten Zeichensatzes sowie die entsprechenden Codepunktzuordnungen.

INI-Datei. Siehe *Konfigurationsdatei*.

Initialisierungwarteschlange. Eine lokale Warteschlange, in die der WS-Manager Auslösenachrichten stellt.

Installierbare Services. In MQSeries auf UNIX-Systemen, MQSeries for Compaq, MQSeries for OS/2 Warp und MQSeries for Windows NT and Windows 2000 zusätzliche Funktionen, die als jeweils eigene Komponenten zur Verfügung gestellt werden. Die Installation dieser Komponenten ist wahlfrei; stattdessen können auch eigene Komponenten oder Komponenten anderer Hersteller verwendet werden. Siehe auch *Berechtigungsservice*, *Namensservice* und *Benutzer-ID-Service*.

Instrumentierungsereignis. Eine Einrichtung, die die Überwachung des WS-Managerbetriebs in einem Netz aus MQSeries-Systemen ermöglicht. MQSeries stellt Instrumentierungsereignisse zur Überwachung von Ressourcendefinitionen von WS-Managern, des Leistungsstatus und des Kanalstatus zur Verfügung. Die Instrumentierungsereignisse können von benutzerdefinierten Meldeverfahren in Verwaltungsan-

wendungen verwendet werden, die dem Systembediener aufgetretene Ereignisse anzeigen. Darüber hinaus ermöglichen sie es, dass Anwendungen von anderen Verwaltungsnetzen als Agenten für die Überwachung von Meldungen und die Erstellung der entsprechenden Alerts eingesetzt werden können.

Internet Protocol (IP). Ein Protokoll, das zur Datenübertragung von Daten in einer Internet-Umgebung verwendet wird. Hierbei handelt es sich um die Basischicht, auf der andere Protokollschichten wie TCP und UDP aufbauen.

IP. Siehe *Internet Protocol*.

K

Kanal. Siehe *Nachrichtenkanal*.

Kanaldefinitionsdatei (CDF). In MQSeries eine Datei mit Übertragungskanaldefinitionen, über die den Übertragungswarteschlangen Kommunikationsverbindungen zugeordnet werden.

Kanalereignis. Ein Ereignis, das anzeigt, dass ein Kanalexemplar jetzt zur Verfügung steht bzw. nicht mehr verfügbar ist. Kanalereignisse werden in den WS-Managern an beiden Kanalenden generiert.

Kanal für Client-Verbindungen. Die MQI-Kanaldefinition, die einem MQI-Client zugeordnet ist. Siehe auch *Kanal für Serververbindungen*.

Kanal für Serververbindungen. Eine MQI-Kanaldefinition, die dem Server zugeordnet ist, auf dem ein WS-Manager läuft. Siehe auch *Kanal für Client-Verbindungen*.

Kanalsteuerfunktion (CCF, Channel Control Function). Ein Programm zum Übertragen von Nachrichten aus einer Übertragungswarteschlange an eine Kommunikationsverbindung und von dieser in eine lokale Warteschlange; dazu gehört eine Steuerkonsolenschnittstelle für den Aufbau und die Steuerung von Kanälen.

Kennung. Siehe *Verbindungskennung* und *Objektkennung*.

Konfigurationsdatei (auch INI-Datei). Eine Datei, die Konfigurationsdaten z. B. zu Protokollen, zur Kommunikation oder zu installierbaren Services enthält. Siehe auch *Zeilengruppe*.

Kontext. Informationen über den Ursprung einer Nachricht.

Kontextsicherheit. Ein sicherungstechnisches Verfahren, bei dem Nachrichten im Nachrichtendeskriptor Angaben zu ihrem Ursprung enthalten müssen.

L

Leistungsereignis. Eine Ereignisart, die bei Auftreten einer Grenzbedingung ausgegeben wird.

Leistungs-Trace. Eine Trace-Option in MQSeries, bei der Trace-Daten für die Leistungsanalyse und -optimierung herangezogen werden.

Lineare Protokollierung. In MQSeries auf UNIX-Systemen, MQSeries for OS/2 Warp und MQSeries for Windows NT and Windows 2000 das Speichern von Daten für den Wiederanlauf in einer Dateifolge. Je nach Bedarf werden neue Dateien erstellt. Der Speicherbereich, in den die Daten geschrieben werden, wird erst nach einem Wiederanlauf des WS-Managers freigegeben. Vgl. *Zyklische Protokollierung*.

Logische Arbeitseinheit (LUW). Siehe *Arbeitseinheit*.

Lokale Definition. Ein MQSeries-Objekt, das einem lokalen WS-Manager zugeordnet ist.

Lokale Definition einer fernen Warteschlange. Ein MQSeries-Objekt, das einem lokalen WS-Manager zugeordnet ist. Dabei definiert dieses Objekt die Attribute einer Warteschlange, die einem anderen WS-Manager zugeordnet ist. Darüber hinaus wird dieses Objekt zur Aliasnamensumsetzung für WS-Manager und für Warteschlangen für zu beantwortende Nachrichten verwendet.

Lokaler WS-Manager. Der Warteschlangenmanager, zu dem eine Verbindung mit einem Programm besteht und der diesem Programm Services für das Message-Queuing zur Verfügung stellt. Warteschlangenmanager, zu denen für ein Programm keine Verbindungen bestehen, werden als *ferne WS-Manager* bezeichnet, auch wenn sie auf demselben System wie das Programm aktiv sind.

Lokale Warteschlange. Eine Warteschlange, die zu einem lokalen WS-Manager gehört. Eine lokale Warteschlange kann eine Liste der Nachrichten enthalten, die zur Verarbeitung anstehen. Vgl. *ferne Warteschlange*.

M

MCA (Message Channel Agent). Siehe *Nachrichtenkanalagent*.

MCI (Message Channel Interface). Die MQSeries-Schnittstelle, mit der benutzerdefinierte bzw. kommerzielle Programme kompatibel sein müssen, die Nachrichten zwischen dem MQSeries-WS-Manager und anderen Nachrichtenübertragungssystemen übertragen. Bestandteil von MQSeries Framework.

Message-Queuing. Eine Programmiermethode, bei der jedes Programm innerhalb einer Anwendung mit anderen Programmen kommuniziert, indem Nachrichten in Warteschlangen eingereiht werden.

Modellwarteschlangenobjekt. Eine Gruppe von Warteschlangenattributen, die bei Erstellung einer dynamischen Warteschlange durch das Programm als Schablone dienen.

MQAI. MQSeries Administration Interface (siehe *MQSeries-Verwaltungsschnittstelle*).

MQI-Client. Eine Komponente des MQSeries-Produkts, die auf einem System installiert werden kann, ohne dass ein vollständiger WS-Manager installiert sein muss. Der MQSeries-Client empfängt MQI-Aufrufe von Anwendungen und kommuniziert mit einem WS-Manager auf einem Serversystem.

MQI-Kanal. Stellt die Verbindung zwischen einem MQI-Client und einem WS-Manager auf einem Serversystem her; über diesen Kanal werden nur MQI-Aufrufe und Antworten in beide Richtungen übertragen. Vgl. *Nachrichtenkanal*.

MQI (Message Queuing Interface). Die von MQSeries-WS-Managern zur Verfügung gestellte Programmierschnittstelle. Mit dieser Programmierschnittstelle erhalten Anwendungsprogramme Zugriff auf die Message-Queuing-Services.

MQI-Server. Bei einem MQI-Server handelt es sich um einen WS-Manager, der Queuing-Services für einen oder mehrere Clients bereitstellt. Alle MQSeries-Objekte, z. B. Warteschlangen, sind nur auf dem WS-Managersystem, d. h. auf dem MQI-Serversystem, vorhanden. Ein Server kann in der Regel auch normale lokale MQI-Anwendungen unterstützen.

MQM (Message Queue Management). Siehe *Verwaltung von Nachrichtenwarteschlangen*.

MQSC. MQSeries-Befehle (MQSeries Commands).

MQSeries. Eine Produktfamilie lizenzierter IBM Programme, die Message-Queuing-Services zur Verfügung stellen.

MQSeries-Befehle (MQSC). Benutzerlesbare Befehle, mit deren Hilfe die Bearbeitung von MQSeries-Objekten möglich ist. Vgl. *Programmable Command Format (PCF)*.

MQSeries-Verwaltungsschnittstelle (MQSeries Administration Interface, MQAI). Eine Programmierschnittstelle in MQSeries.

N

Nachricht. Bei Anwendungen für das Message-Queuing Informationen, die zwischen Programmen ausgetauscht werden. Siehe auch *Permanente Nachricht* und *Nicht permanente Nachricht*. Bei der Systemprogrammierung handelt es sich hier um Informationen für den Terminalbediener bzw. den Systemadministrator.

Nachrichtendeskriptor. Steuerinformationen, die Nachrichtenformat und -darstellung beschreiben und als Teil der MQSeries-Nachricht übertragen werden. Das Format des Nachrichtendeskriptors hängt von der MQMD-Struktur ab.

Nachrichtenfolgenummerierung. Ein Programmierverfahren, bei dem Nachrichten während ihrer Übertragung über eine Kommunikationsverbindung eindeutige Nummern zugeordnet werden. Der empfangende Prozess kann anhand dieser Nummern feststellen, ob alle Nachrichten eingegangen sind, diese in ihrer ursprünglichen Reihenfolge in eine Warteschlange stellen und doppelt vorhandene Nachrichten löschen.

Nachrichtenkanal. Beim verteilten Message-Queuing ein Verfahren zur Weiterleitung von Nachrichten zwischen den verschiedenen WS-Managern. Ein Nachrichtenkanal besteht aus zwei Nachrichtenkanalagenten (ein Sender an einem, ein Empfänger am anderen Ende) sowie einer Kommunikationsverbindung. Vgl. *MQI-Kanal*.

Nachrichtenkanalagent (MCA, Message Channel Agent). Ein Programm, das vorbereitete Nachrichten aus einer Übertragungswarteschlange an eine Kommunikationsverbindung bzw. von einer Kommunikationsverbindung an eine Bestimmungswarteschlange überträgt. Siehe auch *Schnittstelle für Nachrichtenwarteschlangen (MQI)*.

Nachrichtenkanalschnittstelle (MCI, Message Channel Interface). Die MQSeries-Schnittstelle, mit der benutzerdefinierte bzw. kommerzielle Programme kompatibel sein müssen, die Nachrichten zwischen dem MQSeries-WS-Manager und anderen Nachrichtenübertragungssystemen übertragen. Bestandteil von MQSeries Framework.

Nachrichtenpriorität. In MQSeries ein Nachrichtenattribut, das die Abrufreihenfolge aus einer Warteschlange festlegt und das angibt, ob ein Auslöseereignis generiert wird.

Nachrichtenübertragung. Siehe *Synchrone Nachrichtenübertragung* und *Asynchrone Nachrichtenübertragung*.

Nachrichtenwarteschlange. Synonym für *Warteschlange*.

Namensliste. Ein MQSeries-Objekt, das eine Liste mit Namen (z. B. Warteschlangennamen) enthält.

Namensserviceschnittstelle (NSI). Eine MQSeries-Schnittstelle, mit der benutzerdefinierte oder kommerzielle Programme, die das Eigentumsrecht an Warteschlangennamen auflösen, kompatibel sein müssen. Bestandteil von MQSeries Framework.

NetBIOS. Network Basic Input/Output System. Eine Betriebssystemschnittstelle für Anwendungsprogramme auf IBM PCs, die mit einem IBM Token-Ring-Netz verbunden sind.

Nicht permanente Nachricht. Eine Nachricht, die nach dem Wiederanlauf eines WS-Managers nicht mehr vorhanden ist. Vgl. *Permanente Nachricht*.

NSI. Name Service Interface (siehe *Namensservice-schnittstelle*).

Nullzeichen. Das Zeichen, das mit 'X'00' dargestellt wird.

O

Objekt. In MQSeries handelt es sich bei einem Objekt um einen WS-Manager, eine Warteschlange, eine Prozessdefinition, eine Namensliste oder einen Kanal.

Objektberechtigungsmanager (OAM). In MQSeries auf UNIX-Systemen, MQSeries for Compaq und MQSeries for Windows NT and Windows 2000 der Standardberechtigungs-service für die Befehls- und Objektverwaltung. Der OAM kann durch benutzereigene Sicherheitsservices ersetzt oder in Kombination mit diesen eingesetzt werden.

Objektdeskriptor. Eine Datenstruktur zur Beschreibung eines bestimmten MQSeries-Objekts (MQOD). Der Deskriptor enthält Objektname und -art.

Objektkennung. Die Kennung bzw. das Token, über die bzw. das ein Programm Zugriff auf das MQSeries-Objekt erhält, das es bearbeitet.

P

PCF. Programmable Command Format.

PCF-Befehl. Siehe *Programmable Command Format*.

Permanente dynamische Warteschlange. Eine dynamische Warteschlange, die nach dem Schließen nur gelöscht wird, wenn dies explizit gefordert wird. Permanente dynamische Warteschlangen werden im Anschluss an einen WS-Managerausfall wiederhergestellt, so dass sie auch permanente Nachrichten enthalten können. Vgl. *Temporäre dynamische Warteschlange*.

Permanente Nachricht. Eine Nachricht, die auch nach dem Wiederanlauf eines WS-Managers noch vorhanden ist. Vgl. *Nicht permanente Nachricht*.

Ping. Bei der verteilten Steuerung von Warteschlangen eine Diagnosehilfe, die über eine Testnachricht feststellt, ob ein Nachrichtenkanal einwandfrei arbeitet.

Plattform. In MQSeries das Betriebssystem, unter dem ein WS-Manager ausgeführt wird.

Programmable Command Format (PCF, programmierbares Befehlsformat). Eine MQSeries-Nachrichtenart, die von folgenden Komponenten verwendet wird:

- von Anwendungen für die benutzerspezifische Verwaltung, die PCF-Befehle in die systemspezifische Befehlseingabewarteschlange eines bestimmten WS-Managers einreihen;
- von Anwendungen für die benutzerspezifische Verwaltung, um das Ergebnis eines PCF-Befehls aus einem bestimmten WS-Manager abzurufen;
- von einem WS-Manager, um auf das Auftreten eines Ereignisses hinzuweisen.

Vgl. *MQSC*.

Protokoll. In MQSeries eine Datei, in der die Aktivitäten der WS-Manager beim Empfangen, Übertragen und Zustellen von Nachrichten aufgezeichnet werden. Dies ermöglicht bei Auftreten eines Fehlers die Wiederherstellung.

Prozessdefinitionsobjekt. Ein MQSeries-Objekt, das die Definition einer MQSeries-Anwendung enthält. Diese Definition wird beispielsweise von einem WS-Manager bei der Verwendung von Auslösenachrichten verwendet.

Prüfpunkt. Der Zeitpunkt, zu dem wichtige Informationen in das Protokoll geschrieben werden. Vgl. *Synchronisationspunkt*. In MQSeries auf UNIX-Systemen der Zeitpunkt, zu dem ein Datensatz im Protokoll mit dem Datensatz in der Warteschlange identisch ist. Prüfpunkte werden automatisch generiert und werden beim Systemneustart verwendet.

PTF. Program Temporary Fix (siehe *Vorläufige Programmkorrektur*).

R

RBA. Relative Byte Address (siehe *Relative Byteadresse*).

Regeltabelle. Eine Steuerdatei mit einer oder mehreren Regeln, die von der Steuerroutine für Warteschlangen für nicht zustellbare Nachrichten auf die Nachrichten in dieser Warteschlange angewandt werden.

Repository. Eine Reihe von Informationen zu den WS-Managern, die Teil eines Clusters sind. Zu diesen Informationen gehören Name, Adresse und Kanäle der WS-Manager, die Namen der Hosts, in denen sich die Warteschlangen befinden usw.

Requester-Kanal. Beim Message-Queuing ein Kanal, der von einem Senderkanal fern gestartet werden kann. Der Requester-Kanal empfängt über eine Kommunikationsverbindung Nachrichten von dem Senderkanal und reiht diese in die lokale Warteschlange ein, die in den Nachrichten angegeben ist. Siehe auch *Serverkanal*.

Responder. Bei der verteilten Steuerung von Warteschlangen ein Programm, das auf Netzverbindungsanforderungen anderer Systeme antwortet.

Resynchronisation. In MQSeries eine Option, mit der ein Kanal gestartet und angewiesen werden kann, alle unbestätigten Statusnachrichten ohne einen Neustart der Nachrichtenübertragung aufzulösen.

ROLLBACK-Operation. Synonym für *Zurücksetzen*.

Rückkehrcodes. Sammelbegriff für Beendigungscode und Ursachencode.

Rückruf. Ein Requester-Nachrichtenkanal leitet Übertragungen von einem Senderkanal ein, indem er zunächst den Sender aufruft. Anschließend wird der Requester-Nachrichtenkanal beendet und wartet auf einen Rückruf.

S

Schnittstelle für Nachrichtenwarteschlangen (MQI, Message Queue Interface). Die von MQSeries-WS-Managern zur Verfügung gestellte Programmierschnittstelle. Mit dieser Programmierschnittstelle erhalten Anwendungsprogramme Zugriff auf die Message-Queuing-Services.

Schnittstelle für Sicherheitsaktivierung (SEI, Security Enabling Interface). Eine Schnittstelle in MQSeries, mit der benutzerdefinierte oder kommerzielle Programme, die Berechtigungen prüfen, Benutzer-IDs zur Verfügung stellen oder Authentifizierungsvorgänge durchführen, kompatibel sein müssen. Bestandteil von MQSeries Framework.

SEI. Security Enabling Interface (siehe *Schnittstelle für Sicherheitsaktivierung*).

Senderkanal. Beim Message-Queuing ein Kanal, der Übertragungen einleitet sowie Nachrichten aus Übertragungswarteschlangen entfernt und über eine Kommunikationsverbindung in einen Empfänger- oder Requester-Kanal stellt.

Sequenzielle Zustellung. In MQSeries ein Verfahren, bei dem Nachrichten mit einer Folgenummer übertragen werden; dies ermöglicht es dem Empfängerkanal, die Nachrichten in der richtigen Reihenfolge zu speichern. Dies ist erforderlich, wenn Nachrichten nur einmal und in der richtigen Reihenfolge übertragen werden können.

Server. (1) In MQSeries ein WS-Manager, der einer Client-Anwendung, die auf einer fernen Workstation ausgeführt wird, Warteschlangenservices zur Verfügung stellt. (2) Ein Programm, das in einer Client/Server-Umgebung (ein Zweiprogrammmodell, bei dem zwischen beiden Programmen Informationen ausgetauscht werden) auf Informationsanforderungen antwortet. Siehe auch *Client*.

Serverkanal. Beim Message-Queuing ein Kanal, der auf einen Requester-Kanal reagiert sowie Nachrichten

aus Übertragungswarteschlangen entfernt und über eine Kommunikationsverbindung in einem Requester-Kanal stellt.

Serviceintervall. Ein Zeitintervall, mit dem die abgelaufene Zeit zwischen einem PUT- bzw. GET-Vorgang und einem anschließenden GET-Vorgang vom WS-Manager verglichen wird. Anhand dieses Wertes entscheidet der WS-Manager ob ein Serviceintervallereignis ausgegeben wird oder nicht. Das Serviceintervall für eine Warteschlange wird über ein Warteschlangenattribut angegeben.

Serviceintervallereignis. Ein Ereignis, das in Zusammenhang mit dem Serviceintervall ausgegeben wird.

SIT. System Initialization Table (siehe *Systeminitialisierungstabelle*).

SNA. Systems Network Architecture (Systemnetzwerkarchitektur).

Sofortiger Abschluss. In MQSeries der Systemabschluss eines WS-Managers, ohne eine Verbindungsunterbrechung der Anwendungen abzuwarten. Aktuelle MQI-Aufrufe können abgeschlossen werden, neue MQI-Aufrufe hingegen können bei Anforderung eines sofortigen Abschlusses nicht ausgeführt werden. Vgl. *Gesteuerter Abschluss* und *Erzwungener Abschluss*.

Speichern und weiterleiten. Die temporäre Speicherung von Datenpaketen, Nachrichten oder Blöcken in einem Datennetz, bevor sie an die Zieladresse weitergeleitet werden.

Standardobjekt. Die Definition eines Objekts (z. B. einer Warteschlange), in der alle Attribute angegeben sind. Bei der Definition eines Objekts durch den Benutzer, bei der nur Angaben zu bestimmten Attributen erfolgen, übernimmt der WS-Manager für alle nicht angegebenen Attribute die Standardwerte.

Statusserver. Der Statusserver unterstützt die Statusinformationen für alle Objekte, bei denen es sich nicht um lokale Warteschlangen handelt. Darüber hinaus verwaltet der Standardstatusserver auch noch Kanalstatusinformationen für den WS-Manager.

Steuerroutine der Warteschlange für nicht zustellbare Nachrichten. Ein von MQSeries zur Verfügung gestelltes Dienstprogramm zur Überwachung von Warteschlangen für nicht zustellbare Nachrichten und zur Verarbeitung von Nachrichten in der Warteschlange anhand einer benutzerdefinierten Regeltable.

Steuerung von fernen Warteschlangen. Beim Message-Queuing Services, die es Anwendungen ermöglichen, Nachrichten in Warteschlangen einzureihen, die anderen WS-Managern zugeordnet sind.

Steuerung von Warteschlangen (Queuing). Siehe *Message-Queuing*.

Stilllegung. In MQSeries der Status eines WS-Managers, bevor er gestoppt wird. In diesem Status können alle Programme ordnungsgemäß beendet werden, es können jedoch keine neuen Programme gestartet werden.

Symptomzeichenfolge. Diagnoseinformationen, die in einem strukturierten Format angezeigt werden, das für die Suche in der Datenbank der IBM Softwareunterstützung konzipiert wurde.

Synchrone Nachrichtenübertragung. Ein Kommunikationsverfahren zwischen verschiedenen Programmen, bei dem von den einzelnen Programmen Nachrichten in Nachrichtenwarteschlangen eingereicht werden. Bei der synchronen Nachrichtenübertragung können die sendenden Programme mit der Verarbeitung momentan anstehender Vorgänge erst fortfahren, nachdem eine Antwort auf die Nachricht eingegangen ist. Vgl. *Asynchrone Nachrichtenübertragung*.

Synchronisationspunkt. Ein Punkt im Verlauf bzw. am Ende der Verarbeitung einer Transaktion, an dem sich die geschützten Ressourcen der Transaktion in einem konsistenten Zustand befinden. An einem Synchronisationspunkt können Änderungen an den Ressourcen ohne Bedenken festgeschrieben oder auf den vorherigen Synchronisationspunkt zurückgesetzt werden.

Systemabschluss. Siehe *Sofortiger Abschluss*, *Präventiver Abschluss* und *Gesteuerter Abschluss*.

Systembehälter (System Bag). Eine besondere Art von Datenbehälter (Data Bag), die von der Verwaltungsschnittstelle MQAI erstellt wird.

SYSTEM.COMMAND.INPUT. Eine lokale Warteschlange, in die Anwendungsprogramme MQSeries-Befehle stellen können. Die Befehle werden vom Befehlsserver aus der Warteschlange abgerufen, überprüft und anschließend zu ihrer Ausführung an den Befehlsprozessor übergeben.

Systemsteuerbefehle. Befehle für die Manipulation plattformspezifischer Einheiten wie beispielsweise Pufferpools, Speicherklassen oder Dateien.

T

TACL. Tandem Advanced Command Language.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

Temporäre dynamische Warteschlange. Eine dynamische Warteschlange, die beim Schließen gelöscht wird. Temporäre dynamische Warteschlangen werden im Anschluss an einen WS-Managerausfall nicht wieder-

hergestellt, so dass sie nur nicht permanente Nachrichten enthalten können. Vgl. *Permanente dynamische Warteschlange*.

Thread. In MQSeries die niedrigste Ebene, auf der in einem Betriebssystem eine parallele Verarbeitung möglich ist.

TMF. Transaction Management Facility.

TMI. Trigger Monitor Interface (siehe *Auslösemonitor-schnittstelle*).

TM/MP. NonStop Transaction Manager/MP.

tranid. Siehe *Transaktions-ID*.

Transmission Control Protocol/Internet Protocol (TCP/IP). Eine Reihe von Übertragungsprotokollen die Peer-zu-Peer-Konnektivitätsfunktionen für LAN- und WAN-Netze unterstützt.

Transmission Control Protocol (TCP). Bestandteil der TCP/IP-Protokollfamilie. Ein Host-zu-Host-Protokoll für die Kommunikation zwischen Hosts in einem DVF-Netz mit Paketvermittlung. TCP ermöglicht die verbindungsorientierte Zustellung von Datenströmen. Die Zustellung ist zuverlässig und sicher.

U

Übertragungsprogramm. Siehe *Nachrichtenkanalagent*.

Übertragungswarteschlange. Eine lokale Warteschlange, in der vorbereitete Nachrichten an einen ferneren WS-Manager temporär gespeichert werden.

UDP. User Datagram Protocol.

Ursachencode. Ein Rückkehrcode, der Aufschluss über die Ursache eines fehlgeschlagenen bzw. teilweise fehlgeschlagenen MQI-Aufrufs gibt.

User Datagram Protocol (UDP). Bestandteil der TCP/IP-Protokollfamilie. Ein Protokoll auf Paketebene, das direkt auf die IP-Schicht aufsetzt. Bei UDP handelt es sich um eine nicht verbindungsorientierte Alternative zu TCP, die jedoch weniger zuverlässig ist. Dieses Protokoll wird für die Kommunikation zwischen Anwendungsprogrammen auf verschiedenen TCP/IP-Host-Systemen verwendet.

V

Verbinden. Der Vorgang, bei dem die Kennung für eine WS-Managerverbindung zur Verfügung gestellt wird, die anschließend von einer Anwendung in MQI-Aufrufen verwendet wird. Die Verbindung wird entweder durch den MQCONN-Aufruf oder automatisch durch den MQOPEN-Aufruf hergestellt.

Verbindungskennung. Die Kennung bzw. das Token, über die bzw. das ein Programm Zugriff auf den WS-Manager erhält, mit dem es verbunden ist.

Verteilte Anwendung. Beim Message-Queuing eine Gruppe von Anwendungsprogrammen, die einzeln jeweils mit verschiedenen WS-Managern verbunden sein können, zusammen jedoch eine einzige Anwendung darstellen.

Verteilte Warteschlangenverwaltung (DQM). Beim Message-Queuing die Konfiguration und Steuerung von Nachrichtenkanälen zu WS-Managern auf anderen Systemen.

Verwaltungsbefehle. MQSeries-Befehle zum Verwalten von MQSeries-Objekten wie beispielsweise Warteschlangen, Prozesse oder Namenslisten.

Verwaltungsbehälter (Administration Bag). In Zusammenhang mit der Verwaltungsschnittstelle MQAI ein Datenbehälter (Data Bag), der für die MQSeries-Verwaltung erstellt wird und für den die Optionen zum Ändern der Reihenfolge der Datenelemente, zum Erstellen von Listen und zum Überprüfen der Selektoren in einer Nachricht automatisch aktiviert sind.

Verwaltung von Nachrichtenwarteschlangen (MQM). Die Verwaltung von Nachrichtenwarteschlangen (MQM, Message Queue Management) ist eine Einrichtung in MQSeries for Compaq NSK Version 2.2, die PCF- und Steuerbefehle verwendet. MQM wird als PATHWAY-SCOBOL-Requester unter TCP (Terminal Control Process) ausgeführt und verwendet einen MMQ-Serverklassenserver, der die C-Anwendungsprogrammierschnittstelle zur Ausführung von PCF-Befehlen aufruft. Für jeden auf einem System konfigurierten WS-Manager ist ein eigenes MQM-Exemplar vorhanden, da jeder WS-Manager über eine eigene PATHWAY-Konfiguration verfügt. Daher kann mit MQM nur der jeweils zugeordnete WS-Manager verwaltet werden.

Vorläufige Programmkorrektur (PTF). Die temporäre Lösung bzw. Umgehung von Problemen, die von IBM Ingenieuren im Außendienst gefunden und durch Fehler in einem aktuellen, unveränderten Programm-Release verursacht wurden.

W

Warteschlange. Ein MQSeries-Objekt. Anwendungen für Message-Queuing können Nachrichten in eine Warteschlangen einreihen bzw. aus ihr abrufen. Eine Warteschlange ist einem WS-Manager zugeordnet, von dem sie verwaltet wird. Lokale Warteschlangen können Listen mit Nachrichten enthalten, die zur Verarbeitung anstehen. Andere Warteschlangenarten können keine Nachrichten enthalten; sie verweisen lediglich auf andere Warteschlangen oder können als Modell für dynamische Warteschlangen verwendet werden.

Warteschlange für nicht zugestellte Nachrichten. Siehe *Warteschlange für nicht zustellbare Nachrichten*.

Warteschlange für nicht zustellbare Nachrichten. Eine Warteschlange, in die von WS-Managern bzw. Anwendungen Nachrichten eingereiht werden, die nicht an die eigentliche Zieladresse übertragen werden können.

Warteschlangen für zu beantwortende Nachrichten. Gibt die Warteschlange an, an die auf Anforderung des Programms, das einen MQPUT-Aufruf abgesetzt hat, eine Antwort- bzw. Berichtsnachricht gesendet werden soll.

Warteschlangenserver. Eine NonStop-Prozesspaar, das alle Messaging-Operationen für lokale Warteschlangen unterstützt.

Wert für Folgenummernserie. In MQSeries ein Verfahren, mit dem gewährleistet wird, dass die Nachrichtenfolgennummern an beiden Enden der Kommunikationsverbindung gleichzeitig zurückgesetzt werden. Durch die Übertragung von Nachrichten mit Folgenummern wird sichergestellt, dass der Empfangskanal die Nachrichten in der ursprünglichen Reihenfolge speichern kann.

WS-Manager. Ein Systemprogramm, das Anwendungen Services für die Steuerung von Warteschlangen zur Verfügung stellt. Der WS-Manager stellt eine Anwendungsprogrammierschnittstelle zur Verfügung, die anderen Programmen den Zugriff auf Nachrichten in den Warteschlangen ermöglicht, die dem WS-Manager zugeordnet sind. Siehe auch *Lokaler WS-Manager* und *Ferner WS-Manager*. Ein MQSeries-Objekt, das die Attribute eines bestimmten WS-Managers definiert.

WS-Managerereignis. Ein Ereignis, das auf Folgendes hinweist:

- das Auftreten einer Fehlerbedingung in Zusammenhang mit den Ressourcen, auf die von einem WS-Manager zugegriffen wird. (Z. B. Eine Fehlerbedingung, die auftritt, wenn eine Warteschlangen nicht verfügbar ist.)
- Im WS-Manager ist eine erhebliche Änderung aufgetreten. Beispiel: Der WS-Manager wurde gestoppt bzw. gestartet.

Z

Zeilengruppe. Eine Gruppe von Zeilen in einer Konfigurationsdatei, die einen Wert für einen Parameter festlegt, der das Verhalten eines WS-Managers, Clients oder Kanals bestimmt. In MQSeries-Systemen kann eine Konfigurationsdatei (INI-Datei) eine Reihe von Zeilengruppen enthalten.

Zeitunabhängige Nachrichtenübertragung. Siehe *Asynchrone Nachrichtenübertragung*.

Zurücksetzen. Ein Vorgang, bei dem alle Änderungen wieder rückgängig gemacht werden, die während der aktuellen Arbeitseinheit mit Wiederherstellung bzw. der aktuellen Arbeitseinheit vorgenommen wurden. Nach Abschluss dieses Vorgangs beginnt eine neue Arbeitseinheit mit Wiederherstellung bzw. eine neue Arbeitseinheit.

Zweiphasige Festschreibung. Ein Protokoll für die Koordinierung von Änderungen an wiederherstellbaren Ressourcen, wenn von einer einzelnen Transaktion mehrere Ressourcenmanager verwendet werden. Vgl. *Einphasige Festschreibung*.

Index

A

- Abrufalgorithmus für Nachrichten 10
 - ADD SERVER, PATHCOM-Befehl 34
 - Agentenprozesse 28, 54
 - Aktionsschlüsselwörter 168
 - Aktivieren
 - Instrumentierungsereignisse 177
 - Sicherheit 146
 - Aktuelle Warteschlangenlänge 107
 - Aliasname
 - Alias 13
 - Anwendung, für Auslösefunktion definieren 117
 - arbeiten mit 105
 - Attribute 12, 353
 - Attribute, ändern 108
 - Befehl 15
 - Berechtigungen für 150
 - Beschreibung 9
 - Cluster-
 - Übertragungswarteschlange 229
 - definieren 12
 - durchsuchen 111
 - dynamisch 10
 - Ereignis 16, 176
 - ferne WS-Manager 12, 13
 - Aliasnamen für WS-Manager 136
 - arbeiten mit 136
 - erstellen 133
 - für MQI-Anwendungen 95
 - für nicht zustellbare Nachrichten 15, 47
 - für zu beantwortende Nachrichten 16, 136
 - Initialisierung
 - Auslösenachrichten 14
 - definieren 119
 - Länge 106
 - Länge ermitteln 233
 - lokale 12, 13
 - definieren 105
 - Inhalt löschen 110
 - kopieren 108
 - löschen 111
 - mit Aliasnamen arbeiten 114
 - Modell 14
 - arbeiten mit 116
 - definieren 116
 - nicht zugestellte Nachricht 15, 47
 - Objekte
 - Alias 13
 - ferne WS-Manager 13
 - lokale 13
 - Modell 14
 - physische Länge 106
 - temporär 10
 - Übertragung 15
 - definieren 129
 - erstellen 136
 - Fernverwaltung 128
 - Standardeinstellung 47
 - Aliasname (*Forts.*)
 - Übertragung 15 (*Forts.*)
 - standardmäßige 136
 - verteilte, falsche Ausgaben 217
 - vordefiniert 10
 - Aliasnamen
 - Warteschlangen für zu beantwortende Nachrichten 137
 - WS-Manager 136
 - Aliaswarteschlangen
 - Berechtigungen für 150
 - Beschreibung 13
 - AllQueueManagers, Zeilengruppe in der Datei MQSINI 194
 - ALTER SERVER
 - PATHCOM-Befehl 35
 - Alternative Benutzerberechtigung 151
 - altmqfls, Befehl 109, 260
 - Datenstruktur beim Start des Warteschlangenservers erstellen 232
 - Prüfpunktinformationen für nicht permanente Nachrichten 232
 - Schwellenwert für Suchvorgänge festlegen 231
 - Speichersperre des Warteschlangenservers 232
 - Warteschlangendateien verschieben 230
 - Warteschlangenlänge ermitteln 233
 - zugehörige Befehle 264
 - altmqsr, Befehl 265
 - zugehörige Befehle 266
 - ändern
 - CCSID 53
 - Kanal 85
 - Warteschlange 82
 - Ändern, Warteschlangenattribute 108
 - Ändern, WS-Managerattribute 100
 - Änderungsdatei 228, 245
 - verschieben 230
 - Angegebene Betriebsumgebung 323
 - Anwendung
 - ausführen 362
 - ausgelöste 358
 - Auswirkung beim Durchsuchen permanenter Nachrichten in einer Warteschlange 231
 - Daten 8
 - Einschränkungen bei Verwendung von FASTPATH 236
 - FASTPATH-Bindung 361
 - Wiederherstellung nach Ausfall 246
 - gleichzeitige Transaktionen verwalten 245
 - Guardian 360
 - kompilieren und binden 361
 - MQI-Verwaltung, Unterstützung 95
 - OpenTMF verwenden 244
 - optimieren 225
 - Programmierfehler, Beispiele 209
- Anwendung (*Forts.*)
 - schreiben 355
 - sicher 234
 - STANDARD-Bindung 361
 - Wiederherstellung nach Ausfall 246
 - Überlegungen zum Design 214
 - zeitunabhängig 7
 - Anwendungen kompilieren und binden 361
 - Anzeigen
 - Befehlsserver, Befehl 283
 - Berechtigung, Befehl 279
 - formatierte MQSeries-Trace-Ausgabe anzeigen, Befehl 287
 - Kanalstatus 90
 - MQM 73
 - MQSeries-Dateien, Befehl 284
 - Prozessdefinitionen 120
 - Status des Befehlsservers 124
 - WS-Managerattribute 99
 - APPLIDAT-Schlüsselwort, Regeltabelle 166
 - APPLNAME-Schlüsselwort, Regeltabelle 167
 - APPLTYPE-Schlüsselwort, Regeltabelle 167
 - Arbeitseinheit
 - Beschreibung 21
 - Arbeitseinheiten verwalten 356
 - Arten, Objekte 10
 - Attribute von MQSeries-Objekten
 - Aliasname 12
 - ALL, Attribut 107
 - ändern, lokale Warteschlange 108
 - MQSC und PCF im Vergleich 33
 - Standardgruppe 107
 - WS-Manager
 - ändern 100
 - anzeigen 99
 - Auflösen, Kanal 93
 - Aufzeichnung von Ereignissen 176
 - Ausfall einzelner Komponenten 238
 - Ausfall mehrerer Komponenten 238
 - Ausführungs-Controller
 - Funktion 24
 - Steuerdatei 245
 - Wiederherstellung nach Ausfall 246
 - Ausführungs-Controller-Prozesse, Anzahl 50
 - Ausgangsdatenträger des WS-Managers 50
 - Ausgelöste Programme 358
 - USERDATA übergeben an 345
 - Auslagerungsspeicher zuordnen 59
 - Auslösefunktion
 - Ereignisse 176
 - Monitor
 - Beschreibung 14
 - mit PATHCOM-Befehlen konfigurieren 35

Auslösefunktion (*Forts.*)
 Startbefehl 305
 Nachrichten in einer Initialisierungswarteschlange 14

Auslösen
 Anwendungswarteschlange definieren für 117
 Definition 8
 Objekte verwalten 117

Authority, Zeilengruppe in der Datei QMINI 196

Automatische Definition von Kanälen 131

B

Bedienerbefehle, ohne Antwort 211

Bedienernachrichten 220

Beenden, WS-Manager 71

Befehle
 Attribute der Warteschlangendateien ändern (altmqfls) 260

Auslösemonitor starten (runmqtrm) 305

Befehle für Sicherheitsfunktion
 dspmqaut 149
 setmqaut 146

Befehlsserver anzeigen (dspmqcsv) 283

Befehlsserver beenden (endmqcsv) 290

Befehlsserver starten (strmqcsv) 312

Berechtigung anzeigen (dspmqaut) 279

Berechtigung setzen/zurücksetzen (setmqaut) 148, 306

Client-Kanaldefinitionen konvertieren (cnvclchl) 269

DLQ-Steueroutine ausführen (runmqdlq) 163

Empfangsprogramm ausführen (runmqlsr) 300

formatierten MQSeries-Trace anzeigen (dspmqtrc) 287

Informationen zu einem bestimmten Principal anzeigen (dspmqusr) 288

Kanal ausführen (runmqchl) 297

Kanalinitiator ausführen (runmqchi) 296

MQSC-Befehl
 ALTER QLOCAL 108
 ALTER QREMOTE 135
 Befehlsdateien 101
 DEFINE CHANNEL 129
 DEFINE QALIAS 114
 DEFINE QLOCAL 108
 DEFINE QLOCAL LIKE 108
 DEFINE QLOCAL REPLACE 108
 DEFINE QMODEL 116
 DEFINE QREMOTE 133
 DELETE QLOCAL 111
 DISPLAY QREMOTE 135
 prüfen 103
 verwenden 33

MQSeries for Compaq NSK installieren (instmqm) 295

Befehle (*Forts.*)
 MQSeries-Trace beenden (endmqtrc) 294

MQSeries-Trace starten (strmqtrc) 315

PCF-Befehle 33

Principal, der der Compaq NSK-Benutzer-ID entspricht, definieren (altmqusr) 265

RDF-Verwaltungsaufgaben ausführen (cleanrdf) 267

runmqsc 97

Steueroutine der DLQ ausführen 298

Steuerung 31
 altmqfls 260
 altmqusr 265
 cleanrdf 267
 cnvclchl 269
 crtmqcvx 271
 crtmqm 273
 dltnmqm 277
 dspmqaut 279
 dspmqcsv 283
 dspmqfls 284
 dspmqtrc 287
 dspmqusr 288
 endmqcsv 290
 endmqm 292
 endmqtrc 294
 instmqm 295
 runmqchi 296
 runmqchl 297
 runmqdlq 298
 runmqlsr 300
 runmqsc 302
 runmqtrm 305
 setmqaut 306
 strmqcsv 312
 strmqm 313
 strmqtrc 315
 upgmqm 318

Übersicht über die Befehlssätze 333

Upgrade eines WS-Managers der Version 2.2.0.1 (upgmqm) 318

WS-Manager beenden (endmqm) 292

WS-Manager erstellen (crtmqm) 273

WS-Manager löschen (dltnmqm) 277

WS-Manager starten (strmqm) 313

Befehle, Fehler 210

Befehlsdateien 101

Befehlssatz
 Übersicht 333
 Verwaltung 31

Befehlsserver
 Anzeigebefehl 283
 Beenden, Befehl 290
 Fernverwaltung 123
 mit PATHCOM-Befehlen konfigurieren 34
 Startbefehl 312
 starten 123
 Status anzeigen 124
 stoppen 124
 Wiederherstellung nach Ausfall 246

Befehlswarteschlange 15

Beispiel für Trace-Daten 223

Beispiele
 altmqfls, Befehl 264
 altmqusr, Befehl 266
 cleanrdf, Befehl 268
 cnvclchl, Befehl 270
 crtmqcvx, Befehl 271
 crtmqm, Befehl 276
 dltnmqm, Befehl 277
 dspmqaut, Befehl 282
 dspmqcsv, Befehl 283
 dspmqfls, Befehl 285
 dspmqusr, Befehl 289
 endmqcsv, Befehl 291
 endmqm, Befehl 293
 endmqtrc, Befehl 294
 Fehlerprotokoll 220
 Kommunikationskonfiguration 390
 Programmierfehler 209
 runmqsc, Befehl 304
 setmqaut, Befehl 311
 strmqcsv, Befehl 312
 strmqm, Befehl 313
 strmqtrc, Befehl 317
 upgmqm, Befehl 319

Beispielprogramme
 C++-Versionen erstellen 368
 COBOL-Versionen erstellen 369
 im Lieferumfang von MQSeries for Compaq NSK enthalten 365
 TAL-Versionen erstellen 372
 Versionen in C erstellen 366
 Versionen unter OSS erstellen 372

Benutzer
 Gruppen 143
 mehreren Benutzergruppen zugeordnet 143

Benutzer-Exit
 Beschreibung 19
 Cluster-Workload 19
 Datenkonvertierungs-Exit 19
 Kanal-Exit 19

Benutzer-Exits 373

Benutzer-ID
 Berechtigung 141, 150
 Compaq NSK, angemeldeter Benutzer 150

Benutzerdefinierte Nachrichtenformate 139

Benutzerdefinierte Serverklassen 45

Benutzergruppe
 für Berechtigungen 144
 MQM 141

Berechtigung
 alternativer Benutzer 151
 Befehle 149
 Benutzergruppen 144
 Dateien 159
 dspmqaut, Befehl 149
 Ereignisse 405
 installierbare Services 149
 Kontext 151
 Listen 147
 MQI 154
 setmqaut, Befehl 149
 setzen/zurücksetzen, Befehl 306
 Verwaltung 157

Berechtigungsservice 20

Berechtigungsservice 20 (*Forts.*)
 Schnittstelle 412

Bibliotheken
 MQI 361

Bindungen 25

BookManager 421

Bücher im PostScript-Format 421

C

C++ (nativ)
 unterstützte Programmiersprache 359

C (nativ)
 unterstützte Programmiersprache 359

C (nicht nativ)
 unterstützte Programmiersprache 359

CCSID, Datei 138

CCSID, Parameter 53

CCSIDs
 Konvertierungsverfahren 54
 unterstützteMQSeries for Compaq NSK 325

ChannelInitiator, Zeilengruppe in der Datei QMINI 196

Channels, Zeilengruppe in der Datei QMINI 197

Channels Menu, MQM 83

cleanrdf, Befehl 267

Client-Kanaldefinitionen 245

Clients 19, 403

Cluster
 aus WS-Managern 10
 Beschreibung 126
 fernes Queuing 125
 Netzverfügbarkeit erhöhen 253
 Workload-Management-Exits 373
 WS-Manager
 Workload-Exit 20

Cluster-Übertragungswarteschlange 229

cnvclchl, Befehl 269

COBOL (nativ)
 unterstützte Programmiersprache 359

COBOL (nicht nativ)
 unterstützte Programmiersprache 359

Compaq NonStop Server for Java 339

Compaq NSK, angemeldete Benutzer-ID 150

Compaq NSK Fix Command 32

Configuration, Zeilengruppe in der Datei QMINI 195

CONNAME
 Prozessattribut 344

CorrelId
 Leistungsüberlegungen bei Verwendung von 214

CPU
 gleichmäßige Auslastung 36
 hohe Belegung 234
 Konfigurationsoptionen für Verfügbarkeit 253
 optimieren 225
 Warteschlangenserver 229

crtmqcvx, Befehl 271

crtmqm, Befehl 273

CURDEPTH 233

D

Datei
 Änderungsdatei 228, 245
 Adresse 230
 für Aliaswarteschlangen und ferne Warteschlangen 30

Berechtigung 159

ENSCRIBE 29

Initialisierung 27

Kanaldefinitionen, für andere als Client-Kanäle 245

Konfiguration
 Fehlerbestimmung 222

Nachrichtenüberlauf 228
 Adresse 230
 Schwellenwert ändern 231

Namensliste 30

Namenslistendefinitionen 245

OAM-Datenbank 245

Objektänderungsdatei 245

Objektkatalog 245

Principal-Datenbank 245

protokollierte Datenbankdateien 245

Steuerdatei für Ausführungs-Controller 245

Warteschlangendatei 228, 245
 Adresse 230

Warteschlangenüberlaufdatei 228, 245
 Adresse 230

Dateien
 Konfiguration
 MQSeries (MQSINI) 193
 WS-Manager (QMINI) 195

Datenbank
 durch TM/MP geschützt 242
 externe Konsistenz 243
 Konsistenz 242
 protokollierte Datenbankdateien 245

Datendateien, Teildatenträger des WS-Managers 65

Datenintegrität
 Definition 238
 Konfigurationsoptionen 254

Datenkonvertierung 138, 354
 benutzerdefinierte Nachrichtenformate konvertieren 139

crtmqcvx, Befehl 271

EBCDIC 53

Standarddatenkonvertierung 138

Datenkonvertierungs-Exits 373

Datenträger
 Struktur 64
 Warteschlangendateien partitionieren 230

Datenträger, ändern 109

Datentypen
 Strukturdatentypen 347

Deaktivieren, Objektberechtigungsmanager (OAM) 146

Debug
 erste Überprüfungen 205

Debug (*Forts.*)
 typische Programmierfehler 209
 weitere Überprüfungen 210, 214

DefaultPrefix (Parameter), Befehl crtmqm 50

DefaultProcess, Zeilengruppe in der Datei QMINI 195

DefaultQueueManager, Zeilengruppe in der Datei MQSINI 194

Definieren, Warteschlangen 79

DESTQ-Schlüsselwort, Regeltabelle 167

DESTQM-Schlüsselwort, Regeltabelle 167

DISPLAY CHSTATUS, Befehl 344

DLQ-Header, MQDLH 163

DLQ-Steueroutine
 aufrufen 163
 Regeltabelle 164

dltmqm, Befehl 277

dspmqaout, Befehl 279
 verwenden 146, 149

dspmqcvs, Befehl 283

dspmqlfs, Befehl 284

dspmqrtr, Befehl 287

dspmqrtr, Befehl 288

Dynamische Definition von Kanälen 131

Dynamische Warteschlangen 10
 Berechtigungen für 150
 Beschreibung 10

Dynamisches Binden 25

E

EBCDIC
 Datenkonvertierung 53

EC, Zeilengruppe in der Datei QMINI 196

ECBoss
 Funktion 24
 Wiederherstellung nach Ausfall 246

ECBoss, Funktion 51

ECBoss, Zeilengruppe in der Datei QMINI 196

Eingabe und Ausgabe umleiten, bei MQSC-Befehlen 100

Eingeben, MQSeries-Befehle 96

Einphasige Festschreibung 183

Einschränkungen
 Objektnamen 258
 Zugriff auf MQM-Objekte 141

Empfängerkanal, automatische Definition 131

Empfangsprogramm ausführen (runmqslr), Befehl 300

EMS-Ereignisschablone, MQSeries 413

EMS-Ereignisse 178, 221
 anderen Sammler angeben 182
 Programme für ihre Verarbeitung erstellen 182
 Standardsammler 182
 Umgebungsvariable MQEMSEVENTS festlegen 181

EMSCollectorName 54

endmqcvs, Befehl 290

endmqm, Befehl 71, 292

endmqtr, Befehl 294

ENSCRIBE-Dateien 22, 29

- Entfernen (manuell), WS-Manager 337
- Entfernen, Statusserver 61
- Entfernen, Warteschlangenserver 64
- Entfernen, zusätzliche Statusserver 60
- Entfernen, zusätzliche Warteschlangenserver 63
- Ereignisarten 176
- Ereignisgesteuerte Verarbeitung 8
- Ereignisnachrichten, Format 406
- Ereignisse
 - Aliasname 176
 - Auslösefunktion 176
 - EMS 413
 - Instrumentierung
 - aktivieren 177
 - Beschreibung 175
 - Funktion 175
 - Gründe für die Verwendung 175
 - Nachricht 177
 - Typen 176
 - Kanal 176
 - Typen 176
 - Unterstützung in MQSeries for Compaq NSK 405
- Ereigniswarteschlange 16
- Erneut starten, WS-Manager 72
- Erstellen
 - Kanaldefinition 86
 - Principals 69
 - Prozessdefinitionen 119
 - Übertragungswarteschlange 136
 - Warteschlange 79
 - WS-Manager 47, 68
- Erzwungene Beendigung von Kanälen (sofortiger Stopp) 91
- Erzwungener Abschluss des WS-Managers 71
- Euro-Unterstützung 325
- Exit
 - Benutzer-Exit 19
 - Benutzer-Exits 373
 - Cluster-Workload-Exit 19
 - installieren 378
 - Kanal-Exit 19
 - MQ_LOAD_ENTRY_POINT_EXIT 374
 - Namensformat 374
- ExpectedNumECs 50
- ExtPoolSize, Eintrag QMINI, Datei 59

F

- Falsche Ausgabe 215
- FASTPATH-Bindung 234, 361
 - aktivieren 236
 - Einschränkungen 236
 - Systemaufwand reduzieren 235
- FEEDBACK-Schlüsselwort, Regeltabelle 167
- Fehlerbehebung 187
- Fehlerbestimmung
 - falsche Ausgabe 217
 - keine Antwort von Befehlen 211
 - Konfigurationsdateien 222
 - Programmierfehler 209

- Fehlerbestimmung (*Forts.*)
 - was zuerst überprüft werden muss 205
 - weitere Überprüfungen 210, 214
- Fehlernachrichten 98
- Fehlerprotokoll 219
 - Beispiel 220
 - Fehler in der Startphase 220
 - Teildatenträger 66
- Ferne Warteschlangen 125
 - Beschreibung 12, 13
- Ferne WS-Manager
 - Aliasname
 - als WS-Managername 136
 - Berechtigungen für 150
 - für Warteschlange für zu beantwortende Nachrichten 137
 - Definition erstellen 133
 - Ereignisse 405
 - mit Objekt arbeiten 136
 - MQSC-Befehle 131
 - Objektverwaltung 125
 - Queueing
 - Empfehlungen 132
 - Sicherheitsüberlegungen 152
- Fernverwaltung
 - Befehlsserver 123
 - Startprobleme 133
- Festschreiben und zurücksetzen 183
- FFST
 - Teildatenträger 65
 - überprüfen 223
- FORMAT-Schlüsselwort, Regeltabelle 167
- FREEZE SERVER, PATHCOM-Befehl 34
- FWDQ-Schlüsselwort, Regeltabelle 168
- FWDQM-Schlüsselwort, Regeltabelle 169

G

- Gemeinsam benutzte Ressourcenbibliothek 25
- Gemeinsam benutzte Speichersegmente 27
- Geschützte Ressourcen 144
- Gesteuerter Abschluss 71
- Gleichmäßige Auslastung 51, 225
 - durch Cluster 10
- Globale Arbeitseinheit
 - Beschreibung 21
- Glossar 423
- Groß-/Kleinschreibung
 - MQSC-Befehle 97
 - Steuerbefehle 31
- Guardian-Benutzer-ID 245
- Guardian-Segment-IDs 56

H

- HARDENBO
 - Prozessattribut 344
- HEADER-Schlüsselwort, Regeltabelle 169
- HomeTerminalName, Parameter 51

- HTML (Hypertext Markup Language) 420
- Hypertext Markup Language (HTML) 420

I

- ICE-Kommunikationskonfiguration, Beispiel 396
- ICE TP 385
- Inaktive Agentenprozesse 54
- Inhalt einer lokalen Warteschlange löschen 110
- Inhalt von
 - MQSINI 194
 - QMINI 195
- Inhibit-Ereignisse 405
- Initialisierungswarteschlange
 - Beschreibung 14
 - definieren 119
 - Kanalinitiator, ändern 35
- INPUTQ-Schlüsselwort, Regeltabelle 165
- INPUTQM-Schlüsselwort, Regeltabelle 165
- Installierbare Services
 - Berechtigungs-service 412
 - Namensservice 412
 - Objektberechtigungsmanager (OAM) 142
 - inaktivieren 146
- instmqm, Befehl 295
- Instrumentierungseignisse
 - aktivieren 177
 - Beschreibung 175
 - Funktion 175
 - Nachrichten 177
 - Typen 176
 - unterstützteMQSeries for Compaq NSK 405
- Interaktive MQSC-Befehle
 - beenden 99
 - Rückmeldung 98
 - verwenden 97
- Interaktive MQSC-Befehle beenden 99

J

- Java
 - kompatibel 339
 - unterstützte Programmiersprache 359
- Java Transaction Services 360

K

- Kanal
 - Anzahl begrenzen 56
 - Ausführungsbefehl 297
 - automatische Definition 131
 - Befehle 152
 - Berechtigungen für Escape-Befehle 157
 - Beschreibung 16, 125
 - definieren 129
 - Ereignisse 176, 405
 - Exits 373

- Kanal (*Forts.*)
 - fernes Queuing 125
 - Fernverwaltung 128
 - ICE 399
 - Initiator ausführen, Befehl 296
 - Konfiguration 197
 - Nachrichtenkanäle 227
 - Sicherheit 153
 - Sicherheitsanforderungen 152
 - starten 130
 - TCP/IP-Empfängerkanal definieren 400
 - TCP/IP-Senderkanal definieren 400
 - zwischen WS-Managern definieren 13
 - Kanal für Serververbindung, automatische Definition 131
 - Kanaldefinitionen, für andere als Client-Kanäle 245
 - Kanalinitiator
 - mit PATHCOM-Befehlen konfigurieren 35
 - Wiederherstellung nach Ausfall 246
 - Kanalsynchronisation, Teildatenträger 67
 - Kommunikationskonfiguration 383
 - Kommunikationskonfiguration, Beispiele
 - ICE 396
 - SNAX 391
 - TCP/IP 399
 - Komponente für prozessübergreifende Kommunikation (IPCC) 28
 - Konfigurationsdateien
 - editieren 203
 - für WS-Manager (QMINI)
 - Objektberechtigungsmanager (OAM) inaktivieren 146
 - Sicherungskopie erstellen 47
 - MQSeries (MQSINI) 193
 - Inhalt 194
 - Pfad 104
 - Sicherungskopie erstellen 47
 - Übersicht 193
 - Übersicht 193
 - WS-Manager (QMINI)
 - Inhalt 195
 - Zeilengruppen 195
 - Konfigurationsoptionen für die Datenintegrität 254
 - Konfigurationsoptionen für Verfügbarkeit 253
 - Kontextberechtigung 151
 - Kopieren, Kanal 94
 - Kopieren, Warteschlange 81
- L**
- Längenzähler 233
 - Leistung 225
 - Überlegungen bei der Verwendung der Trace-Funktion 222
 - Überlegungen zu Anwendungsentwürfen 214
 - Leistungsereignisse 176, 405
 - LIKE-Attribut, DEFINE QLOCAL 108
 - Lizenzverwaltung 325
 - Lokale Arbeitseinheit
 - Beschreibung 21
 - Lokale Ereignisse 405
 - Lokale Verwaltung 95
 - Lokale Warteschlangen
 - Attribute 353
 - Befehl 15
 - Beschreibung 12, 13
 - definieren 105
 - Definitionen kopieren 108
 - für nicht zustellbare Nachrichten 15
 - Inhalt löschen 110
 - Initialisierung 14
 - löschen 111
 - nicht zugestellte Nachricht 15
 - Übertragung 15
 - Lokaler WS-Manageragent (LQMA) 235
 - Funktion 24
 - Wiederherstellung nach Ausfall 246
 - Löschen
 - Kanal 90
 - lokale Warteschlange 82, 111
 - WS-Manager 73, 277
 - LQMAgent, Zeilengruppe in der Datei QMINI 196
 - LQMAgentPriority 55
 - LU6.2-Empfangsprogramm 27
 - LU6.2-Responder-Prozess 384
- M**
- Make-Datei
 - Java 339
 - MaxActiveChannels 56
 - Maxapplio, Wert 395
 - MaxChannels 56
 - MAXHANDS
 - Prozessattribut 344
 - MaxIdleAgentReuse 55
 - MaxIdleAgents 55
 - Maximale Zeilenlänge für MQSC-Befehle 102
 - Maximum
 - Anzahl der Nachrichten 10
 - Größe der Warteschlange 9
 - maxrcv, Umgebungsvariable 397
 - MAXUMSGS
 - Prozessattribut 344
 - MCAAgentPriority 55
 - MCA Caller, Zeilengruppe in der Datei QMINI 196
 - MICALU62Responder, Zeilengruppe in der Datei QMINI 196
 - MCATCPResponder, Zeilengruppe in der Datei QMINI 196
 - Message Queue Management (MQM) 73
 - MinIdleLQMAgents 54
 - MinIdleMCACallers 54
 - MinIdleMICALU62Responders 54
 - MinIdleMCATCPResponders 54
 - Modellwarteschlangen
 - arbeiten mit 116
 - Attribute 353
 - Beschreibung 14
 - definieren 116
 - MQ_LOAD_ENTRY_POINT_EXIT 374
 - MQAI (MQSeries Administration Interface)
 - Beschreibung 122, 363
 - MQCFH, PCF-Header 408
 - MQCFIL, PCF-Parameter für Integerliste 409
 - MQCFIN, PCF-Integer 409
 - MQCFSL, PCF-Zeichenfolgeliste 410
 - MQCFST, PCF-Zeichenfolgeparameter 409
 - MQCNO 348
 - MQCNO_FASTPATH_BINDING 236
 - MQCONN 235
 - MQCONNECTTYPE 236
 - MQCONNX 236
 - MQDATA CONVEXIT 354
 - MQDEFAULTPREFIX 329
 - MQDISC 352
 - MQDLH, DLQ-Header 163
 - MQEMSEVENTS 329
 - MQEMSEVENTS, Umgebungsvariable 181
 - MQGMO 349
 - MQGMO_BROWSE_* 358
 - MQGMO_SET_SIGNAL, Option 349
 - MQI
 - Aufrufe 350
 - Berechtigungen 153, 154
 - Beschreibung 7
 - Bibliotheken 361
 - Programmiersprache TAL
 - MQCLOSE 352
 - MQDISC 352
 - MQINQ 352
 - MQSET 353
 - Unterstützung für lokale Verwaltung 95
 - WS-Manageraufrufe 12
 - MQINQ 352
 - MQIServer, Zeilengruppe in der Datei QMINI 196
 - MQLISTENPORTNUM 329
 - MQM
 - Benutzer-ID 141, 150
 - Benutzergruppe 141
 - MQM-Schnittstelle 73
 - MQMACHINIFILE 329
 - MQMACHINIFILE, Umgebungsvariable 193
 - MQMC, Funktion 73
 - MQMD 350
 - MQMREFRESHINT
 - PATHWAY-Parameter 38
 - MQOPEN-Berechtigungen 154
 - MQPMO 350
 - MQPUT-Berechtigungen 154
 - MQPUT und MQPUT1, Leistungsüberlegungen 215
 - MQQSHKEEPINT 331
 - MQQSMAXBATCHEXP 332
 - MQQSMAXMSGSEXP 332
 - MQQSSIGTIMEOUT 331
 - MQRC_Q_SPACE_NOT_AVAILABLE, Rückkehrcode 106
 - MQRDF 329
 - MQRDFUPPROCESSNAME 329
 - MQRDFUPPROGRAM 329
 - MQS-CHANINIT00
 - TS/MP-Serverklasse 35

MQS-CMDSERV00
 TS/MP-Serverklasse 34

MQS-EC00
 TS/MP-Serverklasse 36

MQS-STATUS00, Serverklasse 37

MQS-TCPLIS00
 TS/MP-Serverklasse 34

MQS-TCPLISnn
 rekonfigurieren 58

MQS-TRIGMON00
 TS/MP-Serverklasse 35

MQSC-Befehl 33
 Befehle eingeben 96
 Befehle interaktiv verwenden 97
 Befehle prüfen 103
 Befehle verwenden 100
 Befehlsdateien
 ausführen 103
 Ausgabeberichte 102
 Eingabe und Ausgabe umleiten 100
 Fehler
 ferne Befehle 133
 lokale 104
 ferne Ausführung 131
 interaktive Eingabe beenden 99
 maximale Zeilenlänge 102
 Sicherheitsanforderungen für Kanäle 153
 Zulässiges Zeitlimit für Befehlsantworten überschritten 131

MQSC-Befehle
 ALTER QLOCAL 108
 ALTER QREMOTE 135
 DEFINE CHANNEL 129
 DEFINE QALIAS 114
 DEFINE QLOCAL 108
 DEFINE QLOCAL LIKE 108
 DEFINE QLOCAL REPLACE 108
 DEFINE QMODEL 116
 DEFINE QREMOTE 133
 DELETE QLOCAL 111
 DISPLAY QREMOTE 135
 Groß-/Kleinschreibung 97
 maximale Zeilenlänge 102
 unterstützte in MQSeries for Compaq NSK 341
 verwenden 33

MQSeries for Compaq NSK
 Anwendungen ausführen 355
 Anwendungen erstellen 355
 Architektur 23
 Benutzer-Exits 373
 Client-Unterstützung 403
 Dateien 29
 Datenträgerstruktur 64
 Komponenten 28
 Leistung 225
 Skalierbarkeit 225
 Superuser, MQM 141
 Teildatenträger 29
 TM/MP(TM/-)Unterstützung) 183
 Übersicht 323
 unterstützte MQSC-Befehle 341
 verwendete EMS-Ereignisschablone 413

MQSeries-Trace beenden 294

MQSeries-Trace starten, Befehl 315

MQSeries-Veröffentlichungen 419

MQSET 353

MQSINI, Konfigurationsdatei
 Beispiel 194
 editieren 203
 Pfad 104
 Übersicht 193

MQSNOAUT 329

MQSNOAUT, Umgebungsvariable 146

MQSS-Server 27

MQTRANSACTIONLIFE
 TM/MP-Transaktion aktualisieren 355

MQZAO-Konstanten und Berechtigungen 154

MsgId
 Leistungsüberlegungen bei Verwendung von 214

MSGTYPE-Schlüsselwort, Regeltabelle 167

MVS/ESA, WS-Manager 132

N

Nachricht
 Abrufalgorithmus 10
 Bediener 220
 Beschreibung 8
 Deskriptor 8
 erscheinen nicht in Warteschlangen 215
 für Instrumentierungsereignisse 177
 Größe 228
 Länge 9, 214
 Leistungsüberlegungen 214
 mit unerwarteten Informationen 216
 nach einer bestimmten suchen 214
 nicht zugestellt 222
 Queuing 7
 Segment 9
 variable Länge 214
 Zusammenfassung 9

Nachrichten
 Nicht permanente Nachrichten 227
 permanent 226

Nachrichtengesteuerte Verarbeitung 7

Nachrichtenkanal 227

Nachrichtenkanalagent (MCA)
 Funktion 25
 Wiederherstellung nach Ausfall 246

Nachrichtenlänge erweitern 109

Nachrichtenüberlaufdatei 228
 Konfigurationsoptionen für die Datenintegrität 254
 Schwellenwert ändern 231
 verschieben 230

Nachrichtewarteschlangen, Teildatenträger 66

Name des Standardstatusservers 59

Name des Standardwarteschlangenservers 62

Namen
 Objekte 11
 zulässige Namen für Objekte 258

Namenskonvention
 Statusserverklassen 61
 Warteschlangenserverklassen 64

Namenskonventionen
 Unterstützung in der Landessprache 258

Namenslistendefinitionen 245

Namensservice 20

Namensservice, Schnittstelle 412

Namensumwandlung für Objekte 67

Natives Binden 25

Nicht natives Binden 25

Nicht permanente Daten 239

Nicht permanente Nachrichten
 Konfigurationsoptionen für die Datenintegrität 254
 optimieren 227
 Prüfpunktinformationen 232
 speichern 229
 synchronisierte Protokollierung 227
 Unterschiede zu permanenten Nachrichten 8
 Verfügbarkeit 241

NOHARDENBO
 Prozessattribut 344

NonStop Java
 unterstützte Programmiersprache 359

NonStop-Prozesspaar 26

NonStop-TM/MP (Transaktionsmanager) 183

NonStop Tuxedo
 TM/MP-Transaktionsumgebung 244

NonStop-Warteschlangenserver 227

NPMSPEED, Kanalattribut 227

NSKSegidRange 56

NumECs (Parameter), Befehl crtmqm 50

O

OAM (Object Authority Manager, Objektberechtigungsmanager)
 dspmqaut, Befehl 149
 ersetzen 412
 Funktionsweise 143
 Gruppen 143
 inaktivieren 146
 kritische Operationen 150
 setmqaut, Befehl 146, 148

OAM-Datenbank 245

OAM-Principal 245

Objektänderungsdatei 245

Objektberechtigungsmanager (OAM, Object Authority Manager)
 Einführung 21

Objekte
 Arten 10
 Fernverwaltung 125
 für Auslösefunktion 117
 Namen 97
 Namenskonventionen 258
 Prozessdefinition 16
 Standardeinstellung
 wiederherstellen 70
 standardmäßige
 Attribute 107
 System
 Standardgruppe 17, 327
 wiederherstellen 70
 Warteschlange 13

Objekte (*Forts.*)
 WS-Manager
 MQI-Aufrufe 12
 Zugriff auf 141
 Objektkatalog 245
 Objektnamen, umwandeln 67
 OpenTMF 244
 Optimierung von MQSeries for Compaq
 NSK 225

P

Parameter
 altmqfls, Befehl 261
 altmqusr, Befehl 265
 cleanrdf, Befehl 267
 cnvclchl, Befehl 269
 crtmqcvx, Befehl 271
 crtmqm, Befehl 273
 dltnqm, Befehl 277
 dspmqaut, Befehl 279
 dspmqcsv, Befehl 283
 dspmqfls, Befehl 284
 dspmqtrc, Befehl 287
 dspmqusr, Befehl 288
 endmqcsv, Befehl 290
 endmqm, Befehl 292
 endmqtrc, Befehl 294
 instmqm, Befehl 295
 runmqchi, Befehl 296
 runmqchl, Befehl 297
 runmqdlq, Befehl 298
 runmqslr, Befehl 300
 runmqsc, Befehl 303
 runmqtrm, Befehl 305
 setmqaut, Befehl 308
 strmqcsv, Befehl 312
 strmqm, Befehl 313
 strmqtrc, Befehl 315
 upgmqm, Befehl 319
 PARAMs (Umgebungsvariablen) 329
 Partitionieren
 Warteschlangendateien 230
 PATHCOM-Befehle
 ADD SERVER 34
 ALTER SERVER 35
 FREEZE SERVER 34
 START SERVER 34
 STATUS SERVER 34
 STOP SERVER 34
 THAW SERVER 34
 PathmonProcName 52
 PATHWAY
 Beispielkonfiguration 38
 Fehler 209
 Konfigurationsoptionen für Verfügbarkeit 253
 Serverklasse für Warteschlangenserver 229
 Serverklassen 45
 PATHWAY-Definition für APC-Prozess,
 Beispiel 393
 PCF-Befehle
 Attribute in MQSC- und PCF-
 Befehlen 122
 PCF-Escape-Befehle 122
 Verwaltungs-Tasks verwenden 121

PCF-Befehle (*Forts.*)
 Verwendung mit Hilfe von MQAI ver-
 einfachen 122
 PCF-Befehle, Antworten 411
 PCF-Befehle (Programmable Command
 Format) 407
 Einführung 33
 Header (MQCFH) 408
 Integer (MQCFIN) 409
 MQCFH (Header) 408
 MQCFIL (Parameter für Integer-
 liste) 409
 MQCFIN (Integer) 409
 MQCFSL (Zeichenfolge) 410
 MQCFST (Zeichenfolge-
 parameter) 409
 Nachrichtendeskriptor 408
 Parameter für Integerliste (MQC-
 FIL) 409
 unterstützteMQSeries for Compaq
 NSK 410
 Zeichen-
 folgeliste (MQCFSL) 410
 Zeichenfolgeparameter
 (MQCFST) 409
 PCF-Escape-Befehle 33, 122
 PDF (Portable Document Format) 420
 Permanente Daten 239
 Permanente Nachrichten
 optimieren 226
 speichern 228
 standardmäßig maximal zulässige
 Anzahl überschreiben 332
 Unterschiede zu nicht permanenten
 Nachrichten 8
 Verfügbarkeit 240
 Permanente Warteschlangen 10
 PRIDB, Schlüsselwort, Regeltab-
 elle 167
 PMSEARCH 329
 Portable Document Format (PDF) 420
 PRIDB, Principal-Datenbank 159
 Principal-Datenbank 245
 Principals
 Benennung 143
 Datenbank, PRIDB 159
 erstellen 69, 147
 Funktion 21
 Priorität der WS-Managerprozesse 55
 Programmable Command Format (PCF-
 Befehle) 33, 407
 Programme, zur Verfügung gestellte Bei-
 spiele 365
 Programmierfehler, Beispiele 209
 Programmiersprache
 unterstützte 359
 Protokoll, Fehler 219, 220
 Prozessdefinitionen
 anzeigen 120
 Beschreibung 16
 erstellen 119
 Prüfen, MQSC-Befehle 103
 Prüfprotokollgröße, TM/MP 186
 Prüfpunktinformationen
 für nicht permanente Nachrichten
 durch den Warteschlangens-
 server 232

Puffer
 Nachrichten bei Anzeigevorgängen
 speichern in 231
 PUTAUT-Schlüsselwort, Regeltab-
 elle 169

Q

QMDefaultVolume 50
 QMINI, Konfigurationsdatei
 Beispiel 198
 editieren 203
 für die Unterstützung von TCP/IP-
 Empfangsprogrammen konfigurie-
 ren 401
 Übersicht 195
 Queue Manager Menu, MQM 74
 QueueManager, Zeilengruppe in der
 Datei MQSINI 194
 Queues Menu, MQM 77

R

RDF (Remote Database Duplication Faci-
 lity)
 cleanrdf, Befehl 70, 267
 Fehlerbehebung 192
 REASON-Schlüsselwort, Regeltab-
 elle 167
 Referenzliteratur 421
 Regeltabelle, DLQ-Steuerroutine 164
 Beispiel 173
 Muster und Aktionen (Regeln)
 ACTION, Schlüsselwort 168
 APPLIDAT, Schlüsselwort 166
 APPLNAME, Schlüsselwort 167
 APPLTYPE, Schlüsselwort 167
 DESTQ, Schlüsselwort 167
 DESTQM, Schlüsselwort 167
 FEEDBACK, Schlüsselwort 167
 FORMAT, Schlüsselwort 167
 FWDQ, Schlüsselwort 168
 FWDQM, Schlüsselwort 169
 HEADER, Schlüsselwort 169
 MSGTYPE, Schlüsselwort 167
 PERSIST, Schlüsselwort 167
 PUTAUT, Schlüsselwort 169
 REASON, Schlüsselwort 167
 REPLYQ, Schlüsselwort 167
 REPLYQM, Schlüsselwort 168
 RETRY, Schlüsselwort 169
 USERID, Schlüsselwort 168
 Steuerdateneintrag
 INPUTQ, Schlüsselwort 165
 INPUTQM, Schlüsselwort 165
 RETRYINT, Schlüsselwort 165
 WAIT, Schlüsselwort 165
 Syntax 170
 Verarbeitung 171
 REPLACE-Attribut, DEFINE-Befehle 102
 REPLYQ-Schlüsselwort, Regeltabelle 167
 REPLYQM-Schlüsselwort, Regeltab-
 elle 168
 Repository-Cache-Manager
 Wiederherstellung nach Ausfall 246

- Repository-Manager
 - Prozess 25
 - Wiederherstellung nach Ausfall 246
- Ressourcen, schützen 141
- RETRY-Schlüsselwort, Regeltabelle 169
- RETRYINT-Schlüsselwort, Regeltabelle 165
- ROLLBACK-Operation 183
- Rückkehrcodes 206
 - altnqfls, Befehl 264
 - altnqusr, Befehl 266
 - cleanrdf, Befehl 267
 - crtmqcvx, Befehl 271
 - crtmqm, Befehl 276
 - dltmqm, Befehl 277
 - dspmqaout, Befehl 282
 - dspmqcsv, Befehl 283
 - dspmqfls, Befehl 285
 - dspmqusr, Befehl 288
 - endmqcsvg, Befehl 290
 - endmqm, Befehl 293
 - endmqtrc, Befehl 294
 - runmqchi, Befehl 296
 - runmqchl, Befehl 297
 - runmqslr, Befehl 301
 - runmqsc, Befehl 303
 - runmqtrm, Befehl 305
 - setmqaut, Befehl 311
 - strmqcsvg, Befehl 312
 - strmqm, Befehl 313
 - strmqtrc, Befehl 317
- Rückmeldung auf MQSC-Befehle 98
- runmqchi, Befehl 296
- runmqchl, Befehl 297
- runmqdlq, Befehl 163, 298
- runmqslr, Befehl 300
- runmqsc
 - beenden 99
 - Befehl 302
 - Eingabe und Ausgabe umleiten 100
 - Fehler 104
 - interaktiv verwenden 97
 - MQSC-Befehle eingeben 96
 - prüfen 103
 - Rückmeldung 98
 - verwenden 100
 - Warteschlangenmodus 131
 - WS-Manager angeben 100
- runmqtrm, Befehl 305

S

- SAFEGUARD 21, 144, 324
- SAVE-ENVIRONMENT ON 329
- SCF-Konfigurationsdatei, Beispiel 391
- Schablonen, EMS-Ereignis 178
- Schlüsselwörter für Mustererkennung, Regeltabelle 166
- Schnittstelle für Nachrichtenwarteschlangen (MQI, Message Queue Interface) 7
- Segment-IDs 56
- Serverklasse
 - MQS-TCPLISnn 58
 - Namenskonvention
 - Statusserver 61
 - Warteschlangenserver 64
- Serverklassen
 - benutzerdefiniert 45
- Service, Zeilengruppe in der Datei QMINI 196
- ServiceComponent, Zeilengruppe in der Datei QMINI 197
- Servicekomponente 20
- setmqaut, Befehl 306
 - installierbare Services 149
 - verwenden 146, 148
 - zugehörige Befehle 311
- Sicherheit 141
 - aktivieren 146
 - Befehle verwenden 146, 149
 - ferne WS-Manager 152
 - Objektberechtigungsmanager (OAM, Object Authority Manager) 21
 - Principals erstellen 69
- Signaloption 349
- Skalierbarkeit 225
- SNA
 - CPU-Belegung 234
 - SNA-Empfangsprogramm 218
- SNA-Protokoll 383
- SNAX-Kommunikationskonfiguration, Beispiele 391
- Softcopy-Bücher 420
- Speicherpuffer 229
- Sperre im Cache 232
- STANDARD-Bindung 361
- Standarddatenkonvertierung 138
- Standardeinstellung
 - Objekte 70
 - Statusserver, Name 59
 - TCP/IP-Port 57
 - Übertragungswarteschlange 47
 - Warteschlangenserver, Name 62
 - WS-Manager 47
 - ändern 72
 - versehentliche Änderungen 72
- Standardmäßige
 - Objekte 17
 - Systemobjekte 327
 - Übertragungswarteschlange 136
 - WS-Manager
 - ändern 100
 - Befehlsverarbeitung 97
 - versehentliches Löschen 274
- Start- und Stoppereignisse 405
- Start Channel Initiator 411
- Start Channel Listener 411
- START SERVER, PATHCOM-Befehl 34
- Startbefehl, WS-Manager 313
- Starten
 - Kanäle 91, 130
 - Trace 77
 - WS-Manager 70
- STATUS SERVER, PATHCOM-Befehl 34
- Statusserver
 - entfernen 61
 - Name des Standardsservers 59
 - Prozesse
 - hinzufügen 37
 - Serverklasse MQS-STATUS00 37
 - wiederherstellen und erneut starten 191
 - Wiederherstellung nach Ausfall 246

- Statusserver (*Forts.*)
 - zusätzliche hinzufügen und entfernen 60
- Statusserver, zusätzliche hinzufügen und entfernen 60
- stdin (Standardeingabe), im Befehl runmqsc 100
- stdout (Standardausgabe), im Befehl runmqsc 100
- Steuerbefehle 31
 - altnqfls 260
 - altnqusr 265
 - cleanrdf 267
 - cnvclchl 269
 - crtmqcvx 271
 - crtmqm 273
 - dltmqm 277
 - dspmqaout 279
 - dspmqcsvg 283
 - dspmqfls 284
 - dspmqtrc 287
 - dspmqusr 288
 - endmqcsvg 290
 - endmqm 292
 - endmqtrc 294
 - Groß-/Kleinschreibung muss beachtet werden 31
 - instmqm 295
 - runmqchi 296
 - runmqchl 297
 - runmqdlq 298
 - runmqslr 300
 - runmqsc 97, 302
 - runmqtrm 305
 - setmqaut 306
 - strmqcsvg 312
 - strmqm 313
 - strmqtrc 315
 - upgmqm 318
- STOP SERVER, PATHCOM-Befehl 34
- Stoppen
 - Kanäle 91
 - Trace 77
 - WS-Manager 292, 337
- strmqcsvg, Befehl 312
 - zugehörige Befehle 312
- strmqm, Befehl 313
 - zugehörige Befehle 314
- strmqtrc, Befehl 315
 - zugehörige Befehle 317
- Strukturdatentypen 347
- Superuser (MQSeries)
 - MQM 141
- SupportPac 421
- Synchronisationsdateien für Kanäle 245
- Synchronisationspunkt 243
 - Begrenzung 184
- Synchronisationspunkt, Leistungsüberlegungen 215
- Syntaxdiagramme verstehen 258
- Syntaxfehler, in MQSC-Befehlen 98
- system.cluster.transmit.queue 229
- Systemobjekte
 - wiederherstellen 70
- Systemstandardobjekte 17
- Systemstandardwerte 327

T

- TACL-Umgebungsvariablen 329
- TAL (nicht nativ)
 - unterstützte Programmiersprache 359
- TCP/IP
 - Fernverwaltung 18
 - Kanäle 389
- TCP/IP-Empfangsprogramme
 - Konfigurieren 34
 - starten 34
 - stoppen 34
 - Wiederherstellung nach Ausfall 246
- TCP/IP-Kommunikationskonfiguration,
 - Beispiel 399
- TCP/IP-Protokoll 383
- TCP/IP-Prozess 57
- TCP/IP-Prozess, anderer als der Standard-TCP/IP-Prozess 58
- TCPCConfig, Zeilengruppe
 - TCP/IP konfigurieren 399
- TCPCConfig, Zeilengruppe in der Datei QMINI 197
- TCPLListener, Zeilengruppe in der Datei QMINI 196
- TCPLListenerPort 57
- TCPNumListenerPorts 57
- TCPPort 57
- Teildatenträger 29
- Temporäre Warteschlangen 10
- Terminologie im vorliegenden Handbuch 423
- THAW SERVER, PATHCOM-Befehl 34
- Threads
 - in Anwendungen 360
- TM/MP 22
 - Datenbankdateien protokollieren 242
 - Koordination mit externen Datenbanken 243
 - Prüflistendateien 226, 227
- TM/MP(TM/-)Unterstützung 356
- TM/MP(TM/-)Unterstützung) 183
- Trace
 - Beispieldaten 223
 - Leistungsüberlegungen 222
- Trace für MQSeries-Objekte durchführen 76
- Transaktion
 - mehrere verwalten 245
- Transaktionsmanager (NonStop-TM/MP) 183
- Transaktionsunterstützung 183
- TS/MP-Serverklassen
 - MQS-CHANINIT00 35
 - MQS-CMDSERV00 34
 - MQS-EC00 36
 - MQS-TCPLIS00 34
 - MQS-TRIGMON00 35
- TS/MP-Verwaltung 34
- TuningParameters, Zeilengruppe in der Datei QMINI 197
- Tuxedo, NonStop
 - TM/MP-Transaktionsumgebung 244

U

- Übersicht über MQSeries for Compaq NSK 323
- Übertragungswarteschlange
 - Beschreibung 15
 - definieren 129
 - erstellen 136
 - Fernverwaltung 128
 - Standardeinstellung 47
 - standardmäßige 136
 - zwischen WS-Managern definieren 13
- Überwachen
 - TMF-Status 186
- Überwachen, Kanal 90
- Überwachen, Warteschlange 83
- Überwachen, WS-Manager 175
- Umgebungsvariablen 329
- upgmqm, Befehl 318
- Ursachencode 12
- USERDATA
 - Prozessattribut 345
- USERID-Schlüsselwort, Regeltabelle 168

V

- Verarbeitung, ereignisgesteuert 8
- Verfügbarkeit
 - Definition 238
 - Konfigurationsoptionen 253
- Verteiltes Queuing
 - falsche Ausgabe 217
 - Warteschlange für nicht übermittelte Nachrichten 15
 - Warteschlange für nicht zustellbare Nachrichten 15
- Verwalten, Objekte für Auslösefunktion 117
- Verwaltung
 - Befehlssätze 31
 - MQSeries-Befehle (MQSC) 32
 - PCF-Befehle 33
 - Steuerbefehle 31
 - Berechtigungen 157
 - ferne WS-Manager 127
 - Kanäle 128
 - Objekte 125
 - Übertragungswarteschlangen 128
 - lokale 95
- Verzeichnisse, WS-Manager 150
- Voraussetzungen
 - Hardware 323
 - Software 323
- Vordefinierte Warteschlangen 10

W

- WAIT-Schlüsselwort, Regeltabelle 165
- Warteschlange für nicht zustellbare Nachrichten
 - angeben 47
 - Beschreibung 15
 - Steuerroutine 298
- Warteschlange für zu beantwortende Nachrichten 16

- Warteschlange für zu beantwortende Nachrichten, Aliasname 137
- Warteschlangen durchsuchen 111
- Warteschlangendatei 228, 245
 - partitionieren 230
 - verschieben 230
- Warteschlangenlänge 107
- Warteschlangenmodus, runmqsc 131
- Warteschlangenserver
 - Beschreibung 26
 - CPU-Verteilung 229
 - Datenstruktur beim Start des Warteschlangenservers erstellen 232
 - Datenstrukturen und Datenketten im Speicher sperren 232
 - entfernen 64
 - Längenzähler für Warteschlange verwalten 233
 - Nachrichten im Hauptspeicher ablegen 231
 - Name des Standardsservers 62
 - optimieren 228
 - Prozesse
 - hinzufügen 37
 - Prüfpunktinformationen für nicht permanente Nachrichten 232
 - Umgebungsvariablen 331
 - Verwaltung nicht permanenter Nachrichtendaten 241
 - Warteschlangen verteilen auf 229
 - wiederherstellen und erneut starten 191
 - Wiederherstellung nach Ausfall 246
 - zusätzliche hinzufügen und entfernen 63
- Warteschlangenserver, zusätzliche hinzufügen und entfernen 63
- Warteschlangenüberlaufdatei 228, 245
 - verschieben 230
- Wiederherstellen
 - Standardobjekte 70
 - Systemobjekte 70
- Wiederherstellung nach Ausfall 246
- Windows-Hilfe 421
- WS-Manager
 - Aliasname, ferne Warteschlange 136
 - Anzahl 47
 - Attribut 354
 - Ausgangsdatenträger 50
 - beenden
 - gesteuert 71
 - Befehlsserver 123
 - Berechtigungen 150
 - Beschreibung 12
 - eindeutiger Name 47
 - endmqm, Befehl 292
 - Ereignisse 176
 - erneut starten 72
 - erstellen
 - crtmqm, Befehl 273
 - Standardeinstellung 68
 - Vorgehensweise 47
 - erzwungener Abschluss 71
 - Fernverwaltung 125
 - Kanalsynchronisation, Teildatenträger 67
 - Komponenten 27

WS-Manager (Forts.)
 Konfigurationsdatei
 Inhalt von 195
 Sicherungskopie erstellen 47
 konfigurierbare Merkmale 50
 lokale Verwaltung 95
 löschen 73, 277
 manuell entfernen 337
 mit runmqsc angeben 100
 Nachrichtenwarteschlangen, Teil-
 datenträger 66
 Objektberechtigungsmanager (OAM)
 Beschreibung 142
 inaktivieren 146
 Objekte
 MQI-Aufrufe 12
 Prozesse 23
 sofortiger Abschluss 71
 Standardeinstellung 47
 ändern 72
 versehentliche Änderungen 72
 standardmäßige
 versehentliches Löschen 274
 starten 70
 stoppen 71
 manuell 337
 Teildatenträger mit den Daten-
 dateien 65
 Teildatenträger mit Fehler-
 protokoll 66
 Teildatenträger mit FFST-Dateien 65
 überwachen 175
 unter MVS/ESA 132
 Verfügbarkeit 239
 Verzeichnisse 150
 Wiederherstellung nach Ausfall 246
 WS-Managerserver
 Wiederherstellung nach Ausfall 246

X

XA-kompatible Datenbanken 22
 XA-Schnittstelle 358

Z

Zeilengruppen
 MQSINI 194
 QMINI 195
 Zeitunabhängige Anwendungen 7
 ZMQSTMPL, Schablonendatei für EMS-
 Ereignisse 178
 Zugriff, Schutz vor unbefugtem 141
 Zulässiges Zeitlimit für MQSC-Befehle
 überschritten 131
 Zurücksetzen, Nachrichtenfolge-
 nummer 92
 Zusätzliche Statusserver hinzufügen und
 entfernen 60
 Zusätzliche Warteschlangenserver hinzu-
 fügen 63
 Zweiphasige Festschreibung 183

Kommentare an IBM senden

Sie können uns Anmerkungen zu dem vorliegenden Handbuch über die nachfolgend aufgeführten Wege zukommen lassen.

Bitte lassen Sie es uns wissen, wenn Informationen Ihrer Meinung nach fehlerhaft sind oder ganz fehlen, oder wenn Sie Anmerkungen zur Richtigkeit, zum Aufbau, Inhalt oder Vollständigkeit des Handbuchs haben.

Bitte senden Sie uns Kommentare nur im Zusammenhang mit dem vorliegenden Handbuch und nur über die hier aufgeführten Übermittlungskanäle zu.

Wenn Sie Anmerkungen zu Funktionen von IBM Produkten oder IBM Systemen haben, wenden Sie sich bitte an Ihren IBM Ansprechpartner bzw. den zuständigen IBM Vertriebspartner.

Bei IBM eingehende Kommentare können von IBM beliebig verwendet werden, ohne dass hieraus eine Verpflichtung gegenüber dem Absender entsteht.

Ihre Kommentare können Sie IBM auf folgenden Wegen zukommen lassen:

- Per Post an folgende Adresse:

User Technologies Department (MP095)
IBM United Kingdom Laboratories
Hursley Park
WINCHESTER,
Hampshire
SO21 2JN
Großbritannien

- Per Fax:

- Benutzer außerhalb von Großbritannien müssen im Anschluss an die jeweilige internationale Durchwahl (in Deutschland z. B. 00) folgende Nummer wählen: 44-1962-816151
- Benutzer in Großbritannien müssen folgende Nummer wählen: 01962-816151

- Per E-Mail, unter Angabe der entsprechenden Netz-ID:

- IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
- IBMLink: HURSLEY(IDRCF)
- Internet: idrcf@hursley.ibm.com

Unabhängig von der Übertragungsart sind auf jeden Fall folgende Angaben erforderlich:

- Die Bestellnummer sowie der Titel der Veröffentlichung
- Der Abschnitt, auf den Sie sich beziehen
- Ihre Adresse: Name, Adresse, Telefonnummer, Faxnummer, Netz-ID

Antwort

MQSeries for Compaq NonStop Kernel
Systemverwaltung
Version 5 Release 1

IBM Form SC12-2996-00

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 01803/31 32 33) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: ibmterm@de.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

IBM Deutschland GmbH
SW TSC Germany

70548 Stuttgart



Teilenummer: CT8Z1DE

SC12-2996-00



(1P) P/N: CT8Z1DE

