

MQSeries* pour Compaq NonStop* Kernel



Administration du système

Version 5.1

MQSeries* pour Compaq NonStop* Kernel



Administration du système

Version 5.1

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Annexe Q. Remarques» à la page 409.

Première édition – juin 2001

Réf. US : SC34-5886-00

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT". IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
Tour Descartes
92066 Paris-La Défense Cedex 50*

© Copyright IBM France 2001. Tous droits réservés.

© **Copyright International Business Machines Corporation 1993, 2001. All rights reserved.**

Table des matières

Figures xi

Tableaux. xiii

Avis aux lecteurs canadiens xv

Préface xvii

A qui s'adresse ce manuel xvii

Connaissances préalables nécessaires à la lecture
du manuel xvii

Utilisation du manuel xvii

Informations sur MQSeries disponibles sur
Internet xviii

**Nouvelles fonctions de MQSeries pour
Compaq NSK V5.1 xix**

Amélioration des performances xix

Fonctions MQSeries mises à niveau xix

Intercommunications xxi

Fonctions propres à Compaq NSK xxi

Partie 1. Exploitation 1

Chapitre 1. Introduction 7

MQSeries et la mise en file d'attente de messages . . . 7

 Applications asynchrones 7

 Processus gérés par messages. 7

Messages et files d'attente 8

 Définition d'un message 8

 Définition d'une file d'attente. 9

Objets 10

 Noms des objets. 10

 Gestion des objets 11

 Gestionnaires de file d'attente MQSeries. 11

 Files d'attente MQSeries 12

 Définitions de processus 15

 Canaux. 15

 Clusters 16

 Listes de noms 16

Objets système par défaut 16

Administration 17

 Administration locale et éloignée 17

Clients et serveurs 17

 Applications MQI en environnement
 client-serveur. 18

Extension des fonctions du gestionnaire de files
d'attente 18

 Exits utilisateur 18

 Services optionnels 19

Sécurité. 19

 Gestionnaire des droits d'accès aux objets (OAM) 19

Prise en charge transactionnelle. 19

Optimisation des performances, fiabilité, évolutivité
et préévaluation des capacités 20

**Chapitre 2. Architecture MQSeries pour
Compaq NSK V5.1. 21**

Généralités sur le processus de gestionnaire de files
d'attente 21

Mise en forme du produit 23

Fichiers exécutables. 24

Comparaison entre l'architecture de la Version 5.1 et
celle de la Version 2.2.0.1 24

Gestionnaire de files d'attente — vue fonctionnelle 25

Modèle de processus de gestionnaire de files
d'attente 26

Fichiers et sous-volumes MQSeries 27

**Chapitre 3. Utilisation des jeux de
commandes MQSeries 29**

Administration à l'aide des commandes de contrôle 29

 Utilisation des commandes de contrôle 29

Administration à l'aide des commandes MQSC . . . 30

 Exécution de commandes MQSC 30

Administration à l'aide des commandes PCF . . . 31

 Attributs dans les commandes MQSC et PCF . . . 31

 Commandes PCF d'échappement 31

Administration TS/MP (PATHWAY) 32

 Détermination et contrôle des programmes
 d'écoute TCP/IP. 32

 Contrôle du serveur de commandes 32

 Détermination et contrôle des initialisateurs de
 canaux 32

 Détermination et contrôle des moniteurs de
 déclenchement 33

 Répartition des processus sur les processeurs . . 34

 Ajout de nouveaux processus de status server. . 35

 Ajout de nouveaux processus de serveurs de file
 d'attente 35

 Fréquence de rafraîchissement des écrans du
 moniteur MQM 35

 Configuration PATHWAY pour un gestionnaire
 de file d'attente 36

 Modification des paramètres de classe de
 serveurs PATHWAY 42

 Ajout de classes de serveurs personnalisées à un
 PATHWAY MQSeries 43

**Chapitre 4. Gestion des gestionnaires
de files d'attente 45**

Mise en route. 45

Instructions pour la création d'un gestionnaire de
files d'attente. 45

 Sauvegarde des fichiers de configuration après la
 création d'un gestionnaire de files d'attente. . . 48

Modification des propriétés d'un gestionnaire de files d'attente	48
Volume origine du gestionnaire de files d'attente	48
Nombre de processus EC par gestionnaire de files d'attente	49
Terminal d'origine du gestionnaire de files d'attente	49
Nom de processus PATHMON du gestionnaire de files d'attente	50
CCSID du gestionnaire de files d'attente	51
Contrôle de la conversion de données EBCDIC	51
Collecteur EMS du gestionnaire de files d'attente	52
Groupe d'agents à la disposition des EC du gestionnaire de files d'attente	52
Nombre maximum d'agents en veille et réutilisation des processus	53
Priorité de traitement des processus de gestionnaire de files d'attente	53
Nombre maximum de canaux d'un gestionnaire de files d'attente	53
Nombre maximum de canaux actifs d'un gestionnaire de files d'attente	54
ID de segment Guardian utilisés par MQSeries	54
Port TCP/IP par défaut	54
Ports TCP/IP écoutés par le gestionnaire de files d'attente	55
Processus TCP/IP utilisé par le gestionnaire de files d'attente	55
Reconfiguration de la classe de serveur MQS-TCPLISnn pour un port et un processus TCP/IP personnalisés	56
Affectation de l'espace de permutation	56
Nom de status server par défaut	57
Ajout et suppression de status server personnalisés	58
Nom de serveur de file d'attente par défaut	59
Ajout et suppression de serveurs de file d'attente personnalisés	60
Structure du volume	62
Sous-volume FFST d'un gestionnaire de files d'attente	62
Sous-volume de fichiers de données de gestionnaire de files d'attente	62
Sous-volume des journaux d'erreurs des gestionnaires de files d'attente	63
Sous-volume de files d'attente de messages de gestionnaires	63
Sous-volume de synchronisation des canaux des gestionnaires de files d'attente	64
Transformation des noms d'objets	64
Utilisation des gestionnaires de files d'attente	65
Création d'un gestionnaire de files d'attente par défaut	65
Création de principaux MQSeries	66
Exécution de cleanrdf pour un gestionnaire de files d'attente prêt pour RDF	67
Lancement d'un gestionnaire de files d'attente	67
Restauration des objets système et par défaut	67
Consultation des fichiers objets	67
Arrêt d'un gestionnaire de files d'attente	68
En cas d'incident	68

Arrêt immédiat et arrêt forcé d'un gestionnaire de files d'attente	68
Redémarrage d'un gestionnaire de files d'attente	69
Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant	69
Suppression d'un gestionnaire de files d'attente	70
Utilisation de la fonction MQM (Message Queue Management).	70
Utilisation du menu Queue Manager	72
Utilisation du menu Queues	75
Utilisation du menu Channels	81

Chapitre 5. Administration des objets

MQSeries locaux 93

Prise en charge des applications utilisant l'interface MQI	93
Administration locale à l'aide de commandes MQSC	94
Préambule	94
Utilisation de la fonction MQSC de façon interactive	95
Informations renvoyées par les commandes MQSC	96
Fin de l'entrée interactive sur MQSC	96
Affichage des attributs d'un gestionnaire de files d'attente	96
Utilisation d'un gestionnaire différent du gestionnaire par défaut	97
Modification des attributs d'un gestionnaire de files d'attente	98
Exécution de commandes MQSC à partir de fichiers texte	98
Utilisation des fichiers de commandes MQSC	99
Utilisation des rapports MQSC	100
Exécution du fichier de commandes MQSC fourni	101
Utilisation de runmqsc pour vérifier des commandes	101
Résolution des incidents à l'aide des commandes MQSC	101
Manipulation des files d'attente locales	103
Définition d'une file d'attente locale	103
Modification de la taille du fichier physique des files d'attente	103
Définition d'une file d'attente de rebut	104
Affichage des attributs d'objet par défaut	104
Copie d'une définition de file d'attente locale	105
Modification des attributs d'une file d'attente locale	106
Modification du volume d'une file d'attente locale	106
Modification des options d'une file d'attente locale	106
Reassignation d'objets aux serveurs de statut et de file d'attente	107
Nettoyage d'une file d'attente locale	108
Suppression d'une file d'attente locale	108
Visualisation des files d'attente	108
Manipulation des files d'attente alias	111
Définition d'une file d'attente d'alias	112
Utilisation d'autres commandes avec les files d'attente alias	113

Manipulation des files d'attente modèles	113
Définition d'une file d'attente modèle	114
Utilisation d'autres commandes avec les modèles	114
Gestion des objets pour le déclenchement	114
Définition d'une file d'attente d'application pour le déclenchement	115
Définition d'une file d'attente d'initialisation	116
Création d'une définition de processus	116
Affichage de la définition de processus	117

Chapitre 6. Automatisation des tâches d'administration 119

Commandes programmables PCF.	119
Attributs dans les commandes MQSC et PCF	120
Commandes PCF d'échappement.	120
Utilisation de l'interface MQAI pour simplifier l'utilisation des commandes programmables PCF	120
Gestion du serveur de commandes pour l'administration éloignée	121
Lancement du serveur de commandes	121
Affichage de l'état du serveur de commandes	121
Arrêt d'un serveur de commandes	122

Chapitre 7. Administration des objets MQSeries éloignés 123

Canaux, clusters et mise en file d'attente éloignée	123
Administration éloignée à l'aide de clusters	124
Administration d'un gestionnaire de files d'attente	125
Préparation des gestionnaires pour l'administration éloignée	125
Préparation des canaux et des files d'attente de transmission pour l'administration éloignée	126
Définition des canaux et des files d'attente de transmission.	127
Lancement des canaux	128
Emission de commandes MQSC à distance	129
Recommandations pour la mise en file d'attente éloignée	130
En cas d'incident	131
Création d'une définition locale de file d'attente éloignée	131
Fonctionnement de la définition locale d'une file d'attente éloignée	131
Autre mode d'insertion de messages dans une file d'attente éloignée	133
Utilisation d'autres commandes avec les files d'attente éloignées.	133
Création d'une file d'attente de transmission	133
Files d'attente de transmission par défaut	134
Utilisation des définitions de files d'attente éloignées pour les alias	134
Gestionnaire de files d'attente alias	134
Files d'attente de réponse alias	135
Conversion de données	135
Quand le gestionnaire de files d'attente ne peut pas convertir les messages dans des formats livrés d'origine	135
Fichier CCSID	136

Conversion de messages en formats définis par l'utilisateur	136
Modification du CCSID d'un gestionnaire de files d'attente	136

Chapitre 8. Protection des objets MQSeries 139

Nécessité de protéger les ressources MQSeries	139
Présentation des ID utilisateur du groupe d'utilisateurs MQM	139
Informations complémentaires.	140
Présentation du gestionnaire des droits d'accès aux objets (OAM)	140
Mode de fonctionnement de l'OAM	141
Gestion de l'accès par le biais de groupes d'utilisateurs	141
Protection de ressources avec l'OAM	142
Utilisation de groupes pour les droits d'accès	142
Désactivation du gestionnaire des droits d'accès aux objets (OAM)	143
Utilisation des commandes de l'OAM	144
Informations à préciser aux commandes	144
Utilisation de la commande altmqusr	145
Utilisation de la commande dspmqusr	145
Utilisation de la commande setmqaut	146
Droits d'accès	147
Commande d'affichage des droits d'accès	147
Instructions concernant le gestionnaire des droits d'accès aux objets (OAM)	148
ID utilisateur	148
Volumes du gestionnaire de files d'attente.	148
Files d'attente	148
Droits d'accès d'utilisateur secondaire	148
Droits de contexte.	149
Gestion éloignée de la sécurité	150
Sécurité des commandes de canal	150
Tables de spécification des droits d'accès	151
Droits d'accès MQI	151
Droits d'administration	154
Droits des commandes MQSC dans les PCF d'échappement	155
Fichiers de droits d'accès	157
Base de données de principaux	157
Base de données OAM	157
Enregistrements de droits d'accès aux classes	159
Enregistrement de droit d'accès à toutes les classes.	159

Chapitre 9. Gestionnaire de file d'attente de rebut MQSeries 161

Lancement du gestionnaire DLQ	161
Table de règles du gestionnaire DLQ	162
Données de contrôle	162
Règles (modèles et actions).	164
Mots clés de modèle	164
Mots clés d'action	165
Conventions de la table de règles.	167
Traitement de la table de règles	168
Assurance du traitement de tous les messages DLQ	169

Exemple de table de règles pour un gestionnaire DLQ	170
---	-----

Chapitre 10. Événements outils et événements EMS. 173

Événements outils MQSeries	173
Types d'événements	173
Notification d'événements via les files d'attente d'événements	174
Utilisation de files d'attente d'événements déclenchées	174
Activation des événements outil	175
Messages d'événements	175
Événements EMS (Event Management Service)	175
Fichier modèle EMS fourni avec MQSeries pour Compaq NSK	175
Intégration des modèles d'événements EMS pour MQSeries	176
Définition du PARAM MQEMSEVENTS	179
Utilisation d'un autre programme de collecte	179
Ecriture de programmes de traitement des événements EMS MQSeries.	179

Chapitre 11. Prise en charge transactionnelle et messagerie. 181

Utilisation de NonStop TM/MP (gestionnaire de transactions).	181
Limitations du système de points de synchronisation.	182
Opérations sur les messages persistants, sans point de synchronisation	182
Opérations avec point de synchronisation sur les messages non persistants	183
Configuration requise pour TM/MP et MQSeries pour Compaq NSK	183
Contrôle	183
Taille du fichier d'audit	183
Configuration du gestionnaire de ressources	184
Identification des problèmes	184

Chapitre 12. Reprise et redémarrage 185

Tolérance aux pannes et reprise	185
Sauvegarde et restauration de MQSeries	186
Sauvegarde de MQSeries	186
Restauration de MQSeries	186
Reprise et redémarrage des serveurs de statut et serveurs de file d'attente	187
Reprise après incident grave à l'aide de RDF	188

Chapitre 13. Fichiers de configuration 189

Description des fichiers de configuration	189
Fichier de configuration (MQSINI)	189
Contenu du fichier de configuration MQSeries	190
Fichier de configuration du gestionnaire de files d'attente (QMINI)	191
Contenu du fichier de configuration du gestionnaire de files d'attente	191
Exemple de fichier de configuration de gestionnaire de files d'attente	193
Modification des fichiers de configuration	198

Implémentation des modifications dans les fichiers de configuration.	198
Recommandations pour les fichiers de configuration	198

Chapitre 14. Identification des incidents 199

Vérifications préalables	199
MQSeries a-t-il fonctionné correctement auparavant ?	199
Existe-t-il des messages d'erreur ?	199
Existe-t-il des codes retour clarifiant l'incident ?	200
L'incident peut-il être reproduit ?	200
Des modifications ont-elles été apportées depuis la dernière exécution réussie ?	200
L'exécution de l'application a-t-elle déjà abouti ?	201
Si l'application ne s'est jamais exécutée correctement auparavant	201
L'incident affecte-t-il des parties spécifiques du réseau ?	202
L'incident se produit-il à certaines heures de la journée ?	202
L'incident se produit-il par intermittence ?	203
Avez-vous effectué des mises à jour de fonctions ?	203
Erreurs courantes de programmation	203
Problèmes de commandes	204
Vérifications approfondies	204
Avez-vous obtenu une sortie incorrecte ?	204
Une réponse de la commande PCF vous manque-t-elle ?	204
L'incident est-il lié à certaines de vos files d'attente ?	205
L'incident n'affecte-t-il que les files d'attente éloignées ?	206
Votre application ou MQSeries pour Compaq NSK tournent-ils lentement ?	206
Considérations sur le développement d'applications	207
Effet de la longueur des messages	208
Recherche d'un message particulier	208
Files d'attente contenant des messages de différentes longueurs	208
Fréquence des points de synchronisation	208
Utilisation de l'appel MQPUT1	208
Sortie incorrecte	209
Messages qui n'apparaissent pas dans la file d'attente	209
Messages contenant des informations inattendues ou altérées	210
Sortie incorrecte lors de l'utilisation des files d'attente réparties	210
Journaux d'erreurs	212
Fichiers journaux	212
Erreurs précoces	213
Messages opérateur	213
Exemple de journal des erreurs	214
Événement EMS	215
Files d'attente de rebut	215
Fichiers de configuration et identification des incidents	215

Traçage de MQSeries	215
Fichiers de trace	216
Exemple de données de trace	216
First Failure Support Technology™ (FFST)	217
Consultation des fichiers FFST.	217

Chapitre 15. Evolutivité et performances 219

Introduction	219
Conception d'applications optimisées pour les performances et l'évolutivité	219
Réduction ou élimination de l'utilisation des ressources partagées	219
Optimisation des performances par approximation itérative	220
Messages persistants	220
Messages non-persistants	220
Messages non persistants et canaux	221
Serveurs de file d'attente et fichiers de file d'attente	222
Stockage des messages persistants	222
Stockage des messages non persistants	222
Répartition des serveurs de file d'attente entre les processeurs	222
Assignation d'une file d'attente à un serveur de file d'attente (commande altmqfls)	223
File d'attente de transmission de cluster : SYSTEM.CLUSTER.TRANSMIT.QUEUE	223
Déplacement des fichiers de file d'attente (commande altmqfls)	223
Partitionnement de fichiers de file d'attente	224
Fichiers de dépassement de capacité.	224
Mise en mémoire tampon des messages pendant leur consultation	225
Autres options de serveur de file d'attente	225
Affectation de processeurs	227
Bind des applications avec FASTPATH	228
Généralités	228
Réduction de la charge système MQI	228
Activation du bind FASTPATH	229
Restrictions lors de l'utilisation d'un bind FASTPATH	229

Chapitre 16. Intégrité et disponibilité des données. 231

Intégrité des données.	231
Disponibilité.	232
Données persistantes et non persistantes	233
Messages persistants	234
Messages non persistants	235
Cohérence de la base de données.	236
Cohérence des bases de données internes	236
Cohérence des bases de données externes	236
OpenTMF	237
NonStop Tuxedo	238
Transactions d'applications imbriquées	238
Fichiers de bases de données critiques de MQSeries.	238
Processus critiques	240
Clusters	245

Considérations sur la configuration pour la disponibilité.	246
Configuration favorable à l'intégrité des données	246

Partie 2. Section de référence . . . 249

Chapitre 17. Commandes de contrôle MQSeries 251

Résumé des Commandes de contrôle	251
Noms d'objets MQSeries	251
Lecture des diagrammes de syntaxe	252
altmqfls (modification des attributs d'un fichier de file d'attente)	254
altmqusr (modification des informations sur les utilisateurs MQSeries)	259
cleanrdf (exécution de la gestion interne de la RDF)	261
cnvclchl (conversion des définitions de canaux clients)	263
crtmqcvx (conversion de données)	265
crtmqm (création d'un gestionnaire de files d'attente)	267
dltmqm (suppression d'un gestionnaire de files d'attente)	271
dspmqaout (affichage des droits d'accès)	273
dspmqcsv (affichage du serveur de commandes)	277
dspmqls (affichage des fichiers MQSeries)	278
dspmqrtrc (affichage de la sortie de trace formatée MQSeries)	281
dspmqsusr (affichage des informations utilisateur MQSeries)	282
endmqcvs (arrêt d'un serveur de commandes)	284
endmqm (arrêt d'un gestionnaire de files d'attente)	286
endmqtrc (arrêt de la trace MQSeries)	288
instmqm (installation de MQSeries pour Compaq NSK)	289
runmqchi (exécution de l'initialisateur de canal)	290
runmqchl (démarrage d'un canal)	291
runmqdlq (lancement du gestionnaire de files d'attente de rebut)	292
runmqslr (exécution du programme d'écoute)	294
runmqsc (exécution des commandes MQSeries)	296
runmqtrm (lancement du moniteur de déclenchement)	299
setmqaut (définition ou réinitialisation des droits d'accès)	300
strmqcvs (lancement d'un serveur de commandes)	306
strmqm (lancement d'un gestionnaire de files d'attente)	307
strmqtrc (lancement de la trace MQSeries)	309
upgmqm (Mise à jour du gestionnaire de files d'attente V2.2.0.1)	312

Partie 3. Annexes 315

Annexe A. MQSeries pour Compaq NSK, généralités 317

Programme et référence	317
Caractéristiques matérielles requises.	317
Caractéristiques logicielles requises	317

Sécurité	318
Fonctions de maintenance	318
Compatibilité	318
Compilateurs pris en charge	318
Licences	319
Choix de la langue	319
Compatibilité internationale	319

Annexe B. Objets système et objets par défaut 321

Annexe C. Définition de variables d’environnement TACL pour MQSeries pour Compaq NSK 323

Paramètres d’adaptation du serveur de file d’attente	325
--	-----

Annexe D. Comparaison des jeux de commandes 327

Annexe E. Arrêt et suppression manuels de gestionnaires de file d’attente 331

Arrêt manuel d’un gestionnaire de files d’attente	331
Suppression manuelle de gestionnaires de file d’attente	331

Annexe F. MQSeries et Compaq NonStop Server pour Java. 333

Prise en charge des transactions	333
--	-----

Annexe G. Commandes MQSC prises en charge par MQSeries pour Compaq NSK 335

Attributs de MQSC	337
Informations d’état de canal (DISPLAY CHSTATUS).	337
MAXUMSGS et MAXHANDS.	338
HARDENBO et NOHARDENBO.	338
CONNAME.	338
Attribut USERDATA des applications déclenchées	338
Utilisation de noms d’exit comme attributs d’objets	338

Annexe H. Guide de référence de programmation des applications . . . 339

Types de données de structure	339
MQCNO – Options de connexion	340
MQGMO – Options de Get Message.	341
MQMD – Descripteur de message	341
MQPMO – Options de Put Message.	342
Appels MQI.	342
MQCLOSE – Fermeture d’un objet	343
MQDISC – Déconnexion d’un gestionnaire de files d’attente	344
MQINQ – Demande d’informations sur les attributs d’un objet	344

MQOPEN – Ouverture d’un objet	344
MQSET– Définition des attributs d’un objet	344
Attributs des objets MQSeries	345
Attributs de toutes les files d’attente.	345
Attributs des files d’attentes locales et modèles	345
Attributs des gestionnaires de files d’attente	345
Conversion de données	346

Annexe I. Construction et exécution d’applications 347

Ecriture des application	347
Utilisation du délai d’attente WAIT MQGET, et des commandes de canaux DISCONT et HBINT	347
Gestion des unités de travail (transactions)	348
Considérations générales sur le développement d’applications	349
Interface XA.	349
MQGMO_BROWSE_* avec l’option MQGMO_LOCK	349
Applications déclenchées	349
Environnements et langages pris en charge	351
Création d’applications utilisant des unités d’exécution	351
Compilation et lien des applications.	352
Comparaison des options de bind FASTPATH et STANDARD.	352
Exécution d’applications.	353

Annexe J. Interface d’administration MQSeries (MQAI). 355

Annexe K. Programmes exemples MQSeries pour Compaq NSK 357

Construction de programmes exemples en C	358
Non natif (à l’aide de la bibliothèque statique non native MQMLIB).	358
Natif (à l’aide de la bibliothèque statique native MQMLIBN)	359
Natif (à l’aide de SRL MQMLIBN)	359
Construction de programmes exemples en C++	360
Natif (à l’aide de la bibliothèque statique native MQMLIBN)	360
Natif (à l’aide de SRL MQMLIBN)	360
Construction de programmes exemples en COBOL	361
Non natif (à l’aide de la bibliothèque statique non native MQMLIB).	361
Natif (à l’aide de la bibliothèque statique native MQMLIBN)	361
Natif (à l’aide de SRL MQMLIBN)	362
Construction de programmes exemples en TAL	363
Non natif (à l’aide de la bibliothèque statique non native MQMLIB).	363
Constructions de programmes exemples OSS (mode natif uniquement)	363

Annexe L. Exits utilisateur. 365

Exits utilisateur pris en charge.	365
Format de nom d’exit	366

MQ_LOAD_ENTRY_POINT_EXIT - Chargement des exits utilisateur	366
MQLXP - Structure des paramètres de MQ_LOAD_ENTRY_POINT_EXIT	366
Installation des exits utilisateur	370
Installation d'un exit dans la SRL privée de MQSeries	370
Installation d'un exit dans la bibliothèque statique native de MQSeries	371
Installation d'un exit dans la bibliothèque statique non-native de MQSeries	371

Annexe M. Configuration des communications 375

Canaux SNA	375
Processus de canaux répondeurs LU 6.2	376
Canaux TCP/IP	381
Exemples de communication	382
Exemple de communications SNAX	382
Exemple de communications ICE	388
Exemple de communications TCP/IP	391

Annexe N. Clients MQSeries 395

Prise en charge des clients	395
Gestion de la sécurité	395

Annexe O. Programmation de la gestion du système. 397

Evénements outils	397
Types d'événements acceptés parMQSeries pour Compaq NSK	397
Format de message d'événement	398
Commandes programmables PCF	400
Descripteur de message PCF	400
En-tête PCF (MQCFH)	400

Paramètre chaîne PCF (MQCFST).	401
Paramètre PCF liste d'entiers (MQCFIL)	401
Entier PCF (MQCFIN)	401
Liste de chaînes PCF (MQCFSL)	402
Commandes PCF prises en charge par MQSeries pour Compaq NSK	402
Réponses des commandes PCF	404
Services optionnels	404
Interface du service d'autorisation	404
Interface de service annuaire	404

Annexe P. Modèle de fichier d'événement EMS utilisé par MQSeries pour Compaq NSK 405

Annexe Q. Remarques 409

Remarques	411
---------------------	-----

Bibliographie 413

Publications multi-plateformes MQSeries	413
Publications MQSeries spécifiques d'une plateforme	413
Manuel sous forme électronique	414
format HTML	414
Format PDF (Portable Document Format)	414
Format BookManager	415
Format PostScript	415
Format d'aide Windows	415
Informations sur MQSeries disponibles sur Internet	415
Publications connexes	415

Glossaire 417

Index 427

Figures

1. Processus MQSeries pour Compaq NSK	22	28. Ecran Channel Status	88
2. Composants de MQSeries pour Compaq NSK V5.1	26	29. Ecran Start/Stop Channel	89
3. Exemple de configuration PATHWAY	36	30. Ecran Reset Channel	90
4. Menu principal MQMQ de MQSeries pour Compaq NSK	71	31. Ecran Resolve Channel	91
5. Ecran Queue Manager Menu	72	32. Ecran Copy Channel	92
6. Ecran 2 du menu Queue Manager	72	33. Files d'attente, messages et applications	93
7. Menu QUEUE MANAGER TRACE	73	34. Exemple de sortie pour QMGR ALL	97
8. Ecran Search Criteria (queue)	75	35. Extrait du fichier de commandes MQSC monmqscin	99
9. Menu Queue	75	36. Extrait du fichier de rapport MQSC monmqscou	100
10. Ecran Create Queue	76	37. Administration éloignée	126
11. L'écran Create Local Queue	77	38. Configuration des canaux et des files d'attente pour l'administration éloignée	127
12. Ecran 2 Create Local Queue	77	39. Utilisation de groupes avec SAFEGUARD pour accorder un droit d'accès	143
13. Ecran Create Remote Queue	78	40. Exemple de données de contrôle	163
14. Ecran Copy Queue	78	41. Exemple de règle	164
15. Ecran Display/Modify Local Queue	79	42. Exemple de fichier de configuration MQSeries (MQSINI)	190
16. Ecran Monitor Local Queues	80	43. Exemple de fichier de configuration de gestionnaire de files d'attente (QMINI)	194
17. Ecran Search Criteria (canal)	81	44. Exemple de trace	216
18. Menu Channel	82	45. Exemple de rapport FFST	217
19. Ecran Display/Modify Sender Channel (1)	83	46. Exemple de programme MQLOADEXIT	369
20. Ecran Display/Modify Sender Channel (2)	83	47. Exemple de fichier de configuration SNAX MQSeries	378
21. Ecran Create Channel	84	48. Exemple de fichier de configuration SNA MQSeries pour ICE	380
22. Ecran Create Sender Channel	85		
23. Ecran Create Receiver Channel	85		
24. Ecran Create Server Connection Channel	86		
25. Ecran Create Cluster Sender Channel	86		
26. Ecran Create Cluster Receiver Channel	87		
27. Ecran Monitor Channels	87		

Tableaux

1. Fichiers exécutable MQSeries	24	18. Commandes d'administration des files d'attente	328
2. Liste des CCSID ISO possibles	52	19. Commandes d'administration des processus	328
3. Droits d'accès nécessaires pour les appels MQI	152	20. Commandes d'administration des canaux	329
4. Commandes MQSC et droits d'accès nécessaires	155	21. Autres commandes de contrôle	329
5. Commandes PCF et droits d'accès nécessaires	155	22. Interface Java	333
6. Comparaison entre le dépassement de file d'attente et le dépassement de message	234	23. Commandes MQSC prises en charge par MQSeries pour Compaq NSK	335
7. Fichiers de base de données contrôlée critiques	239	24. Liste des langages et environnements supportés	351
8. Méthodes de protection utilisées pour les processus critiques	240	25. Correspondance entre le langage et la version de la bibliothèque MQI	352
9. Droits d'accès de sécurité avec la commande dspmqaut	274	26. Exits utilisateur pris en charge par MQSeries pour Compaq NSK	365
10. Indication des droits d'accès pour différents types d'objets	303	27. Types d'événements acceptés par MQSeries pour Compaq NSK	397
11. Exemples de fichiers pouvant être effacés après une mise à jour	312	28. Structure d'un message d'événement (MQMD)	398
12. Objets système et objets par défaut - Files d'attente	321	29. Structure de l'en-tête (MQCFH)	399
13. Objets système et objets par défaut - Canaux	322	30. Descripteur de message PCF	400
14. Objets système et objets par défaut - Listes de noms	322	31. En-tête PCF	400
15. Objets système et objets par défaut - Processus	322	32. Paramètre PCF chaîne.	401
16. Commandes d'administration des gestionnaires de files d'attente	327	33. Liste d'entiers PCF.	401
17. Commandes d'administration des serveurs de commandes	327	34. Entier PCF	401
		35. liste de chaînes PCF	402
		36. Commandes PCF prises en charge par MQSeries pour Compaq NSK	402

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

Préface

MQSeries pour Compaq NonStop Kernel, Version 5 Edition 1 (également appelé MQSeries pour Compaq NSK ou MQSeries dans le présent manuel lorsque le contexte le permet), fait partie de la gamme de produits MQSeries. Ces produits proposent des services de programmation d'applications qui permettent aux programmes de communiquer entre eux à l'aide de *files d'attente de messages*. Cette forme de communication est appelée *messagerie applicative*. Les applications concernées peuvent se trouver sur différents noeuds et sur divers types de systèmes d'exploitation et de machines. Elles utilisent une interface commune de programmation d'applications, appelée MQI ou interface de file d'attente de messages, qui permet aux programmes développés sur une plateforme d'être facilement transférés sur une autre.

Ce manuel décrit les différents aspects de la gestion système de MQSeries pour Compaq NSK, Version 5 Edition 1, et des services qu'il fournit pour prendre en charge la messagerie interapplicative en environnement Compaq NSK. Il aborde notamment la gestion des files d'attente utilisées par les applications pour recevoir des messages et la façon de permettre à ces dernières d'accéder à ces files.

A qui s'adresse ce manuel

Ce manuel s'adresse principalement aux administrateurs système et aux programmeurs système qui assurent les tâches de configuration et d'administration pour MQSeries. Il peut également s'avérer utile pour les programmeurs d'applications ayant besoin d'avoir des notions sur les tâches d'administration MQSeries.

Connaissances préalables nécessaires à la lecture du manuel

Pour pouvoir utiliser ce manuel dans de bonnes conditions, vous devez avoir une bonne connaissance du système d'exploitation Compaq NSK et des utilitaires qui lui sont associés. Il n'est pas nécessaire d'avoir déjà utilisé des produits de mise en file d'attente de messages, mais vous devez cependant connaître les notions de base du fonctionnement de ces files d'attente.

Utilisation du manuel

Ce manuel :

- Présente MQSeries
- Décrit les tâches de gestion quotidiennes d'un système MQSeries pour Compaq NonStop Kernel, en traitant des sujets tels que l'administration des objets MQSeries locaux et distants, la sécurité, la prise en charge transactionnelle et l'identification des incidents

Informations sur MQSeries disponibles sur Internet

Adresse URL de MQSeries

L'adresse de la page d'accueil de la gamme de produits MQSeries est la suivante :

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

Nouvelles fonctions de MQSeries pour Compaq NSK V5.1

Les nouvelles fonctions qui suivent sont décrites dans cette édition du manuel *MQSeries pour Compaq NSK V5.1 - Administration du système*.

Amélioration des performances

Nouveau processus de serveur de file d'attente

Un nouveau processus de serveur de file d'attente a été ajouté (au gestionnaire de files d'attente). Il permet le stockage des messages d'une ou plusieurs files d'attente locales et gère toutes les opérations GET et PUT de ces files. Il permet une mise en oeuvre efficace des messages non persistants et prend en charge de nouvelles fonctions de messagerie pour la Version 5.1. Pour plus d'informations, reportez-vous à la section «Chapitre 2. Architecture MQSeries pour Compaq NSK V5.1» à la page 21.

Changements dans le fonctionnement du serveur de statut

Le serveur de statut remplace le concept de statut de canal basé sur des fichiers. Il fournit les informations de statut sur les objets situés dans des files d'attente éloignées et permet un accès efficace aux informations de statut de canal. Pour plus d'informations, consultez la section «Chapitre 2. Architecture MQSeries pour Compaq NSK V5.1» à la page 21.

Messages non persistants

Vous bénéficiez à présent des performances supérieures autorisées par les messages non persistants.

Prise en charge de l'édition des liens FASTPATH pour les applications sécurisées

Si votre application le permet, vous pouvez vous connecter à un gestionnaire de files d'attente grâce aux éditions de liens FASTPATH, pour bénéficier d'améliorations de performances significatives. Les applications FASTPATH font l'objet de certaines restrictions et doivent être manipulées avec soin, puisque cette méthode d'édition des liens diminue la protection des données critiques internes du gestionnaire de files d'attente. Pour plus d'informations, reportez-vous à la section «Annexe I. Construction et exécution d'applications» à la page 347.

Amélioration du stockage sur disque des messages persistants

Dans le cadre de la nouvelle architecture de serveur de file d'attente, le stockage des messages sur disque a été modifié pour améliorer les performances des messages, quelle que soit leur taille. Aucun autre fichier clé n'est requis pour les fichiers de files d'attente et un nouveau type de stockage sur disque pour messages très volumineux a été introduit. Il optimise l'efficacité du stockage des messages allant jusqu'à 100 Mo.

Fonctions MQSeries mises à niveau

Clusters de gestionnaires de files d'attente MQSeries

Les gestionnaires de files d'attente MQSeries peuvent être regroupés en un cluster. Au sein d'un cluster, chaque gestionnaire de files d'attente peut mettre ses propres files d'attente à la disposition de tous les autres gestionnaires. Un gestionnaire de files d'attente peut envoyer un message à tout autre gestionnaire du cluster sans avoir besoin de définitions de canal

ou de file d'attente explicites, ou de files d'attente de transmission pour chaque destination. Un cluster offre les avantages suivants :

- Réduction du nombre de tâches d'administration système
- Disponibilité accrue
- Rééquilibrage de la charge de travail

Consultez le manuel *MQSeries Queue Manager Clusters* pour une description complète de cette fonction.

interface d'administration MQSeries (MQAI)

MQSeries pour Compaq NSK V5.1 prend désormais en charge l'interface d'administration MQSeries (MQAI) qui simplifie l'utilisation des messages PCF pour configurer MQSeries. Pour plus d'informations sur MQAI et une description complète de ses commandes, reportez-vous au manuel *MQSeries Administration Interface Programming Guide and Reference*.

Prise en charge MQI étendue

MQSeries pour Compaq NSK V5.1 prend désormais en charge la fonctionnalité de messagerie avancée présente dans les Versions 5.0 et Version 5.1 de MQSeries sur d'autres plateformes. Elle comprend le traitement de la liste de diffusion, les messages de référence, les messages segmentés et de nombreuses autres nouvelles options. Consultez le *MQSeries - Guide de programmation d'applications* et le *MQSeries Application Programming Reference* pour plus d'informations.

Taille supérieure des messages et files d'attente de messages

La taille d'un message peut aller jusqu'à 2 Go. Celle d'une file d'attente de messages peut atteindre 4 Go.

Création automatique d'objets par défaut

Lorsque vous utilisez la commande `crtmqm` pour créer un gestionnaire de files d'attente, les objets système par défaut sont automatiquement créés.

Arrêt synchrone contrôlé d'un gestionnaire de files d'attente

Une nouvelle option a été ajoutée à la commande `endmqm` afin de permettre l'arrêt synchrone contrôlé d'un gestionnaire de files d'attente.

Support Java

MQSeries pour Compaq NSK V5.1 fonctionne à présent avec des compilateurs Java, et autorise donc le codage d'applications en Java. Pour plus d'informations, reportez-vous à l'«Annexe F. MQSeries et Compaq NonStop Server pour Java» à la page 333.

Support d'applications OSS

MQSeries pour Compaq NSK V5.1 fonctionne à présent avec les applications NSK OSS, en C, C++, Cobol et Java.

Administration Web

Avec MQSeries pour Compaq NSK V5.1, vous pouvez effectuer les tâches suivantes sous Microsoft® Windows NT® à l'aide d'un navigateur HTML tel que Netscape Navigator ou Microsoft Internet Explorer :

- Ouverture de session en tant qu'administrateur MQSeries
- Sélection d'un gestionnaire de files d'attente et émission de commandes MQSC relatives à ce dernier
- Création, édition et suppression de scripts MQSC

Amélioration du mécanisme d'exit utilisateur

Le mécanisme de liaison et de configuration de code d'exit utilisateur utilisé avec MQSeries a été considérablement amélioré pour fournir une

interface plus proche du standard, ainsi qu'un mécanisme commun à tous les exits. Consultez l'«Annexe L. Exits utilisateur» à la page 365 pour plus d'informations.

Intercommunications

TCP/IP

MQSeries pour Compaq NSK V5.1 permet désormais à un gestionnaire de files d'attente d'utiliser plusieurs processus serveur TCP/IP Guardian. Il en résulte une meilleure configuration de l'équilibrage de la charge de travail sur le matériel réseau, et une redondance plus efficace des connexions réseau pour un gestionnaire de files d'attente et les applications. Pour plus d'informations, consultez l'«Annexe M. Configuration des communications» à la page 375.

SNA

MQSeries pour Compaq NSK V5.1 a amélioré le mécanisme de gestion et de contrôle de l'initialisation à distance des canaux pour le protocole de transport SNA. Ce nouveau mécanisme utilise un processus d'écoute exécuté sous PATHWAY et pris en charge à la fois par SNAX et InSession. L'attribut de canal non standard AUTOSTART n'est plus pris en charge. Pour plus d'informations, reportez-vous à l'«Annexe M. Configuration des communications» à la page 375.

Canaux

Les canaux utilisent à présent des *signaux de présence* et peuvent transmettre des messages non persistants hors de l'unité de travail afin d'obtenir de meilleures performances.

MQSeries pour Compaq NSK V5.1 prend maintenant en charge la définition automatique des canaux en option, pour les canaux initialisés à distance à partir de gestionnaires de files d'attente ou clients.

Fonctions propres à Compaq NSK

Compaq NSK Fix Command incluse dans runmqsc

`runmqsc` intègre à présent la fonction Compaq NSK Fix Command pour permettre le rappel et la modification de commandes MQSC. Pour plus d'informations, reportez-vous à la section «Utilisation de Compaq NSK Fix Command» à la page 30.

Utilitaire `altnmqfls` étendu

L'utilitaire `altnmqfls` a été modifié de façon conséquente de façon à proposer une administration détaillée des options de stockage des messages. Consultez le «Chapitre 15. Evolutivité et performances» à la page 219 et la section «`dspmqfls` (affichage des fichiers MQSeries)» à la page 278 pour plus de détails.

Ecrans MQMC

Les écrans d'administration MQMC proposés dans l'environnement PATHWAY de gestionnaire de files d'attente ont été mis à niveau pour prendre en charge les fonctions MQSeries avancées de cette version.

Partie 1. Exploitation

Chapitre 1. Introduction	7	Chapitre 3. Utilisation des jeux de commandes MQSeries.	29
MQSeries et la mise en file d'attente de messages	7	Administration à l'aide des commandes de contrôle	29
Applications asynchrones	7	Utilisation des commandes de contrôle	29
Processus gérés par messages.	7	Administration à l'aide des commandes MQSC	30
Messages et files d'attente	8	Exécution de commandes MQSC	30
Définition d'un message	8	Utilisation de Compaq NSK Fix Command	30
Tailles de message	8	Administration à l'aide des commandes PCF	31
Définition d'une file d'attente.	9	Attributs dans les commandes MQSC et PCF	31
Envoi et réception de messages par les applications.	9	Commandes PCF d'échappement	31
Files d'attente prédéfinies et files d'attente dynamiques	9	Administration TS/MP (PATHWAY)	32
Extraction de messages des files d'attente	10	Détermination et contrôle des programmes d'écoute TCP/IP.	32
Objets	10	Contrôle du serveur de commandes	32
Noms des objets.	10	Détermination et contrôle des initialisateurs de canaux	32
Gestion des objets	11	Modification de la file d'attente d'initialisation par défaut pour l'initialisateur de canal	33
Attributs d'objet	11	Détermination et contrôle des moniteurs de déclenchement	33
Gestionnaires de file d'attente MQSeries.	11	Répartition des processus sur les processeurs	34
Appels MQI	12	Ajout de nouveaux processus de status server	35
Files d'attente MQSeries	12	Ajout de nouveaux processus de serveurs de file d'attente	35
Utilisation des objets de file d'attente.	12	Fréquence de rafraîchissement des écrans du moniteur MQM	35
Files d'attente locales utilisées par MQSeries	13	Configuration PATHWAY pour un gestionnaire de file d'attente	36
Définitions de processus	15	Modification des paramètres de classe de serveurs PATHWAY	42
Canaux.	15	Ajout de classes de serveurs personnalisées à un PATHWAY MQSeries	43
Clusters	16		
Listes de noms	16		
Objets système par défaut	16		
Administration	17		
Administration locale et éloignée	17		
Clients et serveurs	17		
Applications MQI en environnement client-serveur.	18		
Extension des fonctions du gestionnaire de files d'attente	18		
Exits utilisateur	18		
Services optionnels	19		
Sécurité.	19		
Gestionnaire des droits d'accès aux objets (OAM)	19		
Prise en charge transactionnelle.	19		
Optimisation des performances, fiabilité, évolutivité et préévaluation des capacités	20		
Chapitre 2. Architecture MQSeries pour Compaq NSK V5.1	21		
Généralités sur le processus de gestionnaire de files d'attente	21		
Mise en forme du produit	23		
Fichiers exécutables.	24		
Comparaison entre l'architecture de la Version 5.1 et celle de la Version 2.2.0.1	24		
Gestionnaire de files d'attente — vue fonctionnelle	25		
Modèle de processus de gestionnaire de files d'attente	26		
Fichiers et sous-volumes MQSeries	27		
		Chapitre 4. Gestion des gestionnaires de files d'attente	45
		Mise en route.	45
		Instructions pour la création d'un gestionnaire de files d'attente.	45
		Sauvegarde des fichiers de configuration après la création d'un gestionnaire de files d'attente.	48
		Modification des propriétés d'un gestionnaire de files d'attente.	48
		Volume origine du gestionnaire de files d'attente	48
		Nombre de processus EC par gestionnaire de files d'attente.	49
		Équilibrage de la charge de travail d'un système.	49
		Terminal d'origine du gestionnaire de files d'attente	49
		Nom de processus PATHMON du gestionnaire de files d'attente.	50
		CCSID du gestionnaire de files d'attente.	51
		Contrôle de la conversion de données EBCDIC	51
		Collecteur EMS du gestionnaire de files d'attente	52
		Groupe d'agents à la disposition des EC du gestionnaire de files d'attente	52

Nombre maximum d'agents en veille et réutilisation des processus	53
Priorité de traitement des processus de gestionnaire de files d'attente	53
Nombre maximum de canaux d'un gestionnaire de files d'attente.	53
Nombre maximum de canaux actifs d'un gestionnaire de files d'attente	54
ID de segment Guardian utilisés par MQSeries	54
Port TCP/IP par défaut	54
Ports TCP/IP écoutés par le gestionnaire de files d'attente	55
Processus TCP/IP utilisé par le gestionnaire de files d'attente	55
Reconfiguration de la classe de serveur MQS-TCPLISnn pour un port et un processus TCP/IP personnalisés	56
Affectation de l'espace de permutation	56
Nom de status server par défaut	57
Ajout et suppression de status server personnalisés	58
Nom de serveur de file d'attente par défaut	59
Ajout et suppression de serveurs de file d'attente personnalisés	60
Structure du volume	62
Sous-volume FFST d'un gestionnaire de files d'attente	62
Sous-volume de fichiers de données de gestionnaire de files d'attente	62
Sous-volume des journaux d'erreurs des gestionnaires de files d'attente	63
Sous-volume de files d'attente de messages de gestionnaires	63
Sous-volume de synchronisation des canaux des gestionnaires de files d'attente	64
Transformation des noms d'objets	64
Utilisation des gestionnaires de files d'attente	65
Création d'un gestionnaire de files d'attente par défaut	65
Création de principaux MQSeries	66
Exécution de cleanrdf pour un gestionnaire de files d'attente prêt pour RDF	67
Lancement d'un gestionnaire de files d'attente.	67
Restauration des objets système et par défaut	67
Consultation des fichiers objets.	67
Arrêt d'un gestionnaire de files d'attente	68
En cas d'incident	68
Arrêt immédiat et arrêt forcé d'un gestionnaire de files d'attente.	68
Redémarrage d'un gestionnaire de files d'attente	69
Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant	69
Suppression d'un gestionnaire de files d'attente	70
Utilisation de la fonction MQM (Message Queue Management).	70
Utilisation du menu Queue Manager	72
Modification des attributs d'un gestionnaire de files d'attente.	73
Traçage des objets MQSeries.	73
Utilisation du menu Queues.	75
Création d'une file d'attente.	76

Copie d'une file d'attente.	78
Modification d'une file d'attente	79
Suppression d'une file d'attente	79
Contrôle d'une file d'attente.	80
Utilisation du menu Channels	81
Modification d'un canal	83
Création d'une définition de processus	84
Contrôle d'un canal	87
Suppression d'un canal	88
Affichage du statut des canaux.	88
Démarrage et arrêt d'un canal	89
Réinitialisation d'un numéro de séquence de message (MSN)	90
Validation ou annulation d'un canal	91
Copie d'un canal	92

Chapitre 5. Administration des objets MQSeries locaux	93
Prise en charge des applications utilisant l'interface MQI.	93
Administration locale à l'aide de commandes MQSC	94
Préambule.	94
Noms des objets MQSeries	95
Utilisation des majuscules et minuscules dans les commandes MQSC.	95
Utilisation de la fonction MQSC de façon interactive.	95
Informations renvoyées par les commandes MQSC	96
Fin de l'entrée interactive sur MQSC	96
Affichage des attributs d'un gestionnaire de files d'attente	96
Utilisation d'un gestionnaire différent du gestionnaire par défaut	97
Modification des attributs d'un gestionnaire de files d'attente.	98
Exécution de commandes MQSC à partir de fichiers texte.	98
Utilisation des fichiers de commandes MQSC	99
Utilisation des rapports MQSC	100
Exécution du fichier de commandes MQSC fourni	101
Utilisation de runmqsc pour vérifier des commandes	101
Résolution des incidents à l'aide des commandes MQSC.	101
Manipulation des files d'attente locales.	103
Définition d'une file d'attente locale.	103
Modification de la taille du fichier physique des files d'attente	103
Définition d'une file d'attente de rebut	104
Affichage des attributs d'objet par défaut	104
Copie d'une définition de file d'attente locale	105
Modification des attributs d'une file d'attente locale	106
Modification du volume d'une file d'attente locale	106
Modification des options d'une file d'attente locale	106
Reassignation d'objets aux serveurs de statut et de file d'attente.	107

Nettoyage d'une file d'attente locale	108
Suppression d'une file d'attente locale	108
Visualisation des files d'attente	108
Manipulation des files d'attente alias	111
Définition d'une file d'attente d'alias	112
Utilisation d'autres commandes avec les files d'attente alias	113
Manipulation des files d'attente modèles	113
Définition d'une file d'attente modèle	114
Utilisation d'autres commandes avec les modèles	114
Gestion des objets pour le déclenchement	114
Définition d'une file d'attente d'application pour le déclenchement	115
Définition d'une file d'attente d'initialisation	116
Création d'une définition de processus	116
Affichage de la définition de processus	117

Chapitre 6. Automatisation des tâches

d'administration	119
Commandes programmables PCF.	119
Attributs dans les commandes MQSC et PCF	120
Commandes PCF d'échappement.	120
Utilisation de l'interface MQAI pour simplifier l'utilisation des commandes programmables PCF	120
Gestion du serveur de commandes pour l'administration éloignée	121
Lancement du serveur de commandes	121
Affichage de l'état du serveur de commandes	121
Arrêt d'un serveur de commandes	122

Chapitre 7. Administration des objets MQSeries éloignés.

.	123
Canaux, clusters et mise en file d'attente éloignée	123
Administration éloignée à l'aide de clusters	124
Administration d'un gestionnaire de files d'attente	125
Préparation des gestionnaires pour l'administration éloignée	125
Préparation des canaux et des files d'attente de transmission pour l'administration éloignée	126
Définition des canaux et des files d'attente de transmission.	127
Lancement des canaux	128
Définition automatique de canaux	129
Emission de commandes MQSC à distance	129
Opérations sur des gestionnaires de files d'attente sous MVS/ESA	130
Recommandations pour la mise en file d'attente éloignée	130
En cas d'incident	131
Création d'une définition locale de file d'attente éloignée	131
Fonctionnement de la définition locale d'une file d'attente éloignée	131
Exemple	131
Fonctionnement	132
Autre mode d'insertion de messages dans une file d'attente éloignée	133
Utilisation d'autres commandes avec les files d'attente éloignées.	133

Création d'une file d'attente de transmission	133
Files d'attente de transmission par défaut	134
Utilisation des définitions de files d'attente éloignées pour les alias	134
Gestionnaire de files d'attente alias	134
Files d'attente de réponse alias	135
Conversion de données	135
Quand le gestionnaire de files d'attente ne peut pas convertir les messages dans des formats livrés d'origine	135
Fichier CCSID	136
Conversion des données par défaut	136
Conversion de messages en formats définis par l'utilisateur	136
Modification du CCSID d'un gestionnaire de files d'attente	136

Chapitre 8. Protection des objets MQSeries

Nécessité de protéger les ressources MQSeries	139
Présentation des ID utilisateur du groupe d'utilisateurs MQM	139
Informations complémentaires.	140
Présentation du gestionnaire des droits d'accès aux objets (OAM)	140
Mode de fonctionnement de l'OAM	141
Gestion de l'accès par le biais de groupes d'utilisateurs	141
Lorsqu'un utilisateur appartient à plus d'un groupe	141
Ensemble de groupes et groupe principal	141
Protection de ressources avec l'OAM	142
Utilisation de groupes pour les droits d'accès	142
Désactivation du gestionnaire des droits d'accès aux objets (OAM)	143
Utilisation des commandes de l'OAM	144
Informations à préciser aux commandes	144
Listes de droits d'accès	145
Utilisation de la commande almqusr	145
Utilisation de la commande dspmqusr	145
Utilisation de la commande setmqaut	146
Commandes de droits d'accès et services optionnels	147
Droits d'accès	147
Commande d'affichage des droits d'accès	147
Instructions concernant le gestionnaire des droits d'accès aux objets (OAM)	148
ID utilisateur	148
Volumes du gestionnaire de files d'attente.	148
Files d'attente	148
Droits d'accès d'utilisateur secondaire	148
Droits de contexte	149
Gestion éloignée de la sécurité	150
Sécurité des commandes de canal	150
Commandes PCF	150
Commandes de canal MQSC	151
Tables de spécification des droits d'accès	151
Droits d'accès MQI	151
Droits d'administration	154
Droits des commandes MQSC dans les PCF d'échappement	155
Droits pour les commandes PCF	155

Fichiers de droits d'accès	157
Base de données de principaux	157
Base de données OAM	157
Enregistrements de droits d'accès aux classes	159
Enregistrement de droit d'accès à toutes les classes.	159

Chapitre 9. Gestionnaire de file d'attente de rebut MQSeries	161
Lancement du gestionnaire DLQ	161
Table de règles du gestionnaire DLQ	162
Données de contrôle	162
Règles (modèles et actions).	164
Mots clés de modèle	164
Mots clés d'action	165
Conventions de la table de règles.	167
Traitement de la table de règles	168
Assurance du traitement de tous les messages DLQ	169
Exemple de table de règles pour un gestionnaire DLQ	170

Chapitre 10. Événements outils et événements EMS	173
Événements outils MQSeries	173
Types d'événements	173
Notification d'événements via les files d'attente d'événements	174
Utilisation de files d'attente d'événements déclenchées	174
Activation des événements outil	175
Messages d'événements	175
Événements EMS (Event Management Service)	175
Fichier modèle EMS fourni avec MQSeries pour Compaq NSK	175
Intégration des modèles d'événements EMS pour MQSeries	176
Définition du PARAM MQEMSEVENTS	179
Utilisation d'un autre programme de collecte	179
Ecriture de programmes de traitement des événements EMS MQSeries.	179

Chapitre 11. Prise en charge transactionnelle et messagerie	181
Utilisation de NonStop TM/MP (gestionnaire de transactions).	181
Limitations du système de points de synchronisation.	182
Opérations sur les messages persistants, sans point de synchronisation	182
Opérations avec point de synchronisation sur les messages non persistants	183
Configuration requise pour TM/MP et MQSeries pour Compaq NSK	183
Contrôle	183
Taille du fichier d'audit	183
Configuration du gestionnaire de ressources	184
Identification des problèmes	184

Chapitre 12. Reprise et redémarrage.	185
Tolérance aux pannes et reprise	185

Sauvegarde et restauration de MQSeries	186
Sauvegarde de MQSeries	186
Restauration de MQSeries	186
Reprise et redémarrage des serveurs de statut et serveurs de file d'attente	187
Reprise après incident grave à l'aide de RDF	188

Chapitre 13. Fichiers de configuration	189
Description des fichiers de configuration	189
Fichier de configuration (MQSINI)	189
Contenu du fichier de configuration MQSeries	190
Fichier de configuration du gestionnaire de files d'attente (QMINI)	191
Contenu du fichier de configuration du gestionnaire de files d'attente	191
Exemple de fichier de configuration de gestionnaire de files d'attente	193
Modification des fichiers de configuration.	198
Implémentation des modifications dans les fichiers de configuration.	198
Recommandations pour les fichiers de configuration	198

Chapitre 14. Identification des incidents	199
Vérifications préalables	199
MQSeries a-t-il fonctionné correctement auparavant ?	199
Existe-t-il des messages d'erreur ?	199
Existe-t-il des codes retour clarifiant l'incident ?	200
L'incident peut-il être reproduit ?	200
Des modifications ont-elles été apportées depuis la dernière exécution réussie ?	200
L'exécution de l'application a-t-elle déjà abouti ?	201
Si l'application ne s'est jamais exécutée correctement auparavant	201
L'incident affecte-t-il des parties spécifiques du réseau ?	202
L'incident se produit-il à certaines heures de la journée ?	202
L'incident se produit-il par intermittence ?	203
Avez-vous effectué des mises à jour de fonctions ?	203
Erreurs courantes de programmation	203
Problèmes de commandes	204
Vérifications approfondies	204
Avez-vous obtenu une sortie incorrecte ?	204
Une réponse de la commande PCF vous manque-t-elle ?	204
L'incident est-il lié à certaines de vos files d'attente ?	205
L'incident n'affecte-t-il que les files d'attente éloignées ?	206
Votre application ou MQSeries pour Compaq NSK tournent-ils lentement ?	206
Considérations sur le développement d'applications	207
Effet de la longueur des messages	208
Recherche d'un message particulier	208
Files d'attente contenant des messages de différentes longueurs	208
Fréquence des points de synchronisation	208

Utilisation de l'appel MQPUT1	208
Sortie incorrecte	209
Messages qui n'apparaissent pas dans la file d'attente	209
Messages contenant des informations inattendues ou altérées	210
Sortie incorrecte lors de l'utilisation des files d'attente réparties	210
Journaux d'erreurs	212
Fichiers journaux	212
Erreurs précoces	213
Messages opérateur	213
Déchiffrement des numéros EC dans le fichier MQERRLG	213
Exemple de journal des erreurs	214
Événement EMS	215
Files d'attente de rebut	215
Fichiers de configuration et identification des incidents	215
Traçage de MQSeries	215
Fichiers de trace	216
Exemple de données de trace	216
First Failure Support Technology™ (FFST)	217
Consultation des fichiers FFST	217

Chapitre 15. Evolutivité et performances 219

Introduction	219
Conception d'applications optimisées pour les performances et l'évolutivité	219
Réduction ou élimination de l'utilisation des ressources partagées	219
Optimisation des performances par approximation itérative	220
Messages persistants	220
Messages non-persistants	220
Messages non persistants et canaux	221
Serveurs de file d'attente et fichiers de file d'attente	222
Stockage des messages persistants	222
Stockage des messages non persistants	222
Répartition des serveurs de file d'attente entre les processeurs	222
Assignation d'une file d'attente à un serveur de file d'attente (commande almqfls)	223
File d'attente de transmission de cluster : SYSTEM.CLUSTER.TRANSMIT.QUEUE	223
Déplacement des fichiers de file d'attente (commande almqfls)	223
Partitionnement de fichiers de file d'attente	224
Fichiers de dépassement de capacité	224
Mise en mémoire tampon des messages pendant leur consultation	225
Autres options de serveur de file d'attente	225
Chargement au démarrage	225
Verrouillage en cache	226
Point de contrôle NPM	226
Compteur Measure	226
Affectation de processeurs	227
Bind des applications avec FASTPATH	228
Généralités	228
Réduction de la charge système MQI	228

Activation du bind FASTPATH	229
Restrictions lors de l'utilisation d'un bind FASTPATH	229

Chapitre 16. Intégrité et disponibilité des données. 231

Intégrité des données.	231
Disponibilité.	232
Données persistantes et non persistantes	233
Messages persistants	234
Messages non persistants	235
Cohérence de la base de données.	236
Cohérence des bases de données internes	236
Cohérence des bases de données externes	236
OpenTMF	237
NonStop Tuxedo	238
Transactions d'applications imbriquées	238
Fichiers de bases de données critiques de MQSeries.	238
Processus critiques	240
Clusters	245
Considérations sur la configuration pour la disponibilité.	246
Configuration favorable à l'intégrité des données	246

Chapitre 1. Introduction

Ce chapitre présente MQSeries pour Compaq NonStop Kernel Version 5.1, du point de vue de l'administrateur. Les concepts de base de MQSeries et des applications de messagerie y sont décrits. Il contient les sections suivantes :

- «MQSeries et la mise en file d'attente de messages»
- «Messages et files d'attente» à la page 8
- «Objets» à la page 10
- «Files d'attente MQSeries» à la page 12
- «Définitions de processus» à la page 15
- «Canaux» à la page 15
- «Objets système par défaut» à la page 16
- «Administration» à la page 17
- «Clients et serveurs» à la page 17
- «Extension des fonctions du gestionnaire de files d'attente» à la page 18
- «Sécurité» à la page 19
- «Prise en charge transactionnelle» à la page 19
- «Optimisation des performances, fiabilité, évolutivité et préévaluation des capacités» à la page 20

MQSeries et la mise en file d'attente de messages

MQSeries permet aux applications d'utiliser la mise en file d'attente pour participer à des processus gérés par messages. Les programmes d'application installés sur différentes plateformes pouvant communiquer par le biais de logiciels de mise en file d'attente de messages. Par exemple, les applications Compaq NSK et MVS/ESA* peuvent respectivement communiquer à l'aide de MQSeries pour Compaq NSK et MQSeries pour OS/390®. Les applications sont isolées de toute l'infrastructure de communication sous-jacente.

Les produits MQSeries mettent en oeuvre une interface de programmation commune, l'interface MQI (Message Queue Interface), utilisée sur toute plateforme sur laquelle s'exécutent des applications. Le portage des programmes d'application d'une plateforme à une autre en est facilité.

L'interface MQI est décrite en détail dans le manuel *MQSeries Application Programming Reference*.

Applications asynchrones

L'échange de messages entre les programmes émetteur et récepteur est libéré des contraintes de temps grâce à la mise en file d'attente. Cela signifie que les programmes sont découplés, et que l'émetteur peut continuer à travailler sans avoir à attendre que le récepteur envoie l'accusé de réception du message. En fait, il n'est même pas nécessaire que l'application cible s'exécute au moment de l'envoi du message. Elle peut récupérer ce dernier après avoir démarré.

Processus gérés par messages

Grâce à un mécanisme appelé *déclenchement*, les applications peuvent être lancées automatiquement par des messages arrivant dans une file d'attente. Si besoin est, elles peuvent aussi être arrêtées une fois que le ou les messages ont été traités.

Messages et files d'attente

Les messages et les files d'attente sont les composants de base d'un système de mise en file d'attente de messages.

Définition d'un message

Un *message* est une chaîne d'octets ayant une signification pour les applications qui l'utilisent. Les messages servent à transférer des informations d'un programme à un autre (ou entre des parties différentes de la même application). Il n'est pas nécessaire que les applications s'exécutent sur la même plateforme.

Les messages MQSeries se composent de deux parties :

- *données d'application*

Le contenu et la structure des données d'application sont définis par les programmes d'application qui les utilisent.

- *descripteur de message*

Le descripteur identifie le message et contient d'autres informations de contrôle, comme le type du message et la priorité qui lui est attribuée par l'application émettrice.

Le format du descripteur de message est défini par MQSeries. Pour une description complète du descripteur, reportez-vous au manuel *MQSeries Application Programming Reference*.

Le format du descripteur de message est défini par MQSeries. Pour une description complète du descripteur, reportez-vous au manuel *MQSeries Application Programming Reference*.

Il existe deux types de messages : messages persistants et messages non persistants. Les différences sont les suivantes :

- Les messages persistants existent encore après le redémarrage du gestionnaire de files d'attente, pas les non persistants.
- Les messages non persistants ne sont normalement pas enregistrés sur le disque, et sont ainsi plus rapides et utilisent moins de ressources pour être ajoutés ou retirés d'une file d'attente.
- Dans certaines conditions, les messages non persistants ne sont pas aussi fiables que les messages persistants.
- Les messages persistants ne peuvent être placés dans une file d'attente dynamique temporaire.

Tailles de message

La taille maximale d'un message est de 100 Mo (où 1 Mo = 1 048 576 octets). En pratique, elle peut être limitée par :

- la taille maximale définie pour la file d'attente réceptrice.
- la taille maximale définie pour le gestionnaire de files d'attente.
- la taille maximale définie par l'application émettrice ou réceptrice.
- la quantité de mémoire disponible pour le message.

Plusieurs messages peuvent être nécessaires pour envoyer toutes les informations dont une application a besoin.

L'augmentation de la taille maximum des messages pourrait avoir des conséquences négatives. Le message pourrait alors être trop long pour la file

d'attente ou le gestionnaire de files d'attente. Un message peut alors être segmenté puis regroupé en un message logique. Le regroupement logique de messages permet aux applications de regrouper des messages similaires et d'assurer la séquence des messages. Pour plus d'informations, reportez-vous au manuel *MQSeries - Guide de programmation d'applications*.

Définition d'une file d'attente

Une *file d'attente* est une structure de données qui sert au stockage des messages. Ceux-ci peuvent être insérés dans la file par des programmes d'application ou par un gestionnaire de files d'attente au cours de son fonctionnement normal.

Chaque file d'attente appartient à un *gestionnaire de files d'attente*. Ce dernier assure la mise à jour des files d'attente qu'il possède et le stockage de tous les messages qu'il reçoit dans les files d'attente appropriées.

La taille maximale d'une file d'attente est de 4 Go. Pour plus d'informations sur la planification de la quantité de mémoire requise pour les files d'attente, reportez-vous au *Guide de planification MQSeries* ou consultez le site Web suivant :

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

Envoi et réception de messages par les applications

Les programmes d'application envoient et reçoivent des messages à l'aide d'*appels MQI*. Par exemple, pour placer un message en file d'attente, l'application effectue les opérations suivantes :

1. Ouverture de la file d'attente requise en émettant un appel MQI MQOPEN.
2. Emission d'un appel MQPUT pour insérer le message dans cette file d'attente
3. Une autre application peut ensuite extraire le message de cette même file d'attente en émettant un appel MQGET.

Pour plus d'informations sur les appels MQI, consultez le manuel *MQSeries Application Programming Reference*.

Files d'attente prédéfinies et files d'attente dynamiques

Les files d'attente peuvent être caractérisées par leur mode de création :

- Les *files d'attente prédéfinies* sont créées par un administrateur à l'aide du jeu de commandes approprié. Vous pouvez créer une file d'attente locale prédéfinie à l'aide de la commande MQSC DEFINE QLOCAL. Les files de ce type sont permanentes ; elles existent indépendamment des applications qui les utilisent et ne sont pas supprimées lors des redémarrages de MQSeries.
- Les *files d'attente dynamiques* sont créées lorsqu'une application émet un appel OPEN indiquant un *nom de file correspondant à un modèle*. La file créée est basée sur une définition type qui est la file d'attente modèle. Vous pouvez créer un tel modèle à l'aide de la commande MQSC DEFINE QMODEL. Les attributs de la file d'attente modèle (par exemple le nombre maximal de messages qu'elle peut contenir) sont transmis à toutes les files dynamiques créées à partir d'elle.

Les modèles possèdent un attribut qui indique si la file dynamique est permanente ou temporaire. A l'inverse des files d'attente permanentes, les files temporaires sont supprimées lors du redémarrage des applications ou du gestionnaire de files d'attente.

Messages et files d'attente

Extraction de messages des files d'attente

Les applications dûment autorisées peuvent récupérer les messages dans une file d'attente selon les algorithmes d'extraction suivants :

- Premier entré, premier sorti (FIFO).
- Priorité des messages, comme définie dans le descripteur de message. Les messages qui ont la même priorité sont extraits en fonction du système FIFO.
- Une requête de programme relative à un message spécifique.

La requête MQGET émise par l'application détermine la méthode utilisée.

Objets

De nombreuses tâches décrites dans ce manuel font appel à la manipulation d'*objets* MQSeries. Dans MQSeries Version 5.1, les objets peuvent être de plusieurs types : gestionnaires de files d'attente, files d'attente, définitions de processus, canaux, clusters et listes de noms.

La manipulation ou l'*administration* d'objets comprend les tâches suivantes :

- le lancement et l'arrêt des gestionnaires de files d'attente
- la création d'objets (notamment des files d'attente) pour des applications ;
- l'utilisation de canaux pour créer des voies de communication vers des gestionnaires de files situés sur d'autres systèmes (éloignés). Pour plus de détails, reportez-vous au manuel *MQSeries Intercommunication*.
- Création de *clusters* de gestionnaires de files d'attente pour simplifier l'ensemble du processus d'administration ou rééquilibrer la charge de travail.

Vous trouverez des informations détaillées sur les tâches d'administration dans les chapitres suivants :

- «Chapitre 3. Utilisation des jeux de commandes MQSeries» à la page 29
- «Chapitre 4. Gestion des gestionnaires de files d'attente» à la page 45
- «Chapitre 5. Administration des objets MQSeries locaux» à la page 93
- «Chapitre 7. Administration des objets MQSeries éloignés» à la page 123

Noms des objets

Chaque instance d'un gestionnaire de files d'attente a un nom d'objet. Ce nom doit être unique au sein du réseau de gestionnaires de files d'attente pour permettre l'identification du gestionnaire de files d'attente cible auquel le message est expédié.

Ces noms doivent être uniques pour un gestionnaire et un type d'objet donné. Par exemple, une file d'attente et un processus peuvent s'appeler de la même façon, mais pas deux files d'attente.

Dans MQSeries, les noms peuvent comporter 48 caractères au maximum, à l'exception des *canaux*, qui ont un maximum de 20 caractères. Pour plus d'informations sur les noms, consultez la section «Noms d'objets MQSeries» à la page 251.

Gestion des objets

Vous pouvez créer, modifier, afficher et supprimer des objets à l'aide des commandes et méthodes suivantes :

- Commandes MQSeries (MQSC), saisies à partir d'un clavier ou lues à partir d'un fichier.
- MQM (interface écran)
- Commandes PCF (format de commande programmable) utilisables par un programme.
- Commande de contrôles que vous pouvez saisir de façon interactive à partir de la ligne de commande du système d'exploitation.

Pour plus d'informations, reportez-vous au «Chapitre 3. Utilisation des jeux de commandes MQSeries» à la page 29.

Attributs d'objet

Les propriétés d'un objet sont définies par ses attributs. Vous pouvez en définir et modifier certains, tandis que d'autres sont en lecture seule. Par exemple, la longueur maximale des messages qu'une file d'attente peut recevoir est définie par l'attribut *MaxMsgLength*. Vous pouvez définir cet attribut lorsque vous créez une file d'attente. L'attribut *DefinitionType* définit comment la file a été créée. Vous pouvez seulement le visualiser.

Il existe deux façons de faire référence à un attribut dans MQSeries :

- à l'aide de son nom PCF ; par exemple *MaxMsgLength*. Le nom formel d'un attribut est son nom PCF.
- à l'aide de son nom MQSC ; par exemple, MAXMSGL.

Le nom formel d'un attribut est son nom PCF. Toutefois, du fait que l'utilisation de la fonction MQSC représente une partie importante de ce manuel, les noms MQSC des attributs y sont peut-être plus fréquemment utilisés que les noms PCF.

Gestionnaires de file d'attente MQSeries

Programme système mettant à la disposition des applications des fonctions de mise en file d'attente. Il contrôle que :

- les attributs des objets ont été modifiés en fonction des commandes reçues ;
- les événements spéciaux (tels que les événements de déclenchement et les événements outils) sont générés lorsque les conditions nécessaires sont réunies ;
- les messages sont insérés dans la file d'attente adéquate, comme demandé par l'application qui a émis l'appel MQPUT. L'application est informée des échecs éventuels et un code anomalie approprié est alors envoyé.

Chaque file d'attente appartient à un seul gestionnaire et est considérée comme *file d'attente locale*. Le gestionnaire de files d'attente auquel une application est connectée est considéré comme local à cette application. Pour cette dernière, les files d'attente appartenant à son gestionnaire local sont des files locales.

Une *file d'attente éloignée* est une file qui appartient à un autre gestionnaire.

Tout gestionnaire de files d'attente autre que le gestionnaire local est dit *gestionnaire de files d'attente éloigné*. Un gestionnaire éloigné peut s'exécuter sur une autre machine du réseau, ou sur la même machine que le gestionnaire local.

MQSeries est capable d'exécuter plusieurs gestionnaires sur la même machine.

Appels MQI

Un objet gestionnaire de files d'attente peut être utilisé dans certains appels MQI. Par exemple, vous pouvez lire les attributs de l'objet gestionnaire à l'aide de l'appel MQINQ.

Remarque : Vous ne pouvez pas insérer de messages dans un objet gestionnaire, car ils doivent toujours être insérés dans des objets files d'attente.

Files d'attente MQSeries

Sur MQSeries pour Compaq NSK les files d'attente sont définies à l'aide des commandes suivantes :

- Commandes MQSC DEFINE
- Service MQM (gestion de file d'attente des messages) de MQSeries pour Compaq NSK
- Commande PCF Create Queue (création d'une file d'attente)
- Commandes MQAI (interface d'administration MQSeries)

La commande indique le type de la file et ses attributs. Par exemple, un objet file d'attente locale a des attributs qui indiquent ce qui se produit quand des applications y font référence dans des appels MQI. Les attributs peuvent, par exemple, définir :

- si les applications peuvent extraire des messages de la file (GET activé) ;
- si les applications peuvent insérer des messages dans la file (PUT activé) ;
- si l'accès à la file est réservé à une application ou est partagé entre plusieurs applications ;
- le nombre maximal de messages pouvant être stockés dans la file à un moment donné (longueur maximale de la file) ;
- la taille maximale des messages qui peuvent être insérés dans la file.

Pour plus d'informations :

- concernant MQSC, reportez-vous au manuel *MQSeries - Guide de référence des commandes MQSC*
- concernant MQM, reportez-vous à la section «Utilisation de la fonction MQM (Message Queue Management)» à la page 70
- concernant les commandes PCF, reportez-vous au manuel *MQSeries Programmable System Management*

Utilisation des objets de file d'attente

MQSeries fait appel à plusieurs types d'objets de file d'attente. Chaque type peut être manipulé par les commandes MQSeries et est associé aux files réelles de différentes manières.

• **Objet file d'attente locale**

Un objet file d'attente locale identifie une file locale appartenant au gestionnaire auquel l'application est connectée. Toutes les files d'attente sont locales dans le sens où chacune appartient à un gestionnaire de files d'attente et que, pour celui-ci, la file est locale.

• **Objet file d'attente éloignée**

Un objet file d'attente éloignée identifie une file appartenant à un autre gestionnaire. Cette file doit être définie comme locale à ce gestionnaire. Les informations spécifiées lors de la définition de l'objet file d'attente éloignée permettent au gestionnaire de files d'attente local de trouver le gestionnaire éloigné adéquat afin de lui faire parvenir les messages destinés à la file éloignée.

Pour que des applications puissent envoyer des messages à la file d'un autre gestionnaire, vous devez également définir une file d'attente et des canaux de transmission entre les gestionnaires à moins que vous ayez regroupé plusieurs gestionnaires de files d'attente dans un cluster. Pour plus d'informations sur les clusters, reportez-vous à la section «Administration éloignée à l'aide de clusters» à la page 124.

- **Objet file d'attente alias**

Un objet file d'attente alias permet aux applications d'accéder à une file en y faisant référence de manière indirecte dans les appels MQI. Quand un nom de file d'attente alias est utilisé dans un appel MQI, ce nom est mis en correspondance (résolu) avec celui d'une file locale ou éloignée au moment de l'exécution. Cela vous permet de changer les files utilisées par les applications sans modifier l'application elle-même ; vous devez simplement modifier la définition de la file d'attente alias de façon à ce qu'elle pointe sur le nom de la nouvelle file.

Un alias n'est pas une file d'attente, mais un objet permettant d'accéder à une autre file.

- **Objet file d'attente modèle**

Un objet file d'attente modèle définit un ensemble d'attributs utilisés comme modèle pour créer une file dynamique. Les files d'attente dynamiques sont créées par le gestionnaire lorsqu'une application émet un appel MQOPEN indiquant un nom de file correspondant à un modèle. La file dynamique créée de cette façon est une file locale dont les attributs sont extraits de la définition du modèle. Le nom de cette file peut être soit spécifié par l'application, soit généré par le gestionnaire de files d'attente (qui le renvoie alors à l'application).

Les files dynamiques créées de cette façon peuvent être permanentes ou temporaires, selon qu'elle subsistent ou non après un redémarrage du produit.

Files d'attente locales utilisées par MQSeries

MQSeries utilise des files locales pour des besoins précis liés à son fonctionnement. Pour que MQSeries puisse utiliser ces files d'attente, vous *devez* préalablement les définir.

Files d'attente d'application : Une file d'attente utilisée par une application (par le biais de l'interface MQI) est appelée file d'attente d'application. Il peut s'agir d'une file locale du gestionnaire de files d'attente auquel l'application est liée, ou d'une file éloignée appartenant à un autre gestionnaire.

Les applications peuvent insérer des messages dans les files locales ou éloignées. Cependant, elles ne peuvent extraire des messages que dans une file locale.

Files d'attente d'initialisation : Les files d'initialisation sont utilisées lors du déclenchement. Un gestionnaire de files d'attente insère un message de déclenchement dans la file d'initialisation quand un événement de déclenchement se produit. Ce type d'événement est une combinaison logique de conditions détectée par un gestionnaire. Par exemple, un événement de déclenchement peut être généré lorsque le nombre de messages dans une file atteint un seuil prédéfini. Cet événement amène le gestionnaire à insérer un message de déclenchement dans la file d'initialisation spécifiée. Ce message est alors extrait par un *moniteur de déclenchement*, c'est-à-dire une application spéciale qui contrôle une file d'initialisation. Le moniteur lance ensuite le programme d'application indiqué dans le message de déclenchement.

Si un gestionnaire de files d'attente doit utiliser le déclenchement, au moins une file d'initialisation doit être définie pour ce gestionnaire.

Files d'attente MQSeries

Consultez la section «Gestion des objets pour le déclenchement» à la page 114. Pour plus d'informations sur le déclenchement, reportez-vous au manuel *MQSeries - Guide de programmation d'applications*.

Files d'attente de transmission : Les files d'attente de transmission stockent temporairement les messages destinés à un gestionnaire de files d'attente éloigné. Vous devez définir au moins une file d'attente de transmission pour chaque gestionnaire éloigné auquel le gestionnaire local doit envoyer directement des messages. Ces files sont aussi utilisées dans le cadre de l'administration à distance. Consultez la section «Administration d'un gestionnaire de files d'attente» à la page 125. Pour plus d'informations sur l'utilisation des files d'attente de transmission dans le cadre de la gestion répartie des files d'attente, reportez-vous au manuel *MQSeries Intercommunication*.

Files d'attente de transmission de cluster : Tout gestionnaire de files d'attente faisant partie d'un cluster possède une file d'attente de transmission appelée `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Une définition de cette file d'attente est créée par défaut sur chaque gestionnaire de files d'attente.

Un gestionnaire de files d'attente faisant partie d'un cluster peut envoyer des messages dans la file d'attente de transmission de tout autre gestionnaire membre du cluster.

Les gestionnaires de files d'attente d'un cluster peuvent communiquer avec les gestionnaires n'en faisant pas partie. Pour cela, vous devez définir des canaux et une file d'attente de transmission pour l'autre gestionnaire de la même façon que dans l'environnement de mise en file d'attente répartie.

Lors de la procédure de résolution du nom, c'est la file d'attente de transmission du cluster qui est prioritaire sur la file d'attente de transmission par défaut. Lorsqu'un gestionnaire de files d'attente ne faisant pas partie d'un cluster envoie un message vers une file d'attente éloignée et qu'il n'existe pas de file d'attente de transmission portant le même nom que le gestionnaire de files d'attente cible, le système utilise la file d'attente de transmission par défaut.

Lorsqu'un gestionnaire de files d'attente fait partie d'un cluster, le système utilise par défaut `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, sauf si le gestionnaire de files d'attente cible ne fait pas partie du cluster.

Files d'attente de rebut : Une file d'attente de rebut stocke les messages qui n'ont pas pu être acheminés vers leur destination. Une telle situation peut se produire lorsque, par exemple, la file de destination est saturée. La file de rebut fournie par défaut s'appelle `SYSTEM.DEAD.LETTER.QUEUE`. Les files de ce type sont parfois désignées sous le nom de files des messages non livrés.

Pour la mise en file d'attente répartie, vous devez définir une file de rebut pour chaque gestionnaire de files d'attente concerné.

Files d'attente de commandes : La file d'attente de commandes, appelée `SYSTEM.ADMIN.COMMAND.QUEUE`, est une file locale à laquelle les applications dûment autorisées peuvent envoyer des commandes MQSeries pour traitement. Ces commandes sont alors extraites par un composant MQSeries appelé serveur de commandes. Ce serveur valide les commandes, transmet celles qui sont correctes au gestionnaire de files d'attente pour traitement et renvoie les éventuelles réponses à la file appropriée.

Files d'attente de réponses : Quand une application émet un message de demande, l'application qui le reçoit peut envoyer un message de réponse à l'application émettrice. Ce message est inséré dans une file d'attente de réponses, qui est normalement locale à l'application émettrice. Le nom de cette file est spécifié par l'application émettrice dans le descripteur de message.

Files d'attente des événements : MQSeries pour Compaq NSK prend en charge les événements outils, qui peuvent servir à contrôler des gestionnaires de files d'attente indépendamment des applications MQI. Les événements outils peuvent être générés dans diverses circonstances :

- une application tente d'insérer un message dans une file d'attente non disponible ou qui n'existe pas ;
- une file d'attente est saturée ;
- un canal est démarré.

Quand un événement outil se produit, le gestionnaire de files d'attente insère un message dans une file d'attente des événements. Ce message peut alors être lu par une application de surveillance qui peut informer un administrateur ou lancer des actions de correction si l'événement indique un incident.

Remarque : Les événements de déclenchement sont très différents des événements outils car ils ne sont pas provoqués par les mêmes conditions et ne génèrent pas de messages d'événement.

Pour plus d'informations sur les événements outils, reportez-vous au manuel *MQSeries Programmable System Management*.

Définitions de processus

Un *objet de définition de processus* définit une application qui doit être lancée en réponse à un événement de déclenchement sur un gestionnaire de files d'attente MQSeries pour Compaq NSK. Pour plus d'informations, reportez-vous à la section «Files d'attente d'initialisation» à la page 13.

Les attributs de la définition de processus comprennent l'ID de l'application, son type et des données spécifiques à l'application.

Pour créer une définition de processus, utilisez la commande MQSC DEFINE PROCESS ou la commande PCF Create Process.

Canaux

Les *canaux* sont des objets qui fournissent des voies de communication entre gestionnaires de files d'attente. Dans la mise en file d'attente répartie, ils servent à déplacer les messages d'un gestionnaire à un autre. Ils isolent les applications des protocoles de communication sous-jacents. Il n'est pas nécessaire que les gestionnaires de files d'attente s'exécutent sur la même plateforme. Pour que ces derniers puissent communiquer entre eux, vous devez définir un objet canal pour le gestionnaire émetteur de messages, et un autre canal complémentaire pour le gestionnaire qui les reçoit.

MQSeries pour Compaq NSK prend en charge les protocoles de transport TCP/IP et SNA LU6.2 (SNAX ou ICE).

Canaux

Pour plus d'information sur les canaux et leur utilisation, reportez-vous au manuel *MQSeries Intercommunication* ainsi qu'à la section «Préparation des canaux et des files d'attente de transmission pour l'administration éloignée» à la page 126.

Clusters

Dans un réseau MQSeries classique utilisant la mise en file d'attente répartie, chaque gestionnaire de files d'attente est indépendant. Pour qu'un gestionnaire de files d'attente puisse envoyer des messages à un autre gestionnaire, il doit posséder une définition de file d'attente de transmission, une définition de canal vers le gestionnaire éloigné et une définition de file d'attente éloignée pour chaque file destinée à recevoir des messages.

Un cluster est un groupe de gestionnaires de files d'attente configuré de manière à ce qu'ils puissent communiquer directement sur un réseau simple sans avoir recours à des définitions de file d'attente de transmission, de canaux et de files d'attente complexes.

Pour plus d'informations sur les clusters, reportez-vous au «Chapitre 7. Administration des objets MQSeries éloignés» à la page 123 et au manuel *MQSeries Queue Manager Clusters*.

Listes de noms

Une liste de noms est un objet MQSeries qui contient une liste d'objets MQSeries. Généralement, les listes de noms sont utilisées par des applications telles que les moniteurs de déclenchement, lesquels permettent l'identification d'un groupe de files d'attente. La liste de noms offre l'avantage d'être gérée indépendamment des applications. Elle peut donc être mise à jour sans arrêter les applications qui l'utilisent. De même, en cas d'échec d'une application, la liste de noms reste intacte et les autres applications peuvent continuer à l'utiliser.

Les listes de noms sont également utiles pour les clusters de gestionnaires de files d'attente car elles permettent de disposer d'une liste de clusters référencée par plusieurs objets MQSeries.

Objets système par défaut

Les *objets système par défaut* représentent un ensemble de définitions créées automatiquement chaque fois qu'un gestionnaire de files d'attente est créé.

La racine du nom des objets par défaut est SYSTEM.DEFAULT ; par exemple, la file d'attente locale par défaut s'appelle SYSTEM.DEFAULT.LOCAL.QUEUE, et le canal récepteur par défaut est SYSTEM.DEFAULT.RECEIVER. Vous ne pouvez pas renommer ces objets qui sont, par ailleurs, obligatoires.

Lorsque vous définissez un objet, tous les attributs qui ne sont pas spécifiés explicitement sont copiés à partir de l'objet par défaut approprié. Par exemple, si vous définissez une file d'attente locale, les attributs non spécifiés sont extraits de la file par défaut SYSTEM.DEFAULT.LOCAL.QUEUE.

Après la création d'un gestionnaire de files d'attente, vous pouvez utiliser la commande **runmqsc** pour remplacer des valeurs par défaut par d'autres définitions.

Pour plus d'informations sur les objets système par défaut, reportez-vous à l'«Annexe B. Objets système et objets par défaut» à la page 321.

Administration

Sous MQSeries, vous exécutez des tâches d'administration en entrant des *commandes*. Quatre jeux de commandes sont disponibles. Le choix du jeu utilisé dépend des tâches que vous voulez effectuer et de la manière dont vous voulez les effectuer. Les jeux de commandes sont décrits dans le «Chapitre 3. Utilisation des jeux de commandes MQSeries» à la page 29. Les tâches d'administration sont les suivantes :

- le lancement et l'arrêt des gestionnaires de files d'attente.
- la création d'objets (notamment des files d'attente) pour des applications ;
- l'utilisation de canaux pour créer des voies de communication vers des gestionnaires de files situés sur d'autres systèmes (éloignés). Cette procédure est expliquée en détails dans le manuel *MQSeries Intercommunication*.

Administration locale et éloignée

L'administration locale recouvre la réalisation de tâches administratives sur n'importe quel gestionnaire de files d'attente défini sur votre système local. Vous pouvez accéder à d'autres systèmes, par exemple par le programme d'émulation de terminal TCP/IP **telnet**, et y effectuer des tâches d'administration. Avec MQSeries il s'agit d'administration locale car aucun canal n'est utilisé, c'est-à-dire que la communication est gérée par le système d'exploitation.

MQSeries prend en charge l'administration à partir d'un point unique par le biais de la méthode d'*administration éloignée*. A partir de votre système local, vous pouvez ainsi émettre des commandes qui sont exécutées sur un autre système. Il n'est pas nécessaire de se connecter à ce système, mais les canaux appropriés doivent être définis. Le gestionnaire de files d'attente et le serveur de commandes du système cible doivent être actifs. Par exemple, vous pouvez émettre une commande éloignée pour modifier la définition d'une file sur un gestionnaire éloigné.

Certaines commandes ne peuvent pas être émises de cette façon, en particulier la création et le lancement de gestionnaires de files d'attente, ainsi que le lancement de serveurs de commandes. Pour effectuer ce type de tâche, vous devez soit vous connecter au système éloigné et y émettre les commandes, soit créer un processus qui exécutera ces commandes à votre place.

Clients et serveurs

MQSeries pour Compaq NSK prend en charge les configurations client-serveur pour les applications MQSeries. Il n'existe pas de clients MQSeries pour Compaq NSK, seulement un serveur MQSeries pour Compaq NSK. Cependant, les clients sur d'autres plateformes peuvent se connecter au serveur MQSeries pour Compaq NSK.

Un *client MQI* est un élément MQSeries installé sur une machine afin de recevoir des appels MQI en provenance d'applications et de les transmettre à un serveur MQI. Sur le serveur, ces appels sont traités par un gestionnaire de files d'attente. En général, le client et le serveur sont installés sur des machines différentes, mais ce n'est pas une obligation.

Un *serveur MQI* est un gestionnaire de files d'attente qui fournit des fonctions de gestion de files à un ou plusieurs clients. Tous les objets MQSeries, par exemple les

Clients et serveurs

files d'attente, n'existent que sur la machine du gestionnaire, c'est-à-dire sur le serveur MQI. Un serveur peut également prendre en charge les applications MQI locales normales.

Pour plus d'informations, reportez-vous au manuel *MQSeries Intercommunication* et au manuel *MQSeries - Clients*.

Applications MQI en environnement client-serveur

Une fois associées à un serveur, les applications MQI clientes peuvent émettre des appels MQI de la même manière que les applications locales. L'application cliente émet un appel MQCONN pour se connecter à un gestionnaire de files d'attente spécifié. Ce dernier traitera par la suite tout appel MQI comportant l'identificateur de connexion renvoyé par la demande de connexion.

Vous devez faire le link de vos applications avec les bibliothèques clientes appropriées. Pour plus d'informations, reportez-vous au manuel *MQSeries - Guide de programmation d'applications*. Aucun client MQI n'existe actuellement pour Compaq NSK. Cependant, puisque Compaq NSK est un serveur MQI, il accepte les connexions issues d'un client MQI MQSeries fonctionnant sur une autre plateforme.

Extension des fonctions du gestionnaire de files d'attente

Les fonctions fournies par le gestionnaire de files d'attente peuvent être étendues par :

- des exits utilisateur ;
- des services optionnels.

Exits utilisateur

Avec ces exits, les utilisateurs peuvent insérer leur propre code dans une fonction d'un gestionnaire de files d'attente. Les exits utilisateur pris en charge sont les suivants :

- *Exits de canal*

Ces exits modifient le mode de fonctionnement des canaux. Ils sont décrits dans le manuel *MQSeries Intercommunication*.

- *Exits de conversion de données*

Ces exits créent des fragments de code source pouvant être inclus dans des programmes d'application en vue de convertir des données d'un format donné à un autre. Ils sont décrits dans le manuel *MQSeries - Guide de programmation d'applications*.

- *Exit de pondération de charge du cluster*

Cet exit peut servir à définir comment un gestionnaire de files d'attente dans un cluster choisit entre plusieurs instances de files d'attente éloignées. Pour plus d'informations, consultez le manuel *MQSeries Queue Manager Clusters*.

Remarque : Le mécanisme d'activation d'exits utilisateurs a été modifié dans la version 5.1. Relisez attentivement la description des mécanismes d'exit afin de déterminer les modifications nécessaires à apporter pour convertir votre code exit à partir des versions précédentes de MQSeries pour Compaq NSK.

Pour plus d'informations, reportez-vous à l'«Annexe L. Exits utilisateur» à la page 365.

Services optionnels

Les services optionnels sont plus puissants que les exits, dans le sens où ils offrent des API formalisées avec plusieurs points d'entrée.

Un service optionnel est mis en oeuvre par l'intermédiaire d'un *composant de service*. Vous pouvez utiliser les composants fournis avec le produit, ou écrire votre propre composant pour effectuer les fonctions souhaitées. Actuellement, MQSeries est livré avec les services optionnels suivants :

- **Service d'autorisation**

Ce service vous permet d'établir votre propre système de sécurité.

Le composant par défaut qui met en oeuvre ce service est le gestionnaire des droits d'accès aux objets (OAM), livré avec le produit. Par défaut, l'OAM est activé. Vous pouvez utiliser l'interface du service d'autorisation pour créer d'autres composants afin de remplacer ou d'étendre l'OAM. Pour plus d'informations sur l'OAM, reportez-vous au «Chapitre 8. Protection des objets MQSeries» à la page 139.

- **Service annuaire**

Le service annuaire active le partage des files d'attente en permettant aux applications d'identifier les files éloignées comme si elles étaient locales.

Vous pouvez écrire votre propre composant service annuaire. Consultez le manuel *MQSeries Programmable System Management* pour plus d'informations.

Sécurité

MQSeries pour Compaq NSK assure la sécurité au moyen du gestionnaire des droits d'accès aux objets (OAM).

Gestionnaire des droits d'accès aux objets (OAM)

Les droits d'utilisation des appels et des commandes MQI et d'accès aux objets sont accordés par le gestionnaire des droits d'accès aux objets (OAM), qui est activé par défaut. L'accès aux entités MQSeries est contrôlé par les principaux et groupes d'utilisateurs MQSeries pour Compaq NSK et par l'OAM. Les noms de groupes et principaux supportés par l'OAM sont convertis en noms de groupes et d'utilisateurs Compaq NSK. Dans la version 5.1, tous les utilisateurs MQSeries doivent avoir un nom principal qui correspond à un nom d'utilisateur Compaq NSK. Ceci est nécessaire, que l'OAM soit activé ou non. Une interface de ligne de commande est fournie pour vous permettre d'ajouter ou supprimer des principaux et d'accorder ou de révoquer des droits selon les besoins.

De plus, les droits d'accès Compaq NSK peuvent être utilisés pour contrôler l'accès aux commandes et fichiers de base de données MQSeries. MQSeries est compatible avec SAFEGUARD, et si ce dernier est installé, peut profiter des nouvelles fonctions qu'il propose. Pour plus d'informations, reportez-vous au «Chapitre 8. Protection des objets MQSeries» à la page 139.

Prise en charge transactionnelle

Un programme peut regrouper un ensemble de mises à jour dans une *unité de travail*. En général, ces mises à jour sont reliées de façon logique et doivent toutes aboutir pour que l'intégrité des données soit préservée. Si une seule opération échoue, les données ne sont plus cohérentes.

Sécurité

Une unité de travail réalise une **validation** lorsque son traitement a abouti. A ce stade, toutes les mises à jour effectuées dans cette unité deviennent permanentes et irréversibles. En revanche, si l'unité de travail échoue, toutes les opérations de mise à jour sont *annulées*. La coordination au point de synchronisation est le processus par lequel les unités de travail sont soit validées soit annulées, avec préservation de l'intégrité des données.

Une unité de travail est dite *locale* lorsque les seules ressources mises à jour sont celles du gestionnaire de files d'attente MQSeries. Dans ce cas, la coordination au point de synchronisation est fournie par le gestionnaire lui-même, par le biais d'un processus de validation en une seule phase.

Une unité de travail est dite *globale* lorsque les ressources qui appartiennent à d'autres gestionnaires de ressources (tels que les bases de données conformes à la norme XA) sont également mises à jour. Dans ce cas, il est indispensable de recourir à un processus de validation à deux phases, et l'unité de travail peut être coordonnée soit par le gestionnaire de files d'attente lui-même, soit en externe par un autre gestionnaire de transactions conforme à la norme XA tel que IBM®, CICS®, Transarc Encina ou BEA Tuxedo.

Sous MQSeries pour Compaq NSK, toutes les unités de travail sont coordonnées par le TM/MP ou un produit en couche compatible utilisant lui-même le TM/MP (par exemple NonStop Tuxedo dans l'environnement OSS). Les applications peuvent utiliser le TM/MP pour coordonner des unités de travail qui contiennent des mises à jour à toute ressource compatible avec TM/MP ainsi que des messages MQSeries. L'environnement Compaq NSK ne contient pas de base de données conforme à la norme XA, mais une interface similaire est mise en oeuvre. Des mises à jour pour bases de données NonStop SQL ou fichiers ENSCRIBE peuvent par exemple être coordonnées avec des opérations de messagerie MQSeries pour conserver l'intégrité de la base de données.

Pour plus d'informations, reportez-vous au «Chapitre 11. Prise en charge transactionnelle et messagerie» à la page 181.

Optimisation des performances, fiabilité, évolutivité et préévaluation des capacités

Vous disposez de fonctions pour permettre à MQSeries pour Compaq NSK de bénéficier des caractéristiques des logiciels et matériels Compaq NSK qui offrent une infrastructure d'applications évolutive, performante et fiable. Ces caractéristiques comprennent :

- la possibilité de configurer les gestionnaires de file d'attente et leurs objets sur plusieurs processeurs et disques afin d'augmenter le parallélisme
- un programme s'assurant que les points de panne unique seront gérés de façon à n'engendrer, au pire, qu'une interruption momentanée des fonctions avec retour automatique à un fonctionnement normal
- l'intégrité de la base de données, protégée par un logiciel et le système de fichiers transactionnels sous-jacents
- la compatibilité avec les produits qui offrent une protection par reprise après sinistre.

Pour plus d'informations, reportez-vous au «Chapitre 15. Evolutivité et performances» à la page 219.

Chapitre 2. Architecture MQSeries pour Compaq NSK V5.1

Ce chapitre décrit l'architecture globale de MQSeries pour Compaq NSK V5.1. Il comprend les sections suivantes :

- «Généralités sur le processus de gestionnaire de files d'attente»
- «Mise en forme du produit» à la page 23
- «Fichiers exécutables» à la page 24
- «Comparaison entre l'architecture de la Version 5.1 et celle de la Version 2.2.0.1» à la page 24
- «Gestionnaire de files d'attente — vue fonctionnelle» à la page 25
- «Modèle de processus de gestionnaire de files d'attente» à la page 26
- «Fichiers et sous-volumes MQSeries» à la page 27

Ces informations vous aideront à configurer le produit de façon optimale dans votre environnement d'exploitation.

Le «Chapitre 15. Evolutivité et performances» à la page 219 et le «Chapitre 16. Intégrité et disponibilité des données» à la page 231 fournissent des informations de configuration spécifiques à cette architecture.

Généralités sur le processus de gestionnaire de files d'attente

La figure ci-dessous décrit les processus qui composent un gestionnaire de files d'attente MQSeries (PATHMON est indiqué dans une souci de précision, puisque le gestionnaire de files d'attente s'exécute sous PATHWAY). La figure décrit également les interactions IPC du processus.

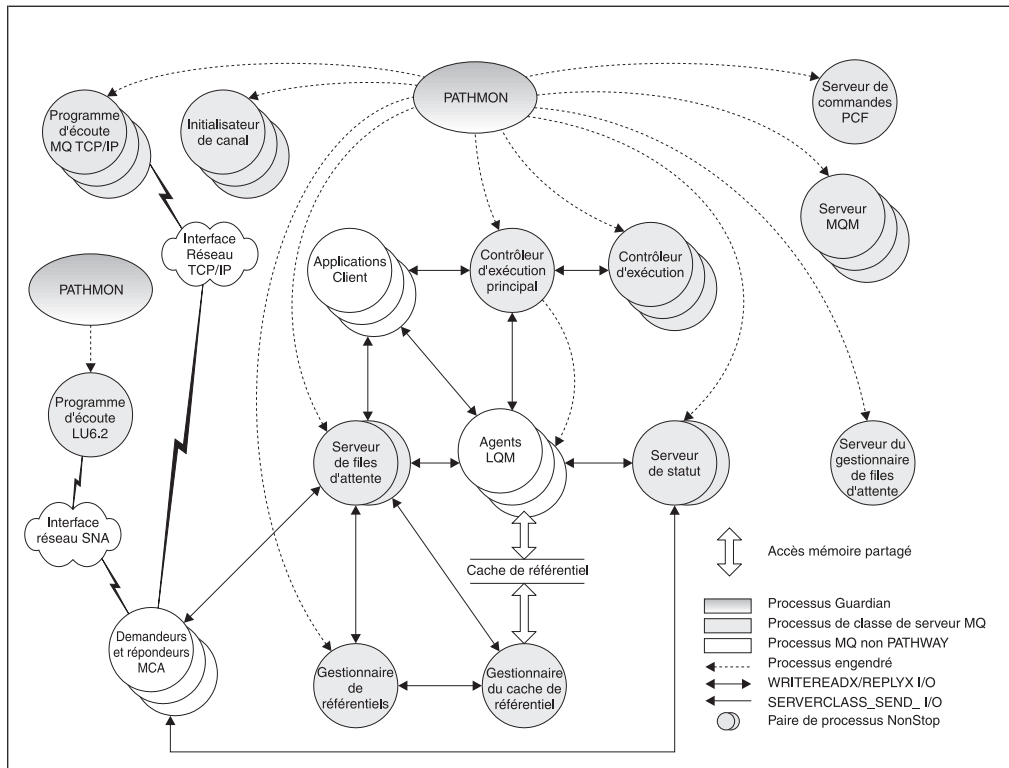


Figure 1. Processus MQSeries pour Compaq NSK

Lorsqu'un gestionnaire de files d'attente est démarré, PATHWAY crée les processus clés de gestionnaire de files d'attente. Dans la configuration par défaut, PATHMON démarre le contrôleur d'exécution principal (ECBoss), les contrôleurs d'exécution (EC), les serveurs de file d'attente et de statut, et le gestionnaire de référentiel.

ECBoss gère toutes les requêtes MQCONN entrantes et les distribue sur les contrôleurs d'exécution disponibles dans un souci de répartition de la charge, en sélectionnant l'EC doté du plus petit nombre de connexions.

Les EC gèrent et contrôlent les autres processus de gestionnaires de files d'attente et applications MQI.

Les LQMA (Local Queue Manager Agents) exécutent les opérations nécessaires au traitement des appels MQI à la place des applications. Les Agents exécutent l'ensemble du code prenant en charge l'interface MQI. Le rôle principal de l'agent est de séparer les programmes d'application des ressources critiques du gestionnaire de files d'attente, afin de les protéger des applications malveillantes et nuisibles.

Le nombre de processus agents dépend du nombre d'applications connectées et des MinIdleLQMAgents indiqués dans le fichier QMINI. Lorsque le gestionnaire de files d'attente démarre, chaque contrôleur d'exécution démarre le nombre spécifié d'agents inactifs. Au fur et à mesure que les agents deviennent actifs, le contrôleur d'exécution démarre de nouveaux agents pour maintenir le nombre minimal d'agents inactifs.

Le *serveur de file d'attente* est l'élément central de l'architecture MQSeries pour la Version 5.1. Le serveur de file d'attente est une paire de processus NonStop prenant en charge toutes les opérations de messagerie des files d'attente locales. A

sa création, un gestionnaire de files d'attente n'est doté que d'un serveur de file d'attente par défaut. En fonction de votre configuration système et des performances requises, vous pouvez configurer des serveurs de file d'attente supplémentaires et leur affecter des files locales.

Le serveur de statut manipule les informations de statut de tous les objets autres que les files d'attente locales. Ce serveur comprend la fonctionnalité du serveur MQSS utilisée dans la Version 2.2.0.1. De plus, le serveur de statut par défaut gère les informations de statut de canal pour le gestionnaire de files d'attente. A sa création, un gestionnaire de files d'attente n'est doté que d'un serveur de statut par défaut. En fonction de votre configuration système et des performances requises, vous pouvez configurer des serveurs de statut supplémentaires et leur affecter des objets MQSeries.

Lors de sa mise en route, le gestionnaire de files d'attente démarre également automatiquement la classe de serveur d'initialisateur de canal. Ceci permet de procéder à des opérations de mise en cluster sans qu'il soit nécessaire de démarrer manuellement l'initialisateur de canal.

La Version 5.1 propose de nouveaux processus de *gestionnaire de référentiel* et de *serveur de cache de référentiel* permettant de manipuler les files d'attente de cluster. Un processus de gestionnaire de référentiel figure sur chaque processeur exécutant un contrôleur d'exécution. Le premier processus démarré joue le rôle de gestionnaire de référentiel et assure la coordination des activités de référentiel du gestionnaire de files d'attente. Les processus suivants gèrent un segment de mémoire partagée contenant le cache de référentiel du processeur dans lequel ils s'exécutent. Il ne peut y avoir qu'un gestionnaire de référentiel par processeur. Le cache de référentiel contient des informations sur les objets MQSeries en cluster, dont les files d'attente et autres gestionnaires de files d'attente du cluster.

Un nouvel élément de la Version 5.1 est le *serveur de gestionnaire de files d'attente*. Ce serveur gère les messages expirés et génère des rapports.

Les agents MCA (Message Channel Agents) transfèrent des messages en provenance et à destination d'autres gestionnaires de files d'attente. Sous MQSeries pour Compaq NSK V5.1, les MCA sont liés avec FASTPATH, pour une meilleure efficacité. Pour plus d'informations sur les applications FASTPATH, consultez la section «Bind des applications avec FASTPATH» à la page 228.

Le programme d'écoute LU6.2 (exécuté dans un environnement PATHWAY distinct) est un nouvel élément de la Version 5.1. L'«Annexe M. Configuration des communications» à la page 375 contient des informations détaillées sur l'utilisation et la configuration du programme d'écoute et de son environnement.

Mise en forme du produit

MQSeries pour Compaq NSK V5.1 propose trois types de liaisons : natives dynamiques, natives statiques et non natives.

Pour les liaisons natives dynamiques, MQSeries propose une bibliothèque de ressources partagées en mode natif (MQSRLLIB). L'un des objectifs de conception de MQSeries pour Compaq NSK V5.1 était de faire tenir la plus grande partie possible du produit dans la bibliothèque de ressource partagée (SRL), et de réduire son encombrement. La liaison dynamique native est la meilleure approche pour utiliser MQSeries, car elle permet d'optimiser les ressources système. Dans l'environnement OSS, il s'agit de la seule liaison supportée.

Pour les liaisons natives statiques, MQSeries pour Compaq NSK propose une bibliothèque reconnectable (MQSRLLNK). Cette bibliothèque est destinée à ceux qui utilisent déjà une SRL et ne peuvent donc pas utiliser la SRL MQSeries.

MQSeries propose une bibliothèque statique non native compatible avec les applications existantes qui ne peuvent pas utiliser les liaisons en mode natif.

Fichiers exécutable

Le tableau suivant présente les fichiers exécutable MQSeries :

Tableau 1. Fichiers exécutable MQSeries

Fichiers exécutable MQSeries	Nom
Contrôleur d'exécution	MQEC
Contrôleur d'exécution principal	MQECBOSS
Agent de gestionnaire de files d'attente locales	MQLQMAG
Serveur de statut	MQSTSVR
Serveur de file d'attente	MQQSSVR
Gestionnaire de référentiel	MQREPSVR
Serveur de gestionnaire de files d'attente	MQMGRSVR
Serveur de commandes	MQCMDSVR
Serveur de gestion MQSeries	MQMSVR
Initialisateur de canal	RUNMQCHI
Agent MCA demandeur	MQMCACAL
Agent MCA répondeur TCP/IP	MQTCPRES
Agent MCA répondeur LU6.2	MQLU6RES
Moniteur de déclenchement	RUNMQTRM

Comparaison entre l'architecture de la Version 5.1 et celle de la Version 2.2.0.1

Le changement majeur entre MQSeries Version 2.2.0.1 et Version 5.1 est l'apparition de serveurs de file d'attente. Chaque serveur de file d'attente gère les données et fichiers associés à un ou plusieurs objets de files d'attente locales. A sa création, un gestionnaire de files d'attente n'est doté que d'un serveur de file d'attente par défaut, responsable de toutes les files d'attente d'un gestionnaire. Vous pouvez ajouter des serveurs de file d'attente supplémentaires (à l'aide de PATHWAY) et leur affecter des files à l'aide de **altmqfls**.

Les serveurs de file d'attente sont des paires de processus NonStop qui prennent en charge toutes les opérations de messagerie des files d'attente locales. Le serveur de file d'attente prend aussi en charge des messages non persistants simplement basés en mémoire, pour une amélioration significative des performances par rapport à la Version 2.2.0.1. MQSeries utilise une interface interne vers TM/MP pour intégrer ces messages basés en mémoire aux transactions. Il permet ainsi d'intégrer des messages persistants et non persistants à la même transaction sans avoir à écrire les messages non persistants sur le disque. Les serveurs de file d'attente reprennent la fonction des serveurs MQSS de la Version 2.2.0.1, pour les files d'attente locales uniquement.

Les serveurs MQSS ont été remplacés par les serveurs de statut. Les serveurs de statut sont des paires de processus NonStop qui gèrent les informations de statut des objets autres que les files d'attente locales.

Une autre modification importante de l'architecture est l'introduction de segments de mémoire partagée. Le fichier d'initialisation du gestionnaire de files d'attente est réparti sur le gestionnaire de files d'attente grâce à de la mémoire partagée en lecture seule. Ceci permet d'obtenir de meilleures performances lors de la connexion à un gestionnaire de files d'attente. Le cache de référentiel MQSeries (utilisé pour mettre en oeuvre des fonctions de mise en cluster) utilise un segment de mémoire partagé en lecture/écriture, sur chaque processeur exécutant des processus MQSeries.

La Version 5.1 permet une prise en charge plus transparente des canaux LU6.2. Elle intègre un programme d'écoute LU6.2, responsable du démarrage des canaux répondeurs LU6.2. L'attribut "AUTOSTART" des canaux LU6.2 a par conséquent été supprimé.

Gestionnaire de files d'attente — vue fonctionnelle

Les principaux composants du gestionnaire de files d'attente sont les suivants :

Interface d'application

Procure l'environnement et les mécanismes permettant l'exécution des appels MQI.

Noyau de gestionnaire de files d'attente

Fournit la plupart des fonctions de MQI. Il permet par exemple de réaliser les déclenchements.

Gestionnaire de droits d'accès aux objets (OAM)

Fournit un contrôle d'accès au gestionnaire de files d'attente et à ses ressources. Il permet de préciser quels utilisateurs et groupes seront autorisés à réaliser telles opérations sur telles ressources.

Persistance et abstraction des données (DAP)

Permet le stockage et la récupération des données d'un gestionnaire de files d'attente. Le DAP détient les messages.

Agents MCA

Ce sont des applications particulières qui utilisent MQI pour la plupart de leurs opérations. Elle assurent une transmission fiable des messages entre les gestionnaires de files d'attente. Les MCA sont liés avec FASTPATH.

Serveur de commandes

Le serveur de commandes est une application MQI spéciale chargée de traiter les messages contenant des commandes de gestion des gestionnaires de files d'attente.

Services communs

Ils isolent le reste du gestionnaire de files d'attente du système d'exploitation. Il s'agit d'un ensemble de services identiques à ceux fournis par un système d'exploitation : ils permettent par exemple la gestion du stockage, la sérialisation et la gestion des processus.

La figure 2 à la page 26 indique les relations entre composants.

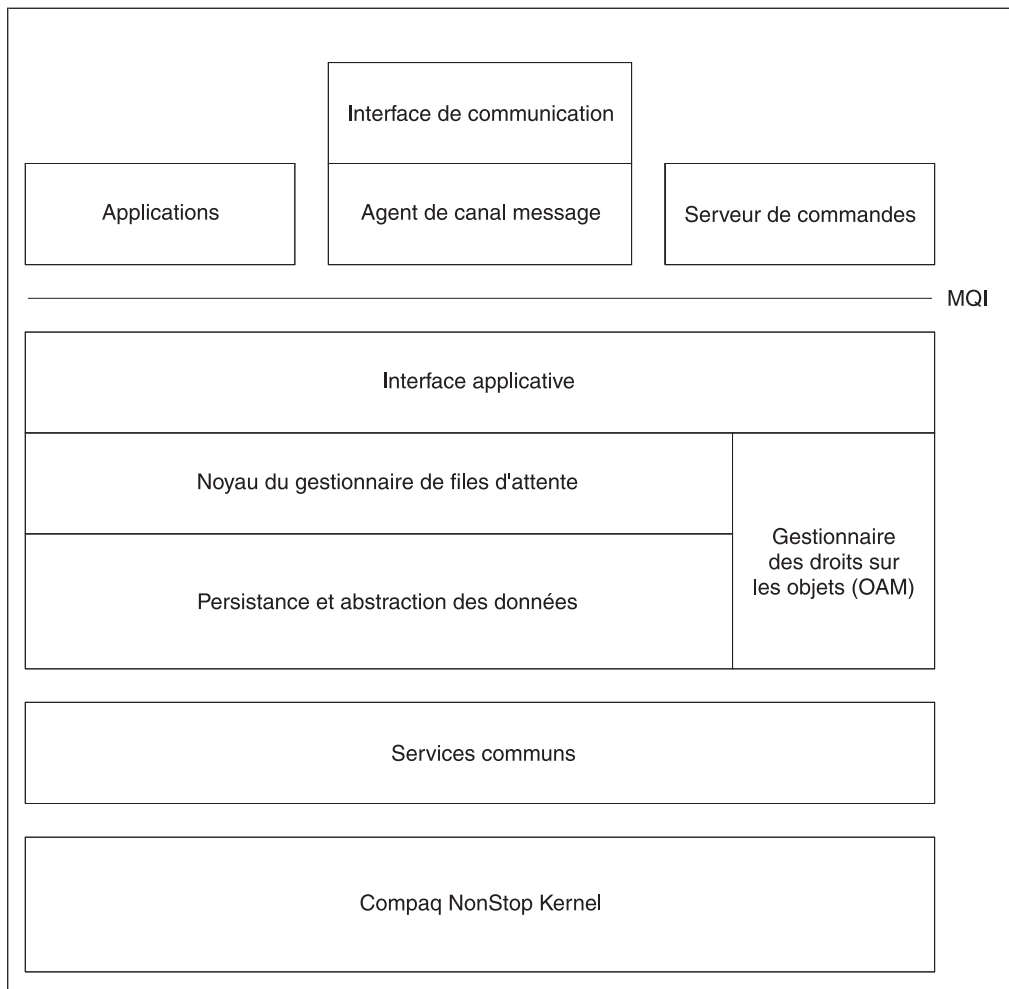


Figure 2. Composants de MQSeries pour Compaq NSK V5.1

Modèle de processus de gestionnaire de files d'attente

L'application communique avec le contrôleur d'exécution principal (ECBoss) lorsqu'elle doit se connecter à un agent. L'ECBoss choisit le contrôleur d'exécution (EC) le moins utilisé (ayant le moins d'applications connectées) et transmet la demande de connexion à l'EC. L'EC choisit un Local Queue Manager Agent (LQMA) inactif. L'EC renvoie une réponse à l'application via l'ECBoss, qui se connecte alors au LQMA choisi.

L'interface d'application se divise en deux :

- La portion de l'application MQI groupe les requêtes MQ et les transmet au processus agent à l'aide du composant de communication inter-processus (IPCC).
- L'IPCC fournit une interface de transmission de messages entre les applications MQI, les agents, les EC et l'ECBoss. Le composant IPCC MQSeries repose sur des IPC GUARDIAN standard.

L'application communique avec ses processus agents par l'intermédiaire de l'IPCC. Le processus agent exécute les appels MQI à la place des applications. Les échanges IPCC entre l'application et l'agent sont des messages de

demande-réponse synchrones. En plus de communiquer avec l'agent, pendant les opérations MQPUT et MQGET, l'application transfère les données de message directement au serveur de file d'attente responsable de la file.

Pour les applications liées avec FASTPATH, le code MQSeries est lié directement à l'application utilisateur. Ceci permet d'améliorer les performances mais fait courir le risque qu'une application malveillante perturbe le fonctionnement du gestionnaire de files d'attente, et provoque la perte de données et d'autres problèmes.

Fichiers et sous-volumes MQSeries

MQSeries utilise différents fichiers ENSCRIBE Compaq NonStop. L'emplacement et les noms des fichiers sont indiqués ci-dessous.

Les fichiers associés à un gestionnaire de files d'attente MQSeries sont répartis sur plusieurs sous-volumes.

<qmgr> D

Sous-volume de données. Utilisé pour des fichiers contenant des informations applicables à tout le gestionnaire de files d'attente.

<qmgr> M

Sous-volume de données de message. Utilisé pour les fichiers associés aux données des messages.

<qmgr> L

Journaux d'erreurs

<qmgr> S

Fichiers de synchronisation de canaux.

<qmgr>

Fichiers FFST™

Catalogue d'objets

Le catalogue d'objet (OBJCAT) contient des informations relatives aux objets MQSeries. Il se situe dans le sous-volume de données du gestionnaire de files d'attente.

Fichiers de file d'attente

Chaque file d'attente locale dispose d'un fichier de file, d'un fichier de dépassement de file et d'un fichier de contact. Ces fichiers commencent respectivement par Q, O et T — le reste du nom est soit une partie du nom MQSeries de l'objet, soit un nom généré par le système. Pour identifier les fichiers associés à une file d'attente précise, exécutez l'utilitaire **dspmqls**. Lorsqu'une file d'attente est créée, le gestionnaire de files d'attente crée les fichiers associés dans le sous-volume de message du gestionnaire de files d'attente (<qmgr>M). **altmqfls** permet de déplacer ces fichiers sur un autre volume, pour des raisons de performances ou de besoin d'espace.

Le gestionnaire de files d'attente crée des fichiers de dépassement pour chaque message dont la taille est supérieure au seuil de dépassement défini par la file d'attente. Les fichiers de dépassement des messages se situent par défaut sur le sous-volume de serveur de file d'attente. Si vous souhaitez modifier le sous-volume dans lequel le serveur de file d'attente crée les fichiers de dépassement des messages, lancez l'utilitaire **altmqfls**. Un fichier de dépassement de message est un fichier non structuré et non contrôlé dédié à un message particulier (reportez-vous à la section «Serveurs de file d'attente et fichiers de file d'attente» à la page 222 pour

plus d'informations sur la configuration des fichiers de configuration). Pour assurer un transfert de données efficace, les données sont écrites dans le fichier de dépassement du message en mode de transfert étendu (SETMODE 141), qui permet de procéder à des transferts directement depuis la mémoire du processus applicatif en segments de 56 Ko.

Pour les messages persistants inférieurs à environ 3 Ko, les données de message sont stockées dans le fichier de files d'attente. Pour les messages persistants dont la taille se situe entre 3 Ko et le seuil de dépassement configuré pour la file d'attente, le serveur de file d'attente écrit des enregistrements supplémentaires dans le fichier de dépassement de files d'attente. Pour les messages persistants dont la taille est supérieure au seuil de dépassement, le serveur de file d'attente crée un fichier de dépassement pour stocker le message.

Si vous souhaitez partitionner des fichiers de file d'attente ou des fichiers de dépassement de file pour des raisons de taille ou de performances, consultez la section «Partitionnement de fichiers de file d'attente» à la page 224. Si vous partitionnez les fichiers, MQSeries répartit les données des messages de façon égale entre les partitions pour des performances optimales.

Le «Chapitre 15. Evolutivité et performances» à la page 219 fournit des informations complémentaires sur les options d'optimisation des performances des files d'attentes et fichiers de files.

Files d'attente d'alias et éloignées

Un fichier de contact est associé aux files d'attente d'alias et éloignées. Il commence par un T et se situe dans le sous-volume de données du gestionnaire de files d'attente.

Fichiers de listes de nom

Un fichier non structuré est associé à chaque liste de noms. Les fichiers commencent par un L et se situent dans le sous-volume de données du gestionnaire de files d'attente. Le reste du nom du fichier est soit une partie du nom de l'objet MQSeries, soit une valeur générée par le système. Pour déterminer le nom du fichier associé à une liste de noms particulière, exécutez l'utilitaire **dspmqls**.

Chapitre 3. Utilisation des jeux de commandes MQSeries

Ce chapitre décrit les commandes à utiliser pour effectuer des tâches d'administration système sur des objets MQSeries. Les tâches d'administration comprennent la création, le démarrage, la modification, la visualisation, l'arrêt et la suppression de gestionnaires de files d'attente, files d'attente, processus, noms de listes, clusters et canaux. Pour cela, vous devez sélectionner la commande appropriée.

MQSeries pour Compaq NSK V5.1 propose les jeux de commandes d'administration suivants pour réaliser les tâches d'administration :

- Commandes MQSC (commandes MQSeries)
- Commandes PCF (Programmable Command Format)
- Commandes de contrôle
- Commandes MQAI (MQSeries Administrator Interface)

De plus :

- Certaines commandes TS/MP (PATHWAY) sont utilisées dans le cadre de l'administration.
- La fonction MQM (Message Queue Management) prend en charge certaines tâches d'administration. Elle est décrite dans la section «Utilisation de la fonction MQM (Message Queue Management)» à la page 70.

Ce chapitre présente les commandes MQSC, PCF et de contrôle, et propose un résumé des fonctions prises en charge par chaque jeu de commandes dans l'«Annexe D. Comparaison des jeux de commandes» à la page 327. L'utilisation des commandes TS/MP est décrite dans la section «Administration TS/MP (PATHWAY)» à la page 32.

Administration à l'aide des commandes de contrôle

Les types de commandes de contrôle suivants sont disponibles :

- Commandes de création, de démarrage, d'arrêt et de suppression des gestionnaires de files d'attente
- Commandes de démarrage, d'arrêt et d'affichage des serveurs de commandes
- Commandes d'utilitaires associées par exemple à l'exécution de commandes MQSC, à la gestion de l'accès aux objets MQSeries, au démarrage et à l'arrêt d'une trace MQSeries et à l'exécution de moniteurs de déclenchement

Utilisation des commandes de contrôle

Les commandes de contrôle s'exécutent à partir de l'invite TACL Compaq. Les noms de commandes ne font pas la distinction entre majuscules et minuscules. (Notez toutefois que les noms de gestionnaires de files d'attente *respectent* les majuscules.) Par exemple :

```
runmqsc
```

Le «Chapitre 17. Commandes de contrôle MQSeries» à la page 251 décrit la syntaxe et l'objet de chaque commande.

Administration à l'aide des commandes MQSC

Les commandes MQSC permettent de gérer des objets gestionnaire de files d'attente, y compris le gestionnaire lui-même, les canaux, les files d'attente et les définitions de processus. Il existe par exemple des commandes MQSC pour définir, modifier, afficher et supprimer une file d'attente spécifique.

Lorsque vous affichez une file d'attente à l'aide de la commande DISPLAY QUEUE, vous affichez les *attributs* de la file. Par exemple, l'attribut MAXMSGL indique la longueur maximale d'un message pouvant être mis en file d'attente. La commande n'affiche pas les messages de la file d'attente. Les principaux aspects de ces commandes sont présentés dans l'«Annexe D. Comparaison des jeux de commandes» à la page 327. Pour des informations détaillées sur chaque commande MQSC, reportez-vous au manuel MQSeries - Guide de référence des commandes MQSC.

Exécution de commandes MQSC

Vous pouvez exécuter MQSC de façon interactive en appelant la commande de contrôle **runmqsc** depuis l'invite Compaq TACL, ou en exécutant un script au cours de l'exécution d'un gestionnaire de files d'attente local. Il existe trois modes d'exécution de la commande **runmqsc** en fonction des indicateurs définis :

- *Mode vérification* : les commandes MQSC sont vérifiées sur un gestionnaire de files d'attente local, mais ne s'exécutent pas.
- *Mode direct* : les commandes MQSC s'exécutent sur un gestionnaire de files d'attente local.
- *Mode indirect* : les commandes MQSC s'exécutent sur un gestionnaire de files d'attente éloigné.

Pour plus d'informations sur l'utilisation des commandes MQSC et des fichiers textes, consultez la section «Utilisation de la fonction MQSC de façon interactive» à la page 95. Vous trouverez davantage d'informations sur la commande **runmqsc** dans la section «runmqsc (exécution des commandes MQSeries)» à la page 296. Les commandes MQSC sont répertoriées dans l'«Annexe D. Comparaison des jeux de commandes» à la page 327.

Utilisation de Compaq NSK Fix Command

Si vous exécutez la commande **runmqsc** de façon interactive (depuis l'invite NSK TACL), vous pouvez aussi utiliser la fonction Compaq NSK Fix Command afin de rappeler et modifier les commandes MQSC. Par exemple :

- Si vous tapez **history** ou **h**, vous obtenez la liste des dix commandes les plus récentes.
- Si vous tapez **!n** où **n** est le numéro de la commande, la commande est réexécutée.
- Si vous tapez **h n** ou **history n** où **n** est un nombre, vous obtiendrez la liste des **n** commandes les plus récentes
- Si vous tapez **fc**, vous pouvez modifier la dernière commande saisie. Le fait de taper **fc n** où **n** est le numéro de la commande, permet de modifier cette même commande. En tapant **fc chaîne** où **chaîne** est le début ou la totalité d'une commande précédemment entrée, vous pouvez modifier la dernière occurrence de cette commande. La syntaxe est standard à NSK. Par exemple, tapez **d** pour supprimer un caractère, **i** pour en insérer un et **r** pour remplacer ce caractère.

Administration à l'aide des commandes PCF

Les commandes PCF permettent de programmer des tâches d'administration dans vos applications ou dans un programme d'administration. Les commandes programmables PCF regroupent des fonctions similaires à celles fournies par la fonction MQSC. Vous pouvez ainsi écrire un programme chargé d'émettre des commandes PCF vers n'importe quel gestionnaire de files d'attente sur le réseau à partir d'un seul noeud. Vous pouvez aussi centraliser et automatiser les tâches d'administration.

Chaque commande programmable PCF est une structure de données intégrée dans les données d'application d'un message MQSeries. Elle est envoyée au gestionnaire de files d'attente cible à l'aide de la fonction MQI MQPUT. Le serveur de commandes du gestionnaire qui reçoit le message l'interprète comme un message de commande et l'exécute. Pour obtenir les réponses, l'application émet un appel MQGET, et les données de réponse sont renvoyées en tant que structure de données dans la partie réservée aux données d'application du message MQSeries. L'application peut ensuite traiter la réponse et agir en conséquence.

Remarque : A la différence des commandes MQSC, les commandes programmables PCF et leurs réponses ne se présentent pas dans un format texte lisible par l'utilisateur.

Vous devez préciser les éléments suivants pour créer un message de commande PCF :

Descripteur de message

Il s'agit d'un descripteur de message MQSeries standard, dans lequel :

- Le type de message (*MsgType*) est MQMT_REQUEST.
- Le format du message (*Format*) est MQFMT_ADMIN.

Données d'application

Elles contiennent le message PCF, y compris l'en-tête PCF, dans lequel :

- Le type du message PCF (*Type*) est MQCFT_COMMAND.
- L'identificateur de commande indique la commande, par exemple *ChangeQueue* (MQCMD_CHANGE_Q).

Pour une description complète des structures de données PCF et de leur mise en oeuvre, reportez-vous au manuel *MQSeries Programmable System Management*.

Attributs dans les commandes MQSC et PCF

Les attributs d'objet indiqués dans les commandes MQSC sont écrits en majuscules (par exemple, RQMNAME), bien que la distinction majuscules/minuscules ne s'applique pas. Ces attributs sont limités à huit caractères, (par exemple, QDPHIEV). Les attributs d'objets dans PCF apparaissent en italique, et ne sont pas limités à huit caractères. L'équivalent PCF de RQMNAME est *RemoteQMgrName* et celui de QDPHIEV est *QDepthHighEvent*.

Commandes PCF d'échappement

Les commandes d'échappement PCF sont des commandes PCF qui contiennent des commandes MQSC à l'intérieur du texte du message. Vous pouvez utiliser les commandes PCF pour envoyer des ordres à un gestionnaire de files d'attente éloigné. Pour plus de détails sur l'utilisation des commandes d'échappement PCF, reportez-vous au manuel *MQSeries Programmable System Management*.

Commandes PCF

MQAI est une interface d'administration MQSeries à présent disponible sur la plateforme Compaq NSK. Il s'agit d'un outil simple permettant de gérer MQSeries. Comme il n'est pas nécessaire d'écrire vos propres messages PCF, vous évitez les problèmes liés aux structures de données complexes. Pour plus d'informations sur l'utilisation de MQAI, reportez-vous à la section «Utilisation de l'interface MQAI pour simplifier l'utilisation des commandes programmables PCF» à la page 120.

Administration TS/MP (PATHWAY)

La plupart des opérations réalisées sur le gestionnaire de files d'attente le sont à l'aide de commandes de contrôle MQSeries lancées depuis TACL. Toutefois, certaines de ces opérations nécessitent l'utilisation de PATHCOM pour agir directement sur les classes de serveur TS/MP. De plus, en raison de certaines modifications de configuration système, vous pouvez être amené à réaliser des opérations d'administration sur la configuration TS/MP elle-même.

Cette section décrit ces activités.

Détermination et contrôle des programmes d'écoute TCP/IP

Pour démarrer des programmes d'écoute TCP/IP, vous pouvez utiliser la commande de contrôle MQSeries **runmqtsr** (décrite dans la section «runmqtsr (exécution du programme d'écoute)» à la page 294), ou les commandes PATHCOM, THAW SERVER et START SERVER. Pour arrêter les programmes d'écoute TCP/IP, utilisez les commandes PATHCOM, FREEZE SERVER et STOP SERVER. Utilisez la commande PATHCOM STATUS SERVER pour afficher le nombre de programmes d'écoute TCP/IP exécutés, ainsi que leurs noms de processus.

Par défaut, chaque gestionnaire de files d'attente dispose d'un programme d'écoute dans la classe de serveur MQS-TCPLIS00. Utilisez la commande PATHCOM ADD SERVER pour créer des classes de serveur de programmes d'écoute TCP/IP afin d'utiliser plusieurs ports TCP/IP. Chaque programme d'écoute TCP/IP doit être configuré avec sa propre classe de serveur TS/MP pour une souplesse optimale. Si vous ajoutez des programmes d'écoute TCP/IP, vous devez aussi ajouter des définitions de ports TCP/IP au fichier d'initialisation de gestionnaires de files d'attente (QMINI), selon la procédure indiquée dans la section «Ports TCP/IP écoutés par le gestionnaire de files d'attente» à la page 55. Le premier programme d'écoute à démarrer utilise le premier port de programme d'écoute défini dans QMINI, le second utilise le second port de programme d'écoute, et ainsi de suite. La section «Configuration de QMINI pour prendre en charge plusieurs programmes d'écoute TCP/IP» à la page 393 propose un exemple des entrées QMINI nécessaires pour prendre en charge plusieurs programmes d'écoute.

Contrôle du serveur de commandes

Le serveur de commandes est créé en tant que classe de serveur TS/MP MQS-CMDSERV00. En remplacement des commandes de contrôle **strmqcsv**, **endmqcsv**, et **dspmqcsv**, vous pouvez utiliser les commandes PATHCOM, THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER, et STATUS SERVER.

Détermination et contrôle des initialisateurs de canaux

L'initialisateur de canal par défaut est créé en tant que classe de serveur TS/MP MQS-CHANINIT00. En remplacement de la commande de contrôle **runmqchi** (décrite dans la section «runmqchi (exécution de l'initialisateur de canal)» à la page 290), vous pouvez utiliser les commandes PATHCOM, THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER, et STATUS SERVER pour

contrôler et afficher le statut de l'initialisateur de canal. L'initialisateur de canal par défaut traite la file d'attente d'initialisation de canal par défaut, SYSTEM.CHANNEL.INITQ.

Modification de la file d'attente d'initialisation par défaut pour l'initialisateur de canal

Dans la Version 5.1, le gestionnaire de files d'attente démarre automatiquement l'initialisateur de canal par défaut. Celui-ci doit fonctionner pour que les opérations de cluster puissent avoir lieu.

Si vous souhaitez utiliser une file d'attente d'initialisation différente de celle qui est définie par défaut (SYSTEM.CHANNEL.INITQ), vous devez modifier la configuration PATHWAY.

Remarque : Vous ne devez pas modifier la file d'attente d'initialisation si le gestionnaire de files d'attente fait partie d'un cluster. Le fait de modifier la file d'attente d'initialisation par défaut de l'initialisateur de canal par défaut annule la prise en charge des clusters.

Vous pouvez modifier la file d'attente d'initialisation par défaut pendant que le gestionnaire de files d'attente est en cours d'exécution, mais la classe de serveurs de l'initialisateur de canal lui-même doit être arrêtée. Dans PATHCOM, lancez la commande suivante sur la configuration PATHWAY du gestionnaire de files d'attente :

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

où *<file init>* est le nom de la file d'attente d'initialisation de remplacement. Vous pouvez alors démarrer l'initialisateur de canal et quitter PATHCOM.

Détermination et contrôle des moniteurs de déclenchement

Un seul moniteur de déclenchement par défaut est créé en tant que classe de serveur TS/MP MQS-TRIGMON00. Vous pouvez utiliser les commandes PATHCOM, THAW SERVER, START SERVER, FREEZE SERVER, STOP SERVER, et STATUS SERVER pour administrer cette classe de serveur. Si vous avez besoin de moniteurs de déclenchement supplémentaires, vous pouvez les configurer en tant que classes de serveur supplémentaires, en utilisant MQS-TRIGMON00 comme modèle. Nous vous recommandons d'utiliser des objets de classes de serveur distincts pour bénéficier d'une souplesse optimale. Vous n'êtes pas obligé d'utiliser TS/MP pour contrôler des moniteurs de déclenchement. Par exemple, vous pouvez exécuter un moniteur de déclenchement depuis TACL à l'aide de la commande de contrôle **runmqtrm**.

Le moniteur de déclenchement par défaut traite la file d'attente par défaut, FILE.INIT.DEFAUT.SYSEME. Vous pouvez y remédier en ajoutant ou modifiant le message STARTUP de la classe de serveur contenant le moniteur de déclenchement. Vous devez procéder de cette façon si plus d'un moniteur de déclenchement est configuré pour le gestionnaire de files d'attente. Utilisez la commande PATHCOM ALTER SERVER pour ajouter ou modifier l'attribut STARTUP.

Répartition des processus sur les processeurs

L'affectation d'un processeur à chaque EC d'un gestionnaire de files d'attente est un aspect important de la répartition des tâches sur différents processeurs. Chaque EC crée et gère un ensemble de processus agents sur son propre processeur uniquement. Par conséquent, si les processus EC sont répartis sur les processeurs du système, les processus agents sont distribués de la même façon.

Par défaut, si plusieurs EC sont précisés, les processus EC (dont chacun est une classe de serveur distincte) sont répartis de façon aussi équitable que possible sur les processeurs disponibles du système. Il n'existe aucune limite interne au nombre de processus EC d'un gestionnaire de files d'attente : ce nombre dépend totalement de la charge que le gestionnaire de files d'attente doit gérer. Par défaut, le gestionnaire de files d'attente comprend un processus EC.

La classe de serveur EC par défaut est MQS-EC00. Précisez l'indicateur `-e` dans la commande `crtmqm` pour créer un gestionnaire de files d'attente contenant plusieurs EC. Le nombre de processus EC peut être modifié une fois le gestionnaire de files d'attente créé, en ajoutant ou supprimant des classes de serveur de processus EC, et en faisant la modification correspondante dans l'entrée `ExpectedNumECs` de la strophe `ECBoss` du fichier `QMINI`.

Chaque processus EC *doit* se trouver dans sa propre classe de serveur. Utilisez la classe de serveur MQS-EC00 comme modèle si vous avez besoin de créer des processus EC supplémentaires manuellement.

Chaque processeur hébergeant un EC doit aussi héberger un gestionnaire de référentiels. Lorsqu'un gestionnaire de files d'attente est créé, MQSeries crée une classe de serveur de gestionnaires de référentiel (avec des noms de type MQS-REPSVR00) pour chaque EC. Si vous ajoutez manuellement des classes de serveur EC à des processeurs qui n'hébergeaient pas d'EC jusqu'à présent, utilisez la classe de serveur MQS-REPSVR00 comme modèle pour créer une nouvelle classe de serveur de gestionnaires de référentiels dans le nouveau processeur.

MQSeries nécessite une classe de serveur de gestionnaires de files d'attente de référentiels dans chaque processeur remplissant l'une des conditions suivantes :

- Le processeur héberge un EC, ou
- Le processeur héberge la classe de serveur MQS-QMGRSVR00, ou
- Le processeur exécute des applications liées avec FASTPATH, ou
- Les utilisateurs du processeur, exécutent l'une des applications MQSeries suivantes : `runmqsc`, `runmqchi`, `runmqchl`, ou `mqrepdmp`.

Vous pouvez modifier l'affectation par défaut des processeurs aux processus EC ou à toute autre classe de serveur, à l'aide de la commande `PATHCOM ALTER SERVER` contenant l'attribut du processeur.

Le status server par défaut est automatiquement créé par `crtmqm` dans la classe de serveur MQS-STATUS00. Par défaut, les seules affectations de processeurs effectuées sont les processus primaires attribués au CPU 0 et les processus de sauvegarde au CPU 1, ou CPUS(0:1). L'affectation de processeurs du serveur MQS-STATUS00 peut être modifiée à l'aide de la commande `PATHCOM ALTER SERVER`. Vous pouvez préciser un processeur de sauvegarde spécifique pour le status server en indiquant deux numéros de processeurs séparés par un point virgule, par exemple CPUS(2:12). Dans ce cas, `PATHMON` crée le processus

primaire sur le CPU 2 et le processus de sauvegarde sur le CPU 12. Si aucun processeur de secours n'est indiqué, Compaq NSK décide de l'endroit où créer la sauvegarde.

Le serveur de file d'attente par défaut est également automatiquement créé par **strmqm** dans la classe de serveur MQS-QUEUE00. Par défaut, les seules affectations de processeurs effectuées sont les processus primaires attribués au CPU 0 est les processus de sauvegarde au CPU 1, ou CPUS(0:1). L'affectation de processeurs du serveur MQS-QUEUE00 peut être modifiée à l'aide de la commande **PATHCOM ALTER SERVER**. Vous pouvez préciser un processeur de sauvegarde spécifique pour le serveur de file d'attente en indiquant deux numéros de processeurs séparés par un point virgule, par exemple CPUS(2:12). Dans ce cas, **PATHMON** crée le processus primaire sur le CPU 2 et le processus de sauvegarde sur le CPU 12. Si aucune processeur de secours n'est indiqué, Compaq NSK décide de l'endroit où créer la sauvegarde.

Ajout de nouveaux processus de status server

Pour ajouter des status server à un gestionnaire de file d'attente, créez une classe de serveur en utilisant la classe status server MQS-STATUS00 par défaut comme modèle.

Le nom de la nouvelle classe de status server doit commencer par la chaîne de caractères MQS-STATUS. Si les noms de classes de serveur ne suivent pas cette règle de dénomination, **strmqm** ne les démarrera pas automatiquement au lancement du gestionnaire de files d'attente, et l'accès à tout objet configuré à l'aide de ces classes de status server sera désactivé.

Si des status server supplémentaires sont configurés, ils doivent tous disposer de noms de processus uniques. Nous vous conseillons également de les configurer de façon à ce qu'ils s'exécutent sur des processeurs différents, afin de pouvoir profiter de l'évolutivité de l'architecture de status server.

Ajout de nouveaux processus de serveurs de file d'attente

Pour ajouter des serveurs de file d'attente à un gestionnaire de files d'attente, créez une classe de serveur en utilisant la classe de serveur de files d'attente MQS-QUEUE00 par défaut comme modèle.

Le nom de toute nouvelle classe de serveur de file d'attente doit commencer par la chaîne de caractères MQS-QUEUE. Si les noms de classes de serveur ne suivent pas cette règle de dénomination, **strmqm** ne les démarrera pas automatiquement au lancement du gestionnaire de files d'attente, et l'accès à tout objet configuré à l'aide de ces classes de serveur de file d'attente sera désactivé.

Si des serveurs de file d'attente supplémentaires sont configurés, ils doivent tous disposer de noms de processus uniques. Nous vous conseillons également de les configurer de façon à ce qu'ils s'exécutent sur des processeurs différents, afin de pouvoir profiter de l'évolutivité de l'architecture de serveur de file d'attente.

Fréquence de rafraîchissement des écrans du moniteur MQM

Le paramètre **MQMQMREFRESHINT PATHWAY** de MQS-MQMSVR00 détermine la fréquence de rafraîchissement des écrans de moniteur des canaux et files d'attente. La fréquence par défaut est de 30 secondes. Pour que le rafraîchissement

Administration TS/MP

s'effectue toutes les 10 secondes, par exemple, entrez la commande suivante dans PATHWAY pour votre gestionnaire de files d'attente :

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

Configuration PATHWAY pour un gestionnaire de file d'attente

Un exemple de configuration PATHWAY pour un gestionnaire de files d'attente est présenté ci-dessous. Il a été généré à l'aide d'une séquence de commandes INFO sur les objets d'une configuration de gestionnaire de files d'attente par défaut.

```
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
```

Figure 3. Exemple de configuration PATHWAY (Numéro 1 de 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)
```

Figure 3. Exemple de configuration PATHWAY (Numéro 2 de 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

```

Figure 3. Exemple de configuration PATHWAY (Numéro 3 de 13)

```

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

```

Figure 3. Exemple de configuration PATHWAY (Numéro 4 de 13)

Administration TS/MP

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

Figure 3. Exemple de configuration PATHWAY (Numéro 5 de 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
```

Figure 3. Exemple de configuration PATHWAY (Numéro 6 de 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

```

Figure 3. Exemple de configuration PATHWAY (Numéro 7 de 13)

```

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

```

Figure 3. Exemple de configuration PATHWAY (Numéro 8 de 13)

Administration TS/MP

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

Figure 3. Exemple de configuration PATHWAY (Numéro 9 de 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
```

Figure 3. Exemple de configuration PATHWAY (Numéro 10 de 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

```

Figure 3. Exemple de configuration PATHWAY (Numéro 11 de 13)

```

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

```

Figure 3. Exemple de configuration PATHWAY (Numéro 12 de 13)

Administration TS/MP

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

Figure 3. Exemple de configuration PATHWAY (Numéro 13 de 13)

Modification des paramètres de classe de serveurs PATHWAY

Pour modifier les paramètres des classes de serveur PATHWAY :

1. Arrêtez le gestionnaire de files d'attente en émettant une commande **endmqm**. Elle a aussi pour effet d'arrêter le processus PATHMON.
2. Allez dans le sous-volume *gestionnaire de files d'attenteD*, contenant le fichier PATHCTL. Par exemple :

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

3. Démarrez un PATHMON avec le même nom que le PATHMON du gestionnaire de files d'attente avec l'option NOWAIT. Par exemple :

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

4. Lancez un PATHCOM sur le nouveau PATHMON. Par exemple :

```
>PATHCOM $MT01
```

5. Chargez la configuration PATHWAY existante du gestionnaire de files d'attente, en émettant la commande suivante à partir de l'invite PATHCOM :

```
>START PATHWAY COOL
```

6. Effectuez les modifications nécessaires à l'aide des commandes PATHCOM.

7. Arrêtez le système PATHWAY en émettant la commande suivante :

```
>SHUTDOWN2
```

8. Lancez le gestionnaire de files d'attente à l'aide de la commande **strmqm**.

Ajout de classes de serveurs personnalisées à un PATHWAY MQSeries

Vous pouvez ajouter vos propres définitions de classes de serveur à la configuration PATHWAY MQSeries à l'aide de PATHCOM. Ce n'est toutefois pas recommandé : les serveurs *doivent* fonctionner parfaitement pour que **endmqm** puisse s'exécuter correctement. Notez également que les définitions de classes de serveur définies par l'utilisateur sont perdues en cas de suppression d'un gestionnaire de files d'attente. Pour des raisons pratiques, nous vous recommandons de créer un script réutilisable.

Chapitre 4. Gestion des gestionnaires de files d'attente

Ce chapitre décrit tous les aspects de la gestion des gestionnaires de files d'attente MQSeries.

Il contient les sections suivantes :

- «Mise en route»
- «Instructions pour la création d'un gestionnaire de files d'attente»
- «Modification des propriétés d'un gestionnaire de files d'attente» à la page 48
- «Nom de serveur de file d'attente par défaut» à la page 59
- «Ajout et suppression de serveurs de file d'attente personnalisés» à la page 60
- «Structure du volume» à la page 62
- «Utilisation des gestionnaires de files d'attente» à la page 65
- «Création d'un gestionnaire de files d'attente par défaut» à la page 65
- «Création de principaux MQSeries» à la page 66
- «Exécution de cleanrdf pour un gestionnaire de files d'attente prêt pour RDF» à la page 67
- «Lancement d'un gestionnaire de files d'attente» à la page 67
- «Restauration des objets système et par défaut» à la page 67
- «Consultation des fichiers objets» à la page 67
- «Arrêt d'un gestionnaire de files d'attente» à la page 68
- «Redémarrage d'un gestionnaire de files d'attente» à la page 69
- «Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant» à la page 69
- «Suppression d'un gestionnaire de files d'attente» à la page 70
- «Utilisation de la fonction MQM (Message Queue Management)» à la page 70

Mise en route

Pour pouvoir utiliser des messages et des files d'attente, vous devez au préalable créer au moins un gestionnaire de files d'attente. Une fois l'installation terminée, vous pouvez utiliser les commandes de contrôle MQSeries pour créer un gestionnaire de files d'attente, des principaux MQSeries et démarrer le gestionnaire de files d'attente.

Lorsque vous créez un gestionnaire de files d'attente, les objets par défaut et objets système sont automatiquement créés. Les objets par défaut constituent la base de toute définition d'objet que vous effectuez ; les objets système sont indispensables au bon fonctionnement du gestionnaire de files d'attente. Pour plus de détails sur la restauration des objets système par défaut, consultez la section «Restauration des objets système et par défaut» à la page 67.

Reportez-vous à la section «Chapitre 3. Utilisation des jeux de commandes MQSeries» à la page 29 pour plus d'informations sur les commandes à utiliser avec MQSeries pour Compaq NSK, et les différentes commandes qui permettent de les appeler.

Instructions pour la création d'un gestionnaire de files d'attente

Un gestionnaire de files d'attente gère les ressources qui lui sont associées, telles que les files d'attente qu'il possède. Il offre aux applications des services de mise en file d'attente pour les appels MQI (Message Queuing Interface) et des commandes pour créer, modifier, afficher et supprimer les objets MQSeries. Pour

Création des gestionnaires de files d'attente

créer un gestionnaire de files d'attente, vous devez utiliser la commande **crtmqm**. Tenez-compte des éléments suivants pour créer un gestionnaire de files d'attente :

- **Définition d'un nom de gestionnaire de files d'attente unique**

Lorsque vous créez un gestionnaire de files d'attente, vous devez vous assurer que son nom est unique sur l'ensemble du réseau. Les noms de gestionnaires de files d'attente ne sont pas vérifiés lors de la création, or si vous n'utilisez pas un nom unique, vous ne pourrez pas utiliser de canaux pour la mise en file d'attente répartie.

Pour être sûr qu'un nom de gestionnaire de files d'attente sera unique, vous pouvez le faire précéder du nom de son noeud. Par exemple, si le noeud s'appelle `comptes`, vous pouvez nommer votre gestionnaire `comptes.saturne.gest.files`, où `saturne` identifie un gestionnaire précis et `gest.files` un suffixe que vous pouvez donner à tous vos gestionnaires de files d'attente. Vous pouvez ne pas utiliser ce suffixe. Toutefois, `comptes.saturne` et `comptes.saturne.gest.files` sont des noms de gestionnaires *différents*.

Remarque : Les noms de gestionnaires de files d'attente des commandes de contrôle font la distinction entre majuscules et minuscules. Par exemple, vous pouvez créer deux gestionnaires de files d'attente avec les noms `jupiter.gest.files` et `JUPITER.gest.files`.

- **Limitation du nombre de gestionnaires de files d'attente.**

Sous MQSeries pour Compaq NSK, le nombre de gestionnaires de files d'attente que vous pouvez créer n'est limité que par les contraintes de ressources. Cependant, étant donné que chaque gestionnaire doit disposer de ses propres ressources, il est souvent préférable de créer un gestionnaire avec 100 files d'attente sur un noeud, plutôt que dix gestionnaires avec dix files d'attente chacun. Plusieurs noeuds peuvent être exécutés avec un seul gestionnaire de files d'attente. Toutefois, des serveurs plus importants peuvent fonctionner avec des gestionnaires multiples. Certaines fonctionnalités demandant des performances ou fonctionnalités particulières peuvent nécessiter plusieurs gestionnaires de files d'attente.

- **Définition d'un gestionnaire de files d'attente par défaut.**

Chaque noeud doit avoir un gestionnaire de files d'attente par défaut, bien qu'il soit possible de configurer MQSeries sur un noeud sans gestionnaire par défaut.

Le gestionnaire de files d'attente par défaut est le gestionnaire auquel sont connectées les applications, si elles n'indiquent pas un nom de gestionnaire dans un appel MQCONN. C'est également à lui qu'il revient de traiter les commandes MQSC lorsque vous appelez la commande **runmqsc** sans indiquer un nom de gestionnaire de files d'attente.

La définition d'un gestionnaire de files d'attente comme gestionnaire par défaut *remplace* toute définition antérieure de gestionnaire par défaut pour le noeud.

N'oubliez pas que le fait de changer le gestionnaire de files d'attente par défaut peut avoir des répercussions sur d'autres utilisateurs ou applications. Il n'y a pas de répercussions sur des applications qui sont connectées au moment du changement, car elles peuvent utiliser l'identificateur de leur appel de connexion d'origine pour des appels MQI ultérieurs. Cet identificateur garantit l'acheminement des appels vers le même gestionnaire de files d'attente. Les applications qui se connectent après le changement sont connectées au nouveau gestionnaire de files d'attente par défaut.

Pour créer un gestionnaire de files d'attente par défaut, ajoutez l'indicateur `-q` dans la commande **crtmqm**. Pour une description détaillée de la commande et de ses paramètres, reportez-vous à la section «**crtmqm** (création d'un gestionnaire de files d'attente)» à la page 267.

- **Définition d'une file d'attente de rebut.**

La file d'attente de rebut est une file d'attente locale dans laquelle sont insérés les messages qui ne peuvent pas être acheminés vers leur destination.

Attention: Il est très important de disposer d'une file d'attente de rebut sur chaque gestionnaire de files d'attente du réseau. Si ce n'est pas le cas, des erreurs d'application peuvent se produire entraînant la fermeture de canaux, et l'échec des réponses aux commandes d'administration. Par exemple, si une application tente d'insérer un message dans une file d'attente sur un autre gestionnaire en indiquant un nom de file d'attente erroné, le canal est arrêté et le message reste dans la file d'attente de transmission. Les autres applications ne peuvent plus utiliser ce canal pour leurs messages.

Si les gestionnaires possèdent une file d'attente de rebut, le canal n'est pas affecté par l'incident. Le message dont la transmission a échoué est inséré dans la file d'attente de rebut, à l'extrémité de réception, et le canal ainsi que sa file d'attente de transmission restent disponibles.

Par conséquent, lorsque vous créez un gestionnaire de files d'attente, vous devez utiliser l'indicateur -u pour spécifier le nom de la file d'attente de rebut. Vous pouvez également utiliser une commande MQSC pour modifier les attributs d'un gestionnaire de files d'attente et indiquer la file d'attente de rebut à utiliser. Reportez-vous à la section «Modification des attributs d'un gestionnaire de files d'attente» à la page 98 pour plus d'informations sur la commande MQSC ALTER.

Lorsque vous trouvez des messages en file d'attente de rebut, vous pouvez utiliser le gestionnaire de files d'attente de rebut fourni avec MQSeries pour les traiter. Reportez-vous au «Chapitre 9. Gestionnaire de file d'attente de rebut MQSeries» à la page 161, pour plus d'informations sur les gestionnaires de files d'attente de rebut, et sur la façon de réduire le nombre de messages qui seront destinés à la file d'attente de rebut.

- **Définition d'une file d'attente de transmission par défaut.**

Une file d'attente de transmission est une file d'attente locale dans laquelle les messages envoyés vers un gestionnaire de files d'attente éloigné sont insérés en attendant d'être transmis. La file d'attente de transmission par défaut est la file d'attente utilisée lorsqu'aucune file d'attente de transmission n'a été définie explicitement. On peut définir une file d'attente de transmission par défaut pour chaque gestionnaire de files d'attente.

Lorsque vous créez un gestionnaire de files d'attente, vous devez utiliser l'indicateur -d pour indiquer le nom de la file d'attente de transmission par défaut. L'indicateur -d ne crée pas réellement la file d'attente : vous devrez la créer ultérieurement. Pour plus d'informations, consultez la section «Manipulation des files d'attente locales» à la page 103.

- **Sauvegarde des fichiers de configuration après création d'un gestionnaire de files d'attente.**

Un fichier de configuration MQSeries (MQSINI) est créé lorsque vous installez MQSeries. Il contient une liste des gestionnaires de files d'attente qui est mise à jour chaque fois que vous créez ou supprimez un gestionnaire de files d'attente. Il existe un fichier MQSINI par installation. Par défaut, MQSINI se trouve dans \$SYSTEM.ZMQSSYS.

Un fichier de configuration de gestionnaire de files d'attente (QMINI) est automatiquement créé lorsque vous créez un nouveau gestionnaire de files d'attente. Il contient les paramètres de configuration définis pour le gestionnaire.

Une fois le gestionnaire de files d'attente créé, il est conseillé de sauvegarder vos fichiers de configuration. Si vous créez un autre gestionnaire de files d'attente provoquant des incidents, vous pouvez rétablir les copies de secours après avoir

Création des gestionnaires de files d'attente

supprimé la cause de l'incident. Faites une copie de secours des fichiers de configuration chaque fois que vous créez un nouveau gestionnaire de files d'attente.

Pour plus d'informations sur les fichiers de configuration, reportez-vous au «Chapitre 13. Fichiers de configuration» à la page 189.

Sauvegarde des fichiers de configuration après la création d'un gestionnaire de files d'attente

Vous devez sauvegarder deux fichiers de configuration, MQSINI et QMINI :

1. Le fichier de configuration MQSeries (MQSINI) est créé lorsque vous installez MQSeries. Il contient une liste des gestionnaires de files d'attente qui est mise à jour chaque fois que vous créez ou supprimez un gestionnaire de files d'attente. Il existe un fichier MQSINI par installation. Par défaut, MQSINI se trouve dans `$SYSTEM.ZMQSSYS`.
2. Un fichier de configuration de gestionnaire de files d'attente (QMINI) est automatiquement créé lorsque vous créez un nouveau gestionnaire de files d'attente. Il contient les paramètres de configuration définis pour le gestionnaire.

Une fois le gestionnaire de files d'attente créé, il est conseillé de sauvegarder vos fichiers de configuration. Si vous créez un autre gestionnaire de files d'attente provoquant des incidents, vous pouvez rétablir les copies de secours après avoir supprimé la cause de l'incident. Faites une copie de secours des fichiers de configuration chaque fois que vous créez un nouveau gestionnaire de files d'attente.

Modification des propriétés d'un gestionnaire de files d'attente

Vous pouvez modifier de nombreuses propriétés du gestionnaire de files d'attente au moment de sa création. Certaines propriétés pourront aussi être modifiées par la suite, mais vous devrez probablement arrêter et redémarrer le gestionnaire de files d'attente avant que les modifications ne prennent effet.

Le reste de cette section décrit certaines propriétés de gestionnaires de files d'attente que vous pourrez être amené à modifier.

Volume origine du gestionnaire de files d'attente

Il s'agit du volume dans lequel toutes les bases de données, dont les files d'attente, sont créées. (Toutefois, les files d'attente individuelles peuvent être déplacées dans un autre volume après création à l'aide de la commande de contrôle `altmqfls`, de la façon décrite dans la section «altmqfls (modification des attributs d'un fichier de file d'attente)» à la page 254.)

La valeur par défaut est extraite de l'entrée `QMDefaultVolume` de la strophe `AllQueueManagers`, du fichier MQSINI. Il est possible de la remplacer par le paramètre `-p DefaultPrefix` de la commande `crtmqm`.

Le volume origine peut être indiqué *uniquement* lors de la création d'un gestionnaire de files d'attente. Ce n'est plus possible après la création.

Nombre de processus EC par gestionnaire de files d'attente

Par défaut, le gestionnaire de files d'attente comprend un processus EC. Précisez le nombre de processus EC d'un gestionnaire de files d'attente dans le paramètre `-e NumECs` de la commande `crtmqm`.

Chaque EC est responsable d'un sous-ensemble de processus serveur en charge de la messagerie et de la mise en file d'attente des applications et canaux du même processeur que l'EC. Il est conseillé d'avoir un EC par processeur, à moins que le nombre d'applications par EC soit important, auquel cas l'exécution d'un EC supplémentaire sur le processeur est avantageux. Par exemple, pour les installations importantes, un nombre supérieur de processus EC est souhaitable (souvent répartis sur plusieurs processeurs) afin qu'un grand nombre d'applications et canaux puissent être manipulés simultanément.

Lors de la création d'un gestionnaire de files d'attente, une classe de serveurs TS/MP est créée pour chaque EC indiqué dans la commande `crtmqm`. Les classes de serveurs EC sont réparties par permutation circulaire sur tous les processeurs du système. Par exemple, huit processus EC définis dans un système à quatre processeurs seraient répartis par défaut en deux EC par processeur.

La zone `ExpectedNumECs` de la strophe `ECBoss` du fichier `QMINI` du gestionnaire de files d'attente indique le nombre de processus EC définis à la création. Cette valeur doit à tout moment être cohérente avec la configuration TS/MP.

Le nombre de processus EC peut être modifié une fois le gestionnaire de files d'attente créé, en ajoutant ou supprimant des classes de serveurs de processus TS/MP, et en faisant la modification correspondante dans l'entrée `ExpectedNumECs` de la strophe `ECBoss` du fichier `QMINI`. Cette opération est possible uniquement lorsque le gestionnaire de files d'attente est arrêté.

Équilibrage de la charge de travail d'un système

Le rôle du contrôleur d'exécution principal est de répartir la charge de travail du gestionnaire de files d'attente sur les EC. La charge de traitement d'un gestionnaire de files d'attente peut être répartie de façon équilibrée sur plusieurs processeurs, si les EC sont correctement configurés.

Lorsqu'une nouvelle demande de connexion est émise par une application locale, ou lorsqu'un canal doit être démarré, le contrôleur d'exécution principal affecte la demande à l'EC doté du plus petit nombre d'agents `LQMA` et `MCA` actifs.

Terminal d'origine du gestionnaire de files d'attente

Tous les processus Compaq NSK, dont les processus de serveur de gestionnaire de files d'attente disposent d'un terminal d'origine. Le terminal *doit* exister et être en pause. En général, le terminal d'origine du gestionnaire de files d'attente n'est pas utilisé pour les sorties. Il peut s'agir de tout périphérique valide du terminal, dont le produit Compaq Virtual Hometerm Service (VHS).

Compaq NonStop Kernel permet jusqu'à 255 instructions d'ouverture primaires sur un terminal physique. Par conséquent, une planification soignée est nécessaire pour ne pas dépasser cette limite. Nous vous recommandons d'utiliser le produit VHS de Compaq si plus de 255 instructions d'ouverture sont nécessaires.

Vous pouvez identifier le terminal d'origine d'un gestionnaire de files d'attente d'après le paramètre `-o HomeTerminalName` de la commande `crtmqm`. Il n'y a pas de valeur par défaut ; ce paramètre est obligatoire.

Création des gestionnaires de files d'attente

Les attributs HOMETERM et OUT de toutes les classes de serveur TS/MP n'indiquent pas le périphérique de terminal précisé. Ces attributs peuvent être modifiés à tout moment lorsque les classes de serveur sont arrêtées, normalement lorsque le gestionnaire de files d'attente est arrêté.

L'entrée HomeTerminalName de la strophe Configuration du fichier de configuration QMINI doit aussi être modifiée pour pouvoir changer le terminal d'origine d'un gestionnaire de files d'attente.

Nom de processus PATHMON du gestionnaire de files d'attente

Chaque gestionnaire de files d'attente s'exécute dans sa propre configuration TS/MP (Pathway). PATHMON est le processus de contrôle utilisé. Un nom unique doit être indiqué pour chaque gestionnaire de files d'attente. De plus, le nom doit être unique dans le système.

Vous pouvez indiquer le nom de processus PATHMON dans le paramètre `-n PATHMONProcessName` de la commande `crtmqm`. Il n'y a pas de valeur par défaut ; ce paramètre est obligatoire.

Indiquez un nom de processus spécifique au système et facile à associer au gestionnaire de files d'attente qu'il contrôle.

Vous pouvez modifier le nom de processus PATHMON d'un gestionnaire de files d'attente de la façon suivante :

1. Arrêtez le gestionnaire de files d'attente.
2. Définissez le volume et le sous-volume par défaut sur l'emplacement des fichiers de données du gestionnaire de files d'attente (normalement `<Nom GestFiles>D`).
3. Modifiez l'entrée `PathmonProcName` du fichier QMINI du gestionnaire de files d'attente en fonction du nouveau nom de processus.
4. Exécutez PATHMON pour le gestionnaire de files d'attente en utilisant le *nouveau* nom de processus.

A partir de TAQL, exécutez la commande suivante :

```
PATHMON /nom $<nouveaunom>, nowait/
```

5. Exécutez PATHCOM sur le nouveau PATHMON.

A partir de TAQL, exécutez la commande suivante :

```
PATHCOM $<nouveaunom>
```

6. Chargez la configuration Pathway du gestionnaire de files d'attente et confirmez la modification du nom du processus PATHMON.

A partir de PATHCOM, exécutez la commande suivante :

```
START PATHWAY COOL
```

Création des gestionnaires de files d'attente

Lors du chargement de la configuration, un message d'avertissement indique que le nouveau processus PATHMON est différent de celui qui figure dans le fichier de configuration. Vous êtes ensuite invité à confirmer que vous souhaitez continuer. A l'invite, tapez *y* et la configuration continue de se charger.

7. Réenregistrez les nouvelles informations de configuration PATHWAY dans la base de données.

A partir de PATHCOM, exécutez la commande suivante :

```
SHUTDOWN2
EXIT
```

L'entrée PathmonProcName de la strophe Configuration du fichier QMINI doit aussi être modifiée.

La procédure de modification de nom PATHMON est à présent terminée. Le prochain **strmqm** démarrera le gestionnaire de files d'attente à l'aide du nouveau nom de processus PATHMON.

CCSID du gestionnaire de files d'attente

Il s'agit de l'ID de jeu de caractères codés (CCSID) du jeu de caractères utilisé par le gestionnaire de files d'attente pour stocker les informations sur les messages.

Vous pouvez indiquer le CCSID grâce au paramètre **-l CCSID** de la commande **crtmqm**. La valeur par défaut est 819.

Vous pouvez modifier le CCSID du gestionnaire de files d'attente à tout moment après la création du gestionnaire de files d'attente, à l'aide de **runmqsc**, de la fonction MQM ou des commandes PCF.

Contrôle de la conversion de données EBCDIC

Le mode de gestion des caractères NL (nouvelle ligne) EBCDIC pendant la conversion peut être contrôlé à l'aide de l'élément ConvEBCDICNewline de la strophe AllQueueManagers du fichier de configuration MQSINI.

ConvEBCDICNewline=NL_TO_LF|TABLE|ISO

Les pages de codes EBCDIC contiennent un caractère NL (new line) qui n'est pas pris en charge par les pages de codes ASCII, bien que certaines variantes ISO du code ASCII proposent un équivalent. A l'aide de l'attribut ConvEBCDICNewline indiquez la méthode de conversion utilisée par MQSeries pour convertir le caractère NL d'EBCDIC en ASCII.

NL_TO_LF

Affectez la valeur NL_TO_LF si vous souhaitez convertir le caractère EBCDIC NL (X'15') en caractère d'avancement ligne ASCII, LF (X'0A'), pour toutes les conversions EBCDIC vers ASCII. NL_TO_LF est la valeur par défaut.

TABLE

L'affectation de la valeur NL_TO_LF à l'attribut ConvEBCDICNewline convertira le caractère EBCDIC NL selon les tables utilisées sur votre plateforme, pour toutes les conversions d'EBCDIC en ASCII. Notez que le résultat de ce type de conversion varie d'une plateforme à l'autre et d'une langue à

Création des gestionnaires de files d'attente

l'autre, tandis que sur une même plateforme, le résultat peut différer si vous utilisez différents CCSID.

- ISO** Affectez la valeur ISO si vous voulez que :
- les CCSID ISO soient convertis à l'aide de la méthode TABLE,
 - tous les autres CCSID soient convertis à l'aide de la méthode NL_TO_CF.

La liste des CCSID ISO possibles se trouve dans le tableau 2.

Tableau 2. Liste des CCSID ISO possibles

CCSID	Jeu de codes
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

Si le CCSID ASCII n'est pas un sous-ensemble ISO, ConvEBCDICNewLine utilise par défaut NL_TO_LF.

Vous trouverez davantage de détails concernant la conversion de données, dans le *MQSeries - Guide de programmation d'applications*.

Collecteur EMS du gestionnaire de files d'attente

Vous pouvez configurer le gestionnaire de files d'attente de façon à ce qu'il utilise un autre collecteur si nécessaire. Les événements EMS sont envoyés à \$0 par défaut. L'entrée `EMSCollectorName` de la strophe `Configuration` du fichier de configuration `QMINI` indique le nom du collecteur EMS du gestionnaire de files d'attente.

Le collecteur EMS peut être modifié à tout moment en changeant la valeur de cette entrée, bien qu'elle ne prenne pas effet avant que le gestionnaire de files d'attente ne soit redémarré.

Groupe d'agents à la disposition des EC du gestionnaire de files d'attente

Pour chacun des quatre types d'agents de base, un EC peut maintenir un groupe de processus agents en veille, prêt à être affecté à une nouvelle tâche. La taille de ces groupes peut être configurée de façon à obtenir l'équilibre approprié entre temps de réponse d'une nouvelle tâche et utilisation des ressources. Les valeurs des zones suivantes de la strophe `Configuration` du fichier `QMINI` peuvent être modifiées de façon à indiquer un nombre de processus différents à maintenir en veille :

MinIdleMCALU62Responders

Indique le nombre minimum d'agents MCA répondeurs SNA LU 6.2 à maintenir en état de veille. La valeur par défaut est 0.

Création des gestionnaires de files d'attente

MinIdleMCATCPResponders

Indique le nombre minimum d'agents MCA répondeurs TCP/IP à maintenir en état de veille. La valeur par défaut est 0.

MinIdleMCACallers

Indique le nombre minimum d'agents MCA répondeurs (sans protocole spécifique) à maintenir en état de veille. La valeur par défaut est 0.

MinIdleLQMAgents

Indique le nombre minimum d'agents de gestionnaires de files d'attente locaux (LQMA) à maintenir en état de veille. La valeur par défaut est 1.

Notez que le nombre de processus indiqué dans ces zone s'applique à chaque EC, non à chaque gestionnaire de files d'attente. Par conséquent, pour un gestionnaire de files d'attente à deux EC, deux LQMA en veille sont nécessaires au minimum par défaut.

Ces valeurs peuvent être modifiées à tout moment, bien qu'elles ne prennent effet qu'une fois le gestionnaire de files d'attente redémarré.

Nombre maximum d'agents en veille et réutilisation des processus

Par défaut, un gestionnaire de files d'attente autorise la mise en veille de jusqu'à 10 processus agents de chaque type. Cette valeur est contrôlée par l'entrée `MaxIdleAgents` de la strophe `Configuration` du fichier de configuration `QMINI`.

L'entrée `MaxIdleAgentReuse` détermine le nombre de fois qu'un processus agent peut être réutilisé avant d'être remplacé par un nouveau processus agent. La valeur par défaut de `MaxIdleAgentReuse` est 10.

Priorité de traitement des processus de gestionnaire de files d'attente

Vous pouvez modifier les priorités de façon à équilibrer les ressources entre `MQSeries` et les autres applications. Les priorités de traitement des classes de serveurs TS/MP peuvent être modifiées en exécutant une commande `ALTER` sur les objets TS/MP lorsque le gestionnaire de files d'attente est arrêté.

Pour cela, modifiez les zones `MCAAgentPriority` et `LQMAgentPriority` de la strophe `EC` du fichier `QMINI`.

Hormis les status server et les serveurs de file d'attente qui bénéficient d'une priorité de traitement par défaut de 176, les processus configurés par TS/MP disposent tous d'une priorité par défaut de 175. Par défaut, les agents MCA et LQMA ont une priorité de traitement de 165.

Veillez à ce que les status server et serveurs de file d'attente bénéficient du plus haut niveau de priorité, suivis du contrôleur d'exécution principal et de l'EC, qui doivent quant à eux avoir une priorité supérieure à celle des agents MCA et LQMA.

Nombre maximum de canaux d'un gestionnaire de files d'attente

Le nombre de canaux pouvant être contrôlés à tout moment par un gestionnaire de files d'attente est limité. Si cette limite est trop élevée, les performances peuvent

Création des gestionnaires de files d'attente

s'en ressentir, puisque ce paramètre définit la taille de la table de statut des canaux, sur laquelle de nombreuses recherches sont effectuées. Si la limite est trop basse, vous pouvez ne pas être en mesure de contrôler assez de canaux pour votre application. La zone MaxChannels de la strophe Channels du fichier QMINI définit le nombre maximum de canaux pouvant être contrôlés simultanément.

La valeur par défaut à la création est 10. Il n'existe aucun moyen de remplacer la valeur par défaut à ce moment là.

L'entrée MaxChannels de la strophe Channels du fichier QMINI peut être modifiée à tout moment, bien qu'elle ne prenne effet qu'une fois le gestionnaire de files d'attente redémarré.

Nombre maximum de canaux actifs d'un gestionnaire de files d'attente

Le nombre de canaux actifs en fonction simultanément sur un gestionnaire de files d'attente est limité. Cette limite permet de contrôler les pics de demandes de ressources système par les canaux. L'entrée MaxActiveChannels de la strophe Channels du fichier QMINI définit le nombre maximum de canaux actifs d'un gestionnaire de files d'attente.

La valeur par défaut à la création est 10. Il n'existe aucun moyen de remplacer la valeur par défaut à ce moment là.

L'entrée MaxActiveChannels du fichier QMINI peut être modifiée à tout moment, bien qu'elle ne prenne effet qu'une fois le gestionnaire de files d'attente redémarré.

ID de segment Guardian utilisés par MQSeries

MQSeries affecte des segments en mémoire Guardian à ses propres processus et au processus du programme d'application. Les ID de segment Guardian utilisés par MQSeries pour ces segments sont affectés à partir d'une plage définie dans le fichier de configuration MQSINI. L'élément NSKSegidRange de la strophe AllQueueManagers du fichier de configuration MQSINI définit la plage d'ID de segment utilisée par MQSINI pour l'attribution des segments de mémoire.

Lorsque MQSeries est installé, cette plage est définie par défaut sur NSKSegidRange=10-20.

Puisque ces segments de mémoire sont aussi affectés dans le processus du programme d'application, veillez à ce que les applications n'affectent pas de segments compris dans cette plage. Vous pouvez obliger MQSeries à affecter ses segments dans une plage différente en éditant l'élément NSKSegidRange dans votre fichier MQSINI. Une fois cet élément modifié, vous devez arrêter et redémarrer tous les gestionnaires de files d'attente pour que la nouvelle valeur soit prise en compte.

La plage définie par le paramètre NSKSegidRange doit être comprise dans les limites d'ID de segment Guardian 0-1023. Le paramètre NSKSegidRange doit définir une plage contenant au moins 10 ID de segments utilisés par MQSeries.

Port TCP/IP par défaut

L'entrée TCPPort de la strophe TCPConfig de QMINI détermine le numéro de port par défaut pour les canaux entrants. Par défaut, le port utilisé est le 1414. Cette valeur est remplacée par celles qui figurent dans la zone CONNAME d'un canal.

Ports TCP/IP écoutés par le gestionnaire de files d'attente

Un gestionnaire de files d'attente avec canaux TCP/IP peut être configuré pour écouter les connexions entrantes sur un ou plusieurs ports TCP/IP. Le `TCPNumListenerPorts` et les entrées de la strophe `TCPCConfig` du fichier `QMINI` définissent le nombre de ports à écouter, et à affecter à ce gestionnaire de files d'attente. Pour des exemples d'entrées `QMINI`, consultez la section «Exemple de communications TCP/IP» à la page 391.

Un système peut comprendre plusieurs gestionnaires de files d'attente. Chacun d'entre eux doit comprendre des ensembles distincts de ports TCP/IP à écouter. L'ensemble de ports TCP/IP de chaque gestionnaire de files d'attente peut contenir un seul port, lorsque la fréquence des demandes de connexions TCP/IP est faible, ou plusieurs pour les configurations plus importantes. Le port TCP/IP par défaut est 1414, et un gestionnaire de files d'attente est créé pour écouter uniquement ce port par défaut.

La liste des ports d'écoute peut être modifiée en éditant la strophe `TCPCConfig` du fichier `QMINI` et en redémarrant le gestionnaire de files d'attente. Pour pouvoir écouter plus d'un port, un gestionnaire de files d'attente doit aussi être configuré avec des classes de serveurs d'écoute TCP/IP supplémentaires utilisant TS/MP. Cette opération s'effectue manuellement à l'aide de `PATHCOM`.

Sinon, vous pouvez indiquer un numéro de port dans la commande `runmqtsr` (décrite dans la section «`runmqtsr` (exécution du programme d'écoute)» à la page 294). Les valeurs `TCPListenerPort` sont substituées par un programme de classe de serveur d'écoute, si le paramètre `MQLISTENPORTNUM` est présent dans l'environnement du programme d'écoute.

Processus TCP/IP utilisé par le gestionnaire de files d'attente

Le processus de serveur, aussi appelé TCP/IP, sert d'interface avec le produit TCP/IP Compaq. Par défaut, la valeur `$ZTC0`, est utilisée. Il n'existe aucun moyen de remplacer la valeur par défaut lors de la création du gestionnaire de files d'attente. Les canaux d'un gestionnaire de files d'attente peuvent être configurés pour utiliser un processus TCP/IP particulier si la valeur par défaut du système est insuffisante. La configuration de la classe de serveur TS/MP du programme d'écoute TCP/IP peut être modifiée manuellement si le processus TCP/IP par défaut du système, `$ZTC0`, est insuffisant ou incorrect. Cette configuration de classe de serveurs peut aussi être modifiée manuellement pour permettre à un programme d'écoute d'écouter un port spécifique et de remplacer les ports définis dans le fichier `QMINI`.

La classe de serveur `MQS-TCPLIS00` doit intégrer le `DEFINE TCPIP^PROCESS^NAME` pour référencer le nom de processus TCP/IP de remplacement. Notez que si vous avez plusieurs EC, vous devez tous les mettre à jour. Consultez les manuels des produits TCP/IP NSK de Compaq pour plus d'informations.

La modification des classes de serveur TS/MP n'est possible que lorsque le gestionnaire de files d'attente est arrêté.

`MQSeries` pour Compaq NSK peut prendre en charge plusieurs processus TCP/IP par gestionnaire de files d'attente. Plusieurs ports et processus d'écoute TCP/IP sont également supportés.

Reconfiguration de la classe de serveur MQS-TCPLISnn pour un port et un processus TCP/IP personnalisés

Pour reconfigurer la classe de serveur MQS-TCPLISnn pour un port et un processus TCP/IP personnalisés, suivez les instructions générales fournies dans la section «Modification des paramètres de classe de serveurs PATHWAY» à la page 42.

Pour personnaliser un processus TCP/IP, utilisez les commandes PATHCOM suivantes :

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

où \$ZZZZ est le nom du processus TCP/IP requis.

Pour personnaliser un port, utilisez les commandes PATHCOM suivantes :

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

où nnnn est le numéro du port que vous souhaitez écouter.

Pour les programmes d'écoute exécutés à partir de TACL (**runmqtsr**), et nécessitant un nom de processus TCP/IP différent de la valeur par défaut \$ZTC0, ajoutez la chaîne suivante à l'environnement TACL depuis lequel le programme d'écoute sera exécuté :

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

où \$ZZZZ est le nom du processus TCP/IP requis.

Remarque : Si des classes de serveurs de programmes d'écoute TCP/IP individuelles souhaitent utiliser différents processus TCP/IP avec chaque port, elles doivent définir le port et le nom de processus TCP/IP de chaque instance de classe de serveur. Dans le cas contraire, il est impossible de savoir quel sera le processus TCP/IP utilisé par un port défini dans le fichier QMINI.

Si le paramètre MQLISTENPORTNUM n'est pas défini dans l'environnement du programme d'écoute, le programme d'écoute se procure le port dans le fichier QMINI.

Affectation de l'espace de permutation

MQSeries pour Compaq NSK affecte l'espace de permutation en fonction des valeurs ExtPoolSize des divers exécutables dans le fichier de configuration QMINI. Par conséquent, si votre gestionnaire de files d'attente utilise le fichier QMINI par défaut, lequel attribue 300 Ko à chaque exécutable par défaut, et exécute 10 canaux sortants, 20 agents, et 10 canaux répondeurs TCP, vous aurez besoin d'au moins 12 Mo (40 * 300 Ko) d'espace de permutation. Pour réduire cet espace, vous pouvez diminuer les valeurs du fichier QMINI. Les valeurs ExtPoolSize représentent la quantité minimale de mémoire supplémentaire affectée lorsque la mémoire initiale

Création des gestionnaires de files d'attente

est épuisée. La valeur n'a pas besoin d'être supérieure à la taille maximale des messages du gestionnaire de files d'attente.

Un moyen de contrôler l'affectation des permutations des exécutables MQSeries consiste à modifier les classes de serveurs MQ Pathway, en ajoutant :

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

En ajoutant DEFINE, les processus agents créés par le serveur EC héritent également du volume de permutation défini, favorisant ainsi une utilisation évolutive de la permutation lorsque les classes de serveur EC sont utilisées.

Nom de status server par défaut

Vous devez préciser un nom de processus unique pour la paire de processus de status server par défaut lorsque vous créez un gestionnaire de files d'attente. Indiquez le nom du status server par défaut dans le paramètre *-s StatusServerName* obligatoire de la commande **crtmqm**. Il n'y a pas de valeur par défaut pour ce paramètre. Indiquez un nom de processus spécifique au système et facile à associer au gestionnaire de files d'attente auquel il appartient.

Vous pouvez modifier le nom du processus de status server par défaut d'un gestionnaire de files d'attente de la façon suivante :

1. Arrêtez le gestionnaire de files d'attente.
2. Définissez le volume et le sous-volume par défaut d'après l'emplacement des fichiers de données du gestionnaire de files d'attente (normalement <Nom GestFiles>D).
3. Modifiez l'entrée *DefaultStatusServerName* dans la strophe Configuration du fichier QMINI du gestionnaire de files d'attente pour enregistrer le nouveau nom de processus.
4. Exécutez PATHMON pour le gestionnaire de files d'attente en entrant la commande suivante à partir de l'invite TACL :

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

où <pmon> est le nom du processus PATHMON du gestionnaire de files d'attente.

5. Exécutez PATHCOM sur le processus PATHMON en entrant la commande suivante à partir de l'invite TACL :

```
PATHCOM $<pmon>
```

6. Chargez la configuration Pathway de gestionnaire de files d'attente en entrant la commande suivante à partir de PATHCOM :

```
START PATHWAY COOL
```

Création des gestionnaires de files d'attente

7. Modifiez le MQS-STATUS00 du serveur et réinitialisez le processus en entrant les commandes suivantes à partir de PATHCOM :

```
ALTER SERVER MQS-STATUS00, DELETE PROCESS $<anciennom>  
ALTER SERVER MQS-STATUS00, PROCESS $<nouveaunom>
```

8. Réenregistrez la configuration modifiée sur disque en entrant les commandes suivantes depuis PATHCOM :

```
SHUTDOWN2  
EXIT
```

La procédure de modification de nom de processus de status server par défaut est à présent terminée. La prochaine commande **strmqm** démarrera le gestionnaire de files d'attente à l'aide du nom de processus de status server par défaut.

Ajout et suppression de status server personnalisés

Pour ajouter des status server supplémentaires, prenez modèle sur les status server par défaut existants. Il n'est pas nécessaire d'arrêter et de démarrer le gestionnaire de files d'attente pour pouvoir ajouter un nouveau status server.

La procédure suivante permet d'ajouter un nouveau status server :

Remarque : Cette procédure suppose que le gestionnaire de files d'attente soit en marche. Si ce n'est pas le cas, vous devez démarrer PATHMON et charger les configurations PATHWAY avant de démarrer cette procédure. Sautez également l'étape 5 à la page 61, et enregistrez la configuration Pathway sur disque une fois terminé, à l'aide d'une commande SHUTDOWN2.

1. Exécutez PATHCOM sur le processus PATHMON en entrant la commande suivante à partir de l'invite TACL :

```
PATHCOM $<pmon>
```

2. Créez un ensemble d'attributs de travail en prenant modèle sur la classe de status server par défaut, en entrant les commandes suivantes depuis PATHCOM :

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

3. Modifiez l'ensemble de travail du nouveau serveur en entrant les commandes suivantes dans PATHCOM :

```
SET SERVER CPUS(n:m)  
RESET SERVER PROCESS $<nom de processus de serveur de statut par défaut>  
SET SERVER PROCESS $<nouveau nom de serveur de statut>
```

Création des gestionnaires de files d'attente

4. Ajoutez le nouveau serveur, en lui donnant un nouveau nom de classe de serveur, en entrant la commande suivante dans PATHCOM :

```
ADD SERVER MQS-STATUSxx
```

5. Démarrez le nouveau status server de façon à pouvoir le réutiliser, et quittez PATHCOM en entrant les commandes suivantes :

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

La convention d'appellation recommandée pour les classes de serveur supplémentaires est MQS-STATUS01, MQS-STATUS02, etc. Vous n'êtes toutefois pas obligé de l'utiliser. Si le nom de classe de serveur commence par la chaîne de caractères MQS-STATUS, la classe de serveur sera démarrée par la commande **strmqm**.

Une fois qu'un status server a été ajouté et démarré (explicitement à l'aide de PATHCOM, ou implicitement avec la commande **strmqm**), vous pouvez réaffecter des objets au nouveau serveur de statut avec **almqfls**. Pour plus d'informations sur la réaffectation des objets, reportez-vous à la section «Reassignation d'objets aux serveurs de statut et de file d'attente» à la page 107.

Avant de supprimer un status server, contrôlez que tous ses objets ont été soit supprimés soit réaffectés à un autre status server. Vous **ne devez pas** supprimer le serveur de statut par défaut, sinon le gestionnaire de files d'attente ne fonctionnera plus.

Nom de serveur de file d'attente par défaut

Vous devez préciser un nom de processus unique pour la paire de processus de serveur de file d'attente par défaut lorsque vous créez un gestionnaire de files d'attente. Indiquez le nom de serveur de file d'attente par défaut dans le paramètre obligatoire **-v QueueServer** de la commande **crmqm**. Il n'y a pas de valeur par défaut pour ce paramètre. Indiquez un nom de processus spécifique au système et facile à associer au gestionnaire de files d'attente auquel il appartient.

Vous pouvez modifier le nom de processus de serveur de file d'attente par défaut d'un gestionnaire de files d'attente de la façon suivante :

1. Arrêtez le gestionnaire de files d'attente.
2. Définissez le volume et le sous-volume par défaut d'après l'emplacement des fichiers de données du gestionnaire de files d'attente (normalement <Nom GestFiles>D).
3. Modifiez l'entrée `DefaultQueueServerName` dans la strophe Configuration du fichier QMINI du gestionnaire de files d'attente pour enregistrer le nouveau nom de processus.

Création des gestionnaires de files d'attente

4. Exécutez PATHMON pour le gestionnaire de files d'attente en entrant la commande suivante à partir de l'invite TACL :

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

où <pmon> est le nom du processus PATHMON du gestionnaire de files d'attente.

5. Exécutez PATHCOM sur le processus PATHMON en entrant la commande suivante à partir de l'invite TACL :

```
PATHCOM $<pmon>
```

6. Chargez la configuration Pathway de gestionnaire de files d'attente en entrant la commande suivante à partir de PATHCOM :

```
START PATHWAY COOL
```

7. Modifiez le mqs-queue*nm* du serveur et réinitialisez le processus en entrant les commandes suivantes à partir de PATHCOM:

```
ALTER SERVER MQS-QUEUEnn, DELETE PROCESS $<o1dname>  
ALTER SERVER MQS-QUEUEnn, PROCESS $<nouveaunom>
```

8. Réenregistrez la configuration modifiée sur disque en entrant les commandes suivantes depuis PATHCOM :

```
SHUTDOWN2  
EXIT
```

La procédure de modification de nom de processus de status server par défaut est à présent terminée. La prochaine commande **strmqm** démarrera le gestionnaire de files d'attente à l'aide du nom de processus de status server par défaut.

Ajout et suppression de serveurs de file d'attente personnalisés

Pour ajouter des serveurs de file d'attente, prenez modèle sur le serveur de file d'attente par défaut. Il n'est pas nécessaire d'arrêter et de démarrer le gestionnaire de files d'attente pour pouvoir ajouter un nouveau serveur de file d'attente.

La procédure suivante permet d'ajouter un nouveau serveur de file d'attente :

Remarque : Cette procédure suppose que le gestionnaire de files d'attente soit en marche. Si ce n'est pas le cas, vous devez démarrer PATHMON et charger les configurations PATHWAY avant de démarrer cette procédure. Sautez également l'étape 5 à la page 61, et enregistrez la configuration PATHWAY sur disque une fois terminé, à l'aide d'une commande SHUTDOWN2.

Création des gestionnaires de files d'attente

1. Exécutez PATHCOM sur le processus PATHMON en entrant la commande suivante à partir de l'invite TACL :

```
PATHCOM $<pmon>
```

2. Créez un ensemble d'attributs de travail en prenant modèle sur la classe de serveur de file d'attente par défaut, en entrant les commandes suivantes depuis PATHCOM :

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

3. Modifiez l'ensemble de travail du nouveau serveur en entrant les commandes suivantes dans PATHCOM :

```
SET SERVER CPUS(n:m)  
RESET SERVER PROCESS $<nom de processus de serveur de file d'attente par défaut>  
SET SERVER PROCESS $<nouveau nom de serveur de file d'attente>
```

4. Ajoutez le nouveau serveur, en lui donnant un nouveau nom de classe de serveur, en entrant la commande suivante dans PATHCOM :

```
ADD SERVER MQS-QUEUEnn
```

5. Démarrez le nouveau serveur de file d'attente de façon à pouvoir l'utiliser, et quittez PATHCOM en entrant les commandes suivantes :

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

La convention d'appellation des classes de serveur supplémentaire recommandée est MQS-QUEUE01, MQS-QUEUE02, etc. Vous n'êtes toutefois pas obligé de l'utiliser. Si le nom de classe de serveur commence par la chaîne de caractères MQS-QUEUE, la classe de serveur pourra être démarrée par la commande **strmqm**.

Une fois qu'un serveur de file d'attente a été ajouté et démarré (explicitement à l'aide de PATHCOM, ou implicitement avec la commande **strmqm**), vous pouvez réaffecter des objets au nouveau serveur de file d'attente avec **altmqfls**. Pour plus d'informations sur la réaffectation des objets, reportez-vous à la section «Reassignation d'objets aux serveurs de statut et de file d'attente» à la page 107.

Avant de supprimer un serveur de file d'attente, contrôlez que tous ses objets ont été soit supprimés soit réaffectés à un autre serveur de file d'attente. Vous **ne devez pas** supprimer le serveur de file d'attente, sinon le gestionnaire de files d'attente ne fonctionnera plus.

Structure du volume

Les fichiers pour MQSeries pour Compaq NSK sont répartis sur plusieurs sous-volumes. Vous êtes amené à choisir le volume dans lequel résident ces sous-volumes lorsque vous créez le gestionnaire de files d'attente : il peut s'agir de la valeur de volume par défaut de MQSINI ou d'une valeur précisée par le paramètre *-p DefaultPrefix* de la commande **crtmqm**.

Il y a cinq sous-volumes par gestionnaire de files d'attente. Le contenu des sous-volume est déterminé par le dernier caractère du nom du sous-volume. Par exemple, pour un gestionnaire de files d'attente nommé QMGR, résidant sur un volume \$DATA, les sous-volumes suivants sont présents :

\$DATA.QMGR	Sous-volume FFST
\$DATA.QMGRD	Sous-volume de fichiers de données de gestionnaire de files d'attente
\$DATA.QMGRL	Sous-volume des journaux d'erreurs du gestionnaire de files d'attente
\$DATA.QMGRM	Sous-volume de la file d'attente de messages
\$DATA.QMGRS	Sous-volume de synchronisation des canaux

Si le nom du gestionnaire de files d'attente contient plus de 17 caractères, les noms de sous-volumes sont transformés ou raccourcis. Les strophes QMVolume et QMSubvolume du fichier MQSINI du gestionnaire de files d'attente sont utilisées pour enregistrer les emplacements et noms de ces sous-volumes.

Sous-volume FFST d'un gestionnaire de files d'attente

Le sous-volume FFST contient des fichiers de support d'incident. Ces fichiers commencent tous par les lettres FD. Ils indiquent des incidents sérieux rencontrés par le système MQSeries, tels que des ressources insuffisantes, des erreurs MQSeries internes ou des problèmes avec Compaq NSK.

Sous-volume de fichiers de données de gestionnaire de files d'attente

AMQRFNxx	Fichiers de mémoire partagée de cache de référentiel pour chaque processeur (numéro <i>xx</i>).
CCHDEFS	Fichier de définition de canal de connexion client.
CCSIDMEM	Fichier de mémoire partagée en lecture seule pour le support CCSID de la conversion de données.
CHDEFS	Fichier de définition de canal. Contient les informations de configuration des canaux définis pour un gestionnaire de files d'attente.
Lxxxxxxx	Fichiers de liste de noms. Lxxxxxxx est construit sur la base du nom d'objet ; sinon, la valeur est générée automatiquement.
OAMDB	Base de données de droits d'accès.
OBJCAT	Catalogue d'objets.
ABJCAT	Fichier clé de remplacement pour OBJCAT, contenant un index par noms d'objet.
PATHCTL	Fichier de contrôle PATHWAY.
PRIDB	Base de données des principaux (utilisateurs) MQSeries.

PRIDBA	Fichier de clé de remplacement pour PRIDB.
QMINI	Fichier d'initialisation de gestionnaire de files d'attente.
QMINIMEM	Fichier de mémoire partagée en lecture seule pour les informations de configuration statique du gestionnaire de files d'attente entier.
REPMGR	Fichier utilisé pour coordonner le démarrage de plusieurs processus de serveur de référentiel.
RDFPURGE	Base de données utilisée par le gestionnaire de files d'attente pour enregistrer les fichiers supprimés de façon logique, et qui seront éventuellement supprimés par cleanrdf .
RUNTIME	Fichier de coordination de reprise du contrôleur d'exécution principal et du contrôleur d'exécution.
SHUTDOWN	Fichier permettant de contrôler endmqm .
STATABLE	Fichier de table de statut de canal. Contient les informations dynamiques associées au statut des canaux. Utilisé pour enregistrer les informations de statut des canaux pendant l'arrêt et le redémarrage d'un gestionnaire de files d'attente.
TRACEOPT	Fichier contenant les paramètres de suivi en cours d'un gestionnaire de files d'attente, sous forme d'un enregistrement bitmap non formaté. Les commandes de contrôle strmqtrc et endmqtrc modifient le contenu du fichier à l'aide du mécanisme CONTROL 27, pour notifier tous les processus de la mise à jour.
UMQSINI	Image instantanée d'un fichier MQSINI non structuré, au démarrage du gestionnaire de files d'attente.

Sous-volume des journaux d'erreurs des gestionnaires de files d'attente

Le sous-volume de journaux d'erreurs contient les fichiers de trace. Le préfixe TR identifie les fichiers de trace. (Vous pouvez changer le préfixe en modifiant l'entrée TracePrefix du fichier QMINI.) Les fichiers de trace contiennent des informations de diagnostic, et sont créés uniquement si le traçage est activé à l'aide de la fonction MQM ou de la commande de contrôle **strmqtrc**.

Le format du nom des journaux d'erreurs est MQERRLGn, où n est 1, 2, ou 3. MQERRLG1 est toujours le journal d'erreurs en cours. Son contenu est déplacé vers MQERRLG2 lorsque MQERRLG1 est plein ; MQERRLG2 est déplacé vers MQERRLG3 lorsque MQERRLG1 est vidé. MQERRLG3 est écrasé si nécessaire. Il n'y a jamais plus de trois journaux d'erreurs. Ils doivent donc être dimensionnés correctement afin d'éviter de perdre de précieuses informations sur les erreurs.

Sous-volume de files d'attente de messages de gestionnaires

Le sous-volume de messages contient les fichiers associés au stockage des messages des files d'attente locales. Les formats des noms de fichiers sont les suivants :

Qxxxxxx	Fichiers de files d'attente, contenant les messages persistants.
Txxxxxx	Fichiers d'horodatage. Si un objet est modifié, les fichiers d'horodatage changent la date de l'objet. Txxxxxx est construit sur la base du nom d'objet ; sinon, la valeur est générée automatiquement.
Oxxxxxx	Fichier de dépassement de files d'attente.

Structure du volume

xxxxxxx Nom de la file d'attente, s'il est unique et court ; sinon la valeur est générée automatiquement. (Consultez la section «Transformation des noms d'objets».)

En plus des fichiers décrits ci-dessus commençant par une lettre spécifique, les fichiers de dépassement de message sont créés par défaut dans le sous-volume de files d'attente de messages. Leur rôle est de stocker les messages volumineux. Bien qu'ils n'aient aucun préfixe, ils sont facilement identifiables car ce sont des fichiers non contrôlés et non structurés. L'emplacement des nouveaux fichiers de dépassement de messages peut être modifié à l'aide de **altmqfls**, file d'attente par file d'attente, en précisant le paramètre `--msgofsubvol`. (Pour plus d'informations, consultez la section «**dspmqfls** (affichage des fichiers MQSeries)» à la page 278.)

Sous-volume de synchronisation des canaux des gestionnaires de files d'attente

Le sous-volume de synchronisation de gestionnaires de files d'attente contient des bases de données internes qui enregistrent le statut des unités de travail (ou lots de messages) transmises ou reçues sur les canaux d'un gestionnaire de files d'attente.

Une fois que des canaux ont été utilisés par un gestionnaire de files d'attente, le sous-volume contient les fichiers suivants :

SYNCHIDX

Fichier d'index de synchronisation. Contient une entrée pour chaque fichier de synchronisation créé par le gestionnaire de files d'attente.

Sxxxxxxx

Fichiers de synchronisation individuels. A chaque combinaison unique de canal local et éloigné, utilisée par le gestionnaire de files d'attente correspond un fichier. Ces fichiers enregistrent les identités des messages transmis ou reçus dans un lot de messages. Les informations sont utilisées lors de la resynchronisation des canaux, suite à un incident ou à la résolution de canaux en attente de validation.

Transformation des noms d'objets

Les noms d'objets ne sont pas obligatoirement des noms de systèmes de fichiers corrects. En conséquence, il est parfois nécessaire de modifier les noms d'objets. La méthode utilisée diffère de celle employée pour les noms de gestionnaires de files. En effet, s'il existe un nombre réduit de gestionnaires de files sur un système, il peut y avoir un grand nombre d'objets pour chaque gestionnaire de files. Seules les définitions de processus, les listes de noms et files d'attente sont représentées à l'aide de fichiers distincts dans le système de fichiers ; les canaux et autres objets ne sont pas concernés par ces considérations car ils sont stockés sous forme d'enregistrements dans des bases de données contenant plusieurs définitions d'objets.

Lorsqu'un nouveau nom est généré par le processus de transformation, il ne présente pas de relation avec le nom d'origine de l'objet. Vous pouvez utiliser la commande **dspmqfls** pour effectuer la conversion entre les noms d'objets réels et transformés : **dspmqfls** affiche les noms des principaux fichiers associés à un objet MQSeries.

Utilisation des gestionnaires de files d'attente

MQSeries fournit des commandes de création, de démarrage, d'arrêt et de suppression des gestionnaires de files d'attente. Vous pouvez également afficher les attributs d'un gestionnaire de files d'attente à l'aide de la commande MQSC, DISPLAY QMGR et les modifier à l'aide de ALTER QMGR. Consultez les sections «Affichage des attributs d'un gestionnaire de files d'attente» à la page 96 et «Modification des attributs d'un gestionnaire de files d'attente» à la page 98.

Veillez à ce que la variable d'environnement PMSEARCHLIST indique l'emplacement de vos exécutables MQSeries avant d'utiliser les commandes de contrôle. Pour plus d'informations, reportez-vous à l'«Annexe C. Définition de variables d'environnement TAACL pour MQSeries pour Compaq NSK» à la page 323.

Création d'un gestionnaire de files d'attente par défaut

Pour créer un gestionnaire de files d'attente, vous devez utiliser la commande **crtmqm**. La commande **crtmqm** accompagnée de l'indicateur **-q** :

- crée un gestionnaire de files d'attente par défaut nommé `saturne.gest.files` ;
- crée les objets système et les objets par défaut ;
- définit les noms de la file d'attente de transmission par défaut et de la file d'attente de rebut.

```
crtmqm -q -d MY.DEF.XMITQ -u FILE.ATTENTE.REBUT.SYSTEME -n $PMON -o $TRM01 -s $MQSS
-v $MQQS saturne.gest.files
```

où :

-q précise que ce gestionnaire de files d'attente est le gestionnaire par défaut.

-d MY.DEFXMIT.Q
est le nom de la file d'attente de transmission par défaut.

-u SYSTEM.DEAD.LETTER.QUEUE
est le nom de la file d'attente de rebut.

-n \$PMON
est le nom de processus PATHMON pour le gestionnaire de files d'attente.

-o \$TRM01
est le nom du terminal d'origine (doit être en Pause).

-s \$MQSS
est le nom de processus du status server par défaut.

-v \$MQQS
est le nom de processus du serveur de files d'attente par défaut.

saturne.gest.files
est le nom de ce gestionnaire de files d'attente. Pour **crtmqm**, ce nom doit être le dernier paramètre de la commande.

Un gestionnaire de files d'attente doté du nom et des options que vous avez choisis est créé. En créant un gestionnaire de files d'attente, vous avez également créé automatiquement les éléments suivants :

- Principal MQSeries par défaut pour le créateur du gestionnaire de files d'attente

Utilisation des gestionnaires de files d'attente

- Serveur de statut du gestionnaire de files d'attente
- Serveur de file d'attente du gestionnaire de files d'attente
- Fichier de configuration du gestionnaire de files d'attente

Vous pouvez à présent démarrer le gestionnaire de files d'attente. Pour plus d'informations, reportez-vous au «Lancement d'un gestionnaire de files d'attente» à la page 67.

Vous aurez peut-être besoin de modifier certains attributs d'un gestionnaire de files d'attente. Vous pouvez utiliser pour cela MQM ou la commande de contrôle `strmqm`. Pour plus d'informations, consultez la figure 5 à la page 72 ou la section «`strmqm` (lancement d'un gestionnaire de files d'attente)» à la page 307.

Sauvegardez les deux fichiers de configuration définis lors de la création du gestionnaire de files d'attente. Si vous créez un autre gestionnaire de files d'attente provoquant des incidents, vous pouvez rétablir les copies de secours. Pour plus d'informations, reportez-vous au «Sauvegarde des fichiers de configuration après la création d'un gestionnaire de files d'attente» à la page 48.

Création de principaux MQSeries

La commande `crtmqm` crée automatiquement un principal pour l'utilisateur ayant créé le gestionnaire de files d'attente. Ce principal (aussi nommé principal par défaut) est toujours appelé `mqm` dans un souci de compatibilité avec les implémentations MQSeries.

Une fois un gestionnaire de files d'attente créé, vous pouvez définir des principaux pour d'autres utilisateurs MQSeries. Cette procédure peut être effectuée à tout moment (que le gestionnaire de files d'attente soit démarré ou non). Si aucun autre utilisateur n'est requis pour le gestionnaire de files d'attente, vous pouvez sauter cette étape.

Pour créer un principal MQSeries nommé `MQPRINCIPAL` correspondant à l'utilisateur Compaq NSK `MQM.MQUSER`, entrez la commande :

```
altmqusr -m saturne.gest.files -p MQPRINCIPAL -u MQM.MQUSER
```

Pour afficher tous les principaux actuellement créés, entrez la commande :

```
dspmqsusr -m saturne.gest.files
```

Souvenez-vous que si vous ne créez aucune entrée principale pour un utilisateur, toute tentative d'accès au gestionnaire de files d'attente par l'utilisateur (que l'OAM soit activé ou non) provoquera une erreur de droits d'accès. Cette modification est apparue dans MQSeries pour Compaq NSK Version 2.2.0.1 et figure dans toutes les versions ultérieures.

Exécution de cleanrdf pour un gestionnaire de files d'attente prêt pour RDF

Si vous utilisez un gestionnaire de files d'attente en environnement RDF, et avez autorisé les comportements spécifiques à RDF à l'aide du PARAM MQRDF, exécutez régulièrement l'utilitaire **cleanrdf**, de la façon suivante :

- Après avoir modifié la configuration (en créant ou supprimant des objets, ou en rectifiant le fichier QMINI), exécutez **cleanrdf**.
- Si votre application crée et supprime des objets dans le cadre d'un fonctionnement normal, en particulier si elle utilise des files d'attente dynamiques, exécutez **cleanrdf** pendant le fonctionnement normal selon une fréquence variable en fonction du rapprochement des suppressions. Vous pouvez utiliser NetBatch ou tout autre logiciel de programmation.

Lancement d'un gestionnaire de files d'attente

Un gestionnaire de files d'attente qui vient d'être créé ne peut pas traiter de commandes ni d'appels MQI tant qu'il n'a pas été démarré. Lancez le gestionnaire de files d'attente à l'aide de la commande suivante :

```
strmqm saturne.gest.files
```

La commande **strmqm** ne rend le contrôle que lorsque le gestionnaire a été lancé et qu'il est prêt à accepter des demandes de connexion.

Restauration des objets système et par défaut

Les objets système et par défaut sont automatiquement créés au même moment que les gestionnaires de files d'attente. Toutefois, les objets peuvent être remplacés par d'autres définitions d'objets à tout moment. Pour restaurer les objets système et par défaut pour un gestionnaire de files d'attente nommé saturne.gest.files, émettez la commande **strmqm** avec l'option **-c** :

```
strmqm -c saturne.gest.files
```

Le gestionnaire de files d'attente est lancé, les objets système et par défaut créés en même temps que le gestionnaire de files d'attente sont restaurés, puis le gestionnaire de files d'attente est arrêté.

Consultation des fichiers objets

Chaque objet MQSeries, file d'attente, gestionnaire de files d'attente ou processus, est représenté par un fichier. Comme les noms d'objets ne sont pas forcément des noms de fichiers corrects, le gestionnaire de files d'attente les transforme en des noms de fichiers corrects, le cas échéant. Ce processus est décrit dans la section «Transformation des noms d'objets» à la page 64.

Arrêt d'un gestionnaire de files d'attente

Pour arrêter un gestionnaire de files d'attente, utilisez la commande **endmqm**. Par exemple, pour arrêter un gestionnaire nommé `saturne.gest.files`, utilisez la commande suivante :

```
endmqm saturne.gest.files
```

Par défaut cette commande effectue un arrêt *contrôlé* ou *progressif* du gestionnaire de files d'attente indiqué. L'exécution de cette commande peut mettre un certain temps, car un arrêt progressif n'arrête le gestionnaire que lorsque *toutes* les applications actives ont été déconnectées, et que tous les canaux en cours sont arrêtés.

La section «Arrêt immédiat et arrêt forcé d'un gestionnaire de files d'attente» décrit les indicateurs optionnels pour la commande **endmqm**, indiquant la façon dont l'arrêt doit être effectué.

En cas d'incident

Les incidents survenant lors de l'arrêt d'un gestionnaire de files d'attente sont souvent provoqués par les applications. Par exemple, les applications :

- ne vérifient pas correctement les codes retour MQI,
- ne demandent pas la notification d'une mise au repos.

Arrêt immédiat et arrêt forcé d'un gestionnaire de files d'attente

Si la fermeture d'un gestionnaire de files d'attente est très lente, ou s'il ne s'arrête pas, vous pouvez annuler la commande **endmqm** par BREAK suivi de STOP. Vous pouvez alors émettre une autre commande **endmqm**, en spécifiant cette fois-ci un indicateur correspondant au type d'arrêt souhaité (immédiat ou forcé).

Dans un *arrêt immédiat*, l'exécution de tous les appels MQI en cours est autorisée, mais les nouveaux appels échouent. Ce type d'arrêt arrête le gestionnaire sans attendre que les applications soient déconnectées. Pour exécuter un arrêt immédiat, tapez :

```
endmqm -i saturne.gest.files
```

Si un arrêt immédiat échoue, il vous reste la possibilité de lancer un arrêt *forcé*, avec l'indicateur `-p`. Par exemple :

```
endmqm -p saturne.gest.files
```

Attention

N'utilisez cette méthode que si toutes les autres tentatives d'arrêt par la commande **endmqm** ont échoué. En effet, cette méthode peut avoir des conséquences imprévisibles sur les applications connectées.

Utilisation des gestionnaires de files d'attente

Si cette méthode ne fonctionne toujours pas, reportez-vous à la section «Arrêt manuel d'un gestionnaire de files d'attente» à la page 331 pour savoir comment procéder.

Pour une description détaillée de la commande et de ses paramètres, reportez-vous à la section «endmqm (arrêt d'un gestionnaire de files d'attente)» à la page 286.

Redémarrage d'un gestionnaire de files d'attente

Pour redémarrer un gestionnaire de files d'attente, utilisez la commande :

```
strmqm saturne.gest.files
```

Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant

Lorsque vous créez un gestionnaire de files d'attente par défaut, son nom est inséré dans la strophe *DefaultQueueManager* du fichier de configuration MQSeries (MQSINI). La strophe et son contenu sont créés automatiquement s'ils n'existent pas.

Vous pouvez être amené à modifier la strophe dans les cas suivants :

- Si vous souhaitez définir un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant. Pour cela, vous devez remplacer dans la strophe le nom du gestionnaire de files par le nom du nouveau gestionnaire de files d'attente par défaut. Cette opération doit se faire manuellement avec un éditeur de texte.
- Si vous voulez définir un gestionnaire de files d'attente existant comme gestionnaire par défaut, alors qu'aucun gestionnaire par défaut n'est défini sur le noeud. Pour cela, vous devez créer vous-même la strophe *DefaultQueueManager* avec le nom approprié.
- Si vous avez changé accidentellement le gestionnaire de files d'attente par défaut et que vous voulez revenir au premier gestionnaire par défaut. Pour cela, modifiez la strophe *DefaultQueueManager* du fichier de configuration MQSeries, en remplaçant le nom du gestionnaire par défaut indésirable par celui que vous voulez garder.

Le gestionnaire de files d'attente par défaut est remplacé par celui que vous avez indiqué. Toutes les connexions suivantes pour lesquelles aucun gestionnaire de files d'attente n'aura été spécifié, se connecteront au nouveau gestionnaire de files d'attente par défaut.

Pour plus de détails sur les fichiers de configuration, reportez-vous au «Chapitre 13. Fichiers de configuration» à la page 189.

Lorsque vous avez fourni les données de configuration requises, arrêtez le gestionnaire de files d'attente et redémarrez-le.

Suppression d'un gestionnaire de files d'attente

Pour supprimer un gestionnaire de files d'attente, vous devez d'abord l'arrêter, puis entrer la commande suivante :

```
dltmqm saturne.gest.files
```

Attention

Si vous supprimez un gestionnaire de files d'attente, vous effacerez également toutes les ressources qui lui sont associées, dont toutes les files d'attentes et leurs messages et toutes les définitions d'objets. De plus, il se peut que tous les fichiers des sous-volumes du gestionnaire de files d'attente soient purgés (même s'ils n'ont pas été créés par MQSeries).

Pour une description détaillée de la commande **dltmqm** et de ses options, reportez-vous à la section «dltmqm (suppression d'un gestionnaire de files d'attente)» à la page 271. Vous devez vous assurer que seuls les administrateurs autorisés ont les droits d'accès pour utiliser cette commande.

Si les méthodes habituelles de suppression d'un gestionnaire de files d'attente ne fonctionnent toujours pas, reportez-vous à la section «Suppression manuelle de gestionnaires de file d'attente» à la page 331 pour savoir comment procéder.

Utilisation de la fonction MQM (Message Queue Management)

La fonction MQM (Message Queue Management) de MQSeries pour Compaq NSK, V5.1 s'exécute comme un demandeur PATHWAY SCOBOL sous TCP (Terminal ControlProcess). Elle utilise un serveur de classe de serveur MQM, lequel appelle une API en langage C.

Chaque gestionnaire de files d'attente du système dispose d'une instance séparée de MQM, car chaque gestionnaire de files d'attente est contrôlé sous sa propre configuration PATHWAY. Par conséquent, MQM se limite à gérer le gestionnaire de files d'attente auquel il appartient.

Remarque : Par défaut, un maximum de 10 utilisateurs peuvent utiliser simultanément la fonction MQM. Pour repousser cette limite à 20 utilisateurs par exemple, saisissez la commande suivante depuis le PATHWAY du gestionnaire de files d'attente :

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

Pour plus d'informations, reportez-vous au «Chapitre 3. Utilisation des jeux de commandes MQSeries» à la page 29.

Remarque :

Pour appeler MQM, entrez `run mqmc` à l'invite PATHCOM du gestionnaire de files d'attente.

Message Queue Management (MQM)

Le menu principal de MQM est le suivant :

```
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.
```

Figure 4. Menu principal MQMQ de MQSeries pour Compaq NSK

Les sous-menus suivants sont proposés à partir du menu principal de MQM :

1. Queue Manager
2. Queues
3. Channels

Ceux-ci sont décrits dans le reste de ce chapitre. Vous pouvez revenir à tout moment au menu principal MQM en appuyant sur les touches Alt+F6. Vous pouvez revenir à l'écran précédent en appuyant sur la touche Return (F16). Si vous appuyez sur la touche F16 depuis le menu principal MQM, vous quittez la fonction MQM.

Message Queue Management (MQM)

Utilisation du menu Queue Manager

Pour sélectionner l'option Queue Manager, tapez 1 dans la zone Enter Choice du menu principal, et appuyez sur la touche Enter (F1). L'écran Queue Manager Menu s'affiche :

```
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
```

Figure 5. Ecran Queue Manager Menu

Appuyez sur la touche PGDN pour afficher le second écran d'information.

```
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
```

Figure 6. Ecran 2 du menu Queue Manager

Vous pouvez utiliser l'écran de menu Queue Manager pour :

- Modifier certains attributs du gestionnaire de files d'attente
- Contrôler le traçage des objets MQSeries

Modification des attributs d'un gestionnaire de files d'attente

Réécrivez les valeurs que vous voulez modifier dans l'écran de menu Queue Manager, et appuyez sur la touche Modify (F1). Vous n'avez pas accès aux valeurs dont la modification est interdite.

Traçage des objets MQSeries

Appuyez sur la touche Trace (F2) pour afficher le menu QUEUE MANAGER TRACE :

```
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.
_ LQMFLWS  : 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
```

Figure 7. Menu QUEUE MANAGER TRACE

Les options de trace suivantes sont disponibles :

- API** : Données de sortie des points de trace associés au MQI et aux composants principaux du gestionnaire de files d'attente.
- COMMS** : Données de sortie des points de trace associés aux données transmises par des réseaux de communication.
- CSFLOWS** : Données de sortie des points de trace associés au flot de traitement des fonctions communes.
- LQMFLWS** : Données de sortie des points de trace associés au flot de traitement du gestionnaire de files d'attente local.
- REMOTEFLOWS** : Données de sortie des points de trace associés au flot de traitement du composant de communication.
- ADMINFLOW** : Données de sortie des points de trace associés aux tampons de données internes d'administration.
- OTHERFLOWS** : Données de sortie des points de trace associés au flot de traitement d'autres composants.
- CSDATA** : Données de sortie des points de trace associés aux tampons de données internes des fonctions communes.

Message Queue Management (MQM)

LQMDATA Données de sortie des points de trace associés aux tampons de données internes du gestionnaire de files d'attente local.

REMOTEDATA

Données de sortie des points de trace associés aux tampons de données internes du composant de communication.

ADMINDATA

Données de sortie des points de trace associés aux tampons de données internes du composant de communication.

OTHERDATA Données de sortie des points de trace associés aux tampons de données internes d'autres composants.

VERSIONDATA

Données de sortie des points de trace associés à la version de MQSeries en cours d'exécution.

COMMENTARY

Données de sortie des points de trace associés aux commentaires des composants MQSeries.

ALL Les points de trace sont activés et la trace complète est générée.

Tapez n'importe quel caractère en face des noms de composants pour lesquels vous souhaitez commencer (ou arrêter) le traçage.

Pour commencer le traçage des composants sélectionnés, appuyez sur la touche Start Trace (F1). Pour arrêter le traçage des composants sélectionnés, appuyez sur la touche Stop Trace (F2).

Utilisation du menu Queues

Pour sélectionner l'option Queues, tapez 2 dans la zone Enter Choice du menu principal MQM, et appuyez sur la touche Enter (F1). L'écran Search Criteria s'affiche :

```

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
    
```

Figure 8. Ecran Search Criteria (queue)

```

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.SDRG.LOCAL                    QLOCAL
- ANNE.ET01.SDRG.REMOTE                   QREMOTE
- ANNE.M401.RQSD.LOCAL                    QLOCAL
- ANNE.M401.RQSD.REMOTE                   QREMOTE
- ANNE.M401.RQSV.LOCAL                    QLOCAL
- ANNE.M401.RQSV.REMOTE                   QREMOTE
- ANNE.M401.SDRG.LOCAL                    QLOCAL
- ANNE.M401.SDRG.REMOTE                   QREMOTE

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

Figure 9. Menu Queue

Dans la zone Queue Name de l'écran Search Criteria, tapez un nom de file d'attente complet ou partiel. Vous pouvez aussi préciser un identificateur de type de file d'attente (zone Queue Type) si vous souhaitez limiter votre recherche aux

Message Queue Management (MQM)

files d'attente d'un seul type. Appuyez sur la touche Enter (F1). Le menu Queue, utilisé pour afficher, modifier, créer, copier, supprimer et contrôler les files d'attente MQSeries s'affiche.

Remarque : Vous pouvez créer, modifier et supprimer des files d'attente uniquement sur le gestionnaire associé à la demande MQM utilisée.

Utilisez les touches PGUP et PGDN pour faire défiler la liste des files d'attente.

Création d'une file d'attente

Depuis le menu Queue, appuyez sur la touche Create (F2) pour afficher l'écran Create Queue :

```
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
```

Figure 10. Ecran Create Queue

Pour créer une nouvelle définition de file d'attente :

1. Tapez 1 (pour une file d'attente locale), 2 (pour un modèle de file d'attente), 3 (pour une file d'attente éloignée), ou 4 (pour une file d'attente alias) dans la zone Queue Type.
2. Tapez le nom du gestionnaire de files d'attente dans la zone Name.
3. Si la file d'attente remplace une autre file du même nom et de même type, tapez Y dans la zone Replace.
4. Appuyez sur la touche Enter (F1).

Message Queue Management (MQM)

Si vous créez une file d'attente locale, l'écran Create Local Queue s'affiche :

```
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
```

Figure 11. L'écran Create Local Queue

Appuyez sur la touche PGDN pour afficher le second écran d'information.

```
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
```

Figure 12. Ecran 2 Create Local Queue

Complétez l'écran et appuyez sur la touche Enter (F1).

Message Queue Management (MQM)

Si vous créez une file d'attente éloignée, l'écran Create Remote Queue s'affiche :

```
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
```

Figure 13. Ecran Create Remote Queue

Complétez l'écran et appuyez sur la touche Enter (F1).

Copie d'une file d'attente

A partir du menu Queue, appuyez sur la touche Copy (F3) pour définir une nouvelle file d'attente en copiant une définition existante. L'écran Copy Queue s'affiche :

```
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
```

Figure 14. Ecran Copy Queue

Message Queue Management (MQM)

Tapez le nom de la définition de file d'attente dans la zone Name ; tapez Y dans la zone Replace si la file d'attente doit remplacer une file existante de même nom et du même type ; tapez le nom de la définition copiée dans la zone Like Queue ; tapez le type de file d'attente dans la zone Queue Type. Appuyez sur la touche Enter (F1).

Modification d'une file d'attente

Depuis le menu Queue, appuyez sur la touche Modify (F1) pour afficher l'écran Display/Modify Local Queue :

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

Retention Interval  : 999999999  Queue Definition Type : PREDEFINED
Max Queue Depth    : 5000        Priority/FIFO [P/F]   : P
Max Message Length : 1024        Share [Y/N]          : Y
Backout Threshold  : 0           Usage [N/X]          : N
Backout Requeue Name :
Init. Queue       :
Process Name      :
Trigger Type [N/E/F/D]: F        Trigger/NoTrigger [Y/N] : N
Trigger Depth    : 1            Trigger Priority       : 0
Trig. Data :
Q Depth Max Event : Y           Q Serv. Int. Event [H/O/N]: N
Q Depth High Limit : 80        Q Depth High Event      : N
Q Depth Low Limit  : 20        Q Depth Low Event       : N
Q Service Interval : 999999999  Scope                   : QMGR
F1 - Modify      PGDN - Next Page                F16 - Return
```

Figure 15. Ecran Display/Modify Local Queue

Réécrivez les valeurs que vous voulez modifier, et appuyez sur la touche Modify (F1). Vous n'avez pas accès aux valeurs dont la modification est interdite.

Suppression d'une file d'attente

Dans le menu Queue, entrez tout caractère en face du nom de la file d'attente que vous souhaitez modifier. Appuyez sur la touche Delete (F4), puis de nouveau sur F4 pour confirmer la suppression.

Message Queue Management (MQM)

Contrôle d'une file d'attente

Appuyez sur la touche Monitor (F5) dans le menu Queue pour afficher l'écran Monitor Local Queues :

```
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
```

Figure 16. Ecran Monitor Local Queues

Dans cet exemple, les files d'attente ne sont ouvertes ni pour les entrées, ni pour les sorties. Une file d'attente, ANNE_ME02_SDRD_LOCAL, contient 10 messages.

Le paramètre MQMQMREFRESHINT PATHWAY de MQS-MQMSVR00 détermine la fréquence de rafraîchissement des écrans de moniteur des canaux et files d'attente. La fréquence par défaut est de 30 secondes. Pour que le rafraîchissement s'effectue toutes les 10 secondes, par exemple, entrez la commande suivante dans PATHWAY pour votre gestionnaire de files d'attente :

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


Utilisation du menu Channels

Pour sélectionner l'option Channels, tapez 3 dans la zone Enter Choice du menu principal MQM, et appuyez sur la touche Enter (F1). L'écran Search Criteria de canal s'affiche :

```
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
```

Figure 17. Ecran Search Criteria (canal)

Message Queue Management (MQM)

Dans la zone Channel Name, tapez un nom de canal complet ou partiel. Dans la zone Channel Type, entrez un chiffre compris entre 1 et 5 pour identifier le type de canal que vous souhaitez utiliser. Appuyez sur Enter key (F1) pour afficher le menu Channel :

```
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_SDRS_0001    RECEIVER
- MD01_MT01_RQSD_0001    REQUESTER
- MD01_MT01_RQSV_0001    REQUESTER
- MD01_MT01_SDRS_0001    RECEIVER
- MD01_MT01_SVRC_0001    RECEIVER
- ME02_MT01_RQSD_0001    REQUESTER
- ME02_MT01_RQSV_0001    REQUESTER
- ME02_MT01_SDRS_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
```

Figure 18. Menu Channel

Le menu Channel affiche une liste de canaux répondant à vos critères de recherche.

A partir du menu Channel, vous pouvez :

- Afficher et modifier le statut des canaux.
- Créer une nouvelle définition de canal.
- Copier une définition de canal.
- Supprimer une définition de canal.
- Contrôler le statut des canaux.
- Valider ou annuler un canal.
- Réinitialiser un numéro de séquence de message (MSN).
- Démarrer ou arrêter un canal.

Modification d'un canal

Dans le menu Channel, tapez tout caractère en face du canal que vous souhaitez modifier, et appuyez sur la touche Enter/Display/Modify (F1). L'écran correspondant s'affiche. Par exemple, si vous avez sélectionné un canal émetteur, l'écran Display/Modify Sender Channel s'affiche :

```

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
    
```

Figure 19. Ecran Display/Modify Sender Channel (1)

Appuyez sur la touche PGDN pour afficher le second écran d'information :

```

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
    
```

Figure 20. Ecran Display/Modify Sender Channel (2)

Réécrivez les valeurs que vous voulez modifier, et appuyez sur la touche Modify (F1). Vous n'avez pas accès aux valeurs dont la modification est interdite.

Message Queue Management (MQM)

Création d'une définition de processus

Depuis le menu Channel, appuyez sur la touche Create (F2) pour afficher l'écran Create Channel :

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

Channel Type: 1  1=Sender, 2 = Server, 3=Receiver,
                4=Requester, 5 = Server Connection
                6 = Cluster Sender, 7 = Cluster Receiver
Name: _____
Replace [Y/N]: _

F1 - Enter                                F16 - Return
```

Figure 21. Ecran Create Channel

Pour créer une nouvelle définition de canal :

1. Tapez 1 (pour un canal émetteur), 2 (pour un canal de serveur), 3 (pour un canal récepteur), 4 (pour un canal demandeur), ou 5 (pour une connexion serveur) dans la zone Channel Type.
2. Tapez le nom de la définition de canal dans la zone Name.
3. Appuyez sur la touche Enter (F1).
4. Tapez Y dans la zone Replace si la définition consiste à remplacer une définition existante de même nom et de même type.

Message Queue Management (MQM)

Si vous tapez 1 dans la zone Channel Type, l'écran Create Sender Channel s'affiche :

```
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
```

Figure 22. Ecran Create Sender Channel

Si vous tapez 3 dans la zone Channel Type, l'écran Create Receiver Channel s'affiche :

```
IBM MQSeries for Compaq NonStop Kernel Version 5.1
** Create Receiver Channel **
Channel Name      : SOLARIS_TO_Cmpaq___
Description      : _____
Put Authority D/C : _ NonPersistent Msg Speed [FAST/NORMAL]: FAST__
User Id          : _____ MCA UserID       : _____
Batch Size      : _____ 50 Max Message Size : 4194304
Msg Retry Count : 10 Msg Retry Interval  : 1000
Heartbeat Interval : 300 MSN Wrap Count   : 999999999
Transport Protocol : _ (1=Lu6.2/ 2=TCP/IP)
F1 - Enter      PGDN - Exits                      F16 - Return
```

Figure 23. Ecran Create Receiver Channel

Message Queue Management (MQM)

Si vous tapez 5 dans la zone Channel Type, l'écran Create Server Connection Channel s'affiche :

```
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
```

Figure 24. Ecran Create Server Connection Channel

Pour créer une nouvelle définition de canal, complétez l'écran de demande et appuyez sur la touche Enter (F1).

Si vous tapez 6 dans la zone Channel Type, l'écran Create Cluster Sender Channel s'affiche :

```
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
```

Figure 25. Ecran Create Cluster Sender Channel

Message Queue Management (MQM)

Si vous tapez 7 dans la zone Channel Type, l'écran Create Cluster Receiver Channel s'affiche :

```
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
```

Figure 26. Ecran Create Cluster Receiver Channel

Contrôle d'un canal

Appuyez sur la touche Monitor (F5) dans le menu Channel pour afficher l'écran Monitor Channels :

```
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.....
```

Figure 27. Ecran Monitor Channels

Le paramètre MQMQMREFRESHINT PATHWAY de MQS-MQMSVR00 détermine la fréquence de rafraîchissement des écrans de moniteur des canaux et files d'attente. La fréquence par défaut est de 30 secondes. Pour que le rafraîchissement

Message Queue Management (MQM)

s'effectue toutes les 10 secondes, par exemple, entrez la commande suivante dans PATHWAY pour votre gestionnaire de files d'attente :

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

Suppression d'un canal

Dans le menu Channel, sélectionnez le canal à supprimer en tapant tout caractère dans le nom de canal. Appuyez sur la touche Delete (F4), puis de nouveau sur F4 pour confirmer la suppression.

Affichage du statut des canaux

Appuyez sur la touche Status (F10) dans le menu Channel pour afficher l'écran Channel Status :

```
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
```

Figure 28. Ecran Channel Status

Démarrage et arrêt d'un canal

Appuyez sur la touche Start/Stop (F8) dans le menu Channel pour afficher l'écran Start/Stop Channel :

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

Name: MT01_MA02_SDRG_0001_
Status:
Action: _  choose one of the following:
          1. Start Channel
          2. Stop Immediate
          3. Stop Quiesce

F1 - Enter                                F16 - Return
```

Figure 29. Ecran Start/Stop Channel

Tapez le nom du canal dans la zone Nom, et un chiffre compris entre 1 et 3 dans la zone Action. Appuyez sur la touche Enter (F1).

Message Queue Management (MQM)

Réinitialisation d'un numéro de séquence de message (MSN)

Depuis le menu Channel, appuyez sur la touche Reset MSN (F7) pour afficher l'écran Reset Channel :

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

Name: MT01_M401_RQSD_0001_____
Sequence Number:      1

F1 - Enter                F16 - Return
```

Figure 30. Ecran Reset Channel

Le MSN apporte l'assurance que les messages n'apparaîtront pas en double, et qu'ils seront stockés dans l'ordre dans lequel ils ont été transmis. Cet écran permet de réinitialiser si nécessaire le numéro de séquence d'un canal.

Validation ou annulation d'un canal

Depuis le menu Channel, appuyez sur la touche Resolve (F6) pour afficher l'écran Resolve Channel :

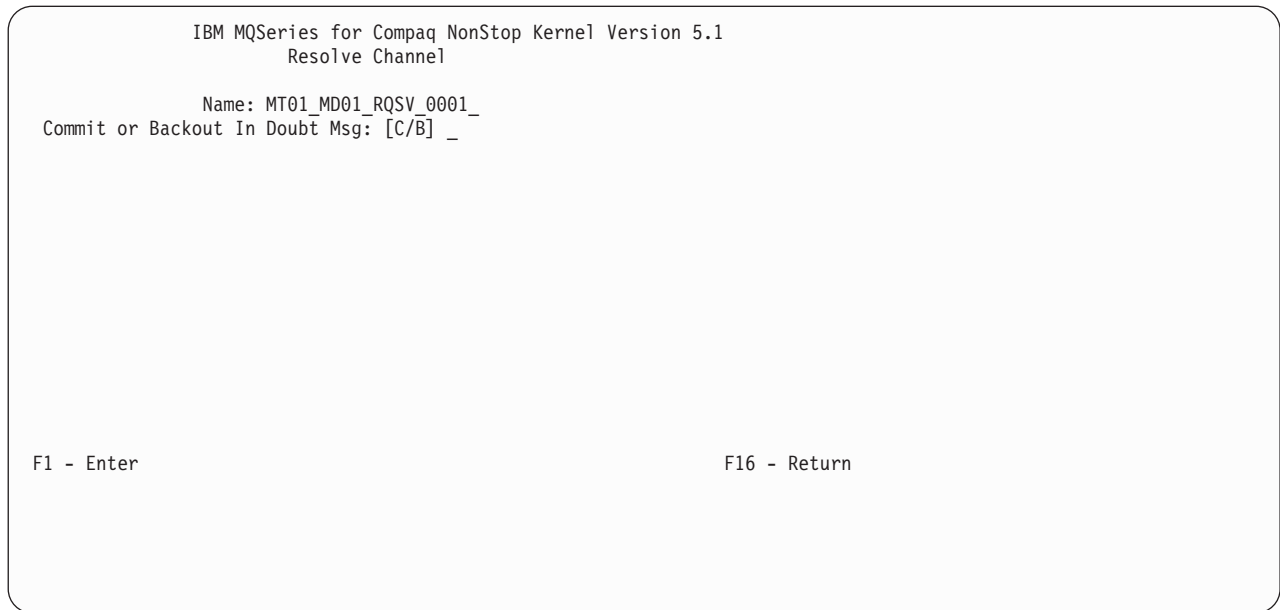


Figure 31. Ecran Resolve Channel

Vous pouvez :

- Annuler un lot de messages en attente de validation (B)
- Valider un lot de messages en attente de validation (C)

Message Queue Management (MQM)

Copie d'un canal

A partir du menu Channel, appuyez sur la touche Copy (F3). L'écran Copy Channel s'affiche :

```
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
```

Figure 32. Ecran Copy Channel

Tapez le nom du nouveau canal dans la zone Nom ; tapez le nom de la définition de canal copiée dans la zone Like Name ; tapez le type de canal dans la zone Channel Type. Appuyez sur la touche Enter (F1) pour copier la définition de canal.

Chapitre 5. Administration des objets MQSeries locaux

Le présent chapitre décrit comment administrer des objets MQSeries locaux pour prendre en charge les programmes d'application utilisant l'interface de mise en file d'attente des messages (MQI). Cette interface de programmation permet aux programmes d'application d'accéder aux fonctions de files d'attente de message.

L'administration locale correspond à la création, l'affichage, la modification, la copie et la suppression d'objets MQSeries.

Ce chapitre contient les sections suivantes :

- «Prise en charge des applications utilisant l'interface MQI»
- «Administration locale à l'aide de commandes MQSC» à la page 94
- «Exécution de commandes MQSC à partir de fichiers texte» à la page 98
- «Résolution des incidents à l'aide des commandes MQSC» à la page 101
- «Manipulation des files d'attente locales» à la page 103
- «Manipulation des files d'attente alias» à la page 111
- «Manipulation des files d'attente modèles» à la page 113
- «Gestion des objets pour le déclenchement» à la page 114

Prise en charge des applications utilisant l'interface MQI

Les programmes d'application MQI nécessitent certains objets pour pouvoir s'exécuter. Une application MQI peut extraire des messages d'une file d'attente, les traiter, puis envoyer les résultats à une autre file d'attente située sur le même gestionnaire.

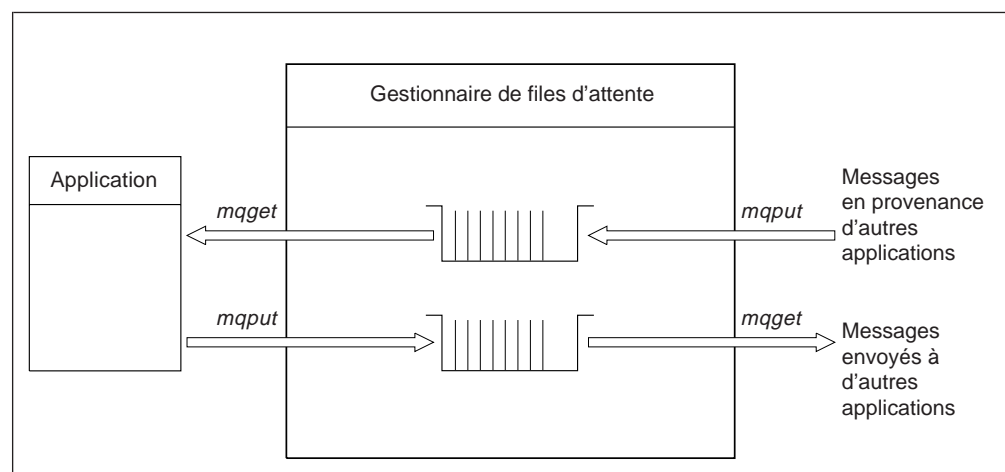


Figure 33. Files d'attente, messages et applications

Les applications peuvent insérer des messages dans des files d'attente locales ou éloignées (à l'aide de MQPUT), mais elles ne peuvent extraire des messages directement que des files d'attente locales (à l'aide de MQGET).

Avant que l'application puisse s'exécuter, les conditions suivantes doivent être remplies :

- Le gestionnaire de files d'attente doit exister et être actif.

Prise en charge d'applications

- La première file d'attente d'application, dans laquelle les messages doivent être récupérés, doit être définie.
- La seconde file d'attente, dans laquelle l'application insère les messages, doit également être définie (à moins qu'il s'agisse d'une file d'attente dynamique).
- L'application doit pouvoir se connecter au gestionnaire de files d'attente. Pour cela, elle doit être associée au code du produit. Pour plus d'informations, consultez la section «Annexe I. Construction et exécution d'applications» à la page 347.
- Les applications qui insèrent les messages dans la première file d'attente doivent également se connecter à un gestionnaire. S'il s'agit d'applications éloignées, leur configuration doit comporter des files d'attente de transmission et des canaux.

Administration locale à l'aide de commandes MQSC

Les commandes MQSeries (MQSC) permettent de manipuler les objets MQSeries. Les commandes se lancent depuis l'invite, à l'aide de la commande **runmqsc**.

Pour plus d'informations concernant l'utilisation de MQSC dans un environnement MQSeries pour Compaq NSK, reportez-vous à l'«Annexe G. Commandes MQSC prises en charge par MQSeries pour Compaq NSK» à la page 335.

Les commandes de script MQSeries (MQSC) permettent de gérer les objets d'un gestionnaire de files d'attente, y compris le gestionnaire lui-même, les clusters, canaux, files d'attente, listes de noms et définitions de processus. Cette section traite des gestionnaires de files d'attente, des files d'attente et des définitions de processus ; pour plus de détails sur l'administration des objets canal, reportez-vous à la section DQM implementation du manuel *MQSeries Intercommunication*.

Utilisez la commande **runmqsc** pour émettre des commandes MQSC vers un gestionnaire de files d'attente. Vous pouvez travailler de façon interactive en entrant les commandes au clavier, ou bien rediriger l'unité d'entrée standard pour exécuter une séquence de commandes à partir d'un fichier texte ASCII. Dans les deux cas, le format des commandes est le même.

Il existe trois modes d'exécution de la commande **runmqsc** en fonction des indicateurs définis :

- *Mode vérification* : les commandes MQSC sont vérifiées sur un gestionnaire de files d'attente local, mais ne s'exécutent pas réellement.
- *Mode direct* : les commandes MQSC s'exécutent sur un gestionnaire de files d'attente local.
- *Mode indirect* : les commandes MQSC s'exécutent sur un gestionnaire de files d'attente éloigné.

Dans ce manuel, les attributs d'objet indiqués dans les commandes MQSC sont écrits en majuscules (par exemple, RQMNAME), bien que la distinction majuscules/minuscules ne s'applique pas. Ces noms d'attribut sont limités à huit caractères.

Préambule

Avant de pouvoir exécuter des commandes MQSC, vous devez créer et démarrer le gestionnaire de files d'attente qui exécutera les commandes. Pour plus d'informations, reportez-vous à la section «Création d'un gestionnaire de files d'attente par défaut» à la page 65.

Noms des objets MQSeries

Dans les exemples, les objets sont désignés par leur nom complet afin de vous aider à identifier le type d'objet avec lequel vous travaillez.

Lorsque vous lancez des commandes MQSC, il suffit d'indiquer le nom local de la file d'attente. Dans les exemples, nous utilisons des noms de file d'attente tels que : FILE.LOCALE.ORANGE

La partie FILE.LOCALE permet tout simplement d'indiquer qu'il s'agit d'une file d'attente locale. Dans l'usage courant, elle *n'est absolument pas* obligatoire pour les noms de file.

De même, nous employons le nom saturne.gest.files pour désigner un gestionnaire de files d'attente.

La partie gest.files permet tout simplement d'indiquer que l'objet en question est un gestionnaire de files d'attente. Dans l'usage courant, elle *n'est absolument pas* obligatoire pour les noms de file.

Vous n'êtes pas obligé d'utiliser ces noms mais, dans ce cas, vous devez apporter les modifications correspondantes aux commandes qui contiennent ce type de nom dans les exemples.

Utilisation des majuscules et minuscules dans les commandes MQSC

Les commandes MQSC et leurs attributs peuvent être indifféremment en majuscules ou en minuscules. Les noms d'objet sont par contre sensibles à la casse (file-entrée n'est pas le même objet que FILE-ENTREE). Les noms d'objets figurant dans les commandes MQSC sont automatiquement convertis en majuscules (file et FILE ne sont pas différenciés) sauf s'ils sont placés entre apostrophes. Si l'on n'utilise pas d'apostrophes, les nom de l'objet est converti en majuscules. Pour plus d'informations, reportez-vous au manuel *MQSeries - Guide de référence des commandes MQSC*.

Certains arguments de la commande `runmqsc` qui appelle la fonction MQSC sont cependant sensibles à la casse ; voir «Utilisation des commandes de contrôle» à la page 29.

Utilisation de la fonction MQSC de façon interactive

Pour exécuter des commandes de façon interactive, entrez le texte suivant dans une session TACL :

```
runmqsc
```

Aucun nom de gestionnaire de files d'attente n'a été indiqué dans cette commande. Par conséquent, les commandes MQSCs seront traitées par le gestionnaire par défaut. Vous pouvez entrer n'importe quelle commande MQSC. Par exemple :

```
MQSD>EFINE QLOCAL (FILE.LOCALE.ORANGE)
```

Pour indiquer qu'une commande doit se poursuivre sur la ligne suivante, utilisez un caractère de continuation.

Emission de commandes MQSC

- le signe moins indique que la commande doit se poursuivre au tout début de la ligne suivante.
- le signe plus indique qu'elle doit se poursuivre à partir du premier caractère non blanc de la ligne suivante.

Une commande se termine normalement au dernier caractère d'une ligne non vide qui ne contient pas de caractère de suite.

La commande **runmqsc** supporte aussi les commandes standard Compaq NSK : historique et Fix. Par exemple :

- tapez `history` ou `h` pour obtenir la liste des dix dernières commandes effectuées
- tapez `!n` où `n` est le numéro de la commande pour réexécuter cette commande
- tapez `h n` ou `history n` où `n` est un numéro pour obtenir la liste des `n` dernières commandes effectuées
- tapez `fc` pour rappeler la dernière commande en mode édition. Tapez `fc n` où `n` est le numéro de commande pour rappeler la commande `n` en mode édition. Tapez `fc string` où `string` est le début ou la commande entière entrée précédemment pour rappelr la dernière occurrence de cette commande en mode édition. La syntaxe est conforme au standard NSK . Par exemple, tapez `d` pour effacer un caractère, `i` pour insérer et `r` pour remplacer un caractère.

Informations renvoyées par les commandes MQSC

Lorsque vous lancez des commandes à partir de la fonction MQSC, le gestionnaire de files d'attente renvoie des messages opérateur pour vous informer du résultat de l'opération ou des erreurs que vous avez faites. Par exemple :

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

Le premier message vous informe que la file d'attente a bien été créée, alors que le second vous indique que vous avez fait une erreur de syntaxe. Si vous n'avez pas entré la commande correctement, consultez le manuel *MQSeries - Guide de référence des commandes MQSC* pour connaître la syntaxe correcte.

Fin de l'entrée interactive sur MQSC

Si vous utilisez MQSC de façon interactive, vous pouvez quitter ce mode en entrant le caractère EOF (CTRL+Y), ou en tapant `exit`, `quit` ou `end` puis en appuyant sur Entrée.

Si vos données en entrée proviennent d'une autre source, telle qu'un fichier texte, MQSC ferme le fichier une fois arrivé à la fin.

Affichage des attributs d'un gestionnaire de files d'attente

Pour afficher les attributs du gestionnaire de files d'attente indiqué dans la commande **runmqsc**, utilisez la commande MQSeries suivante :

```
MQSC>DISPLAY QMGR ALL
```


Vous obtenez une sortie identique à celle présentée à la figure 34.:

```

1 : dis qmgr all
AMQ8408: Détails sur l'affichage du gestionnaire de files d'attente.
DESCR( )                DEADQ(SYSTEM.DEAD.LETTER.QUEUE)
DEFXMITQ(MY.DEFAULT.XMIT.QUEUE)  CHADEXIT( )
CLWLEXIT( )            CLWLDATA( )
REPOS( )              REPOSNL( )
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)  QMNAME(saturne.gest.files)
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)
CLWLLEN(100)        MAXMSGL(100000000)
CCSID(819)          MAXPRTY(9)
CMDLEVEL(510)      PLATFORM(NSK)
SYNCPT              DISTL(YES)

```

Figure 34. Exemple de sortie pour QMGR ALL

Le paramètre ALL de la commande DISPLAY QMGR permet d'afficher tous les attributs du gestionnaire de files d'attente. Cette sortie vous donne notamment le nom du gestionnaire (saturne.gest.files), et les noms de file de rebut (SYSTEM.DEAD.LETTER.QUEUE) et de la file de commandes (SYSTEM.ADMIN.COMMAND.QUEUE). Remarquez que si vous ne précisez pas de nom de file d'attente de rebut à la commande **crtmqm**, il faut modifier le gestionnaire de files d'attente pour lui associer une file d'attente de rebut.

Vous devez confirmer la création de ces files d'attente par la commande :

```
DISPLAY QUEUE (*)
```

Utilisation d'un gestionnaire différent du gestionnaire par défaut

Pour exécuter les commandes MQSC sur un gestionnaire de files d'attente local différent du gestionnaire par défaut, vous pouvez spécifier son nom à la suite de la commande **runmqsc**. Par exemple, si vous voulez utiliser le gestionnaire nommé **jupiter.gest.files**, utilisez la commande :

```
runmqsc jupiter.gest.files
```

Toutes les commandes MQSCs que vous entrerez ensuite seront traitées par le gestionnaire de files d'attente indiqué, à condition qu'il se trouve sur le même noeud et qu'il soit actif.

Vous pouvez également exécuter des commandes MQSC sur un gestionnaire éloigné. Reportez-vous à la section «Emission de commandes MQSC à distance» à la page 129.

Emission de commandes MQSC

Modification des attributs d'un gestionnaire de files d'attente

Pour modifier les attributs du gestionnaire de files d'attente indiqués dans la commande **runmqsc**, utilisez la commande MQSC ALTER QMGR, en spécifiant les attributs et les valeurs que vous souhaitez modifier. Par exemple, utilisez les commandes suivantes pour modifier les attributs du gestionnaire `jupiter.gest.files` :

```
runmqsc jupiter.gest.files
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

La commande ALTER QMGR modifie la file d'attente de rebut utilisée et active les événements Interdiction.

Exécution de commandes MQSC à partir de fichiers texte

L'exécution de commandes MQSCs de façon interactive convient pour effectuer des tests rapides. Cependant, si vous devez lancer des commandes très longues, ou une suite de commandes fréquemment utilisée, il est préférable de rediriger la lecture depuis un fichier texte

Pour ce faire, créez un fichier texte contenant les commandes MQSC, à l'aide de votre éditeur de texte. Lors de l'utilisation de la commande **runmqsc**, utilisez les opérateurs de redirection TACL IN et OUT, ou les indicateurs `-i` et `-o` sous **runmqsc**. Par exemple, l'instruction suivante exécute une séquence de commandes qui se trouvent dans le fichier `monmqscin` :

```
runmqsc /IN monmqscin/
```

ou

```
runmqsc -i monmqscin
```

Vous pouvez également rediriger les données en sortie vers un fichier. Un fichier qui contient les commandes MQSCs d'entrée est appelé un *fichier de commandes MQSC*. Le fichier de sortie contenant les réponses du gestionnaire de files d'attente est appelé le *fichier de rapport*.

Pour rediriger à la fois INPUT et OUTPUT dans la commande **runmqsc**, utilisez la commande suivante :

```
runmqsc /IN monmqscin, OUT monmqscou/
```

ou

```
runmqsc -i monmqscin -o monmqscou
```

Exécution de commandes MQSC

Cette instruction appelle les commandes MQSC contenues dans le fichier de commandes MQSC monmqscin. Comme vous n'avez pas indiqué de gestionnaire de files d'attente, les commandes MQSC sont exécutées sur le gestionnaire par défaut. Les données en sortie sont envoyées vers le fichier de rapport monmqscou. La figure 35 montre un extrait du fichier de commandes MQSC monmqscin, et la figure 36 à la page 100 montre la sortie correspondante extraite du fichier monmqscou.

La redirection des INPUT et OUTPUT, pour une commande **runmqsc** et un gestionnaire de files d'attente (`saturne.gest.files`) qui n'est pas celui par défaut, se fait par la commande ci-dessous :

```
runmqsc /IN monmqscin, OUT monmqscou/ saturne.gest.files
```

ou

```
runmqsc -i monmqscin -o monmqscou saturne.gest.files
```

Utilisation des fichiers de commandes MQSC

Les fichiers de commandes MQSC sont enregistrés comme des fichiers EDIT (code de type de fichier Compaq 101). La figure 35 présente un extrait d'un fichier de commandes MQSeries montrant la commande **DEFINE QLOCAL** avec ses attributs. Le manuel *MQSeries - Guide de référence des commandes MQSC* contient une description de chaque commande MQSC et de sa syntaxe.

```
.  
. .  
. .  
DEFINE QLOCAL(FILE.LOCALE.ORANGE) REPLACE +  
  DESCR(' ') +  
  PUT (ENABLED) +  
  DEFPRTY(0) +  
  DEFPSIST(NO) +  
  GET (ENABLED) +  
  MAXDEPTH(5000) +  
  MAXMSGL(1024) +  
  DEFSOPT(SHARED) +  
  NOHARDENBO +  
  USAGE (NORMAL) +  
  NOTRIGGER  
. . .
```

Figure 35. Extrait du fichier de commandes MQSC monmqscin

Vous devez limiter à 72 caractères la longueur des lignes. Le signe plus indique que la commande se poursuit sur la ligne suivante. Remarquez que ce signe doit être précédé d'un espace.

Exécution de commandes MQSC

Utilisation des rapports MQSC

La commande `runmqsc` génère un *rapport*, qui est envoyé au fichier OUT courant. Ce rapport contient :

- un en-tête indiquant que MQSC est à l'origine du rapport :

```
Starting MQSeries Commands.
```

- une liste numérotée des commandes MQSC émises (en option). Par défaut, le texte en entrée est recopié dans celui en sortie. Dans le rapport, chaque commande est précédée d'un numéro de séquence, comme le montre la figure 36. Cependant, vous pouvez utiliser l'indicateur `-e` dans la commande `runmqsc` pour supprimer cette sortie.
- un message d'erreur pour toutes les commandes dont la syntaxe est erronée.
- un *message opérateur* indiquant le résultat de l'exécution de chaque commande. Par exemple, lorsque la commande `DEFINE QLOCAL` aboutit, le message opérateur envoyé est le suivant :

```
AMQ8006: MQSeries queue created.
```

- d'autres messages provenant des erreurs générales détectées au cours de l'exécution du fichier script.
- un récapitulatif statistique du rapport, indiquant le nombre de commandes lues, le nombre de commandes comportant des erreurs de syntaxe et le nombre de commandes qui n'ont pas pu être traitées.

Remarque : Le gestionnaire de files d'attente ne tente de traiter que les commandes dépourvues d'erreur de syntaxe.

```
Starting MQSeries Commands.
.
.
12:   DEFINE QLOCAL('ROUGE.FILE.LOCALE') REPLACE +
:     DESCR(' ') +
:     PUT(ENABLED) +
:     DEFPRTY(0) +
:     DEFPSIST(NO) +
:     GET(ENABLED) +
:     MAXDEPTH(5000) +
:     MAXMSGL(1024) +
:     DEFSOPT(SHARED) +
:     USAGE(NORMAL) +
:     NOTRIGGER
AMQ8006: MQSeries queue created.
:
.
.
15 MQSC commands read.
0 commands have a syntax error.
0 commands cannot be processed.
```

Figure 36. Extrait du fichier de rapport MQSC `monmqscou`

Exécution du fichier de commandes MQSC fourni

Lors de l'installation de MQSeries pour Compaq NSK, un fichier MQSC appelé **AMQSCOS0** est fourni. Il contient les définitions des objets utilisés par les programmes exemples. Ce fichier se trouve dans le sous-volume des exemples, par défaut **MQS_EXAMPLES** :

Utilisation de `runmqsc` pour vérifier des commandes

Vous pouvez utiliser la commande **runmqsc** pour vérifier les commandes MQSC sur un gestionnaire de files d'attente par défaut local, sans avoir à les exécuter réellement. Pour cela, ajoutez l'indicateur `-v` à la commande **runmqsc**. Par exemple :

```
runmqsc -i monmqscin -o monmqscou -v
```

Lorsque vous lancez **runmqsc** sur un fichier de commandes MQSC, le gestionnaire de files d'attente vérifie chaque commande MQSC sans l'exécuter, et génère un rapport. Vous pouvez ainsi vérifier la syntaxe de toutes les commandes de votre fichier. Cette étape est importante si vous :

- exécutez un grand nombre de commandes MQSC à partir d'un fichier de commandes
- utilisez un fichier de commandes MQSC fréquemment.

Le rapport envoyé est du même type que celui montré dans la figure 36 à la page 100.

Vous ne pouvez pas employer cette méthode pour vérifier des commandes MQSC à distance. Par exemple, si vous lancez cette commande :

```
runmqsc -i monmqscin -o monmqscou -w 30 -v jupiter.gest.files
```

l'indicateur `-w`, (qui sert à signaler que le gestionnaire de files d'attente est éloigné) est ignoré, et la commande est exécutée en local en mode vérification.

Résolution des incidents à l'aide des commandes MQSC

Si vous ne parvenez pas à exécuter des commandes MQSC, consultez la liste suivante pour voir si cet échec est dû à l'un des incidents répertoriés.

Lorsque vous utilisez la commande **runmqsc** :

- vérifiez que `$SYSTEM.ZMQSEXE` est en `PMSEARCH` dans `TACL CSTM`.
- utilisez l'opérateur `IN` ou l'indicateur `-i` pour rediriger la lecture des données à partir d'un fichier. Si vous l'omettez, le gestionnaire de files d'attente interprète le nom du fichier comme celui d'un gestionnaire, et émet le message d'erreur suivant :

```
AMQ8118: MQSeries queue manager does not exist.
```

- Si vous redirigez les données de sortie vers un fichier, utilisez l'opérateur `OUT` ou l'indicateur `-o`. Par défaut, le fichier de sortie est créé à l'aide des `TACL` en

Incidents avec MQSC

vigueur au moment de l'émission de la commande. Indiquez un nom de fichier complet pour envoyer vos données en sortie vers un fichier spécifique.

- Vérifiez que vous avez bien créé le gestionnaire de files d'attente qui doit exécuter les commandes. Pour cela, consultez le fichier de configuration MQS.ini, situé par défaut dans le sous-volume d'installation \$SYSTEM.ZMQSSYS. Il contient le nom des gestionnaires de files d'attente et celui du gestionnaire par défaut, si vous en avez un.
- Le gestionnaire doit déjà être actif, si ce n'est pas le cas, démarrez-le en suivant la description de la section «Lancement d'un gestionnaire de files d'attente» à la page 67. Si vous tentez de le lancer alors qu'il est déjà en cours d'exécution, vous obtiendrez un message d'erreur.
- Si vous n'avez défini aucun gestionnaire de files d'attente par défaut, indiquez un nom de gestionnaire dans la commande **runmqsc**, sinon vous obtiendrez ce message d'erreur :

```
AMQ8146: MQSeries queue manager not available.
```

Pour remédier à ce type d'incident, reportez-vous à la section «Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant» à la page 69.

- Il n'est pas possible de spécifier une commande MQSC comme paramètre de **runmqsc**. Exemple de commande non valide :

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

- Vous ne pouvez pas entrer de commandes MQSC depuis TACL avant d'avoir lancé **runmqsc**. Par exemple :

```
DEFINE QLOCAL(Queue1)
```

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

- Il n'est pas possible d'exécuter des commandes de contrôle à partir de **runmqsc**. Par exemple, vous ne pouvez pas démarrer un gestionnaire de files d'attente lorsque vous utilisez MQSC de façon interactive :

```
runmqsc
(C) Copyright IBM Corp. 1993, 2001. All Rights Reserved
Starting MQSeries Commands.

strmqm saturne.gest.files
  1 : strmqm saturne.gest.files
AMQ8405: Syntax error detected at or near end of command segment below:
```

Consultez aussi la section «En cas d'incident» à la page 131.

Manipulation des files d'attente locales

Cette section contient des exemples de certaines commandes MQSC que vous pouvez utiliser. Consultez le manuel *MQSeries - Guide de référence des commandes MQSC* pour une description complète de ces commandes.

Définition d'une file d'attente locale

Le gestionnaire de files d'attente local d'une application est le gestionnaire auquel cette application est connectée. Les files d'attente gérées par le gestionnaire local sont considérées comme locales à ce gestionnaire.

Pour créer la définition d'une file d'attente locale ainsi que la structure de données appelée file d'attente, utilisez la commande MQSC `DEFINE QLOCAL`. Vous pouvez également modifier les caractéristiques de la file à partir de celles de la file d'attente locale par défaut.

Dans l'exemple ci-dessous, la file que nous définissons, `FILE.LOCALE.ORANGE`, a les caractéristiques suivantes :

- L'extraction de messages est activée, l'insertion est désactivée, et la file d'attente utilise la méthode FIFO (premier entré, premier sorti).
- Il s'agit d'une file 'ordinaire'. C'est-à-dire qu'elle ne sert ni à l'initialisation ni à la transmission, et qu'elle ne peut pas générer de messages de déclenchement.
- La capacité maximale de la file est 1000 messages et la taille maximale de chaque message est 2000 octets.

La commande suivante effectue cette action :

```
DEFINE QLOCAL (FILE.LOCALE.ORANGE) +
  DESCR('Queue for messages from other systems') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL)
```

Remarques :

1. La plupart de ces attributs sont les options par défaut fournies avec le produit. Cependant, nous les avons indiqués ici dans un but d'illustration. Vous pouvez donc les omettre si vous êtes sûr que les options par défaut vous conviennent ou n'ont pas été modifiées.
2. L'attribut `USAGE (NORMAL)` indique que la file d'attente n'est pas une file de transmission.
3. Si vous disposez déjà d'une file d'attente locale nommée `FILE.LOCALE.ORANGE` sur le même gestionnaire, la commande échouera. Utilisez l'attribut `REPLACE` si vous voulez écraser la définition existante d'une file d'attente, mais consultez également la section «Modification des attributs d'une file d'attente locale» à la page 106.

Modification de la taille du fichier physique des files d'attente

Par défaut, le gestionnaire de files d'attente crée des fichiers de données de files d'attente supportant jusqu'à 100 Mo de données. Si cette limite est atteinte, le code

Manipulation des files d'attente locales

de retour MQRC_Q_SPACE_NOT_AVAILABLE apparaît. Pour modifier la capacité de stockage d'une file, commencez par repérer les fichiers physiques qui contiennent les données pour les files d'attente à l'aide de **dspmqls**, puis utilisez la commande **altmqfls** pour redimensionner la taille des domaines primaires et secondaires et les domaines maximums. Si le fichier est déjà segmenté, **altmqfls** redimensionnera tous les segments. Pour plus d'informations, reportez-vous au «altmqfls (modification des attributs d'un fichier de file d'attente)» à la page 254. S'il vous faut plus d'espace, il est possible de segmenter le fichier sur plusieurs volumes.

Définition d'une file d'attente de rebut

Chaque gestionnaire de files d'attente doit avoir une file locale qui sert à stocker les messages qui n'ont pas pu être transmis vers leur cible, en attendant d'en être extraits.

Vous devez attribuer explicitement une file d'attente de rebut au gestionnaire. Pour ce faire, indiquez son nom dans la commande **crtmqm** ou spécifiez-en une ultérieurement à l'aide de la commande ALTER QMGR. Il est indispensable de définir la file d'attente de rebut avant de pouvoir l'utiliser.

Une file de rebut exemple nommée SYSTEM.DEAD.LETTER.QUEUE est fournie avec le produit. Elle est créée par défaut lors de la création d'un gestionnaire de files d'attente. Vous pouvez en modifier la définition si nécessaire, mais il est inutile de la renommer.

Une file d'attente de rebut n'a pas besoin de remplir de conditions particulières, à deux exceptions près

- Il doit s'agir d'une file locale.
- L'attribut MAXMSGL (taille maximale des messages) doit lui permettre de recevoir les messages les plus longs qui passent par le gestionnaire **plus** la taille de l'en-tête de rebut (MQDLH).

MQSeries fournit un gestionnaire de files d'attente de rebut qui vous permet de définir la façon dont les messages qui se trouvent dans une file de rebut doivent être traités ou supprimés. Pour plus d'informations, reportez-vous au «Chapitre 9. Gestionnaire de file d'attente de rebut MQSeries» à la page 161.

Affichage des attributs d'objet par défaut

Lorsque vous définissez un objet MQSeries, la valeur des attributs que vous ne spécifiez pas est récupérée dans l'objet par défaut. Par exemple, si vous définissez une file d'attente locale, les attributs que vous avez omis dans sa définition prennent les valeurs de la file locale par défaut, nommée SYSTEM.DEFAULT.LOCAL.QUEUE. Pour avoir des informations précises sur ces attributs, utilisez la commande suivante :

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Remarque : La syntaxe de cette commande diffère de celle de la commande DEFINE correspondante.

Affichage des attributs d'objet par défaut

Vous pouvez également choisir les attributs à afficher en les spécifiant individuellement. Par exemple :

```
DISPLAY QUEUE (FILE.LOCALE.ORANGE) +  
    MAXDEPTH +  
    MAXMSGL +  
    CURDEPTH
```

Cette commande affiche les trois attributs indiqués, comme suit :

```
AMQ8409: Display Queue details.  
QUEUE(FILE.LOCALE.ORANGE)  
MAXDEPTH (1000)  
MAXMSGL(2000)  
CURDEPTH(0)
```

CURDEPTH représente la longueur actuelle de la file d'attente, c'est-à-dire le nombre de messages qu'elle contient. Il est utile de connaître la valeur de cet attribut, car en contrôlant la longueur de la file, vous pouvez veiller à ce qu'elle n'arrive pas à saturation.

Copie d'une définition de file d'attente locale

Vous pouvez copier la définition d'une file d'attente en utilisant l'attribut LIKE de la commande DEFINE. Par exemple :

```
DEFINE QLOCAL (FILE.MAGENTA) +  
    LIKE (FILE.LOCALE.ORANGE)
```

Cette commande crée une file d'attente ayant les mêmes attributs que FILE.LOCALE.ORANGE, et non ceux de la file locale par défaut du système.

Vous pouvez également employer la syntaxe suivante de la commande DEFINE pour copier une définition en apportant une ou plusieurs modifications aux attributs d'origine. Par exemple :

```
DEFINE QLOCAL (FILE.TROIS) +  
    LIKE (FILE.LOCALE.ORANGE) +  
    MAXMSGL(1024)
```

Cette commande copie les attributs de la file d'attente FILE.LOCALE.ORANGE dans FILE.TROIS, mais la taille maximale des messages pour la nouvelle file est 1024 octets au lieu de 2000.

Remarques :

1. Lorsque vous utilisez l'attribut LIKE dans la commande DEFINE, vous copiez uniquement les attributs de la file d'attente, et non les messages qui se trouvent dans cette file.
2. Si vous définissez une file d'attente locale sans indiquer LIKE, votre commande est identique à DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE).

modification des attributs de file d'attente locale

Modification des attributs d'une file d'attente locale

Il existe deux méthodes pour modifier les attributs d'une file d'attente : la commande ALTER QLOCAL et la commande DEFINE QLOCAL avec l'option REPLACE. A la section «Définition d'une file d'attente locale» à la page 103, nous avons défini la file d'attente FILE.LOCALE.ORANGE. Supposons, par exemple, que vous vouliez augmenter la taille maximale des messages de cette file à 10 000 octets.

- A l'aide de la commande ALTER :

```
ALTER QLOCAL (FILE.LOCALE.ORANGE) MAXMSGL(10000)
```

Cette commande ne modifie qu'un seul attribut, la taille maximale des messages ; tous les autres attributs gardent leur valeur antérieure.

- Avec la commande DEFINE et l'option REPLACE :

```
DEFINE QLOCAL (FILE.LOCALE.ORANGE) MAXMSGL(10000) REPLACE
```

Cette commande ne modifie pas seulement la taille maximale des messages, elle donne aussi à tous les autres attributs leur valeur par défaut. Ainsi, l'insertion de messages est désormais autorisée, alors que cette option était désactivée auparavant. L'autorisation d'insérer des messages (PUT(ENABLED)) est l'option par défaut, comme indiqué dans la définition de SYSTEM.DEFAULT.LOCAL.QUEUE, sauf si vous avez modifié cette option.

Si vous *diminuez* la taille maximale des messages d'une file d'attente existante, cette modification n'a aucune répercussion sur les messages existants. En revanche, tous les nouveaux messages doivent répondre à ce critère.

Modification du volume d'une file d'attente locale

Utilisez la commande **altmqfls** pour changer le volume sur lequel une file locale prédéfinie est stockée. Cela peut être nécessaire pour répartir les E/S sur les différents volumes pour équilibrer le système et obtenir des performances optimales. Le gestionnaire de files d'attente doit être ouvert avant le lancement de cette commande et la file d'attente doit être fermée. Une seule file d'attente peut être nommée sur une commande **altmqfls**. Pour plus d'informations concernant la syntaxe de cette commande, reportez-vous à la section «altmqfls (modification des attributs d'un fichier de file d'attente)» à la page 254.

Modification des options d'une file d'attente locale

Utilisez la commande **altmqfls** pour :

- charger la file du disque dur vers le cache, au démarrage du gestionnaire de files d'attente. Utilisez --qsoptions S.
- conserver en mémoire la file durant le fonctionnement du serveur de file d'attente, après son premier chargement. S'il a été configuré pour cela, le serveur de file d'attente conserve les structures de la file d'attente et toutes les données mises en mémoire une fois qu'il est chargé. Dans le cas contraire, l'emplacement de stockage associé à une file peut être supprimé de l'espace adresse du serveur de file d'attente une fois qu'il n'est plus utilisé. Utilisez --qsoptions L.
- que les messages non-persistents soient contrôlés par le serveur de file d'attente de secours. Ce système assure la tolérance de panne, au prix d'une

Recalcul, mise à jour, et récupération

certaine charge du processeur pour la gestion des points de contrôle supplémentaires, des messages IPC et de la mémoire nécessaire pour stocker les messages. Utilisez `--qsoptions C`.

Remarque : *Toutes* les `--qsoptions SLC` sont modifiées à chaque utilisation de la commande. Par exemple, `--qsoptions S` désactivera L et C. L'option `--qsoptions` ne peut être utilisée qu'une seule fois sur une ligne de commande.

- modifier le nombre maximum d'octets de données que chaque message persistant peut garder dans le cache du serveur de file d'attente (et sur le disque dur). Le nombre d'octets défini est gardé en mémoire et l'opération de récupération renvoie ces données vers l'application sans devoir passer par le disque dur. L'utilisation de ce paramètre accroîtra les ressources mémoire nécessaires au serveur de file d'attente. Utilisez le paramètre `--browse`.
- indiquer la taille minimale de message, lors de l'utilisation d'un fichier de dépassement de capacité de message pour stocker les données du message. Les messages persistants de taille inférieure à ce seuil sont enregistrés dans le fichier de dépassement de capacité de la file d'attente. Les données non formatées des messages persistants de taille supérieure ou égale à ce seuil seront enregistrées dans un fichier de dépassement de capacité spécifique. Utilisez le paramètre `--msgofthresh`.
- modifier le sous-volume du volume où réside la file d'attente sur laquelle le serveur de file d'attente crée de nouveaux fichiers de dépassement de messages. Utilisez le paramètre `--msgofsubvol`.
- modifier le nom d'un compteur mesure. S'il fait partie d'un système de mesure activé, ce compteur est initialisé à la profondeur en cours, puis incrémenté et décrémenté par le serveur responsable de la file d'attente, au fur et à mesure que les messages sont ajoutés ou retirés. Utilisez le paramètre `--meascount`.
- modifier la taille des domaines primaires et secondaires et le nombre maximum de domaines pour le fichier d'attente. Utilisez le paramètre `--qsize` (`primaryextent,secondaryextent,maxextents`). Avec le paramètre `--qsize`, toutes les valeurs doivent être définies.
- modifier la taille des domaines primaires et secondaires et le nombre maximum de domaines pour le fichier de dépassement de capacité de file d'attente. Utilisez le paramètre `--osize` (`domaineprimaire, domainesecondaire, domainesmax`). Avec le paramètre `--osize`, toutes les valeurs doivent être définies.

Reassignation d'objets aux serveurs de statut et de file d'attente

Les serveurs de statut traitent tous les objets sauf les files d'attente modèles et les files locales. Les serveurs de file d'attente traitent tous les objets des files d'attente modèles et locales.

Au départ, tous les objets sont créés pour utiliser le serveur de file d'attente ou le serveur de statut par défaut, selon l'objet. Une fois l'objet créé, utilisez **altmqfls** pour configurer l'objet pour qu'il utilise un serveur approprié plutôt que le serveur par défaut. Il faut que la nouvelle classe de serveur PATHWAY soit configurée pour ce serveur et qu'il soit démarré avant d'utiliser l'objet.

Vous pouvez définir un nom de processus ou le mot DEFAULT sur la ligne de commande pour **altmqfls**. Le système ne vérifie pas que le nouveau serveur est actif ou configuré au moment de la reconfiguration de l'objet.

Utilisez la commande **dspmqls** pour afficher le status server actuel d'un objet.

Reassignation d'objets

Remarquez que le serveur de file d'attente peut être configuré pour des files d'attente modèles ou locales. Les files dynamiques sont gérées par le serveur de file d'attente de la file d'attente modèle utilisée pour leur création.

Nettoyage d'une file d'attente locale

Pour supprimer tous les messages d'une file d'attente locale nommée FILE.MAGENTA, utilisez la commande suivante :

```
CLEAR QLOCAL (FILE.MAGENTA)
```

Cette opération est impossible dans les cas suivants :

- des messages non validés ont été mis dans la file d'attente lors d'un point de synchronisation.
- la file d'attente est ouverte pour une application.

Suppression d'une file d'attente locale

Pour supprimer une file d'attente locale, utilisez la commande MQSC DELETE QLOCAL. Il n'est pas possible de supprimer une file si elle contient des messages non validés. Si elle contient uniquement des messages validés, vous ne pouvez la supprimer qu'en indiquant l'option PURGE. Par exemple :

```
DELETE QLOCAL (FILE.ROSE) PURGE
```

Si vous indiquez NOPURGE au lieu de PURGE, la file d'attente ne sera pas supprimée si elle contient des messages validés.

Visualisation des files d'attente

MQSeries pour Compaq NSK fournit un afficheur dont vous pouvez vous servir pour examiner le contenu des messages d'une file d'attente. Cet afficheur est proposé sous la forme d'un code source et d'un exécutable. Par défaut, les noms de fichier et les chemins d'accès par défaut sont :

Source \$SYSTEM.ZMQSSMPL.AMQSBCG0

Exécutable \$SYSTEM.ZMQSSMPL.AMQSBCG

L'afficheur requiert deux paramètres :

Nom de file d'attente

Par exemple, SYSTEM.ADMIN.RESPQ.tpp01.

Nom du GFA Par exemple, snooker.

Par exemple :

```
AMQSBCG SYSTEM.ADMIN.RESPQ.tpp01 snooker
```


Visualisation des files d'attente

```
MsgId : X'414D5120736E6F6F6B65722020202020ED476901524D200'  
CorrelId : X'414D5120736E6F6F6B65722020202020ED47690071A6D00'  
BackoutCount : 0  
ReplyToQ      : '  
ReplyToQMGr   : 'snooker'  
** Identity Context  
UserIdentifier : 'tpp01'  
AccountingToken :  
X'04373037300000000000000000000000000000000000000000000000000000000000'  
ApplIdentityData : '  
** Origin Context  
PutApplType   : '6'  
PutApplName   : '  
PutDate       : '19941124'   PutTime : '11184035'  
ApplOriginData : ' '
```

**** Message ****

length - 36 bytes

```
00000000: 0000 0002 0000 0024 0000 0001 0000 0015 '.....$.....'  
00000010: 0000 0001 0000 0001 0000 0000 0000 0000 '.....'  
00000020: 0000 0000                                     '....'
```

MQGET of message number 3

****Message descriptor****

```
StrucId : 'MD ' Version : 1  
Report  : 0 MsgType : 8  
Expiry  : -1 Feedback : 0  
Encoding : 273 CodedCharSetId : 850  
Format  : 'AMQMRESP'  
Priority : 5 Persistence : 1  
MsgId   : X'414D5120736E6F6F6B65722020202020ED477D62A9EA100'  
CorrelId : X'00000000000000000000000000000000000000000000000000000000000000000000'  
BackoutCount : 0  
ReplyToQ      : '  
ReplyToQMGr   : 'snooker'  
** Identity Context  
UserIdentifier : 'trevor'  
AccountingToken :  
X'04373037300000000000000000000000000000000000000000000000000000000000'  
ApplIdentityData : '  
** Origin Context  
PutApplType   : '6'  
PutApplName   : '  
PutDate       : '19941124'   PutTime : '11240678'  
ApplOriginData : ' '
```

**** Message ****

length - 188 bytes

```
00000000: 736E 6F6F 6B65 7220 2020 2020 2020 2020 'snooker '  
00000010: 2020 2020 2020 2020 2020 2020 2020 2020 '  
00000020: 2020 2020 2020 2020 2020 2020 2020 2020 '  
00000030: 534E 4F4F 4B45 522E 5749 4748 542E 5443 'SNOOKER.WIGHT.TC'  
00000040: 5020 2020 2020 2020 2020 2020 2020 2020 'P '  
00000050: 2020 2020 2020 2020 2020 2020 2020 2020 '  
00000060: 0000 0001 0000 0024 0000 0001 0000 0015 '.....$.....'  
00000070: 0000 0001 0000 0001 0000 0000 0000 0000 '.....'  
00000080: 0000 0002 0000 0004 0000 0028 0000 0DAD '.....(.....'  
00000090: 0000 0000 0000 0014 534E 4F4F 4B45 522E '.....SNOOKER.'  
000000A0: 5749 4748 542E 5443 5020 2020 0000 0003 'WIGHT.TCP ....'  
000000B0: 0000 0010 0000 05E7 0000 0001 '.....'
```

```

MQGET of message number 4
****Message descriptor****

StrucId : 'MD ' Version : 1
Report  : 0 MsgType : 2
Expiry  : -1 Feedback : 0
Encoding : 273 CodedCharSetId : 850
Format  : 'MQADMIN '
Priority : 8 Persistence : 1
MsgId   : X'414D5120736E6F6F6B6572202020202020ED477D63826C000'
CorrelId : X'414D5120736E6F6F6B65722020202020ED477D62A9EA100'
BackoutCount : 0
ReplyToQ      : '
ReplyToQMgr   : 'snooker
** Identity Context
UserIdentifier : 'tiger
AccountingToken :
X'0437303730000000000000000000000000000000000000000000000000000000'
ApplIdentityData : '
** Origin Context
PutApplType      : '6'
PutApplName     : '
PutDate          : '19941124' PutTime : '11240694'
App1OriginData  : '

**** Message ****

length - 36 bytes

00000000: 0000 0002 0000 0024 0000 0001 0000 0015 '.....$......'
00000010: 0000 0001 0000 0001 0000 0000 0000 0000 '.....'
00000020: 0000 0000 '.....'

No more messages
MQCLOSE
MQDISC
    
```

Manipulation des files d'attente alias

Une file d'attente alias est une méthode permettant de rediriger des appels MQI. Il ne s'agit pas d'une file d'attente réelle, mais d'une définition qui renvoie à une file réelle. La définition de l'alias contient un nom de file d'attente cible indiqué par l'attribut TARGQ (*BaseQName* sous PCF). Lorsqu'une application indique une file d'attente alias dans un appel MQI, le gestionnaire identifie le nom de la file d'attente réelle lors de l'exécution.

Par exemple, une application a été développée pour insérer des messages dans une file d'attente nommée MA.FILE.ALIAS. L'application indique le nom de cette file lorsqu'elle lance une requête MQOPEN et, de façon indirecte, si elle insère un message dans cette file. L'application ne sait pas qu'il s'agit d'un alias. Pour chaque appel MQI utilisant cet alias, le gestionnaire de files d'attente identifie le nom de la file réelle, qui peut correspondre à une file locale ou éloignée définie sur ce gestionnaire.

En modifiant la valeur de l'attribut TARGQ, vous pouvez rediriger des appels MQI vers une autre file d'attente, qui peut éventuellement se trouver sur un autre gestionnaire. Cette fonction est utile pour la maintenance, la migration et l'équilibrage de la charge de travail.

Définition d'une file d'attente alias

Définition d'une file d'attente d'alias

La commande suivante crée une file d'attente alias :

```
DEFINE QALIAS (MA.FILE.ALIAS) TARGQ (FILE.JAUNE)
```

Cette commande redirige vers la file d'attente FILE.JAUNE les appels MQI pointant vers MA.FILE.ALIAS. Cependant, elle ne crée pas la file cible ; les appels MQI échoueront si FILE.JAUNE n'existe pas lors de l'exécution.

Si vous modifiez la définition de l'alias, vous pouvez rediriger les appels MQI vers une autre file d'attente. Par exemple :

```
DEFINE QALIAS (MA.FILE.ALIAS) TARGQ (FILE.MAGENTA) REPLACE
```

Cette commande redirige les appels MQI vers une autre file d'attente nommée FILE.MAGENTA.

Vous pouvez également utiliser les files d'attente alias pour qu'une même file (la cible) semble avoir des attributs différents pour des applications différentes. Pour cela, vous devez définir un alias par application. Supposons que vous ayez deux applications :

- L'application ALPHA peut insérer des messages dans la file d'attente FILE.JAUNE, mais elle n'est pas autorisée à en extraire de cette même file.
- L'application BETA peut récupérer des messages dans FILE.JAUNE, mais elle n'est pas autorisée à en insérer dans cette même file.

Pour définir ces alias, utilisez les commandes suivantes :

```
* Cet alias autorise l'insertion et interdit l'extraction pour ALPHA
DEFINE QALIAS (FILE.ALIAS.ALPHA) +
  TARGQ (FILE.JAUNE) +
  PUT (ENABLED) +
  GET (DISABLED)

* Cet alias interdit l'insertion et autorise l'extraction pour BETA
DEFINE QALIAS (FILE.ALIAS.BETA) +
  TARGQ (FILE.JAUNE) +
  PUT (DISABLED) +
  GET (ENABLED)
```

L'application ALPHA utilise le nom de file d'attente FILE.ALIAS.ALPHA dans ses appels MQI, et l'application BETA emploie le nom FILE.ALIAS.BETA. Elles ont accès à la même file, mais de deux façons différentes.

Vous pouvez utiliser les options LIKE et REPLACE lors de la définition de files d'attente alias, de la même manière qu'avec des files locales.

Utilisation d'autres commandes avec les files d'attente alias

Vous pouvez utiliser les commandes MQSC correspondantes pour afficher ou modifier les attributs de file d'attente alias, ou pour supprimer l'objet file d'attente alias. Par exemple :

```
* Display the queue alias' attributes
* ALL = Display all attributes

DISPLAY QUEUE (ALPHAS.ALIAS.QUEUE) ALL

* ALTER the base queue name, to which the alias resolves.
* FORCE = Force the change even if the queue is open.

ALTER QALIAS (FILE.ALIAS.ALPHA) TARGQ(FILE.LOCALE.ORANGE) FORCE

* Delete this queue alias, if you can.

DELETE QALIAS (FILE.ALIAS.ALPHA)
```

Vous ne pouvez pas supprimer une file d'attente alias si, par exemple, cette file est ouverte pour une application, ou si une file qui renvoie à celle-là est ouverte pour une application. Pour plus d'informations sur ce point et sur d'autres commandes relatives aux alias, consultez le manuel *MQSeries - Guide de référence des commandes MQSC*.

Manipulation des files d'attente modèles

Un gestionnaire de files d'attente crée une *file d'attente dynamique* si une application lui envoie un appel MQI indiquant le nom d'une file qui a été définie comme modèle. Le nom de la nouvelle file dynamique est généré par le gestionnaire lors de la création. Une *file d'attente modèle* est un gabarit contenant des attributs qui s'appliquent à toutes les files dynamiques qu'il sert à créer.

Les files d'attente modèles offrent aux applications une méthode pratique pour créer des files adaptées à leurs besoins.

Définition d'une file d'attente modèle

Définition d'une file d'attente modèle

Pour définir une file d'attente modèle avec un ensemble d'attributs, procédez de la même façon que pour une file locale. Ces deux types de file d'attente ont le même ensemble d'attributs. Cependant, les modèles permettent aussi d'indiquer si la file créée est temporaire ou permanente. (Contrairement aux files d'attente temporaires, les permanentes ne sont pas supprimées lors des redémarrages du gestionnaire de files d'attente.) Par exemple :

```
DEFINE QMODEL (FILE.MODELE.VERTE) +
  DESCR('Queue for messages from application X') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL) +
  DEFTYPE (PERMDYN)
```

Cette commande crée une définition de file d'attente modèle. L'attribut DEFTYPE indique que les files créées à partir de ce modèle sont dynamiques et permanentes.

Remarque : Les attributs non définis sont automatiquement copiés à partir de la file d'attente par défaut SYSTEM.DEFAULT.MODEL.QUEUE.

Vous pouvez utiliser les options LIKE et REPLACE lors de la définition de files d'attente modèles, de la même manière qu'avec des files locales.

Utilisation d'autres commandes avec les modèles

Vous pouvez utiliser les commandes MQSC correspondantes pour afficher ou modifier les attributs d'une file d'attente modèle, ou pour supprimer l'objet file d'attente modèle. Par exemple :

```
* Display the model queue's attributes
* ALL = Display all attributes

DISPLAY QUEUE (FILE.MODELE.VERTE) ALL

* ALTER the model to enable puts on any
* dynamic queue created from this model.

ALTER QMODEL (FILE.MODELE.BLEUE) PUT(ENABLED)

* Delete this model queue:

DELETE QMODEL (FILE.MODELE.ROUGE)
```

Gestion des objets pour le déclenchement

MQSeries offre une fonction permettant de lancer automatiquement une application lorsqu'une file d'attente remplit certaines conditions. Par exemple, une condition peut être remplie lorsque le nombre de messages se trouvant en file d'attente atteint une valeur déterminée. Cette fonction s'appelle le *déclenchement* ; elle est décrite de façon détaillée dans le manuel *MQSeries - Guide de programmation*

d'applications. La présente section décrit comment configurer les objets nécessaires pour la prise en charge du déclenchement sur MQSeries pour Compaq NSK

Définition d'une file d'attente d'application pour le déclenchement

Une file d'attente d'application est une file locale utilisée pour la messagerie par des applications, via l'interface MQI. Le déclenchement exige qu'un certain nombre d'attributs soient définis pour la file d'attente d'application. Le déclenchement lui-même est activé par l'attribut *Trigger* (ou TRIGGER sur MQSC).

Dans l'exemple suivant, un événement de déclenchement doit être généré lorsque la file d'attente locale FILE.ASS.AUTO contient 100 messages de priorité cinq ou supérieure :

```
DEFINE QLOCAL(FILE.ASS.AUTO) +
        PROCESS (EVALUATION.ASS.AUTO) +
        MAXMSGL (2000) +
        DEFPSIST (YES) +
        INITQ (FILE.INIT.ASS.AUTO) +
        TRIGGER +
        TRIGTYPE (DEPTH) +
        TRIGDPTH (100)+
        TRIGMPRI (5)
```

Où :

QLOCAL (FILE.ASS.AUTO)

indique le nom de la file d'attente d'application définie.

PROCESS (EVALUATION.ASS.AUTO)

indique le nom de l'application devant être démarrée par un programme de déclenchement.

MAXMSGL (2000)

indique la taille maximale des messages en file d'attente.

DEFPSIST (YES)

indique que les messages de la file sont persistants par défaut.

INITQ (FILE.INIT.ASS.AUTO)

indique le nom de la file d'attente d'initialisation dans laquelle le gestionnaire doit insérer le message de déclenchement.

TRIGGER

indique la valeur de l'attribut de déclenchement.

TRIGTYPE (DEPTH)

indique qu'un événement de déclenchement est généré lorsque le nombre de messages du niveau de priorité requis (TRIGMPRI) atteint la valeur indiquée dans TRIGDPTH.

TRIGDPTH (100)

indique le nombre de messages nécessaires pour générer un événement de déclenchement.

TRIGMPRI (5)

indique le niveau de priorité des messages comptabilisés par le

File d'attente d'application pour le déclenchement

gestionnaire de files d'attente pour générer un événement de déclenchement. Seuls les messages ayant une priorité égale ou supérieure à 5 sont comptabilisés.

Définition d'une file d'attente d'initialisation

Lorsqu'un événement de déclenchement se produit, le gestionnaire de files d'attente insère un message de déclenchement dans la file d'initialisation indiquée dans la définition de la file d'application. Il n'existe pas de configuration spéciale pour les files d'attente d'initialisation, mais vous pouvez vous aider de la définition de la file locale FILE.INIT.ASS.AUTO ci-dessous :

```
DEFINE QLOCAL(FILE.INIT.ASS.AUTO) +
  GET (ENABLED) +
  NOSHARE +
  NOTRIGGER +
  MAXMSGL (2000) +
  MAXDEPTH (10)
```

Création d'une définition de processus

Pour créer une définition de processus, utilisez la commande DEFINE PROCESS. Une définition de processus associe une file d'attente d'application à l'application qui doit traiter les messages de cette file. Cette opération est effectuée par l'attribut PROCESS de la file d'attente d'application FILE.ASS.AUTO. La commande MQSC suivante définit le processus approprié, EVALUATION.ASS.AUTO, de cet exemple :

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
  DESCR ('Insurance request message processing') +
  APPLTYPE (NSK) +
  APPLICID ('$DATA1.TEST.IRMPO1') +
  USERDATA ('open, close, 235')
```

Où :

MOTOR.INSURANCE.QUOTE.PROCESS

est le nom de la définition de processus.

DESCR ('Insurance request message processing')

est le texte qui décrit le programme d'application auquel cette définition fait référence. Ce texte apparaît lorsque vous utilisez la commande DISPLAY PROCESS. Cela peut vous aider à identifier le type des opérations effectuées par le processus. Si la chaîne de caractères contient des espaces, vous devez l'écrire entre apostrophes.

APPLTYPE(NSK)

indique le type de l'application à démarrer sous Compaq NSK.

APPLICID ('\$DATA1.TEST.IRMPO1')

est le nom du fichier exécutable de l'application sur le système local

USERDATA ('open, close, 235')

indique des données définies par l'utilisateur, qui peuvent être utilisées par l'application.

Affichage de la définition de processus

Pour examiner le résultat de votre définition, utilisez la commande DISPLAY PROCESS avec le mot clé ALL. Par exemple :

```
DISPLAY PROCESS (EVAL.ASS.AUTO) ALL

      24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR (Insurance request message processing)
APPLICID ($DATA1.TEST.IRMP01)
ENVRDATA ( )
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (NSK)
```

USERDATA est une chaîne représentant les arguments transmis à l'application de déclenchement. Voir les programmes exemples AMQSTRG0 et AMQINQA (dans le sous-volume ZMQSSMPL) pour trouver des exemples de syntaxe des moniteurs et applications de déclenchement.

Vous pouvez également utiliser la commande MQSC ALTER PROCESS pour modifier une définition de processus existante, et la commande DELETE PROCESS pour en supprimer une.

Chapitre 6. Automatisation des tâches d'administration

Ce chapitre suppose que vous êtes familiarisé avec l'administration des objets MQSeries.

Vous pouvez être amené à décider de l'automatisation de certaines tâches d'administration ou de contrôle sur votre site. Il est possible d'automatiser les tâches d'administration pour les gestionnaires de files d'attente locaux et éloignés à l'aide de commandes programmables PCF.

Ce chapitre décrit :

- l'utilisation des commandes programmables pour automatiser les tâches d'administration, dans la section Administration à l'aide des commandes PCF.
- l'utilisation du serveur de commandes, dans la section «Gestion du serveur de commandes pour l'administration éloignée» à la page 121.

Commandes programmables PCF

L'objet des commandes programmables PCF de MQSeries est de permettre l'intégration de tâches d'administration dans un programme d'administration. Ainsi, vous pouvez créer des files d'attente, des définitions de processus, des canaux et des listes de noms et modifier des gestionnaires de files d'attente, à partir d'un programme.

Les commandes programmables PCF regroupent des fonctions similaires à celles fournies par la fonction MQSC.

Vous pouvez ainsi écrire un programme chargé d'émettre des commandes PCF vers n'importe quel gestionnaire de files d'attente sur le réseau à partir d'un seul noeud. Vous pouvez donc centraliser et automatiser les tâches d'administration.

Chaque commande programmable PCF est une structure de données intégrée dans les données d'application d'un message MQSeries. Elle est envoyée au gestionnaire de files d'attente cible à l'aide de la fonction MQI MQPUT de la même façon que tout autre message. Le serveur de commandes du gestionnaire qui reçoit le message l'interprète comme un message de commande et l'exécute. Pour obtenir les réponses, l'application émet un appel MQGET, et les données de réponse sont renvoyées dans une autre structure de données. L'application peut ensuite traiter la réponse et agir en conséquence.

Remarque : A la différence des commandes MQSC, les commandes programmables PCF et leurs réponses ne se présentent pas dans un format texte lisible par l'utilisateur.

Les éléments suivants doivent toujours être spécifiés par le programmeur d'applications pour créer un message de commande PCF :

Descripteur de message

- Il s'agit d'un descripteur de message MQSeries standard, dans lequel :
 - Le type de message (*MsgType*) est MQMT_REQUEST.
 - Le format du message (*Format*) est MQFMT_ADMIN.

Commandes PCF

Données d'application

Elles contiennent le message PCF, y compris l'en-tête PCF, dans lequel :

Le type du message PCF (*Type*) est MQCFT_COMMAND.

L'identificateur de commande indique la commande, par exemple *Change Queue* (MQCMD_CHANGE_Q).

Pour une description complète des structures de données PCF et de leur mise en oeuvre, reportez-vous au manuel *MQSeries Programmable System Management*.

Attributs dans les commandes MQSC et PCF

Dans ce manuel, les attributs d'objet indiqués dans les commandes MQSC sont écrits en majuscules (par exemple, RQMNAME), bien que la distinction majuscules/minuscules ne s'applique pas. Ces noms d'attribut sont limités à huit caractères.

Les attributs d'objets pour les commandes PCF sont écrits en italique et ne sont pas limités à huit caractères. Par exemple, l'équivalent PCF de RQMNAME est *NomGestFilesEloignées*.

Commandes PCF d'échappement

Les commandes d'échappement PCF sont des commandes PCF qui contiennent des commandes MQSC à l'intérieur du texte du message. Vous pouvez utiliser les commandes PCF pour envoyer des ordres à un gestionnaire de files d'attente éloigné. Pour plus de détails sur l'utilisation des commandes d'échappement PCF, reportez-vous au manuel *MQSeries Programmable System Management*.

Utilisation de l'interface MQAI pour simplifier l'utilisation des commandes programmables PCF

MQAI est une interface d'administration MQSeries, disponible sur la plateforme Compaq NSK.

Elle exécute des tâches d'administration sur un gestionnaire de files d'attente au moyen de *jeux de données*. Les jeux de données permettent, plus facilement que les commandes PCF, de manipuler les propriétés (ou paramètres) des objets.

MQAI permet de :

- **simplifier l'utilisation des messages PCF.** L'interface MQAI est un outil simple permettant de gérer MQSeries. Comme il n'est pas nécessaire d'écrire vos propres messages PCF, vous évitez les problèmes liés aux structures de données complexes.

Pour transmettre des paramètres dans les programmes écrits à l'aide d'appels MQI, le message PCF doit contenir la commande et le détail de la chaîne ou de l'entier. Pour cela, votre programme doit comporter plusieurs instructions par structure. Il est également nécessaire d'allouer de l'espace mémoire. Cette tâche est longue et laborieuse.

A l'opposé, les programmes écrits à l'aide de l'interface MQAI transmettent les paramètres dans le jeu de données approprié et ne nécessitent qu'une instruction par structure. Les jeux de données MQAI évitent d'avoir à manipuler des tableaux et d'allouer de la mémoire, et masquent, dans une certaine mesure, la complexité des commandes PCF.

- **faciliter la gestion des incidents.** Il est difficile d'obtenir les codes retour des commandes MQSC, mais MQAI rend la gestion des erreurs plus aisée pour le programme.

Après avoir créé et rempli le jeu de données, vous pouvez envoyer un message de commande d'administration au serveur de commandes d'un gestionnaire de files d'attente, à l'aide de l'appel `mqExecute`, qui attendra les réponses éventuelles. L'appel `mqExecute` gère l'échange avec le serveur de commandes et renvoie les réponses dans un jeu de réponses.

Pour plus de détails sur l'utilisation de l'interface MQAI, consultez le manuel *MQSeries Administration Interface Programming Guide and Reference*.

Pour plus de détails sur les commandes PCF, consultez le manuel *MQSeries Programmable System Management*.

Gestion du serveur de commandes pour l'administration éloignée

Un serveur de commandes peut être associé à chaque gestionnaire de files d'attente. Un serveur de commandes traite toutes les commandes en provenance de gestionnaires de files d'attente éloignés, ainsi que les commandes programmables PCF des applications. Il présente les commandes au gestionnaire de files d'attente pour leur traitement et renvoie un code de fin d'exécution ou un message opérateur en fonction de l'origine de la commande.

Un serveur de commandes est obligatoire pour toute administration mettant en oeuvre des commandes PCF et l'interface MQAI, ainsi que pour l'administration éloignée.

Remarque : Pour l'administration éloignée, vous devez vous assurer que le gestionnaire de files d'attente cible est en fonctionnement. Dans le cas contraire, les messages contenant des commandes ne pourront pas quitter le gestionnaire qui les a émis. Les messages sont alors insérés dans la file d'attente de transmission locale qui dessert le gestionnaire de files d'attente éloigné. Dans la mesure du possible, il est préférable d'éviter cette situation.

Lancement du serveur de commandes

Pour lancer le serveur de commandes, utilisez la commande suivante :

```
strmqcsv saturne.gest.files
```

où `saturne.gest.files` est le gestionnaire de file d'attente pour lequel le serveur de commandes est lancé.

Le serveur de commandes peut également être démarré dans PATHCOM en débloquant et redémarrant son Serverclass MQM.

Affichage de l'état du serveur de commandes

Pour l'administration éloignée, assurez-vous que le serveur de commandes sur le gestionnaire de files d'attente cible est en fonctionnement. S'il n'est pas en fonctionnement, les commandes éloignées ne pourront être traitées. Tous les messages contenant des commandes seront insérés dans la file d'attente du gestionnaire cible.

Administration éloignée du serveur de commandes

Pour afficher l'état du serveur de commandes d'un gestionnaire de files d'attente, nommé ici `saturne.gest.files`, la commande est la suivante :

```
dspmqcsv saturne.gest.files
```

Vous devez émettre cette commande depuis la machine cible. Si le serveur de commandes est en fonctionnement, le message suivant est envoyé :

```
AMQ8027    MQSeries Command Server Status ..: Running
```

Arrêt d'un serveur de commandes

Pour arrêter un serveur de commandes, la commande est la suivante (en utilisant l'exemple précédent) :

```
endmqcsv saturne.gest.files
```

Vous pouvez arrêter le serveur de commandes de deux façons différentes :

- Pour effectuer un arrêt contrôlé, utilisez la commande **endmqcsv** avec l'indicateur `-c`. Il s'agit de l'indicateur par défaut.
- Pour effectuer un arrêt immédiat, utilisez la commande **endmqcsv** avec l'indicateur `-i`.

Remarque : L'arrêt d'un gestionnaire de files d'attente met également fin au serveur de commandes qui lui est associé (si un serveur de commandes a été lancé).

Chapitre 7. Administration des objets MQSeries éloignés

Ce chapitre explique comment administrer les objets MQSeries sur un autre gestionnaire de files d'attente. Il décrit également comment utiliser les objets d'une file d'attente éloignée pour contrôler la destination des messages et les messages de réponse.

Il comprend les sections suivantes :

- «Administration d'un gestionnaire de files d'attente» à la page 125
- «Création d'une définition locale de file d'attente éloignée» à la page 131
- «Utilisation des définitions de files d'attente éloignées pour les alias» à la page 134

Pour plus d'informations sur les canaux, leurs attributs et leur configuration, consultez le manuel *MQSeries Intercommunication*.

Canaux, clusters et mise en file d'attente éloignée

Les gestionnaires de files d'attente communiquent entre eux en envoyant un message et, le cas échéant, en recevant une réponse. Le gestionnaire récepteur peut :

- se trouver sur la même machine ;
- se trouver sur une autre machine sur le même site ou sur un site géographiquement éloigné ;
- s'exécuter sur la même plateforme que le gestionnaire de files d'attente local ;
- s'exécuter sur une autre plateforme prise en charge par MQSeries.

Ces messages peuvent provenir de :

- Programmes écrits par l'utilisateur qui transfèrent des données d'un noeud sur un autre.
- Programmes d'administration écrits par l'utilisateur et qui ont recours aux commandes PCF ou à l'interface MQAI.
- Gestionnaires de files d'attente qui envoient :
 - des messages d'événements outils à un autre gestionnaire,
 - des commandes MQSC émises à partir d'une commande **runmqsc** en mode indirect (les commandes sont exécutées sur un autre gestionnaire).

Pour qu'un message puisse être envoyé à un gestionnaire de files d'attente éloigné, le gestionnaire local a besoin d'un mécanisme de détection et de transport des messages entrants. Ce mécanisme doit se composer :

- d'au moins un canal,
- d'une file d'attente de transmission,
- d'un agent MCA,
- d'un programme d'écoute de canal,
- d'un initialiseur de canal.

Un canal est une voie de communication unidirectionnelle entre deux gestionnaires de files d'attente, qui transporte les messages destinés à un nombre quelconque de files d'attente sur le gestionnaire éloigné.

Administration des objets éloignés

Chaque extrémité du canal possède une définition distincte. Par exemple, si une extrémité est un émetteur ou un serveur, l'autre sera un récepteur ou un demandeur. Un canal simple comprend une *définition de canal émetteur* sur le gestionnaire de files d'attente local et une *définition de canal récepteur* à l'extrémité du gestionnaire éloigné. Les deux définitions, qui doivent porter le même nom, forment ensemble un canal unique.

Si le gestionnaire de files d'attente éloigné doit répondre aux messages envoyés par le gestionnaire local, il convient de définir un deuxième canal, qui permettra de renvoyer les réponses.

Les canaux sont définis à l'aide de la commande MQSC DEFINE CHANNEL. Dans ce chapitre, les exemples concernant des canaux utilisent les attributs de canal par défaut, sauf indication contraire.

A chaque extrémité d'un canal se trouve un agent MCA qui contrôle l'envoi et la réception de messages. Le rôle du MCA est de prendre les messages dans la file d'attente de transmission et de les placer sur la voie de communication entre les gestionnaires de files d'attente. Inversement, le MCA est également prévu pour prendre les messages sur la voie de communication et de les placer dans les files d'attente cible.

Une file d'attente de transmission est une file locale spécialisée dans laquelle les messages sont placés temporairement avant d'être récupérés par le MCA et envoyés au gestionnaire de files d'attente éloigné. Vous spécifiez le nom de la file d'attente de transmission dans une *définition de file d'attente éloignée*.

La section «Préparation des canaux et des files d'attente de transmission pour l'administration éloignée» à la page 126 montre comment se servir de ces définitions pour configurer l'administration éloignée.

Pour plus d'informations sur la configuration de la mise en file d'attente répartie, consultez le manuel *MQSeries Intercommunication*.

Administration éloignée à l'aide de clusters

Dans un réseau MQSeries classique utilisant la mise en file d'attente répartie, chaque gestionnaire de files d'attente est indépendant. Pour qu'un gestionnaire de files d'attente puisse envoyer des messages à un autre gestionnaire, il doit posséder une définition de file d'attente de transmission, une définition de canal vers le gestionnaire éloigné et une définition de file d'attente éloignée pour chaque file destinée à recevoir des messages.

Un *cluster* est un groupe de gestionnaires de files d'attente configuré de manière à ce qu'ils puissent communiquer directement sur un réseau simple sans avoir recours à des définitions de file d'attente de transmission, de canaux et de files d'attente complexes. Il est facile de définir des clusters. Ils regroupent en général des gestionnaires de files d'attente entre lesquels il existe un lien logique et qui ont besoin de partager des données ou des applications.

Une fois le cluster créé, les gestionnaires qui le composent peuvent communiquer entre eux *sans passer par des définitions de canal ou de file d'attente complexes*. Même le plus petit des clusters permettra des gains de temps considérables en matière d'administration système.

La création d'un réseau de gestionnaires de files d'attente sous forme de cluster fait appel à moins de définitions que la création d'un environnement classique de mise en file d'attente répartie. Grâce au nombre réduit de définitions à créer, vous pouvez configurer ou modifier votre réseau plus facilement et plus rapidement, tout en diminuant le risque d'erreur dans les définitions.

Pour définir un cluster, vous avez en général besoin d'une définition de cluster émetteur (CLUSDR) et d'une définition de cluster récepteur (CLUSRCVR) par gestionnaire de files d'attente. Les définitions de files de transmission et de files d'attente éloignées sont superflues. Les principes de l'administration éloignées s'appliquent également à un cluster, mais les définitions mêmes en sont grandement simplifiées.

Pour plus d'informations sur les clusters, leurs attributs et leur configuration, consultez le manuel *MQSeries Queue Manager Clusters*.

Administration d'un gestionnaire de files d'attente

La présente section décrit comment administrer un gestionnaire de files d'attente éloigné à partir d'un gestionnaire local. Vous pouvez procéder à partir d'un noeud local à l'aide de :

- Commandes MQSC
- Commandes PCF

La préparation des files d'attente et des canaux est identique pour les deux méthodes. Dans ce manuel, les exemples utilisent les commandes MQSC car elles sont plus faciles à comprendre. Cependant, vous pouvez transposer les exemples sous PCF, si vous préférez. Pour plus de détails sur l'utilisation des commandes PCF pour la rédaction de programmes d'administration, reportez-vous au manuel *MQSeries Programmable System Management*.

Vous envoyez des commandes MQSC à un gestionnaire éloigné, de façon interactive ou à partir d'un fichier de texte contenant les commandes. Le gestionnaire éloigné peut se trouver sur le même ordinateur ou, cas le plus courant, sur un autre ordinateur. Vous pouvez gérer à distance des gestionnaires de files d'attente situés dans différents environnements MQSeries dont UNIX®, Compaq NSK, AS/400®, MVS/ESA, Windows® 2000, and OS/390.

Pour mettre en application l'administration éloignée, vous devez créer des objets spécifiques. Les valeurs par défaut (la longueur des messages, par exemple) devraient vous convenir, à moins que vous n'ayez des besoins particuliers.

Préparation des gestionnaires pour l'administration éloignée

La figure 37 à la page 126 montre la configuration des gestionnaires de files d'attente et des canaux requis pour l'administration éloignée. Le gestionnaire `source.gest.files` est le gestionnaire *source* à partir duquel vous émettez des commandes MQSC et qui recevra, éventuellement, le résultat de ces commandes (par des messages opérateur). `cible.gest.files` est le gestionnaire *cible* qui traite les commandes et génère les messages opérateur.

Remarque : `source.gest.files` doit être le gestionnaire de files d'attente par défaut. Pour plus de détails sur la création d'un gestionnaire de files d'attente, reportez-vous à la section «`crtmqm` (création d'un gestionnaire de files d'attente)» à la page 267.

Administration d'un gestionnaire de files d'attente éloignées

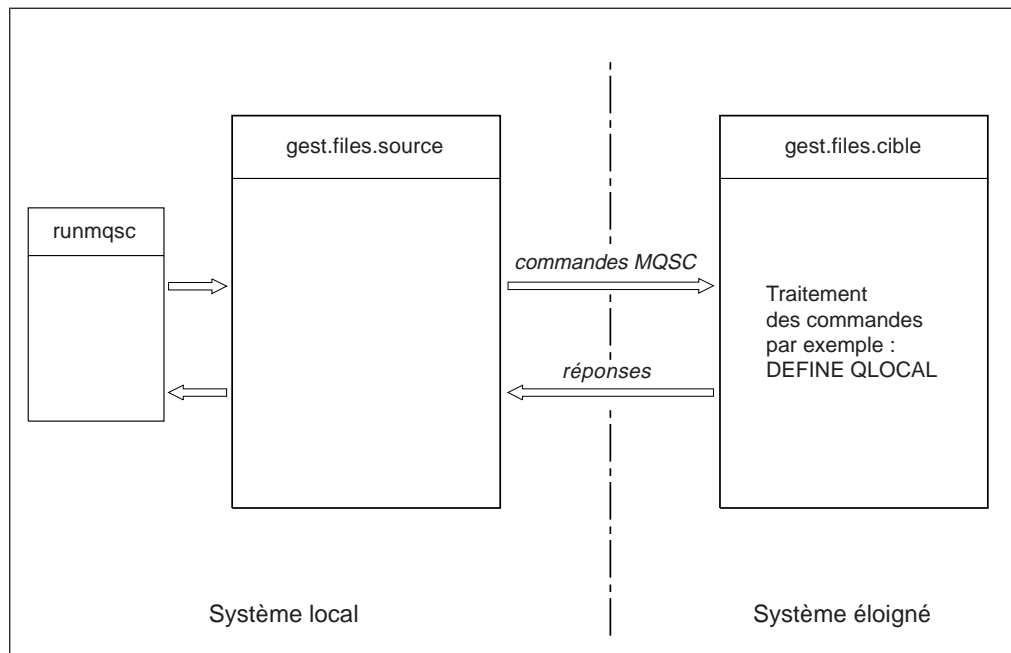


Figure 37. Administration éloignée

Si ce n'est déjà fait, vous devez effectuer les opérations suivantes sur les deux systèmes :

- Créer le gestionnaire de files d'attente, à l'aide de la commande **crtmqm**.
- Lancer le gestionnaire de files d'attente à l'aide de la commande **strmqm**.

Pour plus d'informations, reportez-vous à la section «Restauration des objets système et par défaut» à la page 67. Vous pouvez exécuter ces commandes localement ou par l'intermédiaire d'une fonction réseau, comme Telnet par exemple.

Au niveau du gestionnaire de files d'attente cible :

- La file d'attente de commandes SYSTEM.ADMIN.COMMAND.QUEUE doit être présente. Elle est créée par défaut lors de la création d'un gestionnaire de files d'attente.
- Le serveur de commandes doit être lancé à l'aide de la commande **strmqcsv**.

Préparation des canaux et des files d'attente de transmission pour l'administration éloignée

Pour exécuter des commandes MQSC à distance, vous devez configurer deux canaux, un pour chaque sens, ainsi que les files d'attente de transmission correspondantes. Dans cet exemple, TCP/IP est le type de transmission et l'on suppose que vous connaissez l'adresse TCP/IP correspondante.

Le canal `source.vers.cible` sert à l'envoi des commandes MQSC du gestionnaire source vers le gestionnaire cible. Son émetteur se situe au niveau du gestionnaire `source.gest.files` et son récepteur au niveau du gestionnaire `cible.gest.files`. Le canal `cible.vers.source` sert au retour du résultat des commandes et des messages opérateur générés vers le gestionnaire source. Vous devez également définir une file d'attente de transmission pour chaque émetteur. Il s'agit d'une file d'attente locale portant le nom du gestionnaire récepteur. Pour que l'administration éloignée fonctionne, le nom de la file d'attente de transmission (XMITQ) doit coïncider avec le nom du gestionnaire de files d'attente éloigné, sauf si vous utilisez un gestionnaire alias. Cette configuration est schématisée dans la figure 38 à la page 127

Administration d'un gestionnaire de files d'attente éloignées

à la page 127. Pensez cependant que la SYSTEM.MQSC.REPLY.QUEUE est le nom de la file d'attente modèle utilisée par MQSC pour créer sa propre file d'attente de réponses dynamique. Ce nom de file d'attente peut varier et est interne à MQSC.

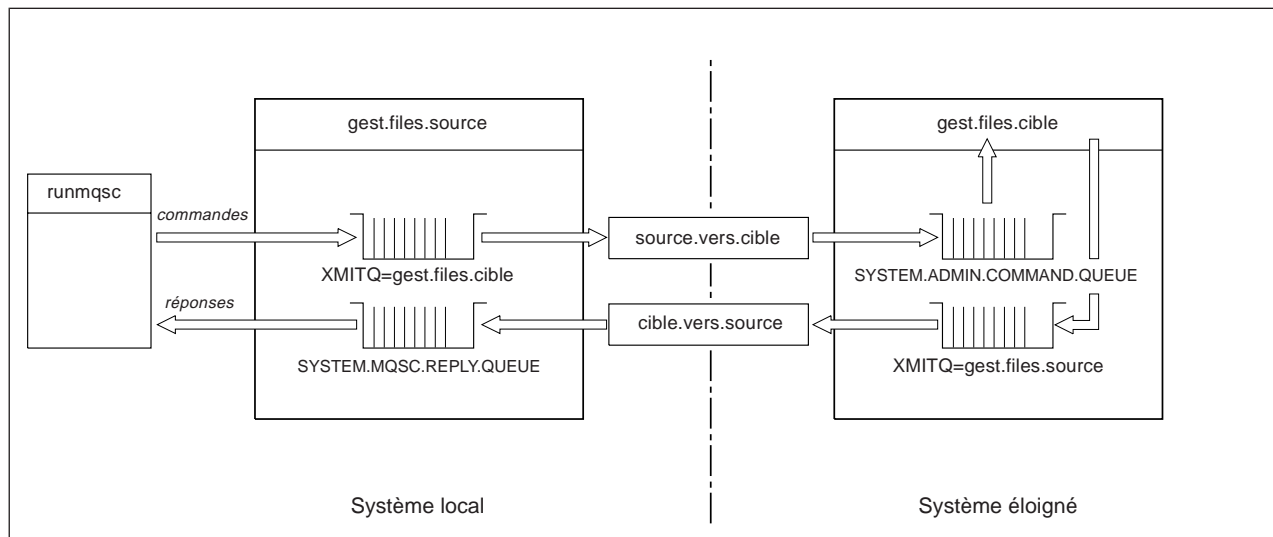


Figure 38. Configuration des canaux et des files d'attente pour l'administration éloignée

Consultez le manuel *MQSeries Intercommunication* pour plus de détails sur la configuration des canaux éloignés.

Définition des canaux et des files d'attente de transmission

Sur le gestionnaire source, lancez les commandes MQSC ci-dessous pour définir les canaux et la file d'attente de transmission :

```
* Define the sender channel at the source queue manager
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)

* Define the receiver channel at the source queue manager

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Define the transmission queue on the source

DEFINE QLOCAL ('target.queue.manager') +
  USAGE (XMITQ)
```

Administration d'un gestionnaire de files d'attente éloignées

Sur le gestionnaire cible (cible.gest.files), émettez ces commandes pour créer les canaux et file d'attente de transmission :

```
* Define the sender channel on the destination queue manager

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(SDR) +
  CONNAME ('198.210.60.37(1414)') +
  XMITQ ('source.queue.manager') +
  TRPTYPE(TCP)

* Define the receiver channel on the destination queue manager

DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Define the transmission queue on the destination queue manager

DEFINE QLOCAL ('source.queue.manager') +
  USAGE (XMITQ)
```

Remarque : Les noms de connexion TCP/IP indiqués pour l'attribut CONNAME dans les définitions du canal émetteur ne sont donnés qu'à titre indicatif. Il s'agit de l'adresse IP ou nom dans le réseau de l'ordinateur à l'autre extrémité de la connexion. Utilisez les valeurs appropriées pour votre réseau.

Lancement des canaux

La description qui suit suppose que les deux extrémités du canal fonctionnent sous MQSeries pour Compaq NSK. Dans le cas contraire, consultez la documentation appropriée pour l'extrémité non-Compaq NSK du canal.

Pour lancer les deux canaux, assurez-vous d'abord que le processus de programme d'écoute TCP Compaq NSK a été configuré pour MQSeries sur les deux noeuds et fonctionne à chaque extrémité des connexions. Lancez ensuite les canaux dans **runmqsc**.

- Sur le gestionnaire source, entrez :

```
start channel ('source.to.target')
```

- Sur le gestionnaire destination, entrez :

```
start channel ('target.to.source')
```


Administration d'un gestionnaire de files d'attente éloignées

Définition automatique de canaux

La définition automatique de canaux ne s'applique que si le gestionnaire de files d'attente cible s'exécute sur un produit MQSeries version 5.1 ou plus. Si une requête de connexion est reçue et qu'aucune définition appropriée de canal récepteur ou de canal de connexion serveur ne peut être trouvée dans le fichier de définition de canal (CDF), MQSeries crée automatiquement une définition et l'ajoute dans le fichier CDF. Les définitions automatiques sont basées sur deux définitions par défaut fournies avec MQSeries : SYSTEM.AUTO.RECEIVER et SYSTEM.AUTO.SVRCONN.

Pour activer la fonction de création automatique des définitions de canaux récepteur et de canal de connexion serveur, vous devez mettre à jour l'objet gestionnaire de files d'attente à l'aide de la commande MQSC ALTER QMGR (ou de la commande PCF Change Queue Manager).

Pour plus de détails sur la création automatique de définitions de canaux, reportez-vous au manuel *MQSeries Intercommunication*.

Pour plus d'informations sur la création automatique de définitions de canaux pour les clusters, reportez-vous au manuel *MQSeries Queue Manager Clusters*.

Emission de commandes MQSC à distance

Le serveur de commandes *doit* s'exécuter sur le gestionnaire de files d'attente cible pour pouvoir traiter des commandes MQSC. Il ne doit pas nécessairement s'exécuter sur le gestionnaire source.

- Sur le gestionnaire destination, entrez :

```
strmqcsv cible.gest;files
```

- Sur le gestionnaire source, vous pouvez maintenant exécuter MQSC de façon interactive en mode file d'attente en entrant :

```
runmqsc -w 30 cible.gest.files
```

Cette forme de la commande **runmqsc** avec l'indicateur **-w**, exécute les commandes en mode file d'attente, dans lequel les commandes sont insérées, sous une forme modifiée, dans la file d'attente en entrée du serveur de commandes et exécutées dans l'ordre.

Lorsque vous entrez une commande MQSC, elle est réacheminée vers le gestionnaire de files d'attente éloigné, dans cet exemple `cible.gest.files`. Le délai est fixé à 30 secondes ; si aucune réponse n'est reçue au bout de 30 secondes, le message suivant est généré sur le gestionnaire local source :

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

A la fin de la session MQSC, le gestionnaire de files d'attente local affiche toutes les réponses hors délai qui sont arrivées. Lorsque la session MQSC est terminée, toutes les réponses ultérieures sont supprimées.

Administration d'un gestionnaire de files d'attente éloignées

En mode file, vous pouvez également exécuter un fichier de commandes MQSC sur un gestionnaire de files d'attente éloigné. Par exemple :

```
f unmqsc /IN mycmds, OUT rapport/ -w 60 cible.gest.file
```

où mycmds est un fichier contenant les commandes MQSC et rapport est le fichier de rapport.

Opérations sur des gestionnaires de files d'attente sous MVS/ESA

Vous pouvez émettre des commandes MQSC vers un gestionnaire de files d'attente MVS/ESA à partir d'un gestionnaire de files d'attente MQSeries pour Compaq NSK. Toutefois, vous devez pour cela modifier la commande **runmqsc** et les définitions de canal au niveau de l'émetteur.

Vous devez en particulier ajouter l'indicateur -x dans la commande **runmqsc** sur un noeud Compaq NSK :

```
runmqsc -w 30 -x QMRI
```

La définition de canal est la suivante :

```
* Define the sender channel at the source queue manager on Compaq NSK  
  
DEFINE CHANNEL ('source.to.target') +  
  CHLTYPE(SDR) +  
  CONNAME ('198.210.60.37(1414)') +  
  XMITQ (QMRI) +  
  TRPTYPE(TCP) +
```

Vous devez également définir le canal récepteur et la file d'attente de transmission sur le gestionnaire de files d'attente source comme précédemment. Dans cet exemple, le protocole de transmission utilisé est une nouvelle fois TCP/IP.

Recommandations pour la mise en file d'attente éloignée

Lorsque vous effectuez une mise en file d'attente éloignée, suivez les règles suivantes :

1. Placez les commandes MQSC à exécuter sur le système éloigné dans un fichier de commandes.
2. Vérifiez localement vos commandes MQSC, en ajoutant l'indicateur -v dans la commande **runmqsc**.

Vous ne pouvez pas utiliser la commande **runmqsc** pour vérifier les commandes MQSC sur un autre gestionnaire de files d'attente.

3. Vérifiez si possible que le fichier de commandes s'exécute localement sans erreurs.
4. Pour finir, exécutez le fichier de commandes pour le système éloigné.

En cas d'incident

Si vous rencontrez des difficultés à exécuter des commandes MQSC à distance, consultez la liste ci-après pour vérifier si vous avez effectué les opérations suivantes :

- Lancement du serveur de commandes sur le gestionnaire de files d'attente cible.
- Définition d'une file d'attente de transmission correcte.
- Définition des deux extrémités des canaux de communication pour :
 - le canal sur lequel sont transmises les commandes envoyées,
 - le canal sur lequel sont transmises les réponses en retour.
- Indication du nom de connexion (CONNNAME) correct dans la définition de canal.
- Lancement des programmes d'écoute avant de lancer les canaux de communication.
- Vérification afin de s'assurer que le délai de déconnexion n'est pas arrivé à expiration, par exemple si un canal est lancé puis s'arrête au bout d'un certain temps. Ce point est particulièrement important si vous lancez les canaux manuellement.
- Vérification que les requêtes envoyées à partir du gestionnaire source peuvent être correctement interprétées par le gestionnaire cible (par exemple, qu'elles ne contiennent pas de nouveaux paramètres).

Reportez-vous également à la section «Résolution des incidents à l'aide des commandes MQSC» à la page 101.

Création d'une définition locale de file d'attente éloignée

Vous pouvez utiliser une définition de file d'attente éloignée pour définir localement une file d'attente éloignée. La création d'une définition de file d'attente éloignée sur votre gestionnaire de files d'attente local vous permet d'identifier une file d'attente locale sur un autre gestionnaire.

Fonctionnement de la définition locale d'une file d'attente éloignée

Une application se connecte à un gestionnaire de files d'attente local, puis émet un appel MQOPEN. Dans l'appel d'ouverture, le nom de file d'attente indiqué est le nom d'une définition de file d'attente éloignée sur le gestionnaire de files d'attente local. La définition de file d'attente éloignée fournit les noms de la file d'attente cible, du gestionnaire de files d'attente cible et, éventuellement, d'une file d'attente de transmission. Pour insérer un message dans la file d'attente éloignée, l'application émet un appel MQPUT indiquant l'identificateur retourné par l'appel MQOPEN. Le gestionnaire de files d'attente inclut le nom de la file d'attente éloignée et celui du gestionnaire éloigné dans un en-tête de transmission ajouté au début du message. Ces informations sont utilisées pour acheminer le message vers sa cible dans le réseau.

En tant qu'administrateur, vous pouvez contrôler la cible du message en modifiant la définition de file d'attente éloignée.

Exemple

Une application doit insérer un message dans une file d'attente qui appartient à un gestionnaire de files d'attente éloigné.

Création d'une définition locale de file d'attente éloignée

Fonctionnement

L'application se connecte à un gestionnaire de files d'attente, par exemple saturne.gest.files. La file d'attente cible appartient à un autre gestionnaire de files d'attente.

Dans l'appel MQOPEN, l'application indique les zones suivantes :

Valeur de la zone	Description
<i>ObjectName</i> FILE.ELOIGNEE.CYAN	Indique le nom local de l'objet file d'attente éloignée. Cette zone définit la file d'attente et le gestionnaire de files d'attente cibles.
<i>ObjectType</i> (file d'attente)	Indique que l'objet est une file d'attente.
<i>ObjectQmgrName</i> Vide ou saturne.gest.files	Cette zone est facultative. Si elle est vide, le nom du gestionnaire de files d'attente local est implicite. (Il s'agit du gestionnaire de files d'attente sur lequel la définition de la file d'attente éloignée est connectée.) Si elle n'est pas vide, le nom du gestionnaire de files d'attente local doit être précisé.

Ensuite, l'application émet un appel MQPUT pour insérer un message dans cette file d'attente.

Sur le gestionnaire de files d'attente local, vous pouvez créer une définition locale de file d'attente éloignée en utilisant les commandes MQSC suivantes :

```
DEFINE QREMOTE (FILE.ELOIGNEE.CYAN) +  
  DESCR ('File des demandes d'assurance auto des filiales') +  
  RNAME (FILE.COT.ASSUR.AUTO) +  
  RQMNAME ('jupiter.gest.files') +  
  XMITQ ('INQUOTE.XMIT.QUEUE')
```

Où :

QREMOTE (FILE.ELOIGNEE.CYAN)

Indique le nom local de l'objet file d'attente éloignée. Il s'agit du nom que les applications connectées à ce gestionnaire de files d'attente doivent indiquer dans l'appel MQOPEN pour ouvrir la file d'attente FILE.COT.ASSUR.AUTO sur le gestionnaire éloigné jupiter.gest.files.

DESCR ('File des demandes d'assurance auto des filiales')

Texte supplémentaire qui décrit l'utilisation de la file d'attente.

RNAME (FILE.COT.ASSUR.AUTO)

Nom de la file d'attente cible sur le gestionnaire de files d'attente éloigné. Il s'agit de la file cible réelle des messages envoyés par des applications qui indiquent le nom de file d'attente FILE.ELOIGNEE.CYAN. La file d'attente FILE.COT.ASSUR.AUTO doit être définie comme file d'attente locale sur le gestionnaire de files d'attente éloigné.

RQMNAME ('jupiter.gest.files')

Nom du gestionnaire de files d'attente éloigné auquel appartient la file d'attente cible FILE.COT.ASSUR.AUTO.

Création d'une définition locale de file d'attente éloignée

XMITQ ('INQUOTE.XMIT.QUEUE')

Nom de la file d'attente de transmission. Ce nom est facultatif ; s'il n'est pas défini, une file d'attente portant le même nom que le gestionnaire de files d'attente éloigné est utilisée.

Dans les deux cas, la file d'attente de transmission appropriée doit être définie en tant que file d'attente locale avec un attribut *Usage* indiquant qu'il s'agit d'une file d'attente de transmission (sous MQSC, cet attribut est USAGE(XMITQ)).

Autre mode d'insertion de messages dans une file d'attente éloignée

L'utilisation d'une définition locale de file d'attente éloignée ne constitue pas la seule méthode pour insérer des messages dans une file d'attente éloignée. Les applications peuvent indiquer, dans l'appel MQOPEN, le nom complet de la file d'attente, qui comprend le nom du gestionnaire de files d'attente éloigné. Dans ce cas, une définition locale d'une file d'attente éloignée n'est pas nécessaire. Cependant, cette méthode implique que les applications connaissent ou puissent avoir accès au nom du gestionnaire éloigné lors de l'exécution.

Utilisation d'autres commandes avec les files d'attente éloignées

Vous pouvez utiliser les commandes MQSC correspondantes pour afficher ou modifier les attributs d'un objet file d'attente éloignée, ou pour supprimer l'objet file d'attente éloignée. Par exemple :

```
* Display the remote queue's attributes.
* ALL = Display all attributes

DISPLAY QUEUE (FILE.ELOIGNEE.CYAN) ALL

* ALTER the remote queue to enable puts.
* This does not affect the destination queue,
* only applications that specify this remote queue.

ALTER QREMOTE (FILE.ELOIGNEE.CYAN) PUT(ENABLED)

* Delete this remote queue
* This does not affect the destination queue
* only its local definition

DELETE QREMOTE (FILE.ELOIGNEE.CYAN)
```

Remarque : Si vous supprimez une file d'attente éloignée, vous ne supprimez que la représentation locale de celle-ci. Vous ne supprimez pas réellement la file d'attente éloignée ou les messages qu'elle contient.

Création d'une file d'attente de transmission

Une file d'attente de transmission est une file d'attente locale utilisée lorsqu'un gestionnaire de files d'attente envoie des messages vers un gestionnaire éloigné via un canal de communication. Le canal offre une liaison unidirectionnelle au gestionnaire de files d'attente éloigné. Les messages sont mis en attente sur la file d'attente de transmission jusqu'à ce que le canal accepte de les transmettre.

Création d'une file d'attente de transmission

Lorsque vous définissez un canal, vous devez indiquer un nom de file d'attente de transmission à l'extrémité émettrice du canal de communication.

L'attribut *Usage* (USAGE sous MQSC) indique si la file d'attente est une file d'attente normale ou une file d'attente de transmission.

Files d'attente de transmission par défaut

Vous pouvez indiquer, si vous le désirez, une file d'attente de transmission dans un objet file d'attente éloignée, en utilisant l'attribut *XmitQName* (XMITQ sous MQSC). Si aucune file d'attente de transmission n'a été définie, le programme utilise une file d'attente par défaut. Lorsque des applications insèrent des messages dans une file d'attente éloignée, la file d'attente de transmission utilisée est celle qui porte le même nom que le gestionnaire de files d'attente cible, si elle existe. Si elle n'existe pas, c'est la file d'attente indiquée par l'attribut *DefaultXmitQ* (DEFXMITQ sous MQSC) sur le gestionnaire de files d'attente local qui est utilisée.

Par exemple, la commande MQSC suivante crée une file d'attente de transmission par défaut sur le gestionnaire `source.gest.files` pour les messages destinés au gestionnaire `cible.gest.files` :

```
DEFINE QLOCAL ('target.queue.manager') +
DESCR ('Default transmission queue for target qm') +
USAGE (XMITQ)
```

Les applications peuvent insérer des messages directement dans une file d'attente de transmission, ou indirectement, par exemple par le biais d'une définition de file d'attente éloignée. Reportez-vous également à la section «Création d'une définition locale de file d'attente éloignée» à la page 131.

Utilisation des définitions de files d'attente éloignées pour les alias

Outre la localisation d'une file d'attente sur un autre gestionnaire, une définition locale de file d'attente éloignée permet de définir :

- les gestionnaires de files d'attente alias,
- les files d'attente de réponses alias

Les deux types d'alias sont identifiés par le biais de la définition locale d'une file d'attente éloignée.

Tout comme dans la mise en file d'attente éloignée, les canaux appropriés doivent être configurés pour que le message arrive à destination.

Gestionnaire de files d'attente alias

La définition d'un alias est la procédure par laquelle le nom du gestionnaire de files d'attente cible, indiqué dans le message, est modifié par un gestionnaire de files d'attente sur l'itinéraire du message. Les gestionnaires de files d'attente alias sont importants car vous pouvez les utiliser pour contrôler la destination des messages à l'intérieur d'un réseau de gestionnaires de files d'attente.

Pour cela, il suffit de modifier la définition de file d'attente éloignée sur le gestionnaire au point de contrôle. L'application émettrice n'est pas informée du fait que le nom du gestionnaire de files d'attente indiqué est un alias.

Pour plus de détails sur les gestionnaires de files d'attente alias, reportez-vous au manuel *MQSeries Intercommunication*.

Files d'attente de réponse alias

De façon facultative, une application peut indiquer le nom d'une file d'attente de réponses lorsqu'elle insère un *message de demande* dans une file d'attente. Si l'application qui traite le message extrait le nom de la file d'attente de réponses, elle saura où envoyer le *message de réponse*, le cas échéant.

La définition d'une file d'attente de réponses alias est la procédure par laquelle une file d'attente de réponses, indiquée dans le message de demande, est modifiée par un gestionnaire de files d'attente sur l'itinéraire du message. L'application émettrice n'est pas informée du fait que le nom de la file d'attente de réponses indiqué est un alias.

Une file d'attente de réponses alias vous permet de modifier le nom de la file d'attente de réponses et, éventuellement, celui de son gestionnaire. Par conséquent, vous contrôlez l'itinéraire utilisé pour les messages de réponse.

Pour plus de détails sur les messages de demande, les messages de réponse et les files d'attente de réponses, reportez-vous au manuel *MQSeries Application Programming Reference*. Pour plus d'informations sur les files d'attente de réponses alias, reportez-vous au manuel *MQSeries Intercommunication*.

Conversion de données

Les données de message en format défini par MQSeries (ou "formats livrés d'origine"), peuvent être converties par le gestionnaire de files d'attente d'un jeu de caractères codés vers un autre, si les deux jeux de caractères codés se rapportent à une même langue ou à un groupe de langues similaires.

Par exemple, la conversion entre les jeux de caractères codés (CCSID) 850 et 500 est prise en charge car ces deux jeux se rapportent à des langues d'Europe occidentale.

Pour la conversion du caractère NL EBCDIC vers l'ASCII, reportez-vous à la section «Contenu du fichier de configuration MQSeries» à la page 190.

les conversions prises en charge sont définies dans l'annexe F : Tables de conversion des pages de code du manuel *MQSeries Application Programming Reference*.

Quand le gestionnaire de files d'attente ne peut pas convertir les messages dans des formats livrés d'origine

Le gestionnaire de files d'attente ne peut pas effectuer la conversion automatique des messages en format livré d'origine si leurs CCSID représentent des groupes de langues nationales différentes. Par exemple, la conversion entre les CCSID 850 et 1025 (jeu de caractères codés EBCDIC pour les langues utilisant l'alphabet cyrillique) n'est pas prise en charge car certains caractères de l'un des jeux de caractères codés ne peuvent pas être représentés dans l'autre. Si vous disposez d'un réseau de gestionnaires de files d'attente travaillant avec des langues nationales différentes et que la conversion des données n'est pas prise en charge pour certaines des combinaisons de jeux de caractères codés, vous pouvez mettre en place une conversion par défaut. Pour plus de détails sur cette opération, reportez-vous à la section «Conversion des données par défaut» à la page 136.

Fichier CCSID

Le fichier CCSID précise tout jeu de codes supplémentaire et toute conversion de donnée par défaut. Vous pouvez mettre à jour les informations enregistrées dans le fichier CCSID. Il est recommandé de le faire, par exemple, si une version suivante de votre système d'exploitation prend en charge des jeux de caractères codés supplémentaires. Pour ce faire, vous devez modifier le fichier (le fichier vous fournit des indications sur la procédure à suivre).

Conversion des données par défaut

Pour mettre en oeuvre la conversion de données par défaut, vous devez modifier le fichier CCSID pour spécifier des identificateurs CCSID EBCDIC et ASCII par défaut et pour spécifier les CCSID à utiliser par défaut. Le fichier comprend des indications sur la procédure à suivre pour le mettre à jour.

Si vous mettez le fichier CCSID à jour de sorte à mettre en oeuvre la conversion de données par défaut, vous devez relancer le gestionnaire de files d'attente pour que les modifications prennent effet.

Le processus de conversion de données par défaut est le suivant :

- Si la conversion entre les CCSID source et cible n'est pas prise en charge, mais que les CCSID des environnements source et cible sont tous deux de type EBCDIC ou de type ASCII, les données de caractères sont transmises vers l'application cible sans conversion.
- Si un CCSID représente un jeu de caractères codés ASCII et l'autre un jeu de caractères codés EBCDIC, MQSeries convertit les données à l'aide des CCSID de conversion de données définis comme CCSID par défaut dans le fichier CCSID.

Remarque : Il est recommandé de limiter les caractères à convertir à ceux dont la valeur de code est la même dans le jeu de caractères codés spécifié pour le message et dans le jeu de caractères codés par défaut. Cette condition sera, en règle générale, satisfaite si vous n'utilisez que le jeu de caractères autorisé pour les noms d'objets MQSeries. Les seules exceptions sont les CCSID EBCDIC 290, 930, 1279 et 5026 utilisés au Japon, dans lesquels les caractères en minuscules ont des codes différents de ceux utilisés dans les autres CCSID EBCDIC.

Conversion de messages en formats définis par l'utilisateur

Les messages aux formats définis par l'utilisateur ne peuvent pas être convertis d'un jeu de caractères codés vers un autre par le gestionnaire de files d'attente. Si des données en un format défini par l'utilisateur doivent être converties, vous devez fournir un exit de conversion de données pour chacun de ces formats. Il n'est pas recommandé d'utiliser les CCSID par défaut pour la conversion de données en formats définis par l'utilisateur, bien que ce soit possible. Pour plus d'informations sur la conversion de données en formats définis par l'utilisateur et sur l'écriture d'exits de conversion de données, reportez-vous au manuel *MQSeries - Guide de programmation d'applications*.

Modification du CCSID d'un gestionnaire de files d'attente

Il est recommandé d'arrêter et de redémarrer le gestionnaire de files d'attente en cas de modification du CCSID du gestionnaire, à l'aide de l'attribut CCSID de la commande ALTER QMGR.

Ceci permet d'être certain que toutes les applications en cours, dont le serveur de commandes et les programmes des canaux, sont arrêtés et redémarrés.

Cette opération est nécessaire car toutes les applications ouvertes pendant la modification du CCSID du gestionnaire de files d'attente continueront d'utiliser le CCSID existant.

Chapitre 8. Protection des objets MQSeries

Ce chapitre décrit les fonctions de contrôle de la sécurité de MQSeries pour Compaq NSK, et leur mise en oeuvre.

Ce chapitre contient les sections suivantes :

- «Présentation des ID utilisateur du groupe d'utilisateurs MQM»
- «Nécessité de protéger les ressources MQSeries»
- «Présentation du gestionnaire des droits d'accès aux objets (OAM)» à la page 140
- «Utilisation des commandes de l'OAM» à la page 144
- «Droits d'accès» à la page 147
- «Commande d'affichage des droits d'accès» à la page 147
- «Instructions concernant le gestionnaire des droits d'accès aux objets (OAM)» à la page 148
- «Tables de spécification des droits d'accès» à la page 151
- «Fichiers de droits d'accès» à la page 157

Nécessité de protéger les ressources MQSeries

Etant donné que les gestionnaires de files d'attente MQSeries gèrent le transfert d'informations potentiellement stratégiques, vous devez disposer d'un système de sécurité. Cela permet d'assurer la protection des ressources détenues et traitées par un gestionnaire de files d'attente contre un accès non autorisé qui pourrait entraîner la perte ou la divulgation d'informations. Dans un système sécurisé, une application ou un utilisateur non autorisé ne doit pas pouvoir accéder aux éléments ci-dessous, ni les modifier :

- Connexions à un gestionnaire de files d'attente
- Accès aux objets MQSeries tels que les files d'attente, clusters, canaux et processus.
- Commandes destinées à l'administration du gestionnaire de files d'attente (y compris les commandes MQSC et PCF)
- Messages MQSeries
- Informations de contexte associées aux messages

C'est à vous qu'il revient d'attribuer les droits adéquats aux utilisateurs voulus.

Présentation des ID utilisateur du groupe d'utilisateurs MQM

Toutes les ressources de gestionnaire de files d'attente fonctionnent avec l'identificateur de droits du groupe MQM.

Pour pouvoir accéder à MQSeries pour Compaq NSK, votre ID utilisateur doit correspondre à un principal MQSeries. Au début, seul l'ID utilisateur ayant créé le gestionnaire de files d'attente a le principal MQSeries mqm. Utilisez la commande **altmqusr** pour créer un principal pour chaque utilisateur qui aura accès à MQSeries. Le principal et l'utilisateur doivent tous deux être uniques. Pour afficher les principaux et leurs propriétés pour un gestionnaire de files d'attente, utilisez la commande **dspmqsar**.

Si votre ID utilisateur dispose de l'identificateur de droits MQSeries pour Compaq NSK MQM et si un principal MQSeries a été créé pour votre ID utilisateur, vous

Présentation des ID utilisateur

bénéficiez de tous les droits, pour toutes les ressources MQSeries. Votre ID utilisateur *doit* disposer de l'identificateur de droits du groupe MQM pour pouvoir utiliser toutes les commandes de contrôle MQSeries pour Compaq NSK (à l'exception de **crtmqcvx**). Ce droit d'accès est notamment nécessaire pour :

- utiliser la fonction **runmqsc** pour exécuter les commandes MQSC,
- administrer les droits d'accès sous MQSeries pour Compaq NSK à l'aide de la commande **setmqaut**.

Si vous émettez des commandes de canal vers des gestionnaires de files d'attente situés sur un système Compaq NSK éloigné, vérifiez que votre ID utilisateur dispose de l'identificateur de droits MQM Compaq NSK sur le système cible. Vous trouverez une liste des commandes de canal PCF et MQSC à la section «Sécurité des commandes de canal» à la page 150.

En revanche, il n'est pas nécessaire que votre ID utilisateur appartienne au groupe d'identificateur MQM pour lancer :

- les commandes PCF (y compris les commandes d'échappement PCF) à partir d'un programme d'administration
- les appels MQI à partir d'un programme d'application

Remarque : Les autorisations pour le principal mqm (et les groupes MQM) sont importantes car le principal mqm est utilisé par les composants du gestionnaire de files d'attente interne eux-mêmes pour accéder à des ressources protégées. En cas de suppression des autorisations du principal mqm ou du groupe MQM à des objets faisant partie du gestionnaire de files d'attente, ou de suppression du principal mqm lui-même, vous risquez de vous trouver confronté à un gestionnaire qui ne peut plus être administré ou même plus du tout être utilisé.

Informations complémentaires

Pour plus d'informations sur :

- les jeux de commandes MQSeries pour Compaq NSK, voir le «Chapitre 3. Utilisation des jeux de commandes MQSeries» à la page 29
- les commandes de contrôle MQSeries pour Compaq NSK, voir le «Chapitre 17. Commandes de contrôle MQSeries» à la page 251
- les commandes PCF et commandes d'échappement PCF, consultez le manuel *MQSeries Programmable System Management*.
- les appels MQI, consultez les manuels *MQSeries - Guide de programmation d'applications* et *MQSeries Application Programming Reference*.

Présentation du gestionnaire des droits d'accès aux objets (OAM)

Par défaut, l'accès aux ressources du gestionnaire de files d'attente est contrôlé par un composant optionnel du service. Le composant du service d'autorisation fourni avec MQSeries pour Compaq NSK est automatiquement installé et activé pour chaque gestionnaire de files d'attente que vous créez (sauf indication contraire). Dans ce chapitre, le gestionnaire des droits d'accès aux objets fourni avec MQSeries est désigné par le terme OAM.

L'OAM est un *composant optionnel* du service d'autorisation. Le fait que l'OAM soit un composant optionnel vous permet de :

- remplacer l'OAM livré avec le produit par votre propre système d'autorisation à l'aide de l'interface fournie ;

Gestionnaire des droits d'accès aux objets.

- compléter les fonctions de l'OAM par celles de votre propre système d'autorisation en utilisant l'interface fournie ;
- supprimer ou désactiver l'OAM et utiliser le produit sans système d'autorisation.

Pour plus d'informations sur les services optionnels, reportez-vous au manuel *MQSeries Programmable System Management*.

L'OAM gère les autorisations des utilisateurs à manipuler les objets MQSeries, tels que les files d'attente, les définitions de processus et les canaux. Il offre également une interface de commande par laquelle vous pouvez accorder ou révoquer les droits d'accès d'un groupe d'utilisateurs à un objet. C'est l'OAM qui décide de l'accès à une ressource, et le gestionnaire de files d'attente respecte sa décision. Si l'OAM ne peut pas prendre une décision, le gestionnaire de files d'attente empêche l'accès à la ressource concernée.

Mode de fonctionnement de l'OAM

L'OAM utilise les ID groupe et utilisateur et les dispositifs de sécurité du système d'exploitation Compaq NSK. Les utilisateurs ne peuvent accéder aux objets du gestionnaire de files d'attente que s'ils possèdent les droits requis.

Gestion de l'accès par le biais de groupes d'utilisateurs

La gestion des droits d'accès aux ressources MQSeries est basée sur les *groupes* Compaq NSK. L'OAM gère les droits d'accès au niveau des groupes.

Dans les interfaces de commandes, on utilise les principaux MQSeries plutôt que les ID utilisateur. C'est parce que les droits accordés à un ID utilisateur peuvent également être accordés à d'autres entités (par exemple, un programme d'application qui émet des appels MQI ou un programme d'administration qui émet des commandes PCF). Dans ces cas, le principal associé au programme n'est pas nécessairement l'ID utilisateur utilisé lors du lancement du programme.

Les ID utilisateur Compaq NSK peuvent prendre la forme <groupe>.<nom> où le groupe et le nom peuvent tous deux comporter jusqu'à 8 caractères, tandis que les noms principaux MQSeries peuvent en comporter 12. De plus, le point (.) est interdit pour les ID utilisateur sur d'autres plateformes. Dans MQSeries pour Compaq NSK, la base de données de principaux contient une correspondance des ID utilisateur Compaq NSK pour les noms des principaux MQSeries de 12 caractères ou moins.

Lorsqu'un utilisateur appartient à plus d'un groupe

L'autorisation détenue par l'utilisateur est l'ensemble des autorisations de tous les groupes auxquels il appartient et l'autorisation par défaut pour tous les utilisateurs. Vous pouvez utiliser la commande **setmqaut** pour définir les droits d'accès d'un groupe donné.

Remarque : Toutes les modifications effectuées à l'aide de cette commande ont un effet immédiat, à moins que l'objet ne soit utilisé. Si c'est le cas, la modifications prendra effet à l'ouverture suivante de l'objet.

Ensemble de groupes et groupe principal

La gestion des droits d'accès aux ressources MQSeries est fondée sur les groupes d'utilisateurs Compaq NSK. Lorsque SAFEGUARD est en cours de fonctionnement, un ID utilisateur Compaq NSK peut être associé à plus d'un groupe et le principal MQSeries correspondant est donc aussi associé à ces groupes. Le groupe principal est toujours le groupe d'administration Compaq. Les

Gestionnaire des droits d'accès aux objets.

groupes secondaires sont configurés par la création de groupes de partage de fichiers SAFEGUARD et l'association d'un ID utilisateur Compaq NSK avec ce groupe de partage de fichiers.

L'OAM gère les droits d'accès au niveau des groupes plutôt qu'au niveau des principaux individuels. La conversion de principaux en noms de groupes est effectuée par l'OAM à l'aide de la base de données de principaux et des fonctions Compaq NSK et SAFEGUARD ; les opérations de l'OAM sont effectuées au niveau des groupes. Vous pouvez toutefois afficher les droits d'accès d'un principal individuel.

Protection de ressources avec l'OAM

L'OAM peut contrôler :

- L'accès aux objets MQSeries via l'interface MQI. Lorsqu'un programme tente d'accéder à un objet, l'OAM vérifie si l'ID utilisateur auteur de la demande possède les droits d'accès (via le groupe d'utilisateurs) nécessaires à l'exécution de l'opération demandée.

En particulier, cela signifie que les files d'attente et les messages de ces files peuvent être protégés contre les accès non autorisés.

- L'autorisation d'utiliser les commandes MQSC. Seuls les membres du groupe d'utilisateurs mqm et les personnes autorisées par **setmqaut** peuvent exécuter des commandes d'administration de gestionnaire de files d'attente, comme la création d'une file d'attente.
- L'autorisation d'utiliser les commandes de contrôle. Seuls les membres du groupe mqm peuvent exécuter des commandes de contrôle, comme la création d'un gestionnaire de files d'attente ou le démarrage d'un serveur de commandes.
- L'autorisation d'utiliser les commandes PCF.

Différents groupes d'utilisateurs peuvent obtenir différents droits d'accès pour le même objet. Par exemple, pour une file d'attente particulière, un groupe peut être autorisé à exécuter des opérations d'insertion et d'extraction, tandis que d'autres utilisateurs n'auront qu'un droit de consultation sur la file d'attente (MQGET avec l'option de consultation). De même, certains groupes pourront posséder un droit d'insertion et d'extraction sur une file d'attente, tout en ne possédant pas le droit de modifier ou de supprimer la file.

Utilisation de groupes pour les droits d'accès

L'utilisation de principaux de groupes (plutôt qu'individuels) pour les droits d'accès réduit la quantité d'opérations administratives à effectuer. Généralement, un même type d'accès est requis par plusieurs principaux. Par exemple, vous pouvez définir un groupe se composant d'utilisateurs finaux qui souhaitent utiliser une application particulière. Les nouveaux utilisateurs peuvent alors se voir accorder l'accès à cette application simplement en ajoutant le groupe approprié à leur ID utilisateur Compaq NSK. A moins que MQSeries soit installé sur un système utilisant SAFEGUARD pour créer des groupes de partage de données, chaque ID utilisateur ne peut être associé qu'à un groupe principal.

En l'absence de SAFEGUARD, le nombre de principaux par groupe est limité à 255. Avec le partage de fichiers SAFEGUARD, on peut atteindre 65535 principaux par groupe. SAFEGUARD permet aussi à un principal d'avoir ses propres droits d'accès. Par exemple, même si le groupe d'administration de SAFEGUARD n'a pas le moindre droit d'accès, tout membre appartenant au groupe peut avoir son

Gestionnaire des droits d'accès aux objets.

propre droit d'accès (par exemple, pour le Principal A, le membre TESTSEC1 du Groupe MQSEC ne peut qu'Insérer, et pour le Principal B, le membre TESTSEC2 du Groupe ne peut qu'Extraire).

En l'absence de SAFEGUARD, il faudrait créer deux groupes (deux ID utilisateur Compaq NSK dans des groupes différents, par exemple MQ.TEST (connexion/insertion) et MQ1.TEST (connexion/extraction)) pour cela, car les droits d'accès du groupe sont associés au principal (ID utilisateur Compaq NSK MQM.MANAGER groupe MQM (connexion) principal mqm et ID utilisateur Compaq NSK MQM.APPS groupe MQM principal apps (insertion/extraction)). L'ensemble des droits d'accès du groupe (connexion pour mqm) et du principal (extraction/insertion pour apps) sont ajoutés les uns aux autres (connexion/extraction/insertion). La validation des droits d'accès des utilisateurs se fait alors sur l'ensemble des données. Si une application entreprend un MQGET alors qu'elle est enregistrée sous l'ID utilisateur MQ.TEST, l'opération est rejetée avec un message MQRC_NOT_AUTHORIZED (2035). Un enregistrement MQ1.TEST aurait permis cette opération. Tout utilisateur enregistré sous un ID utilisateur Compaq du groupe MQM aura aussi pu réaliser l'opération. Une fois un droits d'accès accordé à un principal, tous les principaux du groupe ont ce droits d'accès.

Principal	IDutilisateurCompaq	Group	TypeGroup
Securité			
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

Figure 39. Utilisation de groupes avec SAFEGUARD pour accorder un droit d'accès.

Dans la figure 39, tout utilisateur enregistré sous le groupe MQM aurait un droit d'accès pour la connexion, l'extraction, l'insertion et la recherche. L'utilisateur MQ.TEST en connexion et insertion, l'utilisateur MQ1.TEST en connexion et extraction. Les utilisateurs du groupe MQSEC auraient le droit d'accès du membre du groupe de partage de fichiers SAFEGUARD qui a été accordé.

Essayez de réduire le nombre de groupes au minimum, en répartissant par exemple les utilisateurs entre un groupe d'utilisateurs d'applications et un groupe d'administrateurs.

Désactivation du gestionnaire des droits d'accès aux objets (OAM)

Par défaut, l'OAM est activé. Vous pouvez le désactiver en définissant le nom logique MQSNOAUT avant la création du gestionnaire de files d'attente :

```
PARAM MQSNOAUT 1
```

Gestionnaire des droits d'accès aux objets.

Cependant, si vous utilisez cette commande, vous ne pouvez généralement pas redémarrer l'OAM. Il vaut mieux activer l'OAM et s'assurer que tous les utilisateurs et les applications accèdent aux ressources via un ID utilisateur approprié. Vous pouvez également désactiver l'OAM pour pouvoir effectuer des tests, simplement en retirant la strophe correspondant au service d'autorisation dans le fichier de configuration du gestionnaire de files d'attente (QMINI), ou en désactivant MQAUTH dans la strophe Droits d'accès de QMINI, comme il est décrit à la section «Fichier de configuration du gestionnaire de files d'attente (QMINI)» à la page 191.

Remarque : Il ne suffit pas de spécifier PARAM MQSNOAUT 0 pour activer l'OAM. La variable d'environnement ne doit pas exister dans l'environnement si vous souhaitez réactiver l'OAM.

Utilisation des commandes de l'OAM

L'OAM fournit une interface pour l'attribution et la révocation des droits d'accès. Avant d'utiliser ces commandes, vous devez disposer du droit d'accès approprié (votre ID utilisateur doit appartenir au groupe MQM Compaq NSK). (ce groupe doit avoir été installé avant l'installation de MQSeries pour Compaq NSK.)

Si votre ID utilisateur fait partie du groupe MQM, vous disposez d'un droit d'accès de 'super utilisateur' sur le gestionnaire de files d'attente. Vous êtes donc autorisé à émettre les requêtes ou commandes MQI avec votre ID utilisateur.

L'OAM comporte quatre commandes que vous pouvez appeler depuis votre TACL Compaq NSK pour gérer les droits d'accès des utilisateurs. Il s'agit de :

- **altmqsr** (créer, supprimer, ou modifier un principal MQSeries)
- **dspmqsr** (affichage d'un principal)
- **setmqaut** (définition ou réinitialisation des droits)
- **dspmqaut** (affichage des droits d'accès)

La vérification du droit d'accès se produit dans les appels MQCONN, MQOPEN, MQPUT1 et MQCLOSE. Elle s'effectue lors de la première instance de ces appels. Les droits d'accès ne sont modifiés qu'à la réinitialisation suivante (fermeture et réouverture) de l'objet. Toute modification apportée aux droits d'accès d'un objet à l'aide de **setmqaut** ne prend effet que lors de la réinitialisation de cet objet.

Informations à préciser aux commandes

Les commandes des droits d'accès s'appliquent au gestionnaire de files d'attente indiqué. Si vous n'en indiquez aucun, elles s'appliquent au gestionnaire de files d'attente par défaut. Dans ces commandes, vous devez indiquer un objet unique (c'est-à-dire spécifier le nom de l'objet et son type). Vous devez également indiquer l'utilisateur ou le nom de groupe auquel s'appliquent les droits d'accès.

Listes de droits d'accès

Dans la commande `setmqaut`, vous spécifiez une liste des droits d'accès. Il s'agit d'un moyen rapide permettant d'indiquer si les droits doivent être accordés ou révoqués, et les ressources auxquelles ces modifications des droits d'accès s'appliquent. Chaque droit figurant dans la liste est spécifié sous forme d'un mot clé en minuscules, précédé du signe plus + ou moins -. Utilisez le signe + pour ajouter le droit spécifié et le signe - pour le supprimer. Vous pouvez spécifier n'importe quel nombre de droits dans une même commande. Par exemple :

```
+browse -get +put
```

Utilisation de la commande `altmqsr`

Si vous possédez les droits requis, vous pouvez utiliser la commande `altmqsr` pour créer un principal MQSeries et l'associer à un ID utilisateur Compaq NSK (ou alias SAFEGUARD). L'exemple suivant indique comment utiliser la commande `altmqsr` :

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

Dans cet exemple :

Le terme...	désigne...
saturne.gest.files	Nom du GFA
MQPRINCIPAL	Nom principal à créer
MQM.MQUSER	ID utilisateur Compaq NSK

Pour obtenir une description plus formelle de cette commande, reportez-vous à la section «`altmqsr` (modification des informations sur les utilisateurs MQSeries)» à la page 259.

La commande `altmqsr` peut aussi être utilisée pour retirer un principal (et ainsi révoquer tous les droits d'accès à MQSeries). Par exemple :

```
altmqsr -m saturne.gest.file -p MQPRINCIPAL -remove
```

Dans cet exemple :

Le terme...	désigne...
saturne.gest.files	Nom du GFA
MQPRINCIPAL	Nom principal à supprimer
-remove	Instructions pour la suppression du principal

Utilisation de la commande `dspmqsqr`

La commande `dspmqsqr` affiche le contenu de la base de données de principaux, en plus des groupes d'administration et de partage de fichiers Compaq NSK auxquels les ID utilisateur correspondant à chaque principal MQSeries

Utilisation des commandes de l'OAM

appartiennent. Le paramètre -p permet de n'afficher que le principal choisi. Par exemple :

```
dspmqsqr -m saturne.gest.files -p PRINCIPALMQ
```

Dans cet exemple :

Le terme...	désigne...
saturne.gest.files	Nom du GFA
MQPRINCIPAL	Nom principal à afficher

Pour obtenir une description plus formelle de cette commande, reportez-vous à la section «dspmqsqr (affichage des informations utilisateur MQSeries)» à la page 282.

Utilisation de la commande setmqaut

Si vous possédez les droits requis, vous pouvez utiliser la commande **setmqaut** pour accorder ou révoquer le droit d'un principal ou d'un groupe d'utilisateurs à accéder à un objet particulier. L'exemple suivant indique comment utiliser la commande **setmqaut** :

```
setmqaut -m saturne.gest.files -t file -n FILE.LOCALE.ROUGE -g GroupA  
+browse -get +put
```

Dans cet exemple :

Le terme...	désigne...
saturne.gest.files	Nom du GFA
file	Type d'objet
FILE.LOCALE.ROUGE	Nom d'objet
GroupA	L'ID du groupe auquel les droits d'accès sont accordés.
+browse -get +put	La liste des droits d'accès qui s'appliquent à la file d'attente indiquée. Vous ne devez pas laisser d'espaces entre les signes '+' ou '-' et le mot clé.

La liste des droits d'accès indique les droits à accorder, où :

Le terme...	désigne...
+browse	Ajoute les droits d'accès permettant de consulter (MQGET avec option de consultation).
-get	Supprime les droits d'accès permettant d'extraire (MQGET) les messages de la file.
+put	Ajoute les droits d'accès permettant d'insérer (MQPUT) des messages dans la file.

Les applications lancées avec des ID utilisateurs appartenant au groupe d'utilisateurs Compaq NSK GroupeA possèdent ces autorisations.

Utilisation des commandes de l'OAM

Par exemple, la commande suivante révoque le droit d'accès d'insertion dans la file d'attente MaFile aux groupes GroupA et GroupB.

```
setmqaut -m saturne.gest.files -t file -n "MaFile" -g GroupA -g GroupB -put
```

Pour plus de détails sur une définition formelle de la commande et de sa syntaxe, reportez-vous à la section «setmqaut (définition ou réinitialisation des droits d'accès)» à la page 300.

Commandes de droits d'accès et services optionnels

La commande **setmqaut** comporte un paramètre supplémentaire qui indique le nom du service optionnel auquel la modification s'applique. Vous devez indiquer ce paramètre lorsque plusieurs composants optionnels fonctionnent en même temps. Par défaut, ce n'est pas le cas. Si le paramètre est omis, la mise à jour est effectuée sur le premier composant optionnel de ce type (le cas échéant). Par défaut, c'est l'OAM fourni.

Droits d'accès

Les droits d'accès définis par la liste correspondante associée à la commande **setmqaut** se répartissent en plusieurs catégories :

- Droits d'accès liés aux appels MQI
- Droits d'accès liés aux commandes d'administration
- Droits de contexte
- Droits d'accès généraux, c'est-à-dire droits d'accès pour les appels MQI, pour les commandes ou pour les deux.

Chaque droit d'accès est spécifié par un mot clé utilisé avec les commandes **setmqaut** et **dspmqaut**. Celles-ci sont décrites dans la section «setmqaut (définition ou réinitialisation des droits d'accès)» à la page 300.

Commande d'affichage des droits d'accès

Vous pouvez utiliser la commande **dspmqaut** pour visualiser les droits d'accès dont dispose un principal ou un groupe sur un objet particulier. Les indicateurs possèdent la même signification que ceux de la commande **setmqaut**. Vous ne pouvez afficher les droits d'accès que d'un seul principal ou groupe à la fois. Pour obtenir une description plus formelle de cette commande, reportez-vous à la section «dspmqaut (affichage des droits d'accès)» à la page 273.

Par exemple, la commande suivante affiche les droits dont le groupe GpAdmin dispose sur une définition de processus Annuités du gestionnaire de files d'attente GestFiles1.

```
dspmqaut -m GestFiles1 -t process -n "Annuités" -g "GpAdmin"
```

Les mots clés affichés en retour de cette commande indiquent les droits d'accès activés.

Instructions concernant le gestionnaire des droits d'accès aux objets (OAM)

Certaines opérations sont particulièrement sensibles et doivent être limitées à des utilisateurs privilégiés. Par exemple :

- la création, la suppression, le lancement et l'arrêt des gestionnaires de files d'attente
- l'accès à certaines files d'attente spéciales, telles que les files d'attente de transmission ou la file d'attente de commandes SYSTEM.ADMIN.COMMAND.QUEUE
- les programmes qui utilisent des options de contexte totalement MQI.
- les création et copie de files d'attente d'applications

ID utilisateur

Le groupe spécial appelé MQM que vous créez est conçu pour n'être utilisé que par les administrateurs du produit. Il ne doit en aucun cas être accessible aux utilisateurs non privilégiés.

Volumes du gestionnaire de files d'attente

Le volume comportant des files d'attente et d'autres données du gestionnaire de files d'attente est propre au produit. Les objets de ce répertoire sont soumis à des droits d'accès utilisateur Compaq NSK, liés à leurs droits d'accès OAM. Les commandes Compaq NSK standard ne peuvent pas être utilisées pour accorder ou révoquer des droits d'accès aux ressources MQI car :

- les objets MQSeries ne sont pas forcément identiques au nom d'objet système correspondant. Pour plus d'informations, reportez-vous à la section «Structure du volume» à la page 62.
- Les objets MQSeries ne correspondent pas forcément aux paramètres de sécurité NSK de l'objet.

Files d'attente

Les droits d'accès à une file d'attente dynamique sont basés sur ceux de la file d'attente modèle dont elle est dérivée, sans pour autant être identiques à ces derniers.

Pour les files d'attente alias et les files d'attente éloignées, le droit est celui de l'objet lui-même, et non celui de la file d'attente qui correspond à la file d'attente éloignée ou alias. Il est donc possible d'autoriser l'accès d'un principal à une file d'attente alias qui correspond à une file d'attente locale à laquelle le principal ne peut accéder.

Vous devez limiter les droits de création des files d'attente aux utilisateurs privilégiés. Si vous ne le faites pas, certains utilisateurs peuvent contourner le contrôle d'accès simplement en créant un alias.

Droits d'accès d'utilisateur secondaire

Les droits d'accès d'utilisateur secondaire permettent de déterminer si un ID utilisateur peut utiliser les droits d'un autre ID lors de l'accès à un objet MQSeries. Cela est important lorsqu'un serveur reçoit des demandes d'un programme et doit vérifier les droits de ce programme. Le serveur peut posséder les droits requis, mais il doit tout de même savoir si le programme a le droit d'effectuer les actions qu'il a demandées.

Par exemple :

- Un programme serveur fonctionnant sous l'ID utilisateur SERVPAY récupère dans une file d'attente un message de demande inséré par l'ID utilisateur UTIL1.
- Lorsque le programme serveur récupère le message, il traite la demande et insère la réponse dans la file d'attente des messages de réponse spécifiée dans le message de demande.
- Au lieu d'utiliser son propre ID utilisateur (SERVPAY) pour autoriser l'ouverture de la file d'attente de réponses, le serveur peut indiquer un autre ID utilisateur, dans ce cas, UTIL1. Dans cet exemple, vous pouvez utiliser le droit d'accès d'utilisateur secondaire pour contrôler si SERVPAY est autorisé à indiquer UTIL1 comme ID utilisateur secondaire lorsqu'il ouvre la file d'attente de réponses.

L'ID d'utilisateur secondaire est indiqué dans la zone *AlternateUserId* du descripteur d'objet.

L'ID utilisateur et l'ID utilisateur secondaire doivent être définis comme principaux correspondant aux entrées de la base de données de principaux associée à un ID utilisateur Compaq NSK pour que les droits d'accès soient accordés.

Remarque : Vous pouvez utiliser les ID utilisateur secondaire pour n'importe quel objet MQSeries. Leur utilisation n'affecte pas l'ID utilisateur employé par les autres gestionnaires de ressources.

Droits de contexte

Le contexte est constitué des informations qui s'appliquent à un message particulier. Il est contenu dans le descripteur de message, MQMD, qui fait partie du message. Les informations de contexte se divisent en deux sections :

La section Identité

Cette section indique l'auteur du message. Elle se compose des zones ci-après :

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

La section d'origine

Cette section indique d'où le message provient et à quel moment il a été inséré en file d'attente. Elle se compose des zones ci-après :

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

Les applications peuvent fournir des données de contexte lorsqu'un appel MQOPEN ou MQPUT est effectué. Ces données peuvent être générées par l'application, transmises à partir d'un autre message ou générées par le gestionnaire de files d'attente par défaut. Par exemple, les données de contexte peuvent être utilisées par les programmes serveur pour vérifier l'identité du demandeur, en vérifiant si le message provient d'une application fonctionnant sous un ID utilisateur autorisé.

Un programme serveur peut utiliser *UserIdentifier* pour déterminer l'ID utilisateur d'un utilisateur secondaire. *UserIdentifier* doit être défini comme principal correspondant à une entrée de la base de données de principaux.

Caractéristiques OAM

Vous pouvez utiliser les droits d'accès pour contrôler si l'utilisateur peut indiquer une des options de contexte sur un appel MQOPEN ou MQPUT1. Pour plus d'informations sur les options de contexte, consultez le manuel *MQSeries - Guide de programmation d'applications*. Vous trouverez les descriptions des zones de descripteur de message relatives au contexte dans le manuel *MQSeries Application Programming Reference*.

Gestion éloignée de la sécurité

Pour la gestion à distance de la sécurité, prenez en compte les éléments suivants :

Droit d'insertion

Pour assurer la sécurité au niveau des gestionnaires de files d'attente, vous pouvez spécifier le droit d'insertion à utiliser lorsqu'un canal reçoit un message envoyé par un autre gestionnaire de files d'attente.

Précisez l'attribut de canal PUTAUT comme suit :

DEF ID utilisateur par défaut. Il s'agit de l'ID utilisateur sous lequel l'agent du canal de communication s'exécute.

CTX ID utilisateur dans le contexte de message.

Dans les deux cas, l'ID utilisateur doit être défini comme principal correspondant à une entrée de la base de données de principaux.

Files d'attente de transmission

Les gestionnaires de files d'attente insèrent automatiquement les messages éloignés dans la file d'attente de transmission. Aucun droit particulier n'est nécessaire pour cette opération. Cependant, l'insertion directe d'un message dans une file d'attente de transmission nécessite un droit particulier. Reportez-vous au tableau 3 à la page 152.

Exits de canal

Les exits de canal permettent d'accroître la sécurité.

Pour plus d'informations, reportez-vous au manuel *MQSeries Intercommunication*.

Sécurité des commandes de canal

Les commandes de canal peuvent être émises sous forme de commandes PCF, de commandes MQAI, de commandes MQSC ou de commandes de contrôle.

Commandes PCF

Vous pouvez émettre des commandes de canal PCF en envoyant un message PCF à la file d'attente SYSTEM.ADMIN.COMMAND.QUEUE située sur un système Compaq NSK éloigné. L'ID utilisateur, tel qu'il est spécifié dans le descripteur du message PCF associé, doit être défini comme principal correspondant à une entrée de la base de données de principaux associée à un ID utilisateur Compaq NSK appartenant au groupe MQM sur le système cible. Ces commandes sont les suivantes :

- ChangeChannel
- CopyChannel
- CreateChannel
- DeleteChannel
- PingChannel
- ResetChannel
- StartChannel
- StopChannel
- ResolveChannel

Consultez le manuel *MQSeries Programmable System Management* pour plus de détails sur la sécurité des commandes PCF.

Commandes de canal MQSC

Vous pouvez émettre des commandes de canal MQSC vers un système Compaq NSK éloigné en envoyant la commande directement dans un message d'échappement PCF ou en émettant la commande via **runmqsc** en mode indirect. L'ID utilisateur, tel qu'il est spécifié dans le descripteur du message PCF associé, doit être défini comme principal correspondant à une entrée de la base de données de principaux associée à un ID utilisateur Compaq NSK appartenant au groupe MQM sur le système cible. (Les commandes PCF sont implicites dans les commandes MQSC émises à partir de **runmqsc** en mode indirect.) Ces commandes sont les suivantes :

- ALTER CHANNEL
- DEFINE CHANNEL
- DELETE CHANNEL
- PING CHANNEL
- RESET CHANNEL
- START CHANNEL
- START CHINIT
- STOP CHANNEL
- RESOLVE CHANNEL

En ce qui concerne les commandes MQSC émises à partir de **runmqsc**, l'ID utilisateur du message PCF est normalement celui de l'utilisateur en cours.

Tables de spécification des droits d'accès

Les tableaux de spécification des droits d'accès présentés à partir de la page 152 définissent le fonctionnement des autorisations et les restrictions qui s'appliquent. Ces tableaux concernent les situations suivantes :

- Applications qui émettent des appels MQI
- Programmes d'administration qui émettent des commandes MQSC sous forme de PCF d'échappement
- Programmes d'administration qui émettent des commandes PCF

Dans cette section, les informations sont présentées sous forme de tableaux précisant les éléments suivants :

Action à exécuter	Option MQI, commande MQSC ou commande PCF.
Objet du contrôle d'accès	File d'attente, processus ou gestionnaire de files d'attente
Droit requis	Exprimés sous forme d'une constante 'MQZAO_'

Les constantes préfixées par MQZAO_ correspondent aux mots clés de la liste des droits d'accès de la commande **setmqaut** pour cette entrée particulière. Par exemple, MQZAO_BROWSE correspond au mot clé +browse. De même, le mot clé MQZAO_SET_ALL_CONTEXT correspond au mot clé +setall etc. Ces constantes sont définies dans le fichier d'en-tête CMQZCH, dans le sous-volume ZMQSLIB, qui est fourni avec le produit.

Droits d'accès MQI

Une application n'est autorisée à émettre certains appels et options MQI que si l'ID utilisateur sous lequel elle fonctionne (ou dont elle possède les droits) s'est vu affecter les autorisations requises.

Tables de spécification des droits d'accès

Quatre appels MQI (MQCONN, MQOPEN, MQPUT1 et MQCLOSE) peuvent nécessiter un contrôle des droits.

Pour MQOPEN et MQPUT1, la vérification des droits s'effectue sur le nom de l'objet en cours d'ouverture et non sur le(s) nom(s) résultant de la résolution d'un nom. Par exemple, une application peut se voir accorder le droit d'ouvrir une file d'attente alias sans toutefois posséder le droit d'ouvrir la file d'attente de base à laquelle se rapporte l'alias. Le contrôle est alors effectué sur la première définition rencontrée durant la procédure de résolution de nom (définition différente d'un gestionnaire de files d'attente alias, sauf si la définition de cet alias a été directement ouverte). Le nom doit donc apparaître dans la zone *ObjectName* du descripteur d'objet. Les droits d'accès sont toujours nécessaires pour l'objet ouvert. Dans certains cas, des droits supplémentaires indépendants de la file d'attente sont nécessaires (ils sont obtenus via un droit sur l'objet gestionnaire de files d'attente).

Le tableau 3 résume les droits nécessaires pour chaque appel.

Tableau 3. Droits d'accès nécessaires pour les appels MQI

Droits requis pour :	Objet de file d'attente	Objet processus	Gestionnaire de file d'attente	Listes de noms
option MQCONN	Non applicable	Non applicable	MQZAO_CONNECT	Non applicable
Option MQOPEN				
MQOO_INQUIRE	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)
MQOO_BROWSE	MQOO_BROWSE	Non applicable	Pas de contrôle	Non applicable
MQOO_INPUT_*	MQZAO_INPUT	Non applicable	Pas de contrôle	Non applicable
MQOO_SAVE_ALL_CONTEXT (3)	MQZAO_INPUT	Non applicable	Pas de contrôle	Non applicable
MQOO_OUTPUT (file d'attente normale) (4)	MQOO_OUTPUT	Non applicable	Pas de contrôle	Non applicable
MQOO_PASS_IDENTITY_CONTEXT (5)	MQZAO_PASS_IDENTITY_CONTEXT	Non applicable	Pas de contrôle	Non applicable
MQOO_PASS_ALL_CONTEXT (5, 6)	MQZAO_PASS_ALL_CONTEXT	Non applicable	Pas de contrôle	Non applicable
MQOO_SET_IDENTITY_CONTEXT (5, 6)	MQZAO_SET_IDENTITY_CONTEXT	Non applicable	MQZAO_SET_IDENTITY_CONTEXT	Non applicable
MQOO_SET_ALL_CONTEXT (5, 8)	MQZAO_SET_ALL_CONTEXT	Non applicable	MQZAO_SET_ALL_CONTEXT (7)	Non applicable
MQOO_OUTPUT (File d'attente de transmission) (9)	MQZAO_SET_ALL_CONTEXT	Non applicable	MQZAO_SET_ALL_CONTEXT (7)	Non applicable
MQOO_SET	MQZAO_SET	Non applicable	Pas de contrôle	Non applicable
MQOO_ALTERNATE_USER_AUTHORITY	(10)	(10)	MQZAO_ALTERNATE_USER_AUTHORITY (10, 11)	Non applicable
Option MQPUT1				

Tables de spécification des droits d'accès

Tableau 3. Droits d'accès nécessaires pour les appels MQI (suite)

Droits requis pour :	Objet de file d'attente	Objet processus	Gestionnaire de file d'attente	Listes de noms
MQPMO_PASS_IDENTITY_CONTEXT	MQZAO_PASS_IDENTITY_CONTEXT (12)	Non applicable	Pas de contrôle	Non applicable
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT (12)	Non applicable	Pas de contrôle	Non applicable
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT (12)	Non applicable	MQZAO_SET_IDENTITY_CONTEXT (7)	Non applicable
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT (12)	Non applicable	MQZAO_SET_ALL_CONTEXT (7)	Non applicable
(File d'attente de transmission) (9)	MQZAO_SET_ALL_CONTEXT	Non applicable	MQZAO_SET_ALL_CONTEXT (7)	Non applicable
MQPMO_ALTERNATE_USER_AUTHORITY	(13)	Non applicable	MQZAO_ALTERNATE_USER_AUTHORITY (11)	Non applicable
Option MQCLOSE				
MQCO_DELETE	MQZAO_DELETE (14)	Non applicable	Non applicable	Non applicable
MQCO_DELETE_PURGE	MQZAO_DELETE (14)	Non applicable	Non applicable	Non applicable

Remarques spécifiques :

- Si une file d'attente modèle est en cours d'ouverture :
 - Le droit MQZAO_DISPLAY est nécessaire pour la file d'attente modèle en plus de tout autre droit d'accès (aussi applicable à la file d'attente modèle) requis pour les options d'ouvertures indiquées.
 - Le droit MQZAO_CREATE n'est pas nécessaire pour créer la file d'attente dynamique.
 - L'ID utilisateur utilisé pour ouvrir la file d'attente modèle se voit automatiquement accorder tous les droits (équivalents à MQZAO_ALL) sur la file d'attente dynamique créée.
- Selon le type d'objet en cours d'ouverture, l'objet file d'attente, processus, liste de noms ou gestionnaire de files d'attente est vérifié.
- MQOO_INPUT_* doit également être indiqué. Ceci est valable pour une file d'attente locale, modèle, ou alias.
- Cette vérification est effectuée pour tous les cas de sortie, sauf celui indiqué à la remarque 9.
- MQOO_OUTPUT doit également être indiqué.
- MQOO_PASS_IDENTITY_CONTEXT est également concerné par cette option.
- Ce droit est nécessaire à la fois pour l'objet gestionnaire de files d'attente et la file d'attente particulière.
- MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT et MQOO_SET_IDENTITY_CONTEXT sont également concernés par cette option.
- Cette vérification est exécutée pour une file d'attente modèle ou locale qui dispose d'un attribut de file d'attente *Usage* MQUS_TRANSMISSION et est

Tables de spécification des droits d'accès

directement ouverte pour produire les données de sortie. Elle n'est pas appliqué si une file d'attente éloignée est ouverte (soit en spécifiant les noms du gestionnaire de files d'attente éloigné et la file éloignée, soit en spécifiant le nom d'une définition locale de la file d'attente éloignée).

10. Vous devez également indiquer au moins MQOO_INQUIRE (pour tout type d'objet), ou (pour les files d'attente) MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT, ou MQOO_SET. La vérification effectuée est la même que pour les autres options indiquées, au moyen de l'ID utilisateur secondaire fourni pour le droit sur l'objet désigné et le droit d'application en cours pour la vérification MQZAO_ALTERNATE_USER_IDENTIFIER.
11. Ce droit permet d'indiquer tout *AlternateUserId*.
12. Une vérification est également effectuée, si la file d'attente ne comporte pas d'attribut de file d'attente *Usage* MQUS_TRANSMISSION.
13. La vérification effectuée est la même que pour les autres options indiquées, au moyen de l'ID utilisateur secondaire fourni pour le droit sur la file d'attente désignée et le droit d'application en cours pour la vérification MQZAO_ALTERNATE_USER_IDENTIFIER.
14. La vérification n'est effectuée que si ces deux éléments sont vrais :
 - Une file d'attente dynamique persistante est en cours de fermeture et de suppression.
 - Cette file n'a pas été créée par l'appel MQOPEN qui a renvoyé l'identificateur d'objet utilisé.

Dans le cas contraire, le contrôle n'est pas effectué.

Remarques générales :

1. Le droit spécial MQZAO_ALL_MQI comporte les éléments suivants s'ils correspondent au type d'objet concerné :
 - 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
2. MQZAO_DELETE (voir la remarque 14 à la page 154) et MQZAO_DISPLAY sont classés en tant que droits d'administration. Ils ne figurent donc pas dans MQZAO_ALL_MQI.
3. 'Pas de contrôle' signifie qu'aucun contrôle d'autorisation n'est effectué.
4. 'Non applicable' signifie que le contrôle d'autorisation ne concerne pas cette opération. Par exemple, vous ne pouvez pas émettre d'appel MQPUT sur un objet processus.

Droits d'administration

Ces droits permettent à un utilisateur d'émettre des commandes d'administration. Il peut s'agir d'une commande MQSC sous forme de message PCF d'échappement ou de commande PCF. Cette méthode permet à un programme d'envoyer une commande d'administration sous forme de message à un gestionnaire de files d'attente afin d'obtenir une exécution pour le compte de cet utilisateur.

Droits des commandes MQSC dans les PCF d'échappement

Le tableau 4 résume les droits nécessaires à chaque commande MQSC contenue dans la fonction Escape PCF.

Tableau 4. Commandes MQSC et droits d'accès nécessaires

(2)Droits requis pour :	Objet de file d'attente	Objet processus	Objet gestionnaire de files d'attente	Listes de noms
ALTER object	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
CLEAR QLOCAL	MQZAO_CLEAR	Non applicable	Non applicable	Non applicable
DEFINE object NOREPLACE (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Non applicable	MQZAO_CREATE (4)
DEFINE object REPLACE (3, 5)	MQZAO_CHANGE	MQZAO_CHANGE	Non applicable	MQZAO_CHANGE
DELETE object	MQZAO_DELETE	MQZAO_DELETE	Non applicable	MQZAO_DELETE
DISPLAY object	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY

Remarques spécifiques :

1. L'ID utilisateur, sous lequel le programme (par exemple, **runmqsc**) qui exécute la commande est en cours d'exécution, doit également disposer du droit MQZAO_CONNECT sur le gestionnaire de files d'attente.
2. Selon le type d'objet en cours d'ouverture, l'objet file d'attente, processus, liste de noms ou gestionnaire de files d'attente est vérifié.
3. Pour les commandes DEFINE, le droit MQZAO_DISPLAY est également nécessaire à l'objet LIKE s'il en est précisé un, ou à l'objet SYSTEM.DEFAULT.xxx approprié si LIKE est omis.
4. Le droit MQZAO_CREATE n'est pas propre à un objet ou un type d'objet particulier. Le droit de création est accordé pour tous les objets, pour un gestionnaire de files d'attente donné, en spécifiant le type d'objet QMGR dans la commande SETMQAUT.
5. Ceci s'applique si l'objet à remplacer est déjà existant. S'il n'existe pas, le contrôle est de type DEFINE objet NOREPLACE.

Remarques générales :

1. Pour exécuter une commande PCF, vous devez posséder le droit DISPLAY sur le gestionnaire de files d'attente.
2. Le droit d'exécuter un PCF d'échappement dépend de la commande MQSC figurant dans le texte du message PCF d'échappement.
3. 'Non applicable' signifie que le contrôle d'autorisation ne concerne pas cette opération. Par exemple, vous ne pouvez pas émettre une commande CLEAR QLOCAL sur un gestionnaire de files d'attente.

Droits pour les commandes PCF

Le tableau 5 résume les droits nécessaires pour chaque commande PCF.

Tableau 5. Commandes PCF et droits d'accès nécessaires

(2)Droits requis pour :	Objet de file d'attente	Objet processus	Objet gestionnaire de files d'attente	Listes de noms
Modification d'un objet (Change Object)	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
Effacement d'une file d'attente (Clear Queue)	MQZAO_CLEAR	Non applicable	Non applicable	Non applicable

Tables de spécification des droits d'accès

Tableau 5. Commandes PCF et droits d'accès nécessaires (suite)

(2)Droits requis pour :	Objet de file d'attente	Objet processus	Objet gestionnaire de files d'attente	Listes de noms
Copie d'un objet (Copy object without replace) (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Non applicable	MQZAO_CREATE (4)
Copie d'un objet (Copy object with replace) (3, 6)	MQZAO_CHANGE	MQZAO_CHANGE	Non applicable	MQZAO_CHANGE
Création d'un objet (Create object without replace) (5)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	Non applicable	MQZAO_CREATE (4)
Création d'un objet (Create object with replace) (5, 6)	MQZAO_CHANGE	MQZAO_CHANGE	Non applicable	MQZAO_CHANGE
Suppression d'un objet (Delete object)	MQZAO_DELETE	MQZAO_DELETE	Non applicable	MQZAO_DELETE
Demande sur un objet (Inquire Object)	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY
Demande sur des noms d'objets (Inquire Object Names)	Pas de contrôle	Pas de contrôle	Pas de contrôle	Pas de contrôle
Réinitialisation des statistiques de files d'attente (Reset Queue Statistics)	MQZAO_DISPLAY et MQZAO_CHANGE	Non applicable	Non applicable	Non applicable

Remarques spécifiques :

1. L'ID utilisateur sous lequel le programme émettant la commande s'exécute doit également disposer du droit de se connecter à son gestionnaire de files d'attente local, et d'ouvrir la file d'attente d'administration de la commande pour prendre connaissance des données de sortie.
2. Selon le type d'objet en cours d'ouverture, l'objet file d'attente, processus ou gestionnaire de files d'attente est vérifié.
3. Pour les commandes copy, le droit MQZAO_DISPLAY est également nécessaire pour l'objet From.
4. Le droit MQZAO_CREATE n'est pas propre à un objet ou un type d'objet particulier. Le droit de création est accordé pour tous les objets, pour un gestionnaire de files d'attente donné, en spécifiant le type d'objet QMGR dans la commande SETMQAUT.
5. Pour les commandes Create, le droit MQZAO_DISPLAY est également nécessaire pour l'objet SYSTEM.DEFAULT.* approprié.
6. Cela s'applique si l'objet à remplacer existe déjà. S'il n'existe pas, le contrôle effectué est de type Copie ou Création sans remplacement.

Remarques générales :

1. Pour exécuter une commande PCF, vous devez posséder le droit DISPLAY sur le gestionnaire de files d'attente.
2. L'autorisation spéciale MQZAO_ALL_ADMIN comprend les éléments suivants correspondant au type d'objet concerné :
 - MQZAO_CHANGE

Tables de spécification des droits d'accès

- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY

MQZAO_CREATE n'est pas compris, car il n'est pas spécifique d'un objet ou type d'objet particulier.

3. 'Pas de contrôle' signifie qu'aucun contrôle d'autorisation n'est effectué.
4. 'Non applicable' signifie que le contrôle d'autorisation ne concerne pas cette opération. Par exemple, vous ne pouvez pas utiliser une commande **Clear Queue** sur un objet processus.

Fichiers de droits d'accès

Sous MQSeries pour Compaq NSK, V5.1, toutes les informations relatives aux droits d'accès sont stockées dans les fichiers d'audit TM/MP suivants à l'emplacement \$VOL.<QMgrSubVol>D:

OAMDB Base de données OAM
PRIDB Base de données de principaux
PRIDBA Fichier alternatif de la base de données de principaux

Base de données de principaux

Chaque entrée de la base de données de principaux fait correspondre un ID utilisateur Compaq NSK à un nom principal. La base de données de principaux est un fichier en séquence par clé ENSCRIBE qui fait correspondre les principaux OAM et les ID d'ouverture de session Compaq NSK.

Principal OAM MQPRINCIPAL
ID d'ouverture de session Compaq
0x2CFF

La clé principale est le principal OAM (12 caractères). La clé alternative est l'ID d'ouverture de session Compaq (un entier court sur 2 octets). Le principal OAM est toujours sensible à la casse. Les octets du champ de l'ID d'ouverture de session sont <group>.<user> ; l'exemple ci-dessus concerne l'ID Compaq NSK (44,255).

Base de données OAM

Chaque entrée de la base de données des droits d'accès OAM fait référence à un objet gestionnaire de files d'attente ou à une classe d'objets particuliers. La clé principale correspond au nom d'objet et au type d'objet. Les enregistrements sont de taille variable et la présentation est la suivante :

Nom d'objet	Type	#Entrées CodeAccès	Entrées CodeAccès
QUEUE.AUTH	1	2	PAYROLL 0x00000004, ADMIN 0xFFFFFFFF

Le champ Nom d'Objet est le nom d'objet de 48 caractères complété en blanc. Le champ Type (4 octets) différencie les types d'objets MQSeries des classes d'objets requises par l'OAM.

Le champ Type peut prendre les valeurs suivantes :

- 1 Nom de file d'attente
- 2 Nom de processus
- 4 Nom du GFA
- 128 Classe

Fichiers de droits d'accès

Le champ #CodeAccès (4 octets) définit le nombre de code d'accès individuels de cet enregistrement. Chaque entrée CodeAccès définit un nom de groupe et le code d'accès du groupe pour cet objet :

Nom de groupe

PAYROLL

CodeAccès 0x00000004

Le champ du nom de groupe fait 12 octets de long et contient un nom de groupe de partage de fichiers d'administration ou SAFEGUARD Compaq NSK complété en blanc (12 premiers caractères seulement). Le champ CodeAccès est un masque de contrôle des données de 4 octets (ULONG) contenant les codes d'accès du groupe. Chaque enregistrement peut contenir jusqu'à 250 paires Groupe/CodeAccès.

Plusieurs enregistrements sont utilisés pour le même objet pour contenir les informations concernant les codes d'accès pour plus de 250 groupes le cas échéant.

Les droits d'accès obtenus correspondent à la réunion des schémas de bits individuels basés sur les affectations suivantes :

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

Ces définitions se trouvent dans le fichier d'en-tête cmqzc.h. Dans l'exemple suivant, les droits d'accès accordés à groupB sont basés sur le nombre hexadécimal 0x40007. A savoir :

MQZAO_CONNECT	0x00000001
MQZAO_BROWSE	0x00000002
MQZAO_INPUT	0x00000004
MQZAO_DISPLAY	0x00040000
-----	-----
Authority is:	0x00040007

Ces droits d'accès signifient que toute personne appartenant au groupe groupB peut émettre les appels MQI suivants :

- MQCONN
- MQGET (avec consultation)

Ils disposent également d'un droit DISPLAY sur l'objet associé à ce fichier de droits d'accès.

Enregistrements de droits d'accès aux classes

Les enregistrements de droits d'accès de classe contiennent des droits qui correspondent à la classe entière. Les champs nom et type d'objet correspondent de la manière suivante :

Nom d'objet	Type
@QMGRCLASS	0x80
@PROCESSCLASS	0x80
@QUEUECLASS	0x80

L'entrée MQZAO_CRT dans le champ de code d'accès donne le droit de créer un objet dans cette classe. Il s'agit du seul droit de classe.

Enregistrement de droit d'accès à toutes les classes

L'enregistrement de droits d'accès à toutes les classes contient les droits qui s'appliquent à un gestionnaire de files d'attente entier. Les champs nom et type d'objet correspondent de la manière suivante :

Nom d'objet	Type
@ALLCLASSES	0x80

Les droits suivants s'appliquent au gestionnaire de files d'attente dans son ensemble et sont contenus dans le fichier de droits à toutes les classes.

L'entrée...	autorise à
MQZAO_ALTUSER	Prendre l'identité d'un autre utilisateur en cas d'interaction avec des objets MQSeries.
MQZAO_SET_ALL_CONTEXT	Définir le contexte d'un message lors de l'émission de MQPUT.
MQZAO_SET_IDENTITY_CONTEXT	Définir le contexte d'un message lors de l'émission de MQPUT.

Fichiers de droits d'accès

Chapitre 9. Gestionnaire de file d'attente de rebut MQSeries

MQSeries pour Compaq NSK fournit une file d'attente de rebut (DLQ), (ou file d'attente des messages non livrés), qui contient les messages qui n'ont pas pu être livrés aux files d'attente destinataires. Chaque gestionnaire de files d'attente d'un réseau doit posséder une DLQ associée.

Les messages sont insérés en DLQ par les gestionnaires de files d'attente, par les agents MCA et par les applications. Tous les messages contenus dans la DLQ doivent comporter en préfixe la structure MQDLH, d'*en-tête de rebut*. Les messages mis en DLQ par un gestionnaire de files d'attente ou par un agent MCA ont toujours cet en-tête. Les applications qui placent elles-mêmes des messages dans la DLQ doivent également fournir l'en-tête MQDLH. La zone *Reason* de la structure MQDLH contient un code anomalie qui indique la raison pour laquelle le message a été inséré en DLQ.

Vous devez disposer d'une routine qui s'occupe régulièrement de traiter les messages présents dans la DLQ. MQSeries fournit une routine par défaut appelée *gestionnaire de files d'attente de rebut* (gestionnaire DLQ), que vous lancez à l'aide de la commande **runmqdlq**.

Les instructions de traitement des messages en DLQ sont fournies au gestionnaire par une *table des règles* écrite par l'utilisateur. Le gestionnaire DLQ compare les messages en DLQ avec les entrées de la table des règles lorsqu'un message DLQ correspond à une entrée de la table de règles, le gestionnaire DLQ exécute l'action associée à cette entrée.

Ce chapitre contient les sections suivantes :

- «Lancement du gestionnaire DLQ»
- «Table de règles du gestionnaire DLQ» à la page 162
- «Traitement de la table de règles» à la page 168
- «Exemple de table de règles pour un gestionnaire DLQ» à la page 170

Lancement du gestionnaire DLQ

Pour lancer le gestionnaire DLQ, vous devez utiliser la commande **runmqdlq**. Vous pouvez indiquer la file DLQ à traiter et le gestionnaire de files d'attente à utiliser de deux manières :

- A partir de la ligne de commande, en utilisant des paramètres. Par exemple :

```
runmqdlq /IN qrule/ ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER
```

- Dans la table de règles. Par exemple :

```
INPUTQ(FILE.ATT.REBUT.ABC1) INPUTQM(ABC1.GEST.FILES)
```

Les exemples ci-dessus s'appliquent à la DLQ appelée FILE.ATT.REBUT.ABC1, dont le gestionnaire est ABC1.GEST.FILES.

Lancement du gestionnaire DLQ

Si vous n'indiquez pas la DLQ ou le gestionnaire de files d'attente comme il est indiqué ci-dessus, le gestionnaire de files d'attente par défaut est utilisé, ainsi que la DLQ qui appartient à ce gestionnaire.

La commande **runmqdlq** lit les entrées depuis la table de règles, envoyée au fichier IN standard. Vous associez la table de règles à la commande **runmqdlq** en redirigeant le fichier IN vers le fichier de règles.

Pour lancer le gestionnaire DLQ, vous devez être autorisé à accéder à la DLQ et à toutes les files d'attente de messages auxquelles les messages en DLQ sont transmis. En outre, si le gestionnaire DLQ peut insérer les messages dans des files d'attente avec les droits associés à l'ID utilisateur indiqué dans le contexte de message, vous devez être autorisé à utiliser l'identité des autres utilisateurs.

Vous trouverez davantage d'informations sur la commande **runmqdlq** à la section «runmqdlq (lancement du gestionnaire de files d'attente de rebut)» à la page 292.

Table de règles du gestionnaire DLQ

La table de règles du gestionnaire DLQ définit la façon dont le gestionnaire DLQ doit traiter les messages arrivant en DLQ. Elle contient deux types d'entrées :

- La première entrée de la table, facultative, contient les *données de contrôle*.
- Toutes les autres entrées de la table sont des *règles* que le gestionnaire DLQ doit respecter. Chaque règle se compose d'un *modèle* (série de caractéristiques de message) sur lequel est construit le message, et d'une *action* à exécuter lorsqu'un message en file DLQ correspond au modèle spécifié. Une table de règles doit contenir au moins une règle.

Chaque entrée de la table de règles comporte un ou plusieurs mots clés.

Vos trouverez une description de la syntaxe concernant la table de règles à la section «Conventions de la table de règles» à la page 167.

Données de contrôle

Cette section décrit les mots clés que vous pouvez inclure dans une entrée de données de contrôle d'une table de règles de gestionnaire DLQ. Tenez compte des points suivants :

- La valeur par défaut des mots clés (le cas échéant) est soulignée.
- Une ligne verticale (|) sépare les alternatives, une seule des deux pouvant être spécifiée.
- Tous les mots clés sont facultatifs.

INPUTQ (*NomFile*|' ')

Ce mot clé est le nom de la DLQ à qui la table de règles s'applique. Il permet d'indiquer la file DLQ à traiter :

1. Si vous indiquez un paramètre *NomQ* avec la commande **runmqdlq**, il remplace toute valeur INPUTQ contenue dans la table de règles.
2. Si vous ne précisez aucune valeur *NomQ* comme paramètre de la commande **runmqdlq**, mais que vous précisez une valeur dans la table de règles, c'est cette valeur qui est utilisée comme INPUTQ.
3. Si aucune file DLQ n'est indiquée ou si vous spécifiez INPUTQ(' ') dans la table de règles, c'est le nom de la file DLQ appartenant au gestionnaire de files d'attente dont le nom est indiqué comme paramètre de la commande **runmqdlq**, ou celui qui est associé au mot clé INPUTQM de la table de règles, qui est utilisé.

INPUTQM (*NomGestFiles* | ' _')

Ce mot clé indique le nom du gestionnaire de files d'attente propriétaire de la file d'attente. Ce mot clé indique le nom du gestionnaire de files d'attente propriétaire de la file DLQ indiquée dans le mot clé INPUTQ :

1. Si vous indiquez un paramètre *NomGestFile* avec la commande **runmqdlq**, il remplace toute valeur INPUTQM contenue dans la table de règles.
2. Si vous ne précisez aucune valeur *NomGestFile* comme paramètre de la commande **runmqdlq**, mais que vous précisez une valeur dans la table de règles, c'est cette valeur qui est utilisée comme INPUTQ.
3. Si aucun gestionnaire de files d'attente n'est spécifié ou si vous indiquez INPUTQM(' ') dans la table de règles, c'est le gestionnaire de files d'attente par défaut de l'installation qui est utilisé.

RETRYINT (*Intervalle* | 60)

Ce mot clé indique l'intervalle (en secondes) au bout duquel le gestionnaire DLQ doit tenter à nouveau de traiter les messages en file DLQ qui n'ont pas pu être traités lors d'une première tentative, et pour lesquels plusieurs tentatives sont demandées. L'intervalle par défaut est de 60 secondes.

WAIT (**YES** | **NO** | *nnn*)

Ce mot clé indique si le gestionnaire DLQ doit attendre l'arrivée d'autres messages en DLQ lorsqu'il détecte que la file ne contient plus aucun message à traiter.

YES Le gestionnaire DLQ attend indéfiniment.

NO Le gestionnaire DLQ s'arrête lorsqu'il détecte que la DLQ est vide ou ne contient pas de messages qu'il puisse traiter.

nnn Le gestionnaire DLQ attend pendant *nnn* secondes l'arrivée d'un autre travail avant de s'arrêter, après avoir détecté que la file d'attente est vide ou ne contient plus de messages qu'il puisse traiter.

Il est recommandé de spécifier WAIT (YES) pour les DLQ très actives et WAIT (NO) ou WAIT (*nnn*) pour les DLQ peu actives. Si le gestionnaire DLQ est amené à s'arrêter, il est recommandé de le relancer par déclenchement.

Les données de contrôle présentées à la figure 40 montrent que la table de règles s'applique à la DLQ appartenant au gestionnaire de files d'attente QM1. Le signe plus (+) à la fin de la ligne 1 indique que les données de contrôle continuent à partir du premier caractère non blanc de la ligne 2.

```
INPUTQ' ' +
INPUTQM'QM1'
```

Figure 40. Exemple de données de contrôle

Si vous n'incluez pas les données de contrôle dans la table de règles, vous pouvez fournir les noms de la file DLQ et du gestionnaire de files d'attente correspondant comme paramètres d'entrée de la commande **runmqdlq**. Lorsqu'une valeur est spécifiée à la fois dans la table de règles et en entrée de la commande **runmqdlq** c'est cette dernière qui est prioritaire.

Remarque : Lorsqu'une entrée de données de contrôle figure dans la table des règles, elle doit être la première entrée de la table.

Règles (modèles et actions)

La figure 41 présente un exemple de règle de la table de règles d'un gestionnaire DLQ. Cette règle demande au gestionnaire DLQ d'effectuer trois tentatives d'insertion dans la file d'attente cible, pour les messages persistants insérés en DLQ suite à une désactivation de MQPUT et MQPUT1.

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Figure 41. Exemple de règle. Le signe plus (+) à la fin de la ligne 1 indique que les données de contrôle continuent à partir du premier caractère non blanc de la ligne 2.

La suite de cette section décrit tous les mots clés que vous pouvez utiliser dans une règle. Tenez compte des points suivants :

- La valeur par défaut des mots clés (le cas échéant) est soulignée. Pour la plupart des mots clés, la valeur par défaut est * (astérisque). Ce signe symbolise une valeur quelconque.
- Une ligne verticale (|) sépare les alternatives, une seule des deux pouvant être spécifiée.
- Tous les mots clés sont facultatifs (sauf ACTION).

Cette section débute par une description des mots clés de modèle (ceux auxquels les messages en DLQ sont comparés), et se poursuit avec une description des mots clés d'action (ceux qui déterminent la façon dont le gestionnaire DLQ doit traiter un message conforme au modèle).

Mots clés de modèle

Les mots clés de modèle utilisés pour indiquer les valeurs auxquelles les messages en DLQ doivent être comparés sont décrits ci-dessous. Tous les mots clés de modèle sont facultatifs.

APPLIDAT (*DonnéesIdentitéAppl* | *)

Ce mot clé est la valeur du paramètre *DonnéesIdentitéAppl* précisée dans le descripteur (MQMD) du message en DLQ.

APPLNAME (*NomApplPut* | *)

Nom de l'application qui a émis l'appel MQPUT ou MQPUT1 tel qu'il est indiqué dans la zone *NomApplPut* du descripteur (MQMD) du message en DLQ.

APPLTYPE (*TypeApplPut* | *)

Ce mot clé est la valeur du paramètre *TypeAppPut* précisée dans le descripteur (MQMD) du message en DLQ.

DESTQ (*NomFileAtt* | *)

Nom de la file d'attente de messages à laquelle doit être transmis le message.

DESTQM (*NomGestFiles* | *)

Nom du gestionnaire de la file d'attente de messages à laquelle doit être transmis le message.

FEEDBACK (*Retour* | *)

Lorsque la valeur du paramètre *MsgType* est MQFB_REPORT, le mot clé *Feedback* décrit la nature du rapport.

Des noms symboliques peuvent être utilisés. Par exemple, vous pouvez utiliser le nom symbolique MQFB_COA pour identifier les messages en DLQ qui nécessitent une confirmation de leur arrivée en file d'attente cible.

FORMAT (*Format* | *)

Nom que l'expéditeur du message utilise pour décrire le format des données du message.

MSGTYPE (*TypeMsg* | *)

Type du message en DLQ.

Des noms symboliques peuvent être utilisés. Par exemple, vous pouvez utiliser le nom symbolique `_REQUEST` pour identifier les messages en DLQ qui exigent une réponse.

PERSIST (*Persistence* | *)

Valeur de persistance du message. La persistance d'un message détermine s'il est conservé après une réinitialisation du gestionnaire de files d'attente.

Des noms symboliques peuvent être utilisés. Par exemple, vous pouvez utiliser le nom symbolique `MQPER_PERSISTENT` pour identifier les messages en DLQ qui sont persistants.

REASON (*CodeAnomalie* | *)

Code anomalie qui décrit la raison pour laquelle le message a été inséré en DLQ.

Des noms symboliques peuvent être utilisés. Par exemple, vous pouvez utiliser le nom symbolique `MQRC_Q_FULL` pour identifier les messages insérés en DLQ parce que leur file d'attente cible était pleine.

REPLYQ (*NomFileAtt* | *)

Nom de la file d'attente des messages de réponse indiquée dans le descripteur (MQMD) du message en DLQ.

REPLYQM (*NomGestFiles* | *)

Nom du gestionnaire de la file d'attente de messages de réponse, indiquée dans le descripteur (MQMD) du message en DLQ.

USERID (*UdUtil* | *)

ID de l'utilisateur qui a envoyé le message inséré en DLQ, tel qu'il est indiqué dans le descripteur de message (MQMD).

Mots clés d'action

Les mots clés d'action décrits ci-dessous servent à décrire le traitement à appliquer à tout message en DLQ conforme au modèle.

ACTION (DISCARD | IGNORE | RETRY | FWD)

Action à exécuter pour tout message en DLQ conforme au modèle défini dans cette règle.

DISCARD Le message est supprimé de la DLQ.

IGNORE Le message est laissé en DLQ.

RETRY Le gestionnaire DLQ tente à nouveau d'insérer le message dans sa file d'attente cible.

FWD Le gestionnaire DLQ transmet le message à la file d'attente indiquée dans le mot clé FWDQ.

Le mot clé ACTION doit être spécifié. Le nombre de tentatives d'exécution d'une action est défini par le mot clé RETRY. L'intervalle entre les tentatives est géré par le mot clé RETRYINT des données de contrôle.

table de règles

FWDQ (*NomFileAtt* | **&DESTQ** | **&REPLYQ**)

Nom de la file d'attente de messages à laquelle le message doit être transmis lorsque ACTION (FWD) est demandé.

NomFileAtt

Nom de la file d'attente de messages. FWDQ(' ') n'est pas un nom valide.

&DESTQ

Le nom de la file d'attente est récupéré dans la zone *FileAttCible* de la structure MQDLH.

&REPLYQ

Le nom est récupéré dans la zone *FileAttRép* du descripteur de message, MQMD.

Pour éviter les messages d'erreur lorsqu'une règle spécifie que FWDQ (&REPLYQ) correspond à un message avec une zone *FileAttRép* vide, spécifiez REPLYQ (?*) dans le modèle de message.

FWDQM (*NomGestFiles* | **&DESTQM** | **&REPLYQM** | ' _')

Identifie le gestionnaire de la file d'attente à laquelle un message doit être transmis.

NomGestFiles

Nom du gestionnaire de la file d'attente à laquelle un message doit être transmis lorsque ACTION (FWD) est demandé.

&DESTQM

Le nom du gestionnaire de files d'attente est récupéré depuis la zone *NomGestFilesCible* de la structure MQDLH.

&REPLYQM

Le nom est récupéré dans la zone *FileAttRép* du descripteur de message (MQMD).

' ' FWDQM(' ') est la valeur par défaut, elle identifie le gestionnaire de files d'attente local.

HEADER (**YES** | **NO**)

Indique si le MQDLH doit rester dans un message pour lequel ACTION (FWD) est demandé. Par défaut, le MQDLH reste dans le message. Le mot clé HEADER n'est pas valide pour les actions autres que FWD.

PUTAUT (**DEF** | **CTX**)

Définit les droits avec lesquels les messages doivent être insérés par le gestionnaire DLQ :

DEF Les messages sont insérés en file d'attente avec les droits du gestionnaire DLQ lui-même.

CTX Les messages sont insérés en file d'attente avec les droits de l'ID utilisateur indiqué dans le contexte du message. Si vous spécifiez PUTAUT (CTX), vous devez être autorisé à utiliser l'identité des autres utilisateurs.

RETRY (*NbEssais* | **1**)

RETRY est le nombre de fois qu'une action doit être tentée, de 1 à 999, (avec l'intervalle indiqué par le mot clé RETRYINT des données de contrôle).

Remarque : Le nombre de tentatives faites par le gestionnaire DLQ pour appliquer une règle particulière est spécifique à l'instance en cours du gestionnaire DLQ : ce nombre est supprimé après un

redémarrage. Si le gestionnaire DLQ est redémarré, le nombre de tentatives effectuées pour appliquer une règle est remis à zéro.

Conventions de la table de règles

La syntaxe, la structure et le contenu de la table de règles doivent être conformes aux règles suivantes :

- Une table de règles doit contenir au moins une règle.
- Les mots clés peuvent se présenter dans n'importe quel ordre.
- Un mot clé ne peut figurer qu'une seule fois par règle.
- Les mots clés peuvent être en majuscules ou en minuscules.
- Le mot clé et ses paramètres doivent être séparés des autres mots clés par au moins un blanc ou une virgule.
- Un ou plusieurs blancs peuvent figurer au début ou à la fin d'une règle et entre les mots clés, les ponctuations et les valeurs.
- Chaque règle doit commencer sur une nouvelle ligne.
- Pour des raisons de portabilité, une ligne ne doit pas dépasser 72 caractères.
- Utilisez le signe plus (+) comme dernier caractère non blanc sur une ligne pour indiquer que la règle se poursuit sur la ligne suivante à partir du premier caractère non blanc. Utilisez le signe moins (-) comme dernier caractère non blanc sur une ligne pour indiquer que la règle se poursuit sur la ligne suivante à partir de son début. Les caractères de continuation peuvent figurer dans les mots clés et les paramètres.
- Les lignes de commentaires, qui commencent par un astérisque (*), peuvent figurer à n'importe quel endroit de la table de règles.
- Les lignes vides ne sont pas prises en compte.
- Chaque entrée de la table de règles du gestionnaire DLQ comprend un ou plusieurs mots clés et les paramètres associés. Les paramètres doivent être conformes aux règles de syntaxe suivantes :
 - Chaque valeur de paramètre doit inclure au moins un caractère significatif. L'apostrophe de délimitation n'est pas considérée comme significative. Par exemple, les paramètres ci-après sont corrects :

FORMAT('ABC')	3 caractères significatifs
FORMAT(ABC)	3 caractères significatifs
FORMAT('A')	1 caractère significatif
FORMAT(A)	1 caractère significatif
FORMAT(' ')	1 caractère significatif
 - Les paramètres ci-après sont incorrects car ils ne contiennent aucun caractère significatif :

FORMAT('')
FORMAT()
FORMAT()
FORMAT
 - Les caractères 'joker' sont possibles : vous pouvez utiliser le point d'interrogation (?) à la place d'un caractère quelconque, sauf pour désigner un blanc de fin ; l'astérisque (*) peut remplacer zéro ou plusieurs caractères adjacents. L'astérisque (*) et le point d'interrogation (?) sont *toujours* interprétés comme des caractères 'joker' (ou génériques) dans les valeurs de paramètres.
 - Ils ne peuvent cependant pas figurer dans les paramètres des mots clés suivants : ACTION, HEADER, RETRY, FWDQ, FWDQM et PUTAUT.

Conventions de la table de règles

- Les blancs à la fin des valeurs de paramètres et des zones correspondantes du message en DLQ ne sont pas significatifs lors des recherches par caractères génériques. Cependant, les blancs d'en-tête ou insérés dans des chaînes entre des points d'interrogation sont pris en compte pour les recherches par caractères génériques.
- Les paramètres numériques ne peuvent pas contenir de point d'interrogation (?). L'astérisque (*) peut être utilisé à la place d'un paramètre numérique, mais ne peut pas représenter une partie de ce dernier. Par exemple, les paramètres numériques suivants sont valides :
 - MSGTYPE(2)** Seuls les messages de réponse sont valables.
 - MSGTYPE(*)** Tous les types de messages sont valables.
 - MSGTYPE('*')** Tous les types de messages sont valables.

Cependant, **MSGTYPE('2*')** n'est pas correct car il comprend un astérisque (*) en remplacement d'une partie de ce paramètre numérique.

- Les paramètres numériques doivent être compris entre 0 et 999. Si la valeur du paramètre se situe dans cette plage, elle est acceptée, même si elle n'est pas valide dans la zone à laquelle le mot clé fait référence. Les noms symboliques peuvent être utilisés comme paramètres numériques.
- Si une chaîne est plus courte que la zone du MQDLH ou du MQMD à laquelle le mot clé fait référence, la valeur est complétée par des blancs. Si la chaîne (exclusion faite des astérisques) est plus longue que la zone, une erreur est diagnostiquée. Par exemple, les valeurs de chaîne correcte pour une zone de 8 caractères sont les suivantes :
 - 'ABCDEFGH'** 8 caractères
 - 'A*C*E*G*I'** 5 caractères, exclusion faite des astérisques
 - '*A*C*E*G*I*K*M*O*'** 8 caractères, exclusion faite des astérisques
- Les chaînes qui contiennent des blancs, des caractères en minuscules ou des caractères spéciaux autres que les points (.), les barres obliques (/), les traits de soulignement (_) et les pourcentages (%) doivent être entre apostrophes. Les caractères en minuscules qui ne sont pas entre apostrophes sont mis en majuscules. Si la chaîne comprend une apostrophe, deux apostrophes consécutives doivent être utilisées pour indiquer le début et la fin de la chaîne. Lors du calcul de la longueur de la chaîne, chaque occurrence de deux apostrophes consécutives est comptée comme un caractère simple.

Traitement de la table de règles

Le gestionnaire DLQ recherche dans la table une règle dont le format correspond à un message en DLQ. La recherche commence par la première règle de la table et se poursuit de façon séquentielle. Lorsqu'une correspondance est trouvée, l'action correspondant à la règle est exécutée. Chaque fois qu'il tente d'appliquer une règle, le gestionnaire DLQ augmente d'une unité le compteur d'essais pour cette règle. En cas d'échec de la première tentative, cette tentative est renouvelée autant de fois que le permet le nombre indiqué par le paramètre **RETRY**. En cas d'échec de toutes les tentatives, le gestionnaire DLQ recherche la prochaine correspondance dans la table.

Cette procédure se répète pour les règles suivantes jusqu'à ce qu'une tentative réussisse. Lorsque chaque règle ayant une correspondance a fait l'objet du nombre de tentatives indiquées par le paramètre **RETRY** et que toutes les tentatives ont échoué, **ACTION (IGNORE)** est pris par défaut. **ACTION (IGNORE)** est également pris par défaut si aucune règle correspondante n'est trouvée.

Remarques :

1. Seuls les messages en DLQ qui commencent par un MQDLH sont pris en compte pour les recherches de correspondance de format de règle. Les messages qui ne commencent pas par un MQDLH sont identifiés comme étant erronés et restent définitivement en DLQ.
2. Tous les mots clés de format peuvent être pris à leur valeur par défaut, de telle façon qu'une règle ne se compose plus que d'une action. Veuillez cependant noter que ce type de règle s'applique à tous les messages de la file qui comportent un MQDLH et qui n'ont pas encore été traités conformément aux autres règles de la table.
3. La table des règles est validée lors du démarrage du gestionnaire DLQ et les erreurs sont signalées à ce moment là. (Les messages d'erreur générés par le gestionnaire DLQ sont décrits dans le manuel *MQSeries - Guide des messages*). Vous pouvez modifier la table de règles à tout moment, mais ces modifications ne sont prises en compte que lors du redémarrage du gestionnaire DLQ.
4. Le gestionnaire DLQ ne modifie pas le contenu des messages, du MQDLH ou du descripteur de message. Il insère toujours les messages dans d'autres files d'attente avec l'option MQPMO_PASS_ALL_CONTEXT.
5. Le gestionnaire DLQ ouvre la DLQ avec l'option MQOO_INPUT_AS_Q_DEF.
6. Plusieurs instances du gestionnaire DLQ peuvent s'exécuter en même temps pour la même file d'attente et en utilisant la même table de règles. Il est cependant plus courant de mettre en relation un seul gestionnaire DLQ avec une seule DLQ.

Assurance du traitement de tous les messages DLQ

Le gestionnaire DLQ conserve un enregistrement de tous les messages en DLQ qui ont été consultés, mais pas retirés. Si vous utilisez le gestionnaire DLQ comme filtre pour extraire un petit sous-ensemble de messages de la DLQ, le gestionnaire DLQ doit garder trace des messages en DLQ qu'il n'a pas traités. En outre, le gestionnaire DLQ ne peut pas garantir que les nouveaux messages qui arrivent en DLQ seront détectés, même si la DLQ est définie en mode FIFO (premier entré premier sorti). Ainsi, si la file d'attente n'est pas vide, une consultation périodique de la DLQ est effectuée pour vérifier tous les messages. C'est pourquoi vous devez vous assurer que la DLQ contienne aussi peu de messages que possible. Si les messages qui ne peuvent pas être supprimés ou transmis à d'autres files (pour une raison quelconque) s'accumulent dans la file, la charge de travail du gestionnaire DLQ augmente et la DLQ risque d'être saturée.

Il existe des mesures permettant au gestionnaire DLQ de vider la DLQ. Par exemple, n'utilisez pas ACTION (IGNORE) qui laisse les messages en DLQ. ACTION (IGNORE) est l'action par défaut pour les messages qui ne font pas explicitement l'objet d'autres règles de la table. Pour ces messages (qui autrement ne seraient pas pris en compte), effectuez une opération de transfert dans une autre file d'attente. Par exemple :

```
ACTION (FWD) FWDQ (FILE.ATTENTE.IGNOREE) HEADER (YES)
```

Traitement de la table de règles

De même, la dernière règle de la table doit se comporter comme une règle par défaut de façon à traiter les messages qui n'ont été traités par aucune autre règle de la table. Par exemple, la dernière règle de la table peut se présenter de la façon suivante :

```
ACTION (FWD) FWDQ (FILE.ATTENTE.REELLE) HEADER (YES)
```

Les messages qui arrivent à la dernière règle de la table sont alors transmis à la file d'attente FILE.ATTENTE.REELLE, dans laquelle ils peuvent être traités manuellement. En l'absence d'un tel type de règle, les messages risquent de rester indéfiniment en DLQ.

Exemple de table de règles pour un gestionnaire DLQ

La table de règles présentée en exemple ci-dessous contient une seule entrée de données de contrôle et plusieurs règles :

```
*****
*           An example rules table for the runmqdlq command           *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
  ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never do things by half measures. If
* 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)
```

Exemple de table de règles

- * 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)
```

Exemple de table de règles

Chapitre 10. Événements outils et événements EMS

Le présent chapitre :

- Présente rapidement les événements outil de MQSeries, utilisables pour contrôler le fonctionnement des gestionnaires de file d'attente. Consultez la section «Événements outils MQSeries». Pour plus d'informations sur les événements outils, reportez-vous au manuel *MQSeries Programmable System Management*.
- Décrit l'utilisation des événements EMS (Event Management Service) par MQSeries pour Compaq NSK. Consultez la section «Événements EMS (Event Management Service)» à la page 175.

Événements outils MQSeries

Lorsque le gestionnaire de files d'attente détecte une série de conditions prédéfinies, les événements outils provoquent la création de messages spéciaux appelés *messages d'événements*. Par exemple, les conditions suivantes entraînent un événement *Queue Full (file d'attente saturée)* :

- Les événements Queue Full sont activés pour une file d'attente spécifique.
- Une application émet un appel MQPUT afin d'insérer un message dans cette file, mais l'appel échoue car la file est pleine.

D'autres conditions peuvent engendrer des événements outils :

- Le nombre maximal de messages en file d'attente a été atteint.
- Une file n'est pas desservie pendant une période définie.
- Une instance de canal est démarrée ou arrêtée.
- Une application tente d'ouvrir une file d'attente en spécifiant un ID utilisateur non autorisé.

Tous les événements outils doivent être activés avant d'être générés (sauf en ce qui concerne les événements Canal).

Le message d'événement contient des informations concernant la cause de l'événement. Il est placé dans une *file d'attente des événements*. Une application peut le retirer de la file d'attente d'événements pour l'analyser.

Si vous définissez les files d'attente d'événements en tant que files d'attente éloignées, vous pouvez toutes les placer sur un seul et même gestionnaire de files d'attente (pour les noeuds prenant en charge les événements outils). Vous pouvez ensuite utiliser les événements générés pour contrôler un réseau de gestionnaires de files d'attente à partir d'un noeud.

Types d'événements

Il existe quatre types d'événements :

Événements Gestionnaire de files d'attente

Ces événements concernent les définitions de ressources dans les gestionnaires de files d'attente. Par exemple, si une application tente de mettre un message dans une file d'attente qui n'existe pas.

Evénements outils

Evénements Performances

Ces événements indiquent qu'un seuil a été atteint par une ressource. Par exemple, la limite de longueur d'une file d'attente a été atteinte ou, à la suite d'une demande MQGET, une file d'attente n'a pas été desservie durant une période prédéfinie.

Evénement Canal

Ces événements sont rapportés par les canaux à la suite de conditions détectées durant leur fonctionnement (par exemple lorsqu'une instance de canal est arrêtée).

Evénements de déclenchement

Ce type d'événement se produit lorsqu'un gestionnaire de files d'attente détecte que les conditions d'un événement de déclenchement sont remplies. Par exemple, une file d'attente peut être configurée pour générer un événement de déclenchement chaque fois qu'un message arrive (les conditions requises pour les événements outils et les événements de déclenchement sont donc différentes).

Un événement de déclenchement provoque l'insertion d'un message de déclenchement dans une file d'attente d'initialisation et peut, en option, lancer un programme.

Notification d'événements via les files d'attente d'événements

Lorsqu'un événement survient, le gestionnaire de files d'attente insère un message d'événement dans la file d'attente appropriée, s'il en existe une. Ce message contient des informations concernant l'événement, que vous pouvez récupérer en écrivant un programme MQI qui effectue les opérations suivantes :

- Extraction du message depuis la file d'attente.
- Traitement du message afin d'extraire les données d'événement. Pour obtenir une description des formats de messages d'événements, reportez-vous au manuel *MQSeries Programmable System Management*.

Chaque catégorie d'événement possède sa propre file d'attente d'événements. Chaque événement d'une même catégorie provoque l'insertion d'un message d'événement dans cette file d'attente.

Cette file d'attente d'événement...	Contient les messages provenant des...
SYSTEM.ADMIN.QMGR.EVENT	Evénements Gestionnaire de files d'attente
SYSTEM.ADMIN.PERFM.EVENT	Evénements Performances
SYSTEM.ADMIN.CHANNEL.EVENT	Evénements Canal

Utilisation de files d'attente d'événements déclenchées

Vous pouvez définir les files d'attente d'événements avec une fonction de déclenchement de telle manière que, lorsqu'un événement est généré et insère un message, la file d'attente lance une application de gestion (écrite par l'utilisateur). Cette application peut traiter les messages d'événement et prendre les mesures qui s'imposent. Par exemple, certains événements peuvent nécessiter qu'un opérateur soit informé, d'autres peuvent démarrer une application qui exécute automatiquement des tâches d'administration.

Activation des événements outil

La façon d'activer un événement outil dépend de son type :

- Les événements Gestionnaire de files d'attente sont activés en définissant des attributs du gestionnaire de files d'attente.
- Les événements Performances doivent être activés globalement pour le gestionnaire de files d'attente. Vous pouvez également activer un événement de performances donné en paramétrant l'attribut correspondant de la file d'attente, et en identifiant les conditions qui déclencheront cet événement, comme par exemple un seuil de capacité.
- Les événements Canal se produisent automatiquement. Il n'est pas nécessaire de les activer. Si vous ne souhaitez pas les gérer, vous pouvez désactiver les demandes MQPUT (put-inhibit) vers la file d'attente d'événements Canal.

Pour activer ou désactiver la génération des événements outil, vous pouvez utiliser l'une ou l'autre des méthodes suivantes :

- Des commandes MQSC. Pour plus d'informations, reportez-vous au manuel *MQSeries - Guide de référence des commandes MQSC*.
- Des commandes PCF destinées aux gestionnaires de file d'attente. Pour plus d'informations, reportez-vous au manuel *MQSeries Programmable System Management*.
- Des commandes MQAI. Pour plus d'informations, reportez-vous au manuel *MQSeries Administration Interface Programming Guide and Reference*.

Messages d'événements

Les messages d'événements contiennent des informations concernant l'origine d'un événement (type d'événement, nom de l'application ayant provoqué l'événement et brèves données statistiques concernant la file d'attente).

Le format des messages d'événements est similaire à celui des messages de réponse PCF. Les données de ces messages peuvent être récupérées par des programmes d'administration écrits par l'utilisateur, à l'aide des structures de données décrites dans le manuel *MQSeries Programmable System Management*.

Événements EMS (Event Management Service)

MQSeries pour Compaq NSK génère des messages d'événements (EMS), qui correspondent aux événements de gestionnaire de files d'attente de MQSeries, de canal et de performance. Les messages EMS générés peuvent également correspondre aux entrées dans les journaux de MQSeries, et à des FFST. Ces messages d'événement peuvent servir à prévenir les opérateurs système et les administrateurs de conditions qui pourraient avoir un impact négatif sur l'environnement d'exploitation de MQSeries.

Fichier modèle EMS fourni avec MQSeries pour Compaq NSK

Les fichiers suivants se trouvent dans le sous-volume ZMQSSYS :

ZMQSTMPL (code fichier 839)

Ce fichier contient les modèles de mise en forme des événements EMS générés par MQSeries.

ZMQSDDL (code fichier 101)

Schéma du langage de définition des données pour les événements EMS générés par MQSeries.

Événements EMS

ZMQSC (code fichier 101)

Définitions des événements EMS générés par le produit, compilées en C par le compilateur DDL.

ZMQSCOB (code fichier 101)

Définitions des événements EMS générés par le produit, compilées en COBOL par le compilateur DDL.

ZMQSPAS (code fichier 101)

Définitions des événements EMS générés par le produit, compilées en PASCAL par le compilateur DDL.

ZMQSTACL (code fichier 101)

Définitions des événements EMS générés par le produit, compilées en TACL par le compilateur DDL.

ZMQSTAL (code fichier 101)

Définitions des événements EMS générés par le produit, compilées en TAL par le compilateur DDL.

Le sous-volume ZMQSSYS contient SMQSTMPL, le fichier modèle EMS, à partir duquel le fichier modèle ZMQSTMPL est généré. Le fichier ZMQSTMPL est prêt à être intégré avec les modèles d'événement de votre système, à l'aide des commandes COUP et SYSGEN. Le source des modèles d'événement vous est fournie, vous pouvez donc modifier la mise en forme des événements lorsqu'ils sont utilisés dans votre environnement.

Vous pouvez décider d'ajouter ou de modifier le texte qui est affiché avec les informations de l'événement, ou ne pas vouloir afficher l'ensemble des informations. Vous trouverez dans la documentation Compaq la description du langage source de modèle d'événement EMS, ainsi que celle des méthodes de compilation, qui vous permettront de générer un autre fichier ZMQSTMPL.

Intégration des modèles d'événements EMS pour MQSeries

Le fichier modèle objet doit être intégré aux fichiers modèles EMS (résidants ou non) de votre système. De la sorte, il pourra être utilisé par des programmes comme VIEWPOINT et EMSDIST pour mettre en forme et afficher les événements EMS MQSeries.

Le reste de cette section décrit la procédure d'intégration des modèles EMS pour MQSeries aux modèles du système. Veuillez noter que des méthodes différentes peuvent être mieux adaptées à votre installation.

1. Déterminez les noms des modèles système en cours, à l'aide de la commande `COUP INFO ALLPROCESSORS` : prenez note des valeurs affichées pour le paramètre `EMS^TEMPLATES`. Par exemple :

```

$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. Pour un système d'exploitation série G, déterminez le nom du modèle système courant, à l'aide de la commande `SCF ASSUME SUBSYS $ZZKRN; INFO`. Prenez note des valeurs affichées pour `NONRESIDENT_TEMPLATES` et `RESIDENT_TEMPLATES`. Par exemple :

```

SCF;ASSUME SUBSYS $ZZKRN;INFO
NONSTOP KERNEL - Info SUBSYS \HAWK.$ZZKRN
Current Settings
*DAYLIGHT_SAVING_TIME ..... USA66
*NONRESIDENT_TEMPLATES..... $SYSTEM.SYS01.TEMPLATE
*POWERFAIL_DELAY_TIME..... 30
*RESIDENT_TEMPLATES..... $SYSTEM.SYS01.RTMPLATE
SUPER_SUPER_IS_UNDENIABLE..... OFF
*SYSTEM_NAME..... \HAWK
*SYSTEM_NUMBER..... 2
SYSTEM_PROCESSOR_TYPE ..... NSR-W
*TIME_ZONE_OFFSET..... -05:00

Pending Changes (will take effect at next system load)
None Total Errors = 0 Total Warnings = 0

```

3. Lancez le compilateur `TEMPLI` pour créer de nouveaux fichiers de modèles système, combinant les modèles en cours avec les nouveaux modèles MQSeries. Ceci se fait en deux étapes :
 - a. Créez un fichier texte contenant les commandes suivantes :

```

FILE <fichier de modèle système NONRESIDENT actuel>
FILE <volume d'installation MQSeries>.ZMQSSYS.ZMQSTMPL
EXIT

```

Événements EMS

Par exemple :

```
FILE $SYSTEM.SYS06.TEMPLATE
FILE $DEV2.ZMQSSYS.ZMQSTMPL
EXIT
```

- b. Lancez le compilateur TEMPLI, en lui indiquant le nouveau fichier texte en entrée :

```
TEMPLI /IN <fichier de commande>/<nouveau fichier modèle résident>, <nouveau
fichier modèle non résident>
```

Par exemple, si vous avez appelé TEMGUIDE le fichier de commandes que vous avez créé, et que vous générez les nouveaux fichiers de modèles dans \$SYSTEM.EMS :

```
TEMPLI /IN TEMGUIDE/$SYSTEM.EMS.NEWRES, $SYSTEM.EMS.NEWNRES
```

La compilation des nouveaux fichiers modèles peut prendre quelques minutes, car elle doit traiter tous les modèles d'événements EMS de votre système.

4. Configurez votre système pour utiliser les nouveaux fichiers modèles d'événements EMS à la place des modèles en cours, avec la commande COUP :

```
ASSUME ALLPROCESSORS
ALTER EMS`TEMPLATES(RESIDENT <new resident template file>,
                    NONRESIDENT <new nonresident template file>)
EXIT
```

Pour la série G, c'est une commande SCF qui servira à configurer votre système pour qu'il utilise les nouveaux modèles d'événements EMS :

```
ALTER $ZZKRN, RESIDENT_TEMPLATES $SYSTEM.SYS01.NEWRES
ALTER $ZZKRN, NONRESIDENT_TEMPLATES $SYSTEM.SYS01.NEWNRES
EXIT
```

Remarque : Pour rendre cette modification permanente, vous devez mettre à jour le système avec SYSGEN.

Vous trouverez davantage d'informations concernant les modèles EMS dans le manuel Compaq *DSM Template Services Manual*. Il explique également comment utiliser SYSGEN pour effectuer cette tâche.

Définition du PARAM MQEMSEVENTS

Pour terminer l'activation des événements EMS MQSeries, vous devez vous assurer que le PARAM MQEMSEVENTS est correctement défini. Sa valeur est une chaîne de quatre caractères, codée par les masques binaires suivants :

Message EMS	Masque	Valeur de MQEMSEVENT
FFST	0x00000001	1
START / STOP	0x00000002	2
PERFORMANCE	0x00000004	4
CANAL	0x00000008	8
GESTIONNAIRE DE FILE D'ATTENTE	0x00000010	16
MESSAGE	0x00000020	32
ERREUR	0x00000040	64
TOUS	0x0000007F	127

Par conséquent, pour activer tous les événements EMS pour MQSeries, vous devez définir le PARAM depuis l'environnement TACL (où s'effectuent toutes les commandes d'administration) de la façon suivante :

```
PARAM MQEMSEVENTS 127
```

Cette définition doit également accompagner toute définition de classe de serveur de MQSeries. Chaque classe de serveur peut être configurée avec différentes options. Pour plus d'informations, reportez-vous à la section «Modification des paramètres de classe de serveurs PATHWAY» à la page 42.

Par défaut, aucun PARAM n'est défini, et aucun événement EMS généré.

Utilisation d'un autre programme de collecte

Sur un système Compaq NSK, le collecteur d'événements EMS par défaut est nommé \$0 ; il est toujours présent. Tous les événements EMS générés par un gestionnaire de files d'attente MQSeries sont envoyés au programme de collecte par défaut. Si vous souhaitez qu'un autre programme se charge de collecter les événements EMS d'un gestionnaire de files d'attente, vous devez modifier l'entrée EMSCollector de la strophe Configuration du fichier QMINI, puis redémarrer le gestionnaire de files d'attente. Vous pouvez préciser un programme de collecte différent pour chaque gestionnaire de files d'attente.

Écriture de programmes de traitement des événements EMS MQSeries

Vous pouvez écrire une application qui contrôlera un gestionnaire de files d'attente MQSeries en traitant les messages d'événements EMS. Ce genre d'application peut également modifier le fonctionnement d'un gestionnaire de files d'attente en lançant des commandes PCF qui répondent aux messages d'événements EMS qui sont générés.

Les fichiers ZMQSC, ZMQSTAL, ZMQSCOB, ZMQSPAS et ZMQSTACL, livrés avec MQSeries pour Compaq NSK et qui sont dans le sous-volume ZMQSSYS, définissent les éléments des messages d'événements EMS MQSeries EMS en C,

Événements EMS

TAL, COBOL, PASCAL et TAQL. Un programme d'administration utilisera ces définitions pour comprendre le format des messages.

Vous trouverez davantage d'informations concernant les événements EMS générés par MQSeries, à l'«Annexe P. Modèle de fichier d'événement EMS utilisé par MQSeries pour Compaq NSK» à la page 405.

Chapitre 11. Prise en charge transactionnelle et messagerie

Les applications qui utilisent MQI (Message Queue Interface) permettent d'effectuer des opérations Put et Get sous le contrôle de points de synchronisation. MQSeries pour Compaq NSK dispose de deux opérations transactionnelles :

- Validation (commit) — le fait de terminer une transaction, de manière à ce que les modifications de la base de données soient définitives et enregistrées. Les ressources protégées sont libérées une fois que la transaction est validée.
- Annulation (back out) — le fait d'annuler toutes les modifications effectuées pendant l'unité de reprise ou l'unité de travail en cours. Une fois l'opération terminée, une nouvelle unité de reprise ou de travail peut être démarrée.

Les opérations de validation et d'annulation font partie de l'environnement transactionnel TM/MP (ou TMF) de Compaq NSK. Sur MQSeries pour Compaq NSK, les opérations MQPUT, MQGET et MQPUT1 sont par défaut sous le contrôle de points de synchronisation. Ce qui signifie qu'à moins que le contrôle par points de synchronisation ne soit explicitement requis par une application, il *faudrait* qu'une transaction TMF soit en cours, sinon l'appel MQI échoue.

Une application lance une transaction TM/MP grâce à la procédure BEGINTRANSACTION. Elle valide la transaction avec ENDTRANSACTION et peut annuler la transaction avec ABORTTRANSACTION.

MQSeries pour Compaq NSK prend également en charge les produits qui fonctionnent avec des points de synchronisation via TM/MP, comme NonStop Tuxedo.

Utilisation de NonStop TM/MP (gestionnaire de transactions)

Pour maintenir l'intégrité des transactions, MQSeries pour Compaq NSK V5.1 s'appuie sur les fonctions de gestion de transaction de Compaq NonStop TM/MP.

Le système de transactions NonStop TM/MP assure l'accès concurrent et la protection des transactions, ainsi que l'intégrité des messages et des catalogues des objets.

Les transactions TM/MP peuvent coordonner les opérations Put et Get de MQSeries, en mettant à jour les fichiers de bases de données ENSCRIBE ou NonStop SQL créés par l'application.

MQSeries gère les transactions TM/MP de façon transparente. Si vous avez déjà une transaction TM/MP en cours lors de l'appel d'une fonction MQI, toutes les opérations Put et Get avec l'option de point de synchronisation feront partie de cette transaction. Ce qui signifie que la file d'attente est mise à jour lorsque la transaction est validée. En cas d'échec, TM/MP s'assure que toutes les transactions validées sont appliquées aux fichiers de base de données, et que les transactions non validées sont annulées. L'annulation d'une transaction se fait en appliquant une image de sauvegarde aux enregistrements, qui les remet dans l'état précédant la transaction. Les modifications ne prennent effet qu'après la réussite d'une opération de validation.

Utilisation de TM/MP

Si une application a déjà une transaction en cours, et effectue un appel MQI hors point de synchronisation, MQSeries interrompt la transaction en cours, en démarre une qui lui est propre, la valide, puis relance la première transaction avant de rendre le contrôle à l'application. Les mises à jour de la file d'attente, résultant d'opérations Put ou Get, prennent effet immédiatement.

Les transactions TM/MP servent à coordonner les opérations Put et Get sur tous les messages, persistants ou non. Pour permettre cette coordination, MQSeries pour Compaq NSK utilise une interface particulière vers TM/MP, nommée *OpenTMF*.

Limitations du système de points de synchronisation

Le système de fichiers peut limiter la quantité des données de messages persistants qui peuvent subir une opération Put ou Get, dans le cadre d'une même transaction. Il limite le nombre de verrouillages sur les fichiers physiques qui enregistrent les données des messages persistants.

Cette limite est de 5000 verrous, par transaction et par volume disque. Vous pouvez la modifier avec SCF. Par exemple, la commande suivante change cette limite à 10000 verrous par transaction et par volume disque :

```
ALTER DISK $DISK01, MAXLOCKSPERTCP 10000
```

Le nombre de messages enregistrés dans les fichiers de dépassement de capacité de file d'attente est également limité (parce que leur taille est inférieure au seuil d'utilisation des fichiers de dépassement de capacité de messages).

Nous vous conseillons de ne pas dépasser la valeur de 200 Ko pour ce seuil de dépassement de capacité de message. Pour cette valeur, la limite par défaut du nombre de verrous d'enregistrement permet de gérer environ 100 messages dans le cadre d'une même transaction — ce qui est suffisant dans la plupart des cas. De plus, les avantages des fichiers de dépassement de capacité de message commencent à être significatifs à partir de cette taille de message.

Veuillez noter également que l'utilisation du fichier d'audit de TM/MP est considérablement réduite lorsque les fichiers de dépassement de capacité de message sont utilisés à la place des fichiers de dépassement de capacité de file d'attente.

Vous trouverez davantage d'informations concernant les différences et avantages des fichiers de dépassement de capacité de message et de file d'attente dans la section «Fichiers de dépassement de capacité» à la page 224.

Opérations sur les messages persistants, sans point de synchronisation

Pour que les fichiers ENSCRIBE qui enregistrent les données des messages persistants puissent être mis à jour, il faut que les transactions TM/MP soient démarrées en interne par le serveur de file d'attente. Le système de fichiers NSK impose pour tout processus une limite de 100 transactions simultanées. Par conséquent, un serveur de file d'attente ne peut gérer plus de 100 opérations Put ou Get sur des messages persistants sans point de synchronisation. Cette limite peut être atteinte si de nombreux messages de grande taille (plus de 32 Ko) sont ajoutés ou retirés de la file d'attente en même temps, et via le même serveur de file d'attente.

Dans ce cas, la commande MQPUT ou MQGET sera interrompue et retournera un code anomalie MQRC_SYNCPOINT_LIMIT_REACHED. Pour répartir la charge entre plusieurs processus, il convient d'assigner les files d'attente à d'autres serveurs, ou de modifier les applications pour qu'elles utilisent des file d'attente différentes hébergées par des serveurs de file d'attente différents.

Opérations avec point de synchronisation sur les messages non persistants

Les messages non persistants sont enregistrés en mémoire, et non sur des fichiers disque audités. Par conséquent, ils n'ont pas besoin, en eux-mêmes, d'espace pour le fichier d'audit. MQSeries utilise une interface interne de TM/MP pour contrôler la disponibilité des messages non persistants qui sont ajoutés ou retirés de la file d'attente avec des points de synchronisation. Les mélanges de messages persistants et non persistants doivent faire partie d'une même opération de point de synchronisation. —MQSeries s'assurera qu'à la clôture de la transaction TM/MP, les opérations sur tous les messages non persistants seront validées ou annulées de manière logique, en même temps que celles des messages persistants, conformément au résultat de la transaction.

Configuration requise pour TM/MP et MQSeries pour Compaq NSK

Tous les volumes de votre système NSK qui auront à héberger des gestionnaires de file d'attente ou des files d'attente doivent être configurés avec l'option d'audit validée pour TMF (TM/MP). La commande `status datavols` de TMFCOM vous permet de déterminer l'état de l'audit sur les volumes de votre système. (Attention, pour utiliser TMFCOM vous devez être SUPER.SUPER.) De plus, les fichiers d'audit TMF configurés pour les volumes de données qui supportent les gestionnaires de file d'attente, doivent avoir une capacité suffisante pour gérer les pointes de trafic et la taille des messages que peuvent recevoir tous ces gestionnaires de file d'attente.

Sachant par ailleurs que les applications mal gérées peuvent induire des transactions de longue durée, le système TMF doit être configuré pour annuler ce type de transaction. La taille du fichier d'audit et la limite de durée d'exécution des transactions, sont des paramètres d'optimisation qui dépendent de l'application. Le fichier d'audit configuré pour MQSeries ne doit pas être réglé pour un vidage sur bande.

Contrôle

L'interface TMFCOM sert à contrôler le statut de TMF sous MQSeries. Les commandes `status tmf` et `status datavols` permettent de contrôler le statut général de TMF, ainsi que celui de chaque volume de données.

Le journal d'événements système (EMS) peut également être surveillé. Il peut donner des indications sur des événements TM/MP importants, indiquant des problèmes potentiels de TM/MP, susceptibles d'affecter MQSeries ou les applications qui l'utilisent. TM/MP est une ressource essentielle au fonctionnement correct de MQSeries, et doit rester actif en permanence.

Taille du fichier d'audit

Les règles suivantes permettent de calculer la taille approximative du fichier d'audit de TM/MP :

- L'espace du fichier d'audit n'est requis que pour les opérations de messages persistants (Put et Get destructif).

Configuration requise pour TM/MP

- Cet espace doit être environ égal à la taille totale des données des messages, plus 1500 octets.
- Les opérations sur messages persistants qui dépassent le seuil de capacité n'ont besoin que de 4 Ko de fichier d'audit par opération Put ou Get, quelle que soit leur taille.

Configuration du gestionnaire de ressources

L'interface interne de TM/MP doit être configurée en fonction du volume de transactions prévu pour être géré par MQSeries. Les serveurs de file d'attente MQSeries jouent le rôle de gestionnaire de ressources dans le cadre du sous-système TM/MP. Ce sous-système contient donc plusieurs seuils et limites qui s'appliquent aux gestionnaires de ressources. La configuration requise dépend du nombre de serveurs de file d'attente utilisés, de leur répartition entre les différents systèmes et processeurs, et du nombre d'opérations de point de synchronisation qui sont actives en même temps. La commande ALTER BEGINTRANS de TMFCOM sert à modifier ces valeurs, ainsi qu'il est décrit ci-dessous :

RMOPENPERCPU - au minimum le double du nombre maximal de serveurs de file d'attente qui fonctionneront sur un processeur. La valeur par défaut, 128, est en général satisfaisante.

BRANCHESPERRM - au moins égale au nombre maximal d'opérations de point de synchronisation qui seront traitées par un même serveur de file d'attente. La valeur par défaut, 128, est généralement suffisante, mais peut être augmentée si nécessaire (au maximum 1024). Il est également possible d'assigner des files d'attente à d'autres serveurs de file d'attente, afin de réduire le nombre maximal d'opérations de point de synchronisation traitées par un même serveur de file d'attente.

Pour que les modifications de ces paramètres soient prises en compte, il faut arrêter et relancer TM/MP.

Identification des problèmes

Des événements de type EMS, ou des rapports FFST, indiquant que des commandes BEGINTRANSACTION ont été désactivées par TMF, signifient en général que le fichier d'audit est plein. Ceci peut se produire parce que sa taille est insuffisante, ou parce qu'une application mal gérée a lancé une transaction de trop longue durée, et que TMF ne l'a pas terminée dans les délais.

Dans ce cas :

- Accroissez la taille du fichier d'audit, ou
- Identifiez l'origine de la transaction de longue durée et corrigez-la, ou
- Reconfigurez TMF pour qu'il interrompe les transaction de longue durée plus tôt.

Les événements EMS et rapports FFST qui signalent un arrêt de TMF indiquent un problème de configuration, qui doit être corrigé avant de relancer le gestionnaire de files d'attente. Les gestionnaires de files d'attente MQSeries dépendent du bon fonctionnement de TMF. Bien que les messages ne soient ni perdus ni endommagés, le gestionnaire de files d'attente ne peut fonctionner sans TMF.

Chapitre 12. Reprise et redémarrage

Un système de messagerie assure la livraison à destination des messages entrés dans le système. Il doit également fournir une méthode de suivi des messages sur le système, ainsi qu'une méthode de reprise si le système échoue pour une raison quelconque.

A l'aide de Compaq NonStop Transaction Manager (TM/MP), MQSeries pour Compaq NSK veille à ce que les messages persistants ne soient pas perdus. TM/MP assure la protection des transactions, la cohérence des files d'attente et le rétablissement des fichiers de file d'attente.

MQSeries pour Compaq NSK utilise également une technologie de paire de processus NonStop afin que même les messages non persistants résistent aux pannes. Les serveurs de files d'attente responsables du stockage des messages contrôlent les messages non persistants depuis le processus de sauvegarde exécuté sur un processeur différent.

Le contrôle de messages non persistants est une option propre à la file d'attente pouvant être configurée à l'aide de **altmqfls**. Le contrôle des messages non persistants est activé par défaut.

Le sous-système TM/MP gère les opérations complexes pour une meilleure cohérence des bases de données et transactions actuelles, et des opérations d'utilisateurs et de MQSeries plus efficaces. Elles sont alors transparentes pour les utilisateurs et programmes d'applications.

La reprise restaure le gestionnaire de files d'attente dans l'état qui était le sien au moment de l'arrêt. Les transactions en cours sont abandonnées et tous les messages non validés au moment où le gestionnaire de files d'attente s'est arrêté sont supprimés des files d'attente. La reprise restaure tous les messages persistants mais les messages non persistants sont perdus pendant la procédure.

Le reste du chapitre présente les concepts de reprise et de redémarrage de façon plus détaillée, et vous explique comment procéder en cas d'incident. Les sujets suivants sont traités :

- «Tolérance aux pannes et reprise»
- «Sauvegarde et restauration de MQSeries» à la page 186

Tolérance aux pannes et reprise

Si vous configurez correctement MQSeries Version 5.1, ainsi que le logiciel et le matériel système Compaq NSK (par exemple, si tous les composants sont configurés en tant que dispositifs redondants ou en miroir, selon les indications de Compaq), l'échec de tout composant logiciel ou matériel unique ne provoque aucune perte, duplication ou altération de données, ni perte permanente d'aucune fonction du système (dont la restauration nécessiterait une intervention extérieure). MQSeries pour Compaq NSK V5.1 peut surmonter un point de panne unique tout en maintenant l'intégrité des données, de la façon décrite ci-dessous.

L'échec répété consécutif (par exemple une succession d'échecs et de reprises) du même composant logiciel fait l'objet d'une seule alerte, une fois dépassé le nombre

Tolérance aux pannes

d'échecs maximum configuré. Dans ce cas, ou dans celui d'incidents multiples, MQSeries ne peut pas préserver l'intégrité de la file d'attente.

Pour plus d'informations sur la configuration d'un gestionnaire de files d'attente en vue de préserver l'intégrité et la disponibilité des données, reportez-vous au «Chapitre 16. Intégrité et disponibilité des données» à la page 231.

Sauvegarde et restauration de MQSeries

Vous pouvez régulièrement effectuer une sauvegarde des données de votre gestionnaire de files d'attente afin de pouvoir les récupérer en cas d'incident matériel.

Sauvegarde de MQSeries

Pour faire une copie de sauvegarde des données d'un gestionnaire de files d'attente, suivez les étapes suivantes :

1. Vérifiez que le gestionnaire de files d'attente n'est pas actif.
S'il est en cours de fonctionnement, arrêtez-le à l'aide de la commande **endmqm**.

Remarque : Si vous tentez de faire une copie de sauvegarde d'un gestionnaire de files d'attente en cours de fonctionnement, la sauvegarde risque de ne pas être cohérente à cause des mises à jour en cours au moment de la copie des fichiers.

2. Localisez les volumes et sous-volumes dans lesquels le gestionnaire de files d'attente stocke ses données.

Vous pouvez utiliser les informations des fichiers de configuration pour déterminer quels sont ces répertoires. Pour plus d'informations, reportez-vous au «Chapitre 13. Fichiers de configuration» à la page 189.

Remarque : Les noms qui apparaissent dans le répertoire sont difficilement compréhensibles. En effet, ils ont été transformés pour garantir leur compatibilité avec la plateforme sur laquelle vous utilisez MQSeries. Pour plus d'informations sur la transformation des noms, reportez-vous à la section «Structure du volume» à la page 62.

3. Faites des copies de toutes les données du gestionnaire de files d'attente et des sous-volumes de journalisation.

Assurez-vous que vous n'oubliez aucun fichier.

Restauration de MQSeries

Pour faire une copie de sauvegarde des données d'un gestionnaire de files d'attente, suivez les étapes suivantes :

1. Vérifiez que le gestionnaire de files d'attente n'est pas actif.
2. Localisez les sous-volumes dans lesquels le gestionnaire de files d'attente stocke ses données. Ces informations se trouvent dans le fichier de configuration.
3. Videz les sous-volumes dans lesquels vous allez placer les données sauvegardées.
4. Copiez les données sauvegardées du gestionnaire de files d'attente aux emplacements appropriés.

Contrôlez la structure de répertoires obtenue pour vérifier que tous les répertoires requis sont présents.

Vérifiez que le fichier de configuration MQSeries et celui du gestionnaire de files d'attente sont cohérents pour que MQSeries recherche les données restaurées aux emplacements appropriés.

Si les données ont été sauvegardées et restaurées correctement, le gestionnaire de files d'attente peut maintenant démarrer.

Reprise et redémarrage des serveurs de statut et serveurs de file d'attente

Les processus de serveur de file d'attente et de serveur de statut sont des paires de processus Compaq NSK. Ils sont conçus pour continuer à fonctionner même en cas d'incident d'un processeur ou du processus lui-même. Dans le cas d'un incident unique (par exemple, en cas d'incident du processeur contenant le serveur de statut ou le serveur de file d'attente, ou du processus de serveur de file d'attente ou de statut lui-même), le processus de serveur de statut ou de serveur de file d'attente prend le relai, sans interrompre le traitement du gestionnaire de files d'attente.

En cas d'incident unique, aucune action de reprise spécifique au serveur de statut ou de file d'attente n'est donc nécessaire. Un message est consigné sur le terminal d'origine et dans le fichier de message par un serveur de statut ou de file d'attente, lorsque la sauvegarde doit être redémarrée par le processus principal, ou agit en tant que processus principal.

Dans le cas d'une panne plus grave (par exemple si un incident lié à l'environnement empêche l'initialisation du serveur de statut ou de file d'attente principal ou de sauvegarde), le processus PATHMON TS/MP tente de redémarrer le serveur de statut ou le serveur de file d'attente jusqu'à 10 fois.

Le serveur de statut ou de files d'attente accède uniquement aux bases de données protégées par TM/MP, de façon à ce qu'en cas d'incident affectant l'accès aux disques, la protection assurée par TM/MP et le sous-système de disque DP2 soit totalement fiables.

Le serveur de statut et le serveur de file d'attente peuvent être arrêtés et redémarrés individuellement si nécessaire, à l'aide des commandes PATHCOM TS/MP, bien que ce ne soit pas utile en temps normal. La commande **strmqm** démarre automatiquement toutes les classes de serveur de file d'attente ou de statut dont les noms commencent par la chaîne de caractères MQS-STATUS, ou MQS-QUEUE. A l'arrêt du gestionnaire de files d'attente, toutes les classes de serveur de file d'attente et de serveur de statut coordonnent leur fermeture, et une fois tous les gestionnaires de files d'attente actifs arrêtés, tout serveur de statut ou de file d'attente associé à ces connexions s'arrête.

Compaq NSK abandonne les transactions dans certaines circonstances en cas d'incident du processus principal d'une paire de processus NonStop. Ceci peut provoquer l'échec des opérations MQPUT ou MQGET en cours au moment de l'échec du processus principal. Pour plus d'informations, reportez-vous au «Chapitre 16. Intégrité et disponibilité des données» à la page 231.

Reprise et redémarrage des serveurs de statut et serveurs de file d'attente

Reprise après incident grave à l'aide de RDF

Appliquez les procédures décrites ci-dessous pour remettre en service un gestionnaire de files d'attente sur un site de sauvegarde, au cas où un incident grave rendrait le site principal inutilisable :

1. Assurez-vous que RDF a terminé la mise à jour des bases de données.
2. Utilisez FUP pour placer un indicateur de suivi sur les fichiers suivants :
 - a. Tous les fichiers du sous-volume <qmgr>M.
 - b. Tous les fichiers du sous-volume <qmgr>D, sauf QMINI, AMQRFNxx, CCSIDMEM, QMINIMEM, STABLE, UMQSINI, PATHCTL, TRACEOPT, et SHUTDOWN.
 - c. Tous les fichiers du sous-volume <qmgr>S.
3. Configurez le sous-volume <qmgr>D en tant que volume par défaut. Exécutez PATHMON manuellement, puis lancez PATHCOM et chargez la configuration PATHWAY pour le gestionnaire de files d'attente.
4. Lancez une commande INFO pour tous les objets.
 - a. Vérifiez que toutes les instances du nom de noeud apparaissent sous la forme "`*`" ou ont un nom de noeud accepté par le système de sauvegarde.
 - b. Vérifiez que les numéros de processeurs affectés aux classes de serveur sont toujours valides pour ce site de sauvegarde.
 - c. Vérifiez que les noms de terminal d'origine et de fichiers de sortie conviennent au site de sauvegarde. Si ce n'est pas le cas, modifiez-les.
 - d. Vérifiez que tout autre nom de processus TCP/IP indiqué soit accepté par ce système.
 - e. Si le site de sauvegarde n'est pas configuré pour un fonctionnement RDF, retirez tout paramètre PARAM MQRDF des définitions de classes de serveur EC.
 - f. Une fois la configuration PATHWAY vérifiée, enregistrez-la sur disque à l'aide de la commande shutdown2 et quittez PATHCOM. Si le nom du terminal d'origine a changé, modifiez le fichier QMINI du gestionnaire de files d'attente conformément à la configuration PATHWAY. Si nécessaire, modifiez la configuration des ports d'écoute TCP/IP dans le fichier QMINI.
 - g. Utilisez **strmqm** pour démarrer le gestionnaire de files d'attente.
 - h. A l'aide de **runmqsc**, vérifiez la configuration du canal, et adaptez-la si nécessaire.
 - i. Si vous tentez d'utiliser les mêmes canaux que ceux qui étaient auparavant utilisés, il peut s'avérer nécessaire de modifier la configuration de canal des gestionnaires de files d'attente éloignés, à moins que le système de sauvegarde ne puisse être reconfiguré de façon à utiliser la même adresse IP de nom d'hôte, pour les canaux TCP/IP, ou les mêmes noms de ressource SNAX/APC et ICE (tels que le nom de processus, noms LU etc.) pour les canaux SNA.
 - j. Préparez-vous à rencontrer des erreurs de séquence ou de synchronisation de canal, en particulier si les canaux du site principal fonctionnaient au moment de l'incident. RDF ne garantit pas que les bases de données du site de secours sont à jour (et parfaitement identiques à celles du site principal) : il est donc possible que des données soient perdues en cas de panne totale. Pour réduire les risques, veillez à ce que votre configuration RDF puisse manipuler le volume des mises à jour de bases de données associé au flux des messages.

Chapitre 13. Fichiers de configuration

MQSeries pour Compaq NSK utilise des *fichiers de configuration* qui rassemblent les informations de paramétrage de base du produit. Ce chapitre décrit les fichiers de configuration et indique comment vous pouvez les utiliser pour modifier le fonctionnement des gestionnaires de files d'attente. Ce chapitre contient les sections suivantes :

- «Description des fichiers de configuration»
- «Fichier de configuration (MQSINI)»
- «Fichier de configuration du gestionnaire de files d'attente (QMINI)» à la page 191
- «Modification des fichiers de configuration» à la page 198

Description des fichiers de configuration

Les fichiers de configuration définissent les options des gestionnaires de files d'attente individuels et de MQSeries sur le noeud dans son ensemble. Ces fichiers sont désignés sous le nom de fichiers *ini* ou de fichiers de *strophe*. Un fichier de configuration contient une ou plusieurs strophes. Une strophe est simplement un groupes de lignes du fichier, l'ensemble ayant une fonction commune ou définissant une partie d'un système (par exemple, les fonctions de journalisation, de canal, ou des services optionnels).

Les fichiers de configuration peuvent être modifiés automatiquement par des commandes qui changent la configuration des gestionnaires de files d'attente du noeud, ou en les éditant manuellement. Toutefois, les fichiers de configuration doivent en général être modifiés manuellement pendant l'exécution des gestionnaires de files d'attente.

Il existe deux types de fichiers de configuration :

- Le *fichier de configuration MQSeries*, MQSINI, qui précise les valeurs de MQSeries sur le noeud, celui-ci étant considéré comme un tout. Il existe normalement un fichier de configuration MQSeries par noeud.
- Les *fichiers de configuration de gestionnaire de files d'attente*, QMINI, qui précisent les valeurs propres aux gestionnaires de files d'attente. A chaque gestionnaire de files d'attente du noeud correspond un fichier de configuration de gestionnaire.

Fichier de configuration (MQSINI)

Le fichier de configuration MQSeries, MQSINI, contient des informations applicables à tous les gestionnaires de files d'attente d'un noeud d'installation MQSeries. Il est créé automatiquement au cours de l'installation. Il est utilisé en particulier pour localiser les données associées à chaque gestionnaire de files d'attente. Il se situe dans le sous-volume ZMQSSYS, par défaut \$SYSTEM.ZMQSSYS.MQSINI. Une variable d'environnement, MQMACHINIFILE, est utilisée sur les systèmes dont les fichiers de configuration MQSeries n'utilisent pas les paramètres de nom et d'emplacement par défaut.

Fichier de configuration MQSeries

Contenu du fichier de configuration MQSeries

Le fichier MQSINI contient les valeurs par défaut de l'installation, les noms des gestionnaires de files d'attente, le nom du gestionnaire de files d'attente par défaut, et l'emplacement des fichiers associés à chacun d'entre-eux. Les strophes suivantes peuvent apparaître dans MQSINI :

AllQueueManagers

Précise les valeurs des volumes et adresses de fichiers de toute l'installation.

DefaultQueueManager

Indique le gestionnaire de files d'attente par défaut de l'installation. Ce gestionnaire de files d'attente traite les commandes MQSC pour lesquelles vous n'avez pas indiqué de nom de gestionnaire. La strophe est automatiquement mise à jour lorsque vous créez un nouveau gestionnaire de files d'attente par défaut. Si vous créez par erreur un gestionnaire de files d'attente par défaut et voulez ensuite revenir à l'original, vous devez modifier cette strophe manuellement.

QueueManager

Il existe une strophe de ce type pour chaque gestionnaire de files d'attente. La strophe QueueManager définit le nom du gestionnaire de files d'attente et l'adresse des fichiers associés au gestionnaire. Ces noms sont basés sur celui du gestionnaire, mais ils peuvent être transformés si le nom du gestionnaire n'est pas un nom de fichier correct.

La figure 42 montre un exemple de fichier MQSINI.

```
#####  
#* Module Name: MQSINI                                     #*  
#* Type:          MQSeries machine-wide ini file         #*  
#* Function:     Define configuration data for all queue managers #*  
#*                                                     #*  
#####  
#* Notes      :                                         #*  
#* 1) This file defines configuration data for all queue managers #*  
#*                                                     #*  
#####  
AllQueueManagers:  
    MQSVolume=$DATA00                /Volume for the installation  
    MQSExePath=$DATA00.ZMQSEXE      /Location of product executables  
    QMDefaultVolume=$DATA00         /Default volume for queue manager creation  
    ConvEBCDICNewline=NL_TO_LF      /Data Conversion EBCDIC Newline  
    NSKSegidRange=10-20             /Segment Id Range  
QueueManager:  
    Name=MT01                        /A queue manager called MT01  
    QMVolume=$DATA00                /Volume of the queue manager  
    QMSubVolume=MT01                /Subvolume prefix for the queue manager  
DefaultQueueManager:  
    Name=MT01                        /Name of the default queue manager (optional)
```

Figure 42. Exemple de fichier de configuration MQSeries (MQSINI). Le fichier MQSINI est initialisé pendant l'installation avec les informations de volume et de sous-volume que vous avez indiquées.

Remarque : Etant donné que le fichier de configuration MQSeries est utilisé pour localiser les données associées aux gestionnaires de files d'attente, un fichier de configuration inexistant ou incorrect peut provoquer l'échec de tout ou partie des commandes MQSeries. De même, les

applications ne peuvent pas se connecter à un gestionnaire de files d'attente non défini dans le fichier de configuration MQSeries.

Fichier de configuration du gestionnaire de files d'attente (QMINI)

Chaque fichier de configuration de gestionnaire de files d'attente, QMINI, contient les informations relatives à ce gestionnaire. Il existe un fichier de configuration par gestionnaire de files d'attente. Il est créé automatiquement en même temps que le gestionnaire de files d'attente auquel il est associé.

Le fichier se trouve dans le sous-volume du gestionnaire de files d'attente. Par exemple, l'adresse et le nom du fichier de configuration d'un gestionnaire de files d'attente NOMGESTFILES serait \$VOLUME.NOMGESTFILESETC.QMINI.

Remarque : Le nom du gestionnaire de files d'attente peut comporter jusqu'à 48 caractères. Un nom de sous-volume est généré en fonction du nom du gestionnaire de files d'attente. Ce processus s'appelle la *transformation de nom*, et assure que le nom est valide et unique.

Contenu du fichier de configuration du gestionnaire de files d'attente

Les strophes qui apparaissent dans un fichier de configuration de gestionnaire de files d'attente, QMINI, sont les suivantes :

Configuration

Cette strophe définit les configurations globales du gestionnaire de files d'attente.

Les entrées suivantes peuvent être modifiées :

- DefaultQueueServerName
- DefaultStatusServerName
- HomeTerminalName
- PathmonProcName
- EMSCollectorName
- MinIdleMCALU62Responders
- MinIdleMCATCPResponders
- MinIdleMCAcallers
- MinIdleLQMAgents

Pour plus d'informations sur ces entrées, reportez-vous à la section «Modification des propriétés d'un gestionnaire de files d'attente» à la page 48. Les autres entrées de cette strophe ne peuvent pas être modifiées.

DefaultProcess

Cette strophe définit les valeurs par défaut utilisées par les processus MQSeries. Les entrées de cette strophe ne peuvent pas être modifiées.

ECBoss

Cette strophe définit la configuration du processus EC Boss MQSeries. L'entrée ExpectedNumECs définit le nombre de processus EC de ce gestionnaire de files d'attente. Cette valeur doit correspondre à la configuration PATHWAY du gestionnaire de files d'attente. Pour plus d'informations, reportez-vous à la section «Modification des propriétés d'un gestionnaire de files d'attente» à la page 48. Les autres entrées de cette strophe ne peuvent pas être modifiées.

EC Les entrées MCAAgentPriority et LQMAgentPriority de la strophe EC, qui

Fichier de configuration du gestionnaire de files d'attente.

contrôlent les priorités de traitement des processus agents, peuvent être modifiées. Pour plus d'informations, reportez-vous à la section «Modification des propriétés d'un gestionnaire de files d'attente» à la page 48 . Les autres entrées de cette strophe ne doivent pas être modifiées.

Les strophes suivantes définissent les paramètres d'exploitation spécifiques à chaque type de processus MQSeries. En principe, nous n'avez pas à changer les valeurs de ces paramètres. Toutefois, consultez la section «Exemple de fichier de configuration de gestionnaire de files d'attente» à la page 193.

MCACaller

MCATCPResponder

MCALU62Responder

MQIServer

LQMAgent

ChannelInitiator

TCPListener

Authority

Fournit le mécanisme recommandé pour activer et désactiver l'OAM d'un gestionnaire de files d'attente. Placez l'indicateur MQAUTH sur On ou Off pour activer ou désactiver l'OAM sans avoir à ajouter ou supprimer les strophes Service et Service Component.

Service

Indique le nom d'un service optionnel et le nombre de ses points d'entrée. Il existe une strophe pour chaque service. Les services disponibles sont les suivants :

- Service d'autorisation
- Service annuaire

Le gestionnaire des droits d'accès aux objets (OAM) est activé par défaut : la strophe de service d'autorisation et la strophe ServiceComponent qui lui est associée, figurent par défaut dans QMINI.

Vous pouvez désactiver l'OAM en plaçant simplement l'indicateur MQAUTH de la strophe Authority sur Off et en redémarrant le gestionnaire de files d'attente. Une autre méthode consiste à :

1. Supprimer le gestionnaire de files d'attente (à l'aide de la commande **dltmqm**)
2. Recréer le gestionnaire de files d'attente (à l'aide de la commande **crtmqm**) avec la variable d'environnement MQSNOAUT définie.
3. Supprimer les strophes du service d'autorisation à partir de QMINI.

Vous devez ajouter la strophe de service d'annuaire manuellement à QMINI si vous souhaitez activer le service d'annuaire proposé.

ServiceComponent

Ces strophes définissent le composant de service associé à un service particulier. Il peut exister plusieurs strophes de service par service, mais chacune doit concorder avec la strophe de service correspondante.

Consultez le manuel *MQSeries Programmable System Management* pour plus d'informations.

Fichier de configuration du gestionnaire de files d'attente.

TuningParameters

Cette strophe définit les paramètres de réglage interne utilisés par les agents de gestionnaire de files d'attente locales. Ne modifiez pas ces valeurs.

Channels

Cette strophe contient des informations sur les canaux. Elle détermine le nombre maximum de canaux (MaxChannels) pouvant être définis pour le gestionnaire de files d'attente, et une seconde entrée (MaxActiveChannels) limite le nombre de canaux pouvant être activés simultanément. La valeur de MaxActiveChannels doit être inférieure à celle de MaxChannels. La strophe Channels contient également une entrée (ChanInitDiscInterval) pouvant être utilisée pour régler les performances de l'initialisateur de canal. Pour plus d'informations sur ces entrées, reportez-vous à la section «Modification des propriétés d'un gestionnaire de files d'attente» à la page 48. Les autres entrées de cette strophe ne doivent pas être modifiées.

Consultez le manuel *MQSeries Intercommunication* pour plus d'informations sur les canaux.

TCPCConfig

Indique les paramètres de configuration du protocole réseau. Ces strophes remplacent les paramètres par défaut des canaux. Seules les strophes représentant les valeurs par défaut modifiées sont réellement présentes.

Les valeurs TCPListenerPort sont substituées par le programme d'écoute, si le paramètre MQLISTENPORTNUM est présent dans l'environnement du processus de programme d'écoute.

Consultez le manuel *MQSeries Intercommunication* pour plus d'informations.

Pour plus d'informations sur la modification des entrées TCPPort, TCPNumListenerPorts, et TCPListenerPort, reportez-vous à la section «Modification des propriétés d'un gestionnaire de files d'attente» à la page 48.

Exemple de fichier de configuration de gestionnaire de files d'attente

La figure 43 à la page 194 montre un exemple de fichier de configuration de gestionnaire de files d'attente (QMINI).

Fichier de configuration du gestionnaire de files d'attente.

```
#####  
** Module Name: QMINI *#  
** Type : MQSeries queue manager configuration file *#  
# Function : Define the configuration of a single queue manager *#  
** *#  
#####  
** Notes : *#  
** 1) This file defines the configuration of the queue manager *#  
** *#  
#####  
Configuration:  
  PathmonProcName=$p01p  
  DefaultStatusServerName=$p01s  
  ServerClassName=MQS-ECBOSS  
  EMSCollectorName=$0  
  HomeTerminalName=$ZTN0.#PTY001C  
  ShutdownFileName=SHUTDOWN  
  TraceOptionsFileName=TRACEOPT  
  RuntimeFileName=RUNTIME  
  StatableFileName=STATABLE  
  ChannelDefFileName=CHDEFS  
  DefaultCCSID=819  
  DefaultTraceOptions=0  
  MaxIdleAgents=10  
  MinIdleMCALU62Responders=0  
  MinIdleMCATCPResponders=0  
  MinIdleMCA callers=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  
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
```

Figure 43. Exemple de fichier de configuration de gestionnaire de files d'attente (QMINI)
(Numéro 1 de 4)

Fichier de configuration du gestionnaire de files d'attente.

```
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=MQLU6RES
  MCAAgentPriority=165
  LQMAgentPriority=165
  StopProcessTimer=3000
  IdleProcessTimer=3000
MCACaller:
  ExeFileName=MQMCACAL
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
MCATCPResponder:
  ExeFileName=MQTCPRES
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
MCALU62Responder:
  ExeFileName=MQLU6RES
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
```

Figure 43. Exemple de fichier de configuration de gestionnaire de files d'attente (QMINI)
(Numéro 2 de 4)

Fichier de configuration du gestionnaire de files d'attente.

```
MQIServer:
  ExeFileName=MQMQISER
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=300000
  IniPoolSize=256000
  Priority=175
LQMAgent:
  ExeFileName=MQLQMAG
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=50
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=120000
  IniPoolSize=200000
  Priority=175
ChannelInitiator:
  ExeFileName=RUNMQCHI
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=256000
  IniPoolSize=256000
  Priority=175
TCPListener:
  ExeFileName=RUNMQLSR
  TraceVolSubvol=$DATA1.p101L
  TracePrefix=TR
  ErrorVolSubvol=$DATA1.p101L
  ErrorPrefix=ER
  DebugMode=0
  IPCCTimeOut=10000
  IPCCMemSetSize=32000
  MemSetSize=16000
  ExtPoolSize=256000
  IniPoolSize=256000
  Priority=175
```

Figure 43. Exemple de fichier de configuration de gestionnaire de files d'attente (QMINI)
(Numéro 3 de 4)

Fichier de configuration du gestionnaire de files d'attente.

```
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
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
```

Figure 43. Exemple de fichier de configuration de gestionnaire de files d'attente (QMNI)
(Numéro 4 de 4)

Modification des fichiers de configuration

Vous pouvez modifier les fichiers de configuration par défaut pour changer les valeurs prédéfinies du système. Toutefois, avant de modifier le fichier de configuration, assurez-vous que vous avez fait une sauvegarde (au cas où une restauration s'avérerait nécessaire), et que les gestionnaires de files d'attente concernés sont arrêtés.

Vous pouvez être amené à modifier vos fichiers de configuration dans les cas suivants :

- si vous perdez un fichier de configuration (récupérez-le à partir d'une copie de sauvegarde, si possible).
- si vous devez modifier la répartition de vos gestionnaires de files d'attente sur les processeurs.
- si vous devez changer de gestionnaire de files d'attente par défaut, ce qui peut arriver si vous supprimez accidentellement le gestionnaire de files d'attente existant.
- lorsque le centre d'assistance IBM vous le conseille.

Pour plus d'informations, reportez-vous à la section «Modification des propriétés d'un gestionnaire de files d'attente» à la page 48.

Implémentation des modifications dans les fichiers de configuration

Lorsque vous modifiez un fichier de configuration, ces changements ne sont pas pris immédiatement en compte par le gestionnaire de files d'attente. Les modifications apportées au fichier de configuration MQSeries (MQSINI) deviennent effectives uniquement lorsque des gestionnaires de files d'attente MQSeries sont créés ou démarrés. Les modifications apportées à un fichier de configuration de gestionnaire de files d'attente (QMINI) sont prises en compte au démarrage de ce gestionnaire. Si le gestionnaire de files d'attente est actif lorsque vous effectuez les modifications, vous devez l'arrêter puis le relancer afin que les modifications soient prises en compte par le système.

Recommandations pour les fichiers de configuration

Lorsque vous créez un nouveau gestionnaire de files d'attente, il convient de :

- faire une sauvegarde du fichier de configuration MQSeries (MQSINI)
- faire une sauvegarde du nouveau fichier de configuration du gestionnaire de files d'attente (QMINI)

Chapitre 14. Identification des incidents

Ce chapitre donne des informations concernant le dépannage de MQSeries pour Compaq NSK. Pour identifier un incident, il faut déterminer les symptômes pour ensuite arriver à la cause.

Les problèmes de performances causés par les limitations de votre matériel ne peuvent pas toujours être résolus immédiatement. En outre, si vous pensez que la cause de l'incident réside dans le code MQSeries, prenez contact avec votre centre d'assistance IBM* Support Center. Ce chapitre contient les sections suivantes :

- «Vérifications préalables»
- «Erreurs courantes de programmation» à la page 203
- «Vérifications approfondies» à la page 204
- «Considérations sur le développement d'applications» à la page 207
- «Effet de la longueur des messages» à la page 208
- «Journaux d'erreurs» à la page 212
- «Files d'attente de rebut» à la page 215
- «Fichiers de configuration et identification des incidents» à la page 215
- «Traçage de MQSeries» à la page 215
- «First Failure Support Technology™ (FFST)» à la page 217

Vérifications préalables

Les incidents rencontrés lors de l'utilisation de MQSeries peuvent être liés aux éléments suivants :

- MQSeries
- Votre réseau
- Une application
- Le logiciel de base Compaq

Les sections qui suivent abordent des points essentiels que vous souhaitez peut-être consulter. Répondez aux questions et notez tout ce qui peut concerner votre problème.

MQSeries a-t-il fonctionné correctement auparavant ?

Si MQSeries n'a jamais fonctionné, il est probable que vous ne l'ayez pas encore configuré correctement. Reportez-vous au manuel *MQSeries pour Compaq NSK Quick Beginnings* pour vérifier que vous avez effectué toutes les étapes correctement.

Existe-t-il des messages d'erreur ?

MQSeries utilise des journaux d'erreurs pour consigner les messages concernant son propre fonctionnement, celui des gestionnaires de files d'attente qui ont été démarrés, ainsi que les données d'erreur provenant des canaux en cours d'utilisation. Vérifiez les journaux pour voir si un message concernant l'incident a été enregistré.

Pour plus de détails sur le contenu des journaux d'erreur et leur emplacement, reportez-vous à la section «Journaux d'erreurs» à la page 212.

Vérifications préalables

Existe-t-il des codes retour clarifiant l'incident ?

Si votre application reçoit un code retour indiquant qu'un appel MQI a échoué, reportez-vous au manuel *MQSeries Application Programming Reference* pour une description des codes de retour.

L'incident peut-il être reproduit ?

Si oui, étudiez les questions suivantes :

- L'incident est-il provoqué par une commande ou une requête d'administration équivalente ?
L'opération aboutit-elle si vous l'exécutez selon une autre méthode ? Si la commande fonctionne lorsqu'elle est saisie sur la ligne de commande, mais pas autrement, vérifiez que le serveur de commandes n'est pas arrêté et que la définition de la file d'attente n'a pas été modifiée dans SYSTEM.ADMIN.COMMAND.QUEUE.
- L'incident est-il provoqué par un programme ?
- Pouvez-vous identifier une application qui semble être systématiquement en cours d'exécution dans le système lorsque l'incident se produit ? Si c'est le cas, examinez l'application pour voir si elle ne génère pas une erreur.
- Le volume où doit se trouver la base de données du gestionnaire de files d'attente est-il protégé par TM/MP ? Lancez la commande TMFCOM; STATUS DATAVOLS pour tester si le volume est protégé par TM/MP.
- Les paramètres d'environnement TACL nécessaires décrits dans l'«Annexe C. Définition de variables d'environnement TACL pour MQSeries pour Compaq NSK» à la page 323 ont-ils été ajoutés ? Sélectionnez PARAM à l'invite TACL pour afficher les paramètres actuels.
- Le fichier d'audit est-il conçu pour gérer le chargement ? (Reportez-vous au «Chapitre 11. Prise en charge transactionnelle et messagerie» à la page 181) Exécutez TMFCOM; INFO AUDITTRAIL pour afficher la configuration du fichier d'audit.
- Est-ce que le code d'accès au fichier Compaq autorise l'accès aux fichiers de la base de données qmD ? Un accès au minimum en lecture seule est nécessaire pour permettre aux utilisateurs en dehors du groupe NQM d'accéder aux fichiers. Un code retour 2035 (MQRC_NOT_AUTHORIZED) est retourné suite à une demande MQCONN quand les systèmes de sécurité Compaq refusent l'accès à un fichier.

Des modifications ont-elles été apportées depuis la dernière exécution réussie ?

Lorsque vous étudiez les modifications qui ont pu être apportées récemment, vérifiez le système MQSeries et les autres programmes avec lesquels il s'interface, les éléments matériels et toute application nouvelle. Recherchez également si une nouvelle application dont vous ne connaissiez pas l'existence a été lancée sur le système.

- Avez-vous modifié, ajouté ou supprimé une définition de file d'attente ?
- Avez-vous modifié ou ajouté une définition de canal ? Des modifications ont peut-être été apportées soit aux définitions de canaux MQSeries soit à une définition de communication sous-jacente nécessaire à votre application.
- Vos applications ont-elles rencontré des codes retour provoqués par des modifications que vous avez effectuées ?
- Des modifications ont-elles été apportées au PATHWAY MQSeries ?

- Des modifications ont-elles été apportées aux fichiers d'installation MQSeries, comme par exemple changer le code d'accès au fichier ?
- Des modifications ont-elles été apportées aux fichiers base de données MQSeries Compaq NSK, comme par exemple un changement du code d'accès du fichier ou de l'audit TM/MP ?
- Des modifications ont-elles été appliquées au fichier QMINI du gestionnaire de files d'attente ?

L'exécution de l'application a-t-elle déjà abouti ?

Si l'incident semble impliquer une application spécifique, vérifiez si celle-ci s'est déjà exécutée avec succès.

Vérifiez les points suivants :

- Avez-vous modifié l'application depuis sa dernière exécution réussie ?
Si c'est le cas, il est vraisemblable que l'erreur se produit dans la partie de l'application nouvelle ou modifiée. Etudiez les modifications apportées pour voir si vous pouvez trouver la cause de l'incident. Est-ce possible d'effectuer une nouvelle tentative en utilisant une version précédente de l'application ?
- Toutes les fonctions de l'application ont-elles été utilisées auparavant ?
L'incident s'est-il produit lorsqu'une partie de l'application qui n'avait jamais été appelée auparavant a été utilisée pour la première fois ? Si c'est le cas, il est vraisemblable que l'erreur se trouve dans cette partie de l'application. Tentez de déterminer ce que l'application effectuait au moment de l'incident et vérifiez le code source de cette partie du programme pour savoir s'il comporte des erreurs.
Si l'exécution du programme a abouti plusieurs fois auparavant, vérifiez l'état de la file d'attente et les fichiers en cours de traitement lorsque l'erreur s'est produite. Il est possible qu'ils contiennent certaines données inhabituelles provoquant l'appel d'un chemin rarement utilisé dans le programme.
- L'application vérifie-t-elle tous les codes retour ?
Est-il possible que le système MQSeries ait été modifié, même très peu, de sorte que votre application ne vérifie pas les codes retour provoqués par la modification ? Par exemple, votre application considère-t-elle que les files auxquelles elle accède peuvent être partagées ? Si une file a été redéfinie comme exclusive, est-ce que votre application prend en compte les codes retour indiquant qu'elle n'a plus accès à cette file ?
- L'application s'exécute-t-elle sur d'autres systèmes MQSeries ?
Se peut-il qu'une différence dans la configuration de ce système MQSeries soit la cause de l'incident ? Par exemple, est-ce que les files d'attente ont été définies avec les mêmes longueur de message et la même priorité ?
- Avez-vous activé PARAM SAVE-ENVIRONMENT ?
Si ce n'est pas le cas, un code retour 2058 s'affichera à l'invite MQCONN. Réglez les paramètres de l'environnement de votre application comme indiqué dans l'«Annexe C. Définition de variables d'environnement TACL pour MQSeries pour Compaq NSK» à la page 323.

Si l'application ne s'est jamais exécutée correctement auparavant

Si l'application n'a encore jamais été exécutée avec succès, vérifiez-la attentivement pour voir si vous pouvez détecter des erreurs.

Vérifications préalables

Avant de contrôler le code, et selon le langage de programmation utilisé, examinez les données de sortie du programme de traduction ou du compilateur et de l'éditeur de liens, le cas échéant, pour savoir si des erreurs ont été relevées.

Si la translation, la compilation ou l'édition des liens de l'application avec la bibliothèque de chargement échoue, l'exécution de cette application sera également impossible. Consultez le manuel *MQSeries Application Programming Reference* pour plus de détails sur la création de votre application.

Si la documentation montre que chacune de ces étapes a été effectuée sans erreur, vérifiez la logique de programmation de l'application. Les symptômes de l'incident indiquent-ils la fonction qui échoue et donc la partie erronée du code ? Reportez-vous à la section «Erreurs courantes de programmation» à la page 203 pour quelques exemples d'erreurs habituelles pouvant provoquer des incidents dans les applications MQSeries.

L'incident affecte-t-il des parties spécifiques du réseau ?

Il peut être possible d'identifier les parties spécifiques du réseau affectées par l'incident (files d'attente éloignées, par exemple). Si le lien avec un gestionnaire éloigné de files d'attente de messages ne fonctionne pas, les messages ne pourront pas être transmis à une file d'attente éloignée.

Vérifiez que la connexion entre les deux systèmes est disponible et que le composant d'intercommunication de MQSeries est lancé.

Assurez-vous que les messages atteignent la file d'attente de transmission. Vérifiez également la définition de file locale de la file d'attente de transmission et des files d'attente éloignées.

Avez-vous effectué des modifications concernant le réseau ou modifié des définitions MQSeries qui auraient pu entraîner l'incident ?

Vérifiez que MQS-TCPLISxx, le serveur du programme d'écoute PATHWAY TCP de MQSeries, est démarré. Si le serveur ne peut démarrer, vérifiez qu'il n'y a pas de connexion en cours de fermeture (par exemple un état FIN-WAIT) pour le port utilisant SCF (par exemple SCF; STATUS PROCESS \$ZTC0). Le partenaire distant doit être arrêté et redémarré pour déclencher le port. Pour SNA, vérifiez que le programme d'écoute SNA MQSeries est configuré pour SNAX/APC ou ICE PATHWAY, reportez-vous à la section «Processus de canaux répondeurs LU 6.2» à la page 376.

Vérifiez que le nom de processus, le nom d'hôte ou adresse IP et le port TCP/IP corrects sont utilisés pour le nom de connexion des canaux et correspondent à la définition du canal éloigné.

L'incident se produit-il à certaines heures de la journée ?

Si c'est le cas, il est peut-être lié à la surcharge du système. Le moment de pointe des systèmes se situe généralement en milieu de matinée ou d'après-midi, et ces périodes sont celles où les incidents de surcharge ont le plus de chances de se produire. (Si votre réseau MQSeries couvre plusieurs fuseaux horaires, la période de pointe peut sembler se produire à un autre moment de la journée.)

L'incident se produit-il par intermittence ?

Un incident intermittent peut se produire si vous ne tenez pas compte de l'indépendance d'exécution des processus. Par exemple, un programme peut lancer un appel MQGET, sans indiquer d'option d'attente, avant qu'un processus antérieur ne soit achevé. Un incident intermittent peut aussi survenir si votre application tente d'extraire un message d'une file d'attente alors que l'appel qui a inséré le message est en attente de validation (c'est-à-dire qu'il n'est ni validé, ni annulé).

Des erreurs de PATHWAY peuvent parfois survenir lors de l'exécution de ENDMQM. L'erreur *1018* SERVER FILE (fichier de serveur) (6006) ou *1018* SERVER FILE (7006) peut être signalée lors de l'arrêt du gestionnaire de files d'attente, et principalement lors de l'arrêt des serveurs MQS-QMGRSVR00, MQS-STATUS00 ou MQS-QUEUE00.

Avez-vous effectué des mises à jour de fonctions ?

Si une mise à jour a été effectuée sur un service MQSeries, vérifiez que l'action a abouti et qu'aucun message d'erreur n'a été émis.

- La mise à jour comprenait-elle des instructions spéciales ?
- Est-ce qu'un test a été effectué pour vérifier que la mise à jour a abouti et s'est effectuée correctement ?
- L'incident se produit-il encore si MQSeries est restauré au niveau de service précédent ?
- Si l'installation a réussi, prenez contact avec le centre d'assistance IBM pour savoir s'il n'y a pas d'erreur liée aux patch.
- Si un patch a été appliqué à un autre programme, prenez en compte son éventuelle influence sur la façon dont MQSeries interface avec le programme.
- Niveau de service. Consultez ZMQSSYS.MEMOPTF en lecture seule. Notez l'entrée A) de CSD HISTORY. A l'invite TA CL, entrez VPROC \$vol.ZMQSLIB.MQSRL LIB, où \$vol est votre volume d'installation MQ. Il doit être compatible avec les informations VPROC du MEMOPTF ou être plus récent si un programme efix a été appliqué. Reportez-vous au fichier MEMOEFIX fourni avec le programme efix pour plus d'informations sur le VPROC mis à jour.

Erreurs courantes de programmation

Les erreurs de la liste qui suit illustrent les causes les plus courantes d'incidents rencontrés dans le fonctionnement de programmes MQSeries. Vérifiez que les problèmes de votre système MQSeries ne soient pas causés par une ou plusieurs des erreurs suivantes :

- Supposer que les files d'attente sont partagées alors qu'elles sont en fait en mode exclusif.
- Transmettre des paramètres incorrects dans un appel MQI.
- Transmettre des paramètres insuffisants dans un appel MQI. Cela peut empêcher MQI de définir les codes retour nécessaires au fonctionnement de votre application.
- Oublier de contrôler les codes retour provenant des demandes MQI.
- Transmettre des variables avec des définitions de longueur incorrectes.
- Transmettre des paramètres dans le désordre.
- Omettre d'initialiser correctement les paramètres *MsgId* et *CorrelId*.

Erreurs courantes de programmation

- Omettre d'entrer la commande BEGINTRANSACTION lorsque MQPMO_SYNCPOINT est précisé sur la commande MQPUT.

Problèmes de commandes

Soyez prudent lorsque vous entrez des caractères spéciaux tels que la barre oblique inverse (\) et les guillemets (") dans le texte décrivant les commandes. Pour utiliser l'un de ces deux caractères dans un texte descriptif, faites le précéder du signe \. Il faut donc entrer \\ ou \" si vous souhaitez avoir \ ou " dans le texte.

Vérifications approfondies

Une fois que vous avez établi qu'aucune modification n'était intervenue sur votre système, et que vos programmes d'application ne comportent pas d'erreur, choisissez l'option qui décrit au mieux les symptômes constatés.

- «Avez-vous obtenu une sortie incorrecte ?»
- «Une réponse de la commande PCF vous manque-t-elle ?»
- «L'incident n'affecte-t-il que les files d'attente éloignées ?» à la page 206
- «Votre application ou MQSeries pour Compaq NSK tournent-ils lentement ?» à la page 206

Avez-vous obtenu une sortie incorrecte ?

Dans ce manuel, «sortie incorrecte» fait référence à votre application :

- L'application ne reçoit pas le message prévu.
- L'application reçoit un message contenant une information imprévue ou altérée.
- L'application reçoit un message imprévu, par exemple un message destiné à une autre application.
- La structure que vous utilisez pour afficher les informations est-elle correcte ? Par exemple, la structure MQDLH utilisée pour afficher l'en-tête de la file d'attente de rebut a-t-elle été ajoutée au début du texte de message concernant la file d'attente de rebut ?
- Est-elle reproductible sur l'installation MQSeries d'un autre poste utilisant un système d'exploitation différent ou identique ? Si elle ne peut être reproduite, il peut y avoir une erreur de système d'exploitation qui peut et doit être corrigée à l'aide d'une maintenance logicielle provisoire (IPM) Compaq. Vérifiez auprès des services Compaq.

Dans tous les cas, vérifiez que les files d'attente alias et les gestionnaires de files d'attente alias utilisés par vos applications sont correctement définis et tiennent compte de toutes les modifications effectuées sur votre réseau.

Si des messages d'erreur MQSeries sont générés, et qu'ils sont tous préfixés par «AMQ», consultez le journal des erreurs. Consultez la section «Journaux d'erreurs» à la page 212 pour plus d'informations.

Une réponse de la commande PCF vous manque-t-elle ?

Si vous avez émis une commande mais que vous n'avez pas reçu de réponse, répondez aux questions suivantes:

- Le serveur de commandes fonctionne-t-il ?
Utilisez la commande **dspmqcsv** pour vérifier l'état du serveur de commandes. Si le résultat de cette commande indique que le serveur de commandes n'est pas en cours de fonctionnement, utilisez la commande **strmqcsv** pour le lancer. Si le résultat de la commande indique que la file d'attente

SYSTEM.ADMIN.COMMAND.QUEUE n'est pas activée pour les demandes MQGET, activez-la pour les demandes MQGET.

- Une réponse a-t-elle été transmise à la file d'attente de rebut ?
La structure de l'en-tête de la file d'attente de rebut MQDLH comprend un code anomalie ou code retour décrivant l'incident. Consultez le manuel *MQSeries Application Programming Reference* pour plus d'informations sur cette structure.
Si la file d'attente de rebut contient des messages, vous pouvez utiliser l'application exemple (AMQSBCG) fournie pour consulter les messages à l'aide de l'appel MQGET. L'application exemple passe dans tous les messages d'une file d'attente spécifiée pour un gestionnaire de files déterminé et affiche, pour chaque message, le descripteur et les zones de contexte du message.
- Un message a-t-il été transmis au journal des erreurs ?
Consultez la section «Journaux d'erreurs» à la page 212 pour plus d'informations.
- Les files d'attente sont-elles activées pour des opérations d'insertion et d'extraction ?
- La valeur de l'option *WaitInterval* est-elle suffisante ?
Si le délai imparti à l'appel MQGET a expiré, un code de fin MQCC_FAILED et un code d'anomalie MQRC_NO_MSG_AVAILABLE ont été retournés (consultez le manuel *MQSeries Application Programming Reference* pour plus d'informations sur la zone *WaitInterval* et pour obtenir les codes retour et d'anomalie de MQGET.)
- Si vous utilisez votre propre programme pour insérer des commandes dans la file SYSTEM.ADMIN.COMMAND.QUEUE, faut-il effectuer une opération supplémentaire ?
A moins d'avoir explicitement exclu votre message de demande du point de synchronisation, un point de synchronisation doit être effectué avant toute tentative de réception des messages de réponse.
- Les valeurs des attributs MAXDEPTH et MAXMSGL de vos files d'attente sont-elles suffisantes ?
- Utilisez-vous correctement les zones *CorrelId* et *MsgId* ?
Définissez les valeurs de *MsgId* et *CorrelId* dans votre application de façon à pouvoir recevoir tous les messages de la file d'attente.

Essayez d'arrêter le serveur de commandes et de le redémarrer. Traitez éventuellement tous les messages d'erreurs générés à cette occasion.

Si le système continue de ne pas répondre, l'erreur peut venir d'un gestionnaire de files d'attente. Essayez d'arrêter et de redémarrer le gestionnaire de files d'attente. Si l'incident persiste après le redémarrage, prenez contact avec le centre d'assistance IBM.

L'incident est-il lié à certaines de vos files d'attente ?

Si vous pensez que l'incident ne survient qu'avec un sous-ensemble de files d'attente, procédez comme indiqué ci-dessous :

1. Affichez les informations concernant chaque file d'attente. Vous pouvez utiliser la commande MQSC DISPLAY QUEUE pour effectuer cette opération.
2. Utilisez les données affichées pour effectuer les contrôles ci-dessous :
 - Si CURDEPTH est égal à MAXDEPTH, cela indique que la file d'attente n'est pas traitée. Vérifiez que toutes les applications fonctionnent normalement.

Vérifications approfondies

- Si CURDEPTH n'est pas égal à MAXDEPTH, vérifiez que les attributs de file d'attente ci-dessous sont corrects :
 - Si le déclenchement est utilisé :
 - Le moniteur de déclenchement fonctionne-t-il ?
 - La profondeur de déclenchement n'est-elle pas trop importante ? Autrement dit, les événements de déclenchement sont-ils générés assez souvent ?
 - Le nom du processus est-il correct ?
 - Le processus est-il disponible et opérationnel ?
 - La file d'attente peut-elle être partagée ? Si tel n'est pas le cas, une autre application peut déjà être en train de l'utiliser.
 - Les files d'attente sont-elles correctement activées pour GET et PUT ?
- Si aucun processus d'application n'extrait de message de la file d'attente, déterminez-en la cause. Il se peut que les applications n'aient pas encore été lancées, qu'une connexion ait été coupée ou que l'appel MQOPEN ait échoué pour une raison quelconque.

Vérifiez les attributs IPPROCS et OPPOCS de la file d'attente. Ces attributs indiquent si la file d'attente a été ouverte en entrée ou en sortie. Si la valeur de l'un de ces attributs est zéro, cela indique qu'aucune opération de ce type ne peut avoir lieu. Notez que les valeurs peuvent avoir changé et que la file était ouverte mais qu'elle est maintenant fermée.

Vous devez contrôler l'état au moment où vous prévoyiez l'insertion ou l'extraction d'un message.

Si vous ne pouvez pas résoudre l'incident, prenez contact avec le centre d'assistance IBM.

L'incident n'affecte-t-il que les files d'attente éloignées ?

Si tel est le cas, procédez comme indiqué ci-dessous :

- Vérifiez que les canaux nécessaires ont bien été lancés, qu'ils sont déclenchables et que les éventuels initialisateurs requis sont actifs.
- Vérifiez que les programmes qui doivent insérer des messages dans les files éloignées n'ont pas signalé d'incident.
- Si vous utilisez le déclenchement pour lancer le processus de gestion des files d'attente réparties, vérifiez que le déclenchement est activé au niveau de la file de transmission. Vérifiez également que le moniteur de déclenchement fonctionne.
- Vérifiez dans le journal des erreurs la présence de messages indiquant des erreurs ou des incidents de canal.
- Si nécessaire, lancez le canal manuellement. Consultez le manuel *MQSeries Intercommunication* pour plus d'informations.

Consultez l'«Annexe M. Configuration des communications» à la page 375 et le manuel *MQSeries Intercommunication* pour obtenir des informations sur la définition des canaux.

Votre application ou MQSeries pour Compaq NSK tournent-ils lentement ?

Si votre application tourne lentement, cela peut indiquer qu'elle se trouve dans une boucle ou qu'elle est en attente d'une ressource non disponible.

Vérifications approfondies

Le traçage MQSeries a-t-il été activé à l'aide de **strmqtrc** ou du panneau de gestionnaire de files d'attente ? Ceci entraînerait une perte de performances. Vérifiez si des fichiers TR sont ouverts dans le sous-volume qmL. Désactivez le traçage à l'aide de la commande **endmqtrc** ou du panneau de gestionnaire de files d'attente.

La cause peut également venir d'un problème de performances. Peut-être votre système fonctionne-t-il à la limite de ses capacités. Ce type de problème est probablement pire pendant les périodes de surcharge du système, c'est à dire principalement en milieu de matinée et d'après-midi. (Si votre réseau s'étend sur plus d'un fuseau horaire, les périodes de surcharge peuvent arriver à d'autres moments.)

Examinez la priorité des processus d'applications et de gestionnaires de files d'attente à l'aide de la commande STATUS. La priorité du processus est réduite progressivement par NSK jusqu'à zéro, à l'aide d'une boucle.

Vérifiez que chaque processeur du système NSK est utilisé à son maximum. Si certains processeurs ne sont utilisés que partiellement, il faut équilibrer votre système NSK. Pensez à ajouter des contrôleurs d'exécution à certains processeurs pour répartir la charge MQSeries.

Un problème de performances peut être dû à une limitation de votre matériel.

Remarque : Après une nouvelle Installation de MQSeries ou un chargement à froid du système Compaq NSK, les exécutable MQSeries peuvent mettre plus de temps que prévu à s'exécuter lorsqu'ils sont appelés pour la première fois. C'est parce que le système d'exploitation de Compaq NSK passe par une phase «de réglage», au cours de laquelle il s'assure que toutes les déclarations externes sont résolues.

Si vous pensez que la baisse des performances ne vient pas de la charge du système mais survient parfois lorsque le système est peu chargé, la cause réside probablement dans un programme mal conçu. Cela peut se manifester par un problème qui ne survient que lors de l'accès à certaines files d'attente.

Les symptômes suivants peuvent indiquer que MQSeries tourne lentement :

- votre système met du temps à exécuter les commandes MQSeries ;
- le contrôle fréquent de la longueur de file d'attente indique qu'elle est peu utilisée par une application, alors que vous attendez une grande activité pour cette application.

Si les performances de votre système sont toujours faibles malgré l'exécution des instructions ci-dessus, il est possible que le problème vienne de MQSeries pour Compaq NSK. Si vous pensez que c'est le cas, contactez votre centre d'assistance IBM.

Considérations sur le développement d'applications

Un programme mal conçu peut affecter les performances de différentes façons. Celles-ci ne sont pas toujours évidentes car le programme peut sembler s'exécuter correctement alors qu'il affecte en fait les performances des autres tâches. Les sections qui suivent traitent des divers incidents liés aux programmes effectuant des appels MQSeries.

Considérations sur le développement d'applications

Pour plus d'informations sur le développement d'applications, reportez-vous au manuel *MQSeries - Guide de programmation d'applications*.

Effet de la longueur des messages

Même si MQSeries autorise des messages contenant jusqu'à 100 Mo de données, leur taille peut affecter les performances de l'application qui les traite. Pour obtenir les meilleures performances de votre application, vous ne devez envoyer dans un message que les données essentielles ; par exemple, dans une demande de débit d'un compte bancaire, la seule information qu'il soit nécessaire de transférer du client à l'application du serveur est le numéro de compte et le montant du débit.

Recherche d'un message particulier

L'appel MQGET extrait habituellement le premier message d'une file d'attente. Si vous utilisez les identificateurs de message et de corrélation (*MsgId* et *CorrelId*) dans le descripteur de message pour indiquer un message particulier, le gestionnaire de files le recherche dans la file jusqu'à ce qu'il le trouve. Une telle utilisation de l'appel MQGET affecte les performances de votre application.

Files d'attente contenant des messages de différentes longueurs

Si les messages d'une file ont des longueurs différentes, votre application peut utiliser l'appel MQGET pour déterminer la taille d'un message, avec la zone *BufferLength* définie à zéro afin que, même si l'appel échoue, il renvoie la taille des données du message. L'application peut alors répéter l'appel en indiquant l'identificateur du message repéré lors du premier appel et une mémoire tampon de la taille correcte. Néanmoins, si d'autres applications utilisent la même file, il se peut que les performances de votre application soient dégradées parce que le second appel MQGET perd du temps à rechercher un message qu'une autre application a extrait entretemps.

Si votre application ne peut pas utiliser de messages ayant une taille fixe, une autre solution à ce problème est d'utiliser l'appel MQINQ pour trouver la taille maximale des messages que la file peut accepter, puis de prendre cette valeur pour votre appel MQGET. La taille maximale des messages d'une file d'attente est stockée dans l'attribut *MaxMsgLength*. Toutefois, cette méthode peut induire une forte consommation d'espace, car la valeur de cet attribut de file d'attente peut atteindre 100 Mo, ce qui est le maximum admis par MQSeries pour Compaq NSK.

Fréquence des points de synchronisation

Les programmes qui émettent de nombreux appels MQPUT à l'intérieur d'un point de synchronisation sans les valider peuvent nuire aux performances. Les files d'attente touchées peuvent être saturées de messages actuellement inaccessibles, alors que d'autres tâches attendent de pouvoir extraire ces messages. Cette situation entraîne des répercussions en termes de mémoire, d'utilisation de fichiers d'audit TMF et de processus, liées aux tâches qui tentent d'extraire des messages.

Utilisation de l'appel MQPUT1

N'utilisez l'appel MQPUT1 que si vous n'avez qu'un seul message à insérer dans une file d'attente. Si vous voulez en insérer plusieurs, utilisez l'appel MQOPEN suivi d'une série d'appels MQPUT et d'un seul appel MQCLOSE.

Sortie incorrecte

Le terme de “sortie incorrecte” peut être compris de plusieurs façons. Pour ce qui concerne l’identification des incidents, sa signification est expliquée à la section «Avez-vous obtenu une sortie incorrecte ?» à la page 204.

Deux types de sortie incorrecte sont envisagées dans cette section :

- les messages qui n’apparaissent pas alors que vous les attendez ;
- les messages qui contiennent des informations inappropriées ou altérées.

D’autres incidents pouvant survenir si votre application fait appel aux files réparties sont également commentés ici.

Messages qui n’apparaissent pas dans la file d’attente

Si des messages n’apparaissent pas alors que vous les attendez, vérifiez les points suivants :

- Votre message a-t-il pu être inséré dans la file d’attente ?
- La file a-t-elle été définie correctement ? Par exemple, la valeur de MAXMSGL est-elle suffisante ?
- La file autorise-t-elle les opérations d’insertion ?
- La file d’attente est-elle déjà pleine ? Cela peut signifier qu’une application n’a pas pu insérer le message requis dans la file.
- Pouvez-vous extraire un message de la file d’attente ?
- Avez-vous besoin d’effectuer un point de synchronisation ?

Si des messages sont insérés ou extraits lors d’un point de synchronisation, ils sont indisponibles pour d’autres tâches tant que l’unité de reprise n’est pas validée.

- Votre temps d’attente est-il assez long ?

Vous pouvez définir le temps d’attente comme une option de l’appel MQGET. Vérifiez que la valeur de cet intervalle est suffisante.

- Attendez-vous un message précis identifié par un identificateur de message ou de corrélation (*MsgId* ou *CorrelId*) ?

Vérifiez que vous attendez un message doté des valeurs *MsgId* ou *CorrelId*. Un appel MQGET qui a abouti attribue à ces paramètres les valeurs du message extrait et vous devez peut-être les réinitialiser pour pouvoir extraire un autre message.

Vérifiez aussi que vous pouvez extraire d’autres messages de la file d’attente.

- D’autres applications peuvent-elles extraire des messages de la file d’attente ?
- Le message attendu est-il défini comme persistant ?
Si non, et que MQSeries a été redémarré, le message est perdu.
- Une autre application a-t-elle un accès exclusif à la file d’attente ?

Si vous ne trouvez rien d’anormal dans la file et que MQSeries est en cours d’exécution, faites les vérifications ci-dessous sur le processus qui doit insérer le message dans la file d’attente :

- L’application a-t-elle démarré ?
Si elle doit être déclenchée, vérifiez que les options de déclenchement correctes sont indiquées.
- L’application est-elle arrêtée ?
- Un moniteur de déclenchement fonctionne-t-il ?

sortie incorrecte

- Le processus de déclenchement est-il défini correctement ?
- L'application s'est-elle achevée correctement ?
Recherchez un indicateur de fin anormale dans le journal de la tâche.
- L'application a-t-elle validé ses modifications ou ont-elles été annulées ?

Si plusieurs transactions utilisent la file d'attente, elles peuvent entrer en conflit. Par exemple, supposez qu'une transaction émette un appel MQGET avec une taille de mémoire tampon égale à zéro pour trouver la longueur du message puis émette un appel MQGET spécifique indiquant le *MsgId* de ce message. Pendant ce temps, une autre transaction émet un appel MQGET pour ce message qui aboutit et la première application reçoit un code anomalie MQRC_NO_MSG_AVAILABLE. Les applications devant fonctionner dans un environnement multi-serveur doivent pouvoir gérer ce type de situation.

Il se peut que le message ait été reçu mais que votre application n'a pas pu le traiter. Par exemple, une erreur dans le format attendu du message a-t-elle pu provoquer son rejet par le programme ? Dans ce cas, reportez-vous à la section «Messages contenant des informations inattendues ou altérées».

Messages contenant des informations inattendues ou altérées

Si les informations contenues dans le message ne sont pas celles attendues par l'application, ou si elles ont été altérées, vérifiez les points suivants :

- Votre application, ou l'application qui a inséré le message dans la file, a-t-elle été modifiée ?
Vérifiez que tous les systèmes concernés ont été modifiés en conséquence.
Par exemple, le format des données du message a pu être modifié et, dans ce cas, les deux applications échangeant des messages doivent être recompilées pour tenir compte des modifications. Si une application n'a pas été recompilée, les données paraîtront altérées pour l'autre.
- Une application envoie-t-elle des messages à une file erronée ?
Vérifiez que les messages reçus par votre application ne sont pas en fait destinés à une application utilisant une file différente. Si nécessaire, modifiez les définitions de sécurité pour empêcher des applications non autorisées d'insérer des messages dans des files d'attente erronées.
Si votre application a utilisé une file d'attente alias, vérifiez que l'alias pointe sur la file correcte.
- Les informations de déclenchement ont-elles été indiquées correctement pour cette file ?
Vérifiez que c'est votre application et non une autre qui devait être lancée.

Si ces contrôles ne vous permettent pas de résoudre l'incident, vérifiez la logique du programme d'envoi de messages et du programme de réception.

Sortie incorrecte lors de l'utilisation des files d'attente réparties

Si votre application utilise la gestion de files d'attente réparties, vous devez considérer les points suivants :

- MQSeries est-il correctement installé à la fois sur les systèmes émetteur et récepteur et correctement configuré pour la gestion de files d'attente réparties ?
- Les liaisons sont-elles disponibles entre les deux systèmes ?

Vérifiez que les deux systèmes sont disponibles et connectés à MQSeries.
Vérifiez que la connexion entre les deux systèmes est disponible et que les canaux entre les deux gestionnaires de files d'attente sont actifs.

- Le déclenchement est-il actif sur le système émetteur ?
- Le message attendu est-il une réponse provenant d'un système éloigné ?
Vérifiez que le déclenchement est actif sur le système éloigné.

- La file d'attente est-elle déjà pleine ?

Cela peut signifier qu'une application n'a pas pu insérer le message requis dans la file. Dans ce cas, vérifiez que le message a été inséré dans la file d'attente de rebut.

La structure de l'en-tête de la file d'attente de rebut MQDLH comprend un code anomalie ou de retour décrivant l'incident. Consultez le manuel *MQSeries Application Programming Reference* pour plus d'informations sur cette structure.

- Y a-t-il une incohérence entre le gestionnaire de files d'attente émetteur et le gestionnaire de files d'attente récepteur ?

Par exemple, la longueur du message peut être plus grande que celle que le gestionnaire de files récepteur peut traiter.

- Les définitions de canaux émetteurs et récepteurs sont-elles compatibles ?

Par exemple, une incohérence dans le bouclage des numéros de séquence arrête le composant de gestion de files d'attente réparties. Consultez le manuel *MQSeries Intercommunication* pour plus d'informations sur les files d'attente distribuées.

- Avez-vous lancé un programme d'écoute TCP/IP ?

Si vous utilisez TCP/IP comme protocole de communication pour les communications de MQSeries avec Compaq, un processus de programme d'écoute TCP/IP doit être en cours de fonctionnement. Pour plus d'informations, reportez-vous à la section «Détermination et contrôle des programmes d'écoute TCP/IP» à la page 32.

- Le programme d'écoute TCP/IP est-il en mode écoute sur le bon port TCP/IP ?

Le programme d'écoute TCP/IP doit écouter sur un port défini dans une entrée PortEcouteTCP de la section TCPConfig du fichier QMINI de votre gestionnaire de files d'attente. Pour plus d'informations, reportez-vous à la section «Ports TCP/IP écoutés par le gestionnaire de files d'attente» à la page 55.

- Le nom de processus du TCP/IP est-il correct ?

Si vous utilisez le protocole de communications TCP/IP par défaut, votre système Compaq utilise-t-il le nom de processus par défaut (\$ztc0) pour le processus TCP/IP ? Si ce n'est pas le cas, vous devez modifier certaines classes de serveurs de votre chemin d'accès MQSeries pour permettre aux canaux MQSeries d'utiliser le bon nom de processus. Pour plus d'informations, reportez-vous à la section «Reconfiguration de la classe de serveur MQS-TCPLISnn pour un port et un processus TCP/IP personnalisés» à la page 56.

- Le programme d'écoute SNA MQSeries est-il configuré ?

Si :

- vous vous servez des canaux MQSeries à l'aide d'un SNA comme protocole de communication et si
- le type de canal Compaq est en attente d'initialisation à partir d'un système MQSeries distant (par exemple un RECEPTEUR) et si
- le système distant peine à démarrer le canal,

sortie incorrecte

le programme d'écoute SNA MQSeries PATHWAY ne fonctionne peut-être pas avec votre gestionnaire de files d'attente. Vérifiez que le programme d'écoute SNA MQSeries a été configuré pour le gestionnaire de files d'attente SNAX/APC ou ICE PATHWAY. Pour plus d'informations, consultez la section «Processus de canaux répondeurs LU 6.2» à la page 376.

- La conversion de données est-elle en cause ? Si les formats des données entre les applications émettrice et réceptrice diffèrent, la conversion est obligatoire. Une conversion automatique se produit au moment de l'appel MQGET si le format est reconnu comme un des formats livrés d'origine.

Si le format de données n'est pas reconnu comme format incorporé, un exit de conversion est activé pour vous permettre d'effectuer cette opération avec vos propres sous-programmes. Vérifiez que votre sous-programme est chargé correctement.

Reportez-vous au manuel *MQSeries - Guide de programmation d'applications* pour plus d'informations concernant la conversion de données.

Journaux d'erreurs

MQSeries pour Compaq NSK utilise des journaux d'erreurs pour consigner les messages concernant le fonctionnement de MQSeries et celui des gestionnaires de files d'attente qui ont été démarrés, ainsi que les données d'erreur provenant des canaux en cours d'utilisation.

L'emplacement des journaux d'erreurs dépend de la connaissance ou non du nom du gestionnaire de files d'attente et de l'association ou non de l'erreur avec un client.

- Si le nom du gestionnaire de files d'attente est connu et que le gestionnaire est disponible :

```
<QMVOL>.<SOUSVOL>L.MQERRLG1
```

- Si le gestionnaire de files d'attente n'est pas disponible :

```
<MQSVOL>.ZMQSSYS.MQERRLG1
```

- First Failure Symptom Trap (FFST) dans

```
<QMVOL>.<SOUSVOL>.FDnnnnn
```

- Consultez la section «Consultation des fichiers FFST» à la page 217.

Fichiers journaux

Le sous-répertoire peut contenir trois fichiers de journalisation des erreurs appelés :

- MQERRLG1
- MQERRLG2
- MQERRLG3

Après la création d'un gestionnaire de files d'attente, trois fichiers de journalisation des erreurs sont créés selon les besoins du gestionnaire. Ces fichiers sont nommés

MQERRLG1, MQERRLG2 et MQERRLG3. Ils sont placés dans le sous-volume de chaque gestionnaire de files d'attente que vous créez.

Les messages d'erreur générés sont enregistrés dans MQERRLG1. Quand MQERRLG1 est classé, il est copié dans MQERRLG2. Avant la copie, MQERRLG2 est copié dans MQERRLG3. Le contenu précédent de MQERRLG3, s'il existe, est écrasé.

Les derniers messages d'erreur sont toujours placés dans MQERRLG1, les autres fichiers étant utilisés pour conserver un historique des messages d'erreurs.

Tous les messages relatifs aux canaux sont également enregistrés dans les fichiers d'erreurs du gestionnaire de files concerné, à moins que le nom de leur gestionnaire soit inconnu ou qu'il soit indisponible. Quand le gestionnaire est indisponible ou que son nom ne peut pas être déterminé, les messages relatifs aux canaux sont enregistrés dans le journal des erreurs (ZMQSSYS.MQERRLG1).

Pour examiner le contenu d'un fichier journal d'erreurs, utilisez soit la commande de copie fup, soit votre éditeur Compaq NSK habituel en mode lecture seule. (Si vous ouvrez le journal des erreurs en mode mise à jour, les messages d'erreur seront perdus.)

Erreurs précoces

Dans certains cas, une erreur se produit alors que les journaux ci-dessus n'ont pas encore été créés. MQSeries tente d'enregistrer de telles erreurs dans un journal. L'emplacement du journal dépend du nombre de gestionnaires de fichiers créés.

Si, par exemple à cause d'une détérioration de fichier de configuration, aucune information d'emplacement ne peut être déterminée, les erreurs sont consignées dans le fichier MQERRLG1, créé au moment de l'installation dans le sous-volume ZMQSSYS.

Pour plus d'informations sur les fichiers de configuration, reportez-vous au «Chapitre 13. Fichiers de configuration» à la page 189.

Messages opérateur

Sous MQSeries pour Compaq NSK, Les messages opérateur identifient les erreurs normales, en général provoquées directement par les utilisateurs indiquant, par exemple, des paramètres incorrects dans une commande. Ces messages sont écrits dans la fenêtre associée, si elle existe, ainsi que dans un fichier dans le répertoire du gestionnaire de files d'attente.

Les erreurs pouvant être associées à un gestionnaire de files d'attente donné sont consignées dans le fichier journal MQERRLG1 du sous-volume du gestionnaire de files d'attente. Celles qui ne peuvent être associées à un gestionnaire de files d'attente défini sont consignées dans le fichier MQERRLG1 qui se trouve dans le sous-volume ZMQSSYS.

Déchiffrement des numéros EC dans le fichier MQERRLG

Le numéro EC (ontrôleur d'éécution) dans les messages consignés dans le fichier MQERRLG1 est assigné par l'ECBOSS pour son suivi du processus EC. Il n'y a aucun rapport direct entre le numéro assigné par l'ECBOSS dans MQERRLG1 et le numéro utilisé dans le nom du serveur de PATHWAY à MQS-ECxx. Par exemple, MQS-EC00 n'aura pas forcément le numéro 0. Le numéro EC attribué est lié au

Journaux d'erreurs

nom de processus EC dans le message d'initialisation terminée. Utilisez le nom de processus EC pour déterminer le serveur de PATHWAY à MQS-ECxx et pour analyser les problèmes.

Exemple de journal des erreurs

L'exemple qui suit est une partie d'un journal d'erreur MQSeries pour Compaq NSK :

```
...
02/01/01 11:41:56 AMQ8003: MQSeries queue manager started.
EXPLANATION: MQSeries queue manager janet started.
ACTION: None.
-----
02/01/01 11:56:52 AMQ9002: Channel program started.
EXPLANATION: Channel program 'JANET' started.
ACTION: None.
-----
02/01/01 11:57:26 AMQ9208: Error on receive from host 'camelot
(9.20.12.34)'.
EXPLANATION: An error occurred receiving data from 'camelot
(9.20.12.34)' over TCP/IP. This may be due to a communications failure.
ACTION: Record the TCP/IP return code 232 (X'E8') and tell the
systems administrator.
-----
02/01/01 11:57:27 AMQ9999: Channel program ended abnormally.
EXPLANATION: Channel program 'JANET' ended abnormally.
ACTION: Look at previous error messages for channel program
'JANET' in the error files to determine the cause of the failure.
-----
02/01/01 14:28:57 AMQ8004: MQSeries queue manager ended.
EXPLANATION: MQSeries queue manager janet ended.
ACTION: None.
-----
02/02/01 15:02:49 AMQ9002: Channel program started.
EXPLANATION: Channel program 'JANET' started.
ACTION: None.
-----
02/02/01 15:02:51 AMQ9001: Channel program ended normally.
EXPLANATION: Channel program 'JANET' ended normally.
ACTION: None.
-----
02/02/01 15:09:27 AMQ7030: Request to quiesce the queue manager
accepted. The queue manager will stop when there is no further
work for it to perform.
EXPLANATION: You have requested that the queue manager end when
there is no more work for it. In the meantime, it will refuse
new applications that attempt to start, although it allows those
already running to complete their work.
ACTION: None.
-----
02/02/01 15:09:32 AMQ8004: MQSeries queue manager ended.
EXPLANATION: MQSeries queue manager janet ended.
ACTION: None.
...
```

Événement EMS

Un événement EMS est généré pour chaque entrée d'erreur faite dans le fichier MQERRLG1. Pour plus d'informations sur les événements EMS, voir «Événements EMS (Event Management Service)» à la page 175.

Files d'attente de rebut

Les messages qui ne peuvent pas être transmis pour une raison ou une autre sont insérés dans la file d'attente de rebut. Vous pouvez vérifier si la file contient des messages en émettant la commande MQSC DISPLAY QUEUE. Si la file contient des messages, vous pouvez utiliser l'application exemple (MQSBCG0E) fournie pour consulter les messages à l'aide de l'appel MQGET. L'application exemple passe dans tous les messages d'une file d'attente spécifiée pour un gestionnaire de files déterminé et affiche, pour chaque message, le descripteur et les zones de contexte du message.

Vous devez décider du sort des messages trouvés dans la file d'attente de rebut en fonction des raisons de leur insertion dans cette file.

Des incidents peuvent survenir lorsque vous ne disposez pas d'une file d'attente de rebut pour chaque gestionnaire utilisé. Lors de la création de cette file d'attente de rebut, modifiez l'attribut DEADQ du gestionnaire de files d'attente à l'aide de la commande **runmqsc**.

Fichiers de configuration et identification des incidents

En général, les erreurs dans les fichiers de configuration empêchent de trouver les gestionnaires de files d'attente et génèrent des erreurs du type «gestionnaire de files d'attente indisponible».

Vous pouvez procéder à plusieurs vérifications sur les fichiers de configuration :

- vérifiez que les fichiers de configuration existent,
- vérifiez qu'ils disposent des droits appropriés,
- vérifiez que le fichier de configuration MQSeries référence les bons répertoires de journaux et de gestionnaires de files d'attente.

Traçage de MQSeries

MQSeries pour Compaq NSK utilise les commandes suivantes dans le cadre du traçage :

- **strmqtrc** – reportez-vous à la section «strmqtrc (lancement de la trace MQSeries)» à la page 309
- **dspmqtrc** – reportez-vous à la section «dspmqtrc (affichage de la sortie de trace formatée MQSeries)» à la page 281
- **endmqtrc** – reportez-vous à la section «endmqtrc (arrêt de la trace MQSeries)» à la page 288

La fonction de trace utilise un fichier pour chaque entité tracée, les informations de traçage étant enregistrées dans le fichier approprié.

Les options de traçage sont spécifiées dans le fichier QMINI.

traçage de MQSeries

Remarque : Avec MQSeries pour Compaq NSK, le traçage peut également être contrôlé à l'aide du menu Gestionnaire de file d'attente de la fonction Gestion des files d'attente de messages (MQM).

Fichiers de trace

Les noms de fichiers de trace sont construits selon le modèle suivant :

TRccpppp

où *ccpppp* désigne l'identificateur de processus (PID) produisant la trace. Le PID se compose de :

cc, le numéro de processeur,
pppp, le numéro de processus.

Si la fonction de trace trouve un fichier de trace d'un identificateur de processus identique qui n'a pas été effacé, elle remplace le dernier caractère du numéro de processus par une lettre, offrant à 26 processus du même PID la possibilité d'imprimer leur sortie. Par exemple, le premier fichier de trace pour le PID 00, 0315 serait TR000315. Pour un deuxième processus démarré à la fin du processus 00, 0315 avec le même PID, le fichier de trace serait TR00031A.

Remarque : Pour cette raison, les fichiers de trace doivent être supprimés du système aussitôt après leur examen.

Exemple de données de trace

L'exemple suivant est un extrait de trace :

...	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
...							

Figure 44. Exemple de trace

Remarques :

1. Dans cet exemple, les données sont tronquées. Dans une trace réelle, les noms de fonction complets et codes retour sont présents.
2. Les codes retour sont donnés sous forme de valeurs et non de littéraux.

First Failure Support Technology™ (FFST)

Ces erreurs, généralement graves et irrémédiables, indiquent soit un incident de configuration du système, soit une erreur interne MQSeries. Dans la plupart des cas, le gestionnaire de files d'attente reste opérationnel, bien qu'une brève interruption de la gestion de quelques ou toutes les applications puisse survenir. Les FFST sont référencés dans le fichier ZMQSSYS.MQSYSLOG.

Consultation des fichiers FFST

Les fichiers sont intitulés `FDnnnnn`, où :
`nnnnn` est le numéro d'identification du processus signalant l'erreur

Lorsqu'un processus crée un rapport FFST report, il génère aussi un événement EMS.

La figure 45 montre un rapport FFST typique.

```

+-----+
| 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 fonction 0000000020 |
+-----+

```

Figure 45. Exemple de rapport FFST

Certains problèmes peuvent toutefois être résolus. Si le fichier FFST indique un «manque de ressources» ou un «espace insuffisant sur le disque», il est fort probable que la limite du système ait été dépassée.

Pour résoudre ce problème, augmentez la limite appropriée et redémarrez le gestionnaire de files d'attente.

Chapitre 15. Evolutivité et performances

Ce chapitre présente des techniques pour optimiser les performances et l'évolutivité de vos programmes MQSeries. Il comprend les sections suivantes :

- «Introduction»
- «Messages persistants» à la page 220
- «Messages non-persistants» à la page 220
- «Serveurs de file d'attente et fichiers de file d'attente» à la page 222
- «Affectation de processeurs» à la page 227
- «Bind des applications avec FASTPATH» à la page 228

Introduction

L'optimisation des performances et de l'évolutivité vise à minimiser l'utilisation de deux ressources majeures : les processeurs et les disques. Les applications qui utiliseront au mieux ces deux ressources seront plus efficaces et évolutives (par exemple, elles pourront supporter une plus grande charge de travail en étant configurées pour utiliser de façon optimale le matériel et le logiciel du système).

Ce chapitre présente des techniques qui amélioreront à la fois les performances des applications et celles de MQSeries. Les sections suivantes résument les principes généraux de l'optimisation des performances des applications.

Conception d'applications optimisées pour les performances et l'évolutivité

Vous devez prendre en compte la façon dont sont utilisés MQSeries et les autres sous-systèmes dès le début de la conception d'une application. Ce sont les fonctionnalités et besoins requis qui vont déterminer les fonctions MQSeries nécessaires ou intéressantes pour une application. Certaines fonctions de MQSeries, comme la persistance des messages, assurent une grande intégrité et une garantie de livraison du message, mais utilisent davantage de ressources processeur et disque. Si ces garanties ne sont pas nécessaires pour une application donnée, le fait de configurer MQSeries pour s'en passer peut entraîner d'importants gains de performance. Ce choix est plus facile à faire au début de la phase de conception, lors de l'évaluation des besoins.

Réduction ou élimination de l'utilisation des ressources partagées

Pour gérer la croissance du trafic de messages, les matériels et logiciels doivent être utilisés au maximum de leurs capacités. D'une manière générale, toute ressource partagée devient rapidement un goulet d'étranglement lorsque la charge de travail s'accroît. Que ce soit à cause de l'augmentation du degré de partage (par exemple, davantage d'utilisateurs qui se partagent le même processeur) ou par l'augmentation de la consommation des ressources (si par exemple chaque utilisateur travaille davantage).

Du point de vue des performances et de l'évolutivité, les processeurs et sous-systèmes de stockage sont les ressources les plus partagées, et donc celles qui requièrent le plus d'attention.

Introduction aux notions d'évolutivité et de performances

Vous pouvez répartir les processus MQSeries entre autant de processeurs que vous le voulez. Si votre système possède plusieurs processeurs, le fait de répartir les processus MQSeries entre les processeurs disponibles accroîtra les performances par rapport à la répartition par défaut. De même, pour optimiser l'utilisation des sous-systèmes de disques, il convient de placer les fichiers de files d'attente différentes sur des volumes différents, si possible desservis par des processus disques Compaq NSK différents.

Optimisation des performances par approximation itérative

L'optimisation des performances se fait en alternant les étapes de mesure et de réglage. Chaque cycle mesure-réglage-mesure ne doit modifier qu'une seule variable, afin de pouvoir attribuer les effets observés aux modifications effectuées et comparer l'importance relative des variables. En général, le fait de modifier plusieurs réglages à la fois va à l'encontre de la productivité, certains réglages pouvant améliorer les performances et d'autres les réduire, sans que l'on puisse savoir lesquels. Les performances d'une application sont généralement limitées par un goulet d'étranglement. Le fait de modifier le système pour éliminer ce goulet ne fait généralement que déplacer le problème vers une autre ressource. La complication de ces interactions souligne l'importance d'une approche méthodique d'optimisation, et le respect de la succession des cycles mesure-réglage-mesure.

Supposons par exemple qu'un programme insère un grand nombre de petits messages persistants dans une file d'attente. Il peut être instructif de tester l'effet de rendre ces messages non-persistants, ou de regrouper leurs données dans un plus petit nombre de messages plus longs. Ces deux modifications améliorent (en général) les performances, mais à défaut d'effectuer un cycle mesure-réglage-mesure séparé pour chacune d'elles, il sera impossible de définir la plus efficace.

Messages persistants

Dans le cadre de MQSeries, les messages persistants offrent la plus grande assurance de livraison et de reprise. Ces messages sont toujours enregistrés sur disque, et persistent donc entre l'arrêt et le redémarrage d'un gestionnaire de files d'attente. TM/MP Nonstop surveille les fichiers de file d'attente, et toute écriture ou lecture de message persistant augmente l'activité sur le disque dur, car les fichiers de file d'attente et les fichiers d'audit de TM/MP sont concernés. Le journal d'audit TM/MP sert à conserver l'intégrité transactionnelle des messages persistants, y compris en cas de panne système ou matérielle. Cette journalisation des audits TM/MP doit être prise en considération lors de l'évaluation des performances de la conception d'une application MQSeries.

La persistance est une propriété du message, pas de la file d'attente qui le contient. Les files d'attente peuvent contenir des messages persistants ou non, et l'administrateur peut définir la persistance d'un nouveau message si l'application qui l'insère ne le fait pas. (Consultez la définition de l'attribut de file d'attente DEFPSIST, décrit dans le manuel *MQSeries Application Programming Reference*).

Messages non-persistants

Contrairement aux messages persistants, les messages non-persistants ne sont pas sauvegardés sur disque, et ne résistent pas à l'arrêt d'un gestionnaire de files d'attente. Selon les options définies pour les files d'attente sur le serveur de file d'attente, les messages non persistants peuvent être surveillés par points de contrôle depuis le serveur de file d'attente de sauvegarde. L'utilisation de messages non persistants offre de grands avantages en termes de performances.

Messages non-persistants

Les messages persistants bénéficient d'une livraison et de capacités de reprise garanties, mais toute opération de lecture ou d'écriture les concernant augmente l'activité disque sur les fichiers de file d'attente et les fichiers d'audit TM/MP. Cette activité diminue les performances des applications qui lisent ou écrivent des messages persistants, mais aussi des canaux MQSeries qui déplacent les messages persistants vers d'autres gestionnaires de file d'attente.

L'utilisation des messages non persistants chaque fois que possible améliore considérablement les performances, car ils ne sont pas sauvegardés sur disque par les serveurs de file d'attente Nonstop, ils sont simplement mis en cache mémoire.

La persistance est une propriété du message, pas de la file d'attente qui le contient. Les files d'attente peuvent contenir des messages persistants ou non, et l'administrateur peut définir la persistance d'un nouveau message si l'application qui l'insère ne le fait pas (consultez l'attribut de file d'attente DEFPSIST).

Messages non persistants et canaux

Les canaux de transmission de messages utilisent la journalisation synchronisée du côté émetteur comme du côté récepteur, afin de garantir que les messages envoyés par le réseau sont livrés une fois et une seule. Cette journalisation synchronisée vient s'ajouter à tous les suivis et journalisations réalisés par TM/MP (au bénéfice du gestionnaire de files d'attente) chaque fois qu'un canal lit ou écrit un message sur une file d'attente. Les canaux de transmission de messages peuvent être configurés pour ne pas effectuer cette journalisation lors de l'émission et de la réception de messages non persistants. Il faut pour cela définir l'attribut NPMSPEED à la valeur FAST. L'attribut NPMSPEED contrôle le comportement des canaux émetteurs ou récepteurs qui traitent les messages non persistants. Quand la valeur de l'attribut NPMSPEED d'un canal est NORMAL, les messages non persistants font partie de son lot de traitement (tel qu'il est défini par les attributs BATCHINT et BATCHSZ). Par conséquent, ils requièrent la même synchronisation que les messages persistants. De plus, le canal lit et écrit les messages via des points de synchronisation, ce qui augmente légèrement l'activité du fichier d'audit de TM/MP au début et à la fin d'une transaction.

Lorsque la valeur de NPMSPEED est FAST, les messages non persistants ne font pas partie du lot de traitement du canal, et sont lus et écrits sur les files d'attente sans faire intervenir de points de synchronisation. L'utilisation de la valeur FAST supprime donc deux causes d'activité disque pour un canal : la journalisation du mécanisme de synchronisation de lot de canal, et celle de l'audit TM/MP, qui se feraient sinon à chaque lecture ou écriture de message, avec points de synchronisation.

L'option NPMSPEED(FAST) améliore considérablement les performances, au détriment des capacités de reprise des messages non persistants ; NPMSPEED(FAST) peut entraîner la perte des messages non persistants si le canal ou le réseau tombe en panne.

La valeur par défaut de NPMSPEED est FAST.

Serveurs de file d'attente et fichiers de file d'attente

Un serveur de file d'attente est un processus MQSeries qui gère la lecture, l'écriture et le stockage des messages. C'est donc un important composant de MQSeries, qui doit être considéré attentivement. La configuration des serveurs de file d'attente peut avoir un très grand impact sur les performances d'un système MQSeries très actif.

Les serveurs de file d'attente s'occupent de l'enregistrement physique des messages contenus dans les files d'attente. Les messages persistants et non persistants sont gérés de façon différente, aussi nous les étudierons séparément.

Un serveur de file d'attente peut gérer une ou plusieurs files d'attente. Lors de sa création, une file d'attente est gérée par le serveur de file d'attente par défaut. Par conséquent, toutes les files d'attente sont gérées par ce même serveur, jusqu'à ce qu'elles soient assignées à un autre serveur.

Chaque file d'attente gérée par un serveur de file d'attente utilise les fichiers suivants :

- Un fichier de file d'attente
- Un fichier de dépassement de capacité de file d'attente
- Un fichier d'horodatage (touch)

En outre, il peut y avoir un fichier de dépassement de capacité pour chaque message de la file d'attente qui dépasse le seuil défini pour cette file d'attente. Les fichiers de dépassement de capacité de message sont présentés plus loin.

Stockage des messages persistants

Les messages persistants sont toujours sauvegardés sur disque. La façon dont ils sont enregistrés dépend essentiellement de leur taille :

Taille du message	Enregistré
< 3000 octets (environ)	Entièrement dans le fichier de file d'attente.
> 3000 octets (environ) <= seuil de dépassement de capacité de message	Les 3000 premiers octets (environ) dans le fichier de file d'attente, et le reste dans le fichier de dépassement de capacité de file d'attente
> seuil de dépassement de capacité de message	Les 3000 premiers octets (environ) dans le fichier de file d'attente, le reste dans le fichier de dépassement de capacité de message

Stockage des messages non persistants

Les messages non persistants sont conservés dans les tampons mémoire du processus de serveur de file d'attente. C'est pourquoi il n'y a aucune entrée-sortie lorsque le serveur de file d'attente lit ou écrit des messages non persistants.

Répartition des serveurs de file d'attente entre les processeurs

Les serveurs de file d'attente sont des classes de serveurs PATHWAY. Ils sont donc définis dans le PATHWAY de chaque serveur de file d'attente. Lors de la création d'un gestionnaire de files d'attente, un serveur de file d'attente par défaut est défini dans le PATHWAY, avec son processus primaire tournant sur le processeur 0.

Serveurs de file d'attente et fichiers de file d'attente

Comme ce processeur gère un grand nombre des processus systèmes de priorité élevée, il vaut mieux ne pas l'utiliser pour exécuter un serveur de file d'attente. Au contraire, essayez de repérer plusieurs processeurs relativement peu actifs, et créez un serveur de file d'attente sur chacun d'eux. N'oubliez pas qu'un nouveau serveur de file d'attente ne gère aucune file d'attente qui ne lui aura pas été assignée explicitement.

Assignation d'une file d'attente à un serveur de file d'attente (commande `altmqfls`)

Lorsque vous créez une file d'attente (par la commande `runmqsc`), MQSeries l'assigne au serveur de file d'attente par défaut. Si vous ne créez pas d'autres serveurs de file d'attente pour leur assigner des files d'attente, toutes les files d'attente seront gérées par ce serveur par défaut.

Dans un système MQSeries très actif, cette disposition n'est ni efficace ni évolutive. C'est donc une raison majeure de créer de nouveaux serveurs de file d'attente et de leur assigner des files d'attente, afin de répartir la charge globale entre les processeurs disponibles.

La commande `altmqfls` permet d'assigner une file d'attente à un serveur de file d'attente donné. Par exemple :

```
altmqfls --qmgr QMGR --type QLOCAL --server $QS01 TEST.QUEUE
```

Pour plus d'informations sur l'utilisation de la commande `altmqfls`, consultez la section «`altmqfls` (modification des attributs d'un fichier de file d'attente)» à la page 254.

File d'attente de transmission de cluster : `SYSTEM.CLUSTER.TRANSMIT.QUEUE`

MQSeries utilise une seule file d'attente de transmission de cluster pour toutes les opérations de clustering. Si votre gestionnaire de files d'attente fait partie d'un serveur très actif, cette file d'attente doit être assignée à un serveur de file d'attente dédié. Non seulement les performances des opérations de clustering seront améliorées, mais l'impact sur les autres applications sera réduit.

Déplacement des fichiers de file d'attente (commande `altmqfls`)

Par défaut, les trois fichiers de file d'attente principaux (file d'attente, dépassement de capacité de file d'attente et fichier d'horodatage) sont dans le sous-volume M du gestionnaire de files d'attente. Les fichiers de dépassement de capacité de message sont dans le sous-volume du serveur de file d'attente.

Vous pouvez envisager de les déplacer vers d'autres volumes pour deux raisons :

- Mieux répartir la charge d'entrées-sorties disque
- Eviter les limitations du système d'exploitation concernant le nombre de fichiers ouverts par volume disque

Serveurs de file d'attente et fichiers de file d'attente

La commande **altnmqfls** vous permettra de déplacer des fichiers de file d'attente vers un autre volume disque :

```
altnmqfls --qmgr QMGR --type QLOCAL --volume $DATA01 TEST.QUEUE
```

Pour pouvoir déplacer une file d'attente avec la commande **altnmqfls**, cette file ne doit pas être ouverte.

Vous ne pouvez pas déplacer de fichiers de dépassement de capacité de message. Cependant, vous pouvez créer n'importe où de nouveaux fichiers de dépassement de capacité de message, avec l'option `--msgofsubvol` de la commande **altnmqfls** :

Par exemple, pour créer un nouveau fichier de dépassement de capacité de message à un autre endroit :

```
altnmqfls --qmgr QMGR --type QLOCAL --msgofsubvol $DATA01.TESTMOF TEST.QUEUE
```

Pour plus d'informations sur l'utilisation de la commande **altnmqfls**, consultez la section «`altnmqfls` (modification des attributs d'un fichier de file d'attente)» à la page 254.

Partitionnement de fichiers de file d'attente

Le partitionnement de fichiers consiste à les répartir sur plusieurs volumes disques. Le fichier est alors composé de plusieurs partitions, chacune sur un volume disque différent.

Les avantages de partitionner un fichier de file d'attente sont les suivant :

- La charge d'entrées-sorties d'une file d'attente est répartie entre plusieurs volumes, et donc entre plusieurs chemins d'entrées-sorties
- Un fichier de file d'attente logique peut dépasser la taille maximale d'un fichier physique sur un volume disque.

Le partitionnement d'un fichier de file d'attente se fait par des commandes ACL standard (l'utilitaire FUP). MQSeries propose un script ACL (PARTIT) qui montre comment utiliser FUP pour partitionner un fichier de file d'attente existant.

Dès qu'une file d'attente a été partitionnée, le serveur de file d'attente qui la gère s'efforce de répartir équitablement les nouveaux messages entre toutes les partitions disponibles. Cette méthode accroît encore les avantages du partitionnement.

Remarque : la commande **altnmqfls** ne peut pas déplacer un fichier de file d'attente partitionné.

Fichiers de dépassement de capacité

Les fichiers de dépassement de capacité de message sont créés pour tout message qui dépasse le seuil défini. Ces fichiers ne sont pas structurés et ne sont pas surveillés par TM/MP. Pour les messages de très grande taille, il est plus efficace d'enregistrer la plupart du message dans le fichier de dépassement de capacité de message, puisqu'il n'est pas surveillé par TM/MP. Pour les messages de petite taille, il est plus efficace d'enregistrer la totalité du message dans les fichiers de file

Serveurs de file d'attente et fichiers de file d'attente

d'attente audités par TM/MP (fichiers de file d'attente et de dépassement de capacité de file d'attente). La limite de taille a été déterminée empiriquement à environ 200 Ko.

C'est pourquoi le seuil de dépassement de capacité de message est réglé par défaut à 200 Ko lors de la création d'une file d'attente. Ce seuil peut être modifié avec l'option `--oflowsize` de la commande `altmqfls`.

La commande `altmqfls` sert à définir la valeur du seuil de dépassement de capacité de message :

```
altmqfls --qmgr QMGR --type QLOCAL --oflowsize 400000 TEST.QUEUE
```

Mise en mémoire tampon des messages pendant leur consultation

Le serveur de file d'attente peut conserver en mémoire les n premiers octets de tous les messages persistants d'une file d'attente. Cette possibilité peut avoir un très grand impact sur les performances d'une application qui consulte les messages persistants d'une file d'attente. Si la taille de ces messages est inférieure au seuil de consultation, il est possible de les explorer sans qu'il n'y ait d'accès disque.

Vous pouvez également identifier le contenu d'un message persistant en conservant, par exemple, ses 100 premiers octets. Il sera ainsi possible de parcourir les 100 premiers octets de données de chaque message pour trouver, ceux qui sont importants, sans qu'il y ait d'accès disque. Une fois qu'un message aura été localisé, il sera alors possible de lire l'ensemble de ses données et de le retirer de la file d'attente, si nécessaire.

L'option `--browse` de la commande `altmqfls` définit le seuil de consultation d'une file d'attente. Par défaut, aucune donnée de message persistant n'est conservée en mémoire (le seuil de consultation est à zéro). La valeur maximale est de 25000.

```
altmqfls --qmgr QMGR --type QLOCAL --browse 100 TEST.QUEUE
```

Autres options de serveur de file d'attente

Les autres options utilisent le paramètre `--qsoptions` avec les options L, S et C. Elles définissent si la file d'attente est mise en cache depuis le disque au démarrage du gestionnaire de files d'attente, ou si les messages non persistants sont gérés par des points de contrôle depuis le serveur de file d'attente de secours. Les options L, S et C s'utilisent séparément ou en association, pour ajuster les réglages de fiabilité et de récupération de la file d'attente.

Remarque : *Toutes* les `--qsoptions` SLC sont modifiées à chaque utilisation de la commande. Par exemple, `--qsoptions S` désactivera les options L et C. L'option `--qsoptions` ne peut être utilisée qu'une fois par ligne de commande.

Chargement au démarrage

Cette fonction est contrôlée par le paramètre `--qsoptions S` de la commande `altmqfls`. Elle oblige le serveur de file d'attente à lire les fichiers de file d'attente et à construire ses structures internes de données de message dès son démarrage (en

Serveurs de file d'attente et fichiers de file d'attente

général, au lancement du gestionnaire de files d'attente), au lieu de le faire lorsque la file d'attente est ouverte pour la première fois.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions S TEST.QUEUE
```

Cette option réduit l'activité disque et processeur lors de la première ouverture de la file d'attente, mais les augmente au démarrage du gestionnaire de files d'attente.

Verrouillage en cache

Cette fonction est contrôlée par le paramètre `--qsoptions L` de la commande **altmqfls**. Le serveur de file d'attente verrouille en mémoire les structures de données et chaînes associées à une file d'attente. Par conséquent, les structures de données de la file d'attente ne sont plus recopiées depuis la mémoire vers le disque pour faire de la place aux autres files d'attente. Par défaut, les structures de données d'une file d'attente sont recopiées sur disque en cas de besoin.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions L TEST.QUEUE
```

Cette option accélère l'accès aux structures de données en mémoire d'une file d'attente, au détriment possible des autres files d'attente.

Point de contrôle NPM

Cette fonction est contrôlée par le paramètre `--qsoptions C` de la commande **altmqfls**. Le serveur de file d'attente surveillera les messages non persistants avec des points de contrôle, via son processus de secours non-stop. Cette option est active par défaut.

Cette option améliore la fiabilité des messages non persistants, mais augmente le trafic IPC ainsi que l'utilisation de la mémoire et du processus, pour le processus primaire du serveur de file d'attente comme pour celui de secours.

```
altmqfls --qmgr QMGR --type QLOCAL --qsoptions C TEST.QUEUE
```

Compteur Measure

Cette fonction est contrôlée par le paramètre `--meascount` de la commande **altmqfls**. Le serveur de file d'attente gèrera un compteur contenant la `CURDEPTH` (longueur ou profondeur) de la file d'attente. Ce compteur sert lors de l'évaluation des performances générales du système. Cette valeur vous permet de mettre en correspondance l'activité d'écriture et de lecture de messages sur une file d'attente (fonctions `MQGET` et `MQPUT`), avec des paramètres du système comme les processeurs et les entrées-sorties disque.

```
altmqfls --qmgr QMGR --type QLOCAL --meascount TESTCOUNT1 TEST.QUEUE
```

Cette option permet d'évaluer avec précision les performances et l'évolutivité d'un système qui utilise MQSeries, et donc d'améliorer l'efficacité de vos efforts d'optimisation.

Affectation de processeurs

Lorsque vous démarrez un gestionnaire de files d'attente, MQSeries pour Compaq NSK crée plusieurs processus. Certains s'occupent des opérations principales de messagerie, alors que d'autres effectuent des fonctions qui supportent indirectement les opérations principales. Les processus qui s'occupent des opérations principales sont d'autant plus actifs que les applications font appel aux fonctions de messagerie MQI (MQGET et MQPUT).

Les processus MQSeries et Compaq NSK qui sont impliqués dans les opérations principales sont les suivants :

- Les processus LQMA (Local Queue Manager Agents)
- Les serveurs de file d'attente
- Les serveurs de statut
- Les processus disque NSK

Les processus MQSeries et Compaq NSK qui sont impliqués dans les opérations de file d'attente réparties sont les suivants :

- Les processus MCA
- Les serveurs de statut par défaut
- Les processus NSK, TCPIP ou SNA

Les processus MQSeries qui sont impliqués dans les fonctions de support ou d'administration sont les suivants :

- Les serveurs de gestionnaire de file d'attente
- Les serveurs de référentiels
- Les contrôleurs d'exécution (EC et ECBoss)
- Les initialiseurs de canaux
- Les moniteurs de déclenchement
- Les programmes d'écoute

Un système MQSeries surchargé montre un général une charge processeur très élevée pour les deux premières catégories de processus décrites ci-dessus (processus de messagerie et de files d'attente réparties).

Pour répartir la charge processeur de MQSeries, il convient généralement de répartir entre le plus de processeurs possibles les LQMA, serveurs de file d'attente et MCA des gestionnaires de files d'attente.

Par défaut, un nouveau gestionnaire de files d'attente lance tous ses processus sur le processeur CPU 0 (dans le cas des paires de processus Nonstop, le processus primaire tourne sur le CPU 0 et le processus de secours sur le CPU 1). Cette configuration ne convient pas à une situation de production, et vous devez être prêt à reconfigurer le PATHWAY d'un gestionnaire de files d'attente pour répartir au mieux les processus très actifs entre autant de processeurs que possible. Le processeur CPU 0 traite de nombreux processus à priorité élevée du système d'exploitation, il convient donc mal à l'exécution des processus MQSeries.

La répartition des contrôleurs d'exécution entre les processeurs est définie par le PATHWAY du gestionnaire de files d'attente ; elle contrôle l'exécution des LQMA et des MCA. Les serveurs de file d'attente sont également définis dans le PATHWAY du gestionnaire de files d'attente.

Outre les processus MQSeries, les processus disque du système d'exploitation NSK sont un composant très important des opérations de messagerie (notamment lorsque les serveurs de file d'attente gèrent des messages persistants). Le fait de

Affectation de processeurs

répartir de manière excessive les files d'attente sur un réseau augmente l'utilisation des ressources processeur par les processus TCPIP ou SNA correspondants. Pour évaluer les performances globales d'une installation MQSeries, vous devez prendre en compte le nombre et la répartition par processeur de ces processus.

Vous trouverez des informations sur l'assignation des processus aux processeurs, la configuration des processus disque ou TCPIP dans la documentation système Compaq NSK. Vous trouverez des informations sur la configuration des processus SNA dans la documentation SNAX ou ICE. Pour la configuration des processus TCPIP ou SNA, consultez l'«Annexe M. Configuration des communications» à la page 375.

Bind des applications avec FASTPATH

Le bind FASTPATH est une fonction du MQI conçue pour améliorer l'efficacité des applications MQSeries. Cette méthode sert à réduire le temps système utilisé par toutes les commandes MQI dans les applications MQSeries. Les applications qui ont subi un bind FASTPATH sont dites sécurisées, à cause du partage de la mémoire et des logiciels utilisés par le gestionnaire de files d'attente avec l'application. Des erreurs dans ces applications peuvent par conséquent endommager les structures de données de MQSeries et compromettre l'intégrité du gestionnaire de files d'attente.

Généralités

Quand une application lance une commande MQCONN, MQSeries crée (ou réutilise) un processus spécial nommé LQMA (Local Queue Manager Agent). Ce LQMA dessert tous les appels suivants de l'application à des fonctions MQI en utilisant ce descripteur de connexion. Dans MQSeries pour Compaq NSK, le LQMA peut tourner sur le même processeur que l'application, ou sur un autre.

Comme le LQMA est un processus séparé, l'application n'accède pas directement à la mémoire ni aux fichiers utilisés par MQSeries. Ce qui évite qu'une application mal programmée ne vienne endommager le LQMA. De cette façon, les structures de données et les logiciels essentiels au fonctionnement de MQSeries sont isolés des données et applications de l'utilisateur. Mais cette sécurité ne va pas sans contrepartie. La commande MQCONN ne peut s'exécuter tant qu'un nouveau processus LQMA n'a pas été créé (ou que l'ancien n'est pas réutilisable). Mais le plus sérieux problème vient du fait que l'application doit envoyer des informations au LQMA à chaque fois qu'elle lance une commande MQI.

Ces informations MQI sont passées au LQMA par un mécanisme IPC (Interprocess Communications ou communications entre processus). Les requêtes IPC peuvent être internes au processeur (si le LQMA tourne sur le même processeur que l'application), ou, plus pénalisant, se faire entre processeurs différents.

Réduction de la charge système MQI

Les développeurs utiliseront le bind FASTPATH afin de soulager le système de la charge de travail associée aux IPC et engendrée par les LQMA suite à chaque commande MQI (la création éventuelle d'un processus LQMA est également supprimée). Quand l'option de bind FASTPATH est activée pour une application, elle n'utilise pas de processus LQMA indépendant. Au contraire, les composants MQSeries, qui sont normalement contenus dans le LQMA, sont chargés dans le processus utilisateur (par exemple, le processus de l'application qui s'est connectée).

Bind des applications avec FASTPATH

Les autres commandes MQI lancées par l'application n'entraîneront aucune activité IPC avec le LQMA, puisque les données et logiciels MQSeries (normalement contenues dans le processus LQMA) sont en local, dans le processus de l'application. Veuillez noter qu'il peut néanmoins se produire une certaine activité IPC, lorsque le gestionnaire de files d'attente doit communiquer avec d'autres processus MQSeries comme des serveurs de file d'attente. Le bind FASTPATH ne supprime pas toutes les activités IPC, mais supprime une cause importante de ces activités.

Activation du bind FASTPATH

Pour activer le bind FASTPATH, il faut se connecter au gestionnaire de files d'attente avec la commande MQCONN et l'option MQCNO_FASTPATH_BINDING. Si elle est indiquée, la valeur du PARAM MQCONNECTTYPE influence le comportement de MQCONN. Si le PARAM MQCONNECTTYPE est présent, sa valeur doit être FASTPATH pour que MQCONN puisse définir une connexion FASTPATH. Dès qu'une connexion de bind FASTPATH est établie, toutes les autres commandes MQI se comportent de la même façon que pour une connexion de bind STANDARD, aux exceptions décrites dans la section suivante.

Restrictions lors de l'utilisation d'un bind FASTPATH

En mode de bind STANDARD, les applications sont isolées des données internes à MQSeries. Cette séparation n'existe plus lorsqu'une application sécurisée établit une connexion de bind FASTPATH. Des erreurs dans cette application peuvent par conséquent endommager les structures de données de MQSeries et compromettre l'intégrité du gestionnaire de files d'attente. C'est un point très important, qu'il faut prendre en considération pour décider s'il convient d'utiliser le bind FASTPATH pour une application particulière.

Les considérations suivantes s'appliquent également aux applications sécurisées :

- Les applications sécurisées **doivent** se déconnecter explicitement du gestionnaire de files d'attente (par exemple avec la commande MQDISC).
- Les applications sécurisées doivent être arrêtées avant de lancer la commande **endmqm**. Les applications sécurisées doivent s'exécuter sous un ID utilisateur administrateur (celui qui correspond au principal mqm).
- Les applications sécurisées ne peuvent être exécutées que sur des processeurs contenant un serveur de référentiels MQSeries actif.

Bind des applications avec FASTPATH

Chapitre 16. Intégrité et disponibilité des données

Ce chapitre décrit les concepts d'intégrité et de disponibilité des données, ainsi que leur impact sur la gestion et la configuration de MQSeries pour Compaq NSK V5.1. Il décrit également les différents niveaux d'intégrité et de disponibilité des données que vous êtes en droit d'attendre de MQSeries, et les répercussions que pourront avoir vos choix de configuration sur ces niveaux. Il contient les sections suivantes :

- «Intégrité des données»
- «Disponibilité» à la page 232
- «Données persistantes et non persistantes» à la page 233
- «Cohérence de la base de données» à la page 236
- «Processus critiques» à la page 240
- «Clusters» à la page 245
- «Considérations sur la configuration pour la disponibilité» à la page 246
- «Configuration favorable à l'intégrité des données» à la page 246

Vous devez avoir lu et assimilé les informations fournies dans le «Chapitre 2. Architecture MQSeries pour Compaq NSK V5.1» à la page 21 pour comprendre et utiliser correctement les informations de ce chapitre.

Intégrité des données

L'étude des différents aspects du stockage et de la gestion des données pour les applications de traitement en ligne des transactions vous permettra de mieux appréhender le concept d'intégrité des données :

- Lorsque des données sont écrites ou lues depuis une base de données, un message d'erreur doit obligatoirement être émis au cours du transfert en cas d'altération, de doublon ou de perte de données.
- Lorsque plusieurs processus doivent accéder simultanément aux mêmes données, ils doivent tous avoir la même perception des données, et celles-ci doivent être protégées contre les risques d'altération, de doublon ou de perte.
- Lorsque plusieurs modifications cohérentes doivent être réalisées sur des données présentes dans plusieurs bases de données, alors toutes doivent être appliquées, ou aucune.

Pour MQSeries, les besoins d'intégrité des données en matière de stockage indiqués ci-dessus s'appliquent également aux opérations de messagerie (par exemple, MQPUT et MQGET) sur les files d'attente. Notez que la cohérence des diverses modifications apportées aux bases de données doit être préservée aussi bien au sein des bases de données d'applications et files d'attente MQSeries elles-mêmes, qu'*entre-elles*.

MQSeries pour Compaq NSK V5.1 est conçu pour maintenir l'intégrité des données associées aux opérations de données persistantes pour tout point de défaillance unique (matériel ou logiciel). En fait, l'intégrité des données peut être préservée dans différents cas où plusieurs points de défaillance se produiraient. Ceci n'implique pas pour autant que les messages non persistants ne sont pas fiables ; l'architecture de serveur de file d'attente a la capacité de rendre les messages non persistants aussi fiables que les messages persistants, sauf en cas de

Intégrité des données

panne système grave. (Pour plus d'informations sur l'architecture de serveur de file d'attente, consultez le «Chapitre 2. Architecture MQSeries pour Compaq NSK V5.1» à la page 21.)

Sous MQSeries pour Compaq NSK V5.1, l'intégrité des données est assurée par les caractéristiques fondamentales du logiciel et du matériel système Compaq NSK, ainsi que par le logiciel MQSeries lui-même.

Le niveau d'intégrité des données peut être influencé de plusieurs façons par les choix de configuration de MQSeries:

- Choix du caractère persistant ou non des messages par l'application
- Choix de la technique de mise en mémoire des messages persistants
- Choix des options de réglage des messages non persistants
- Choix des options de configuration du serveur de file d'attente
- Configuration des fichiers de file d'attente pour le matériel
- Utilisation et configuration de NonStop TM/MP
- Utilisation et configuration de la fonction de base de données éloignée (NonStop RDF).

Chacun de ces facteurs est décrit plus loin dans ce chapitre.

Disponibilité

La disponibilité consiste à évaluer la durée pendant laquelle une application ou un service est opérationnel et utilisable, par rapport au temps écoulé. Ainsi, atteindre une *disponibilité continue* est le but ultime de tous ces types de systèmes. Bien sûr, de telles évaluations n'ont aucune signification si aucune période de temps ne lui est associée— il est facile d'obtenir 100% de disponibilité sur une courte période.

En pratique, et sur une période de fonctionnement significative, un système devra faire face à différentes situations susceptibles de remettre en cause sa disponibilité :

- Défaillances logicielles et matérielles
- Défaillances de l'application elle-même
- La nécessité d'apporter des modifications au système dans un souci de maintenance préventive
- Un nombre de transactions ou un trafic dépassant les contraintes de conception ou les ressources disponibles

Comme pour l'intégrité des données, sous MQSeries pour Compaq NSK V5.1, la disponibilité est assurée par les caractéristiques fondamentales du logiciel et du matériel Compaq NSK, ainsi que par le logiciel MQSeries lui-même.

Il est important d'avoir conscience que (au moins sous sa forme actuelle) MQSeries pour Compaq NSK n'a pas pour objectif de fournir un niveau équivalent de disponibilité continue que celui apporté par un système de fichiers Compaq NSK. Certains éléments du gestionnaire de files d'attente proposent ce niveau de disponibilité, mais ce n'est pas le cas du gestionnaire dans son ensemble.

MQSeries pour Compaq NSK, V5.1 est conçu pour fournir un niveau de disponibilité qui permette que sur tout point de défaillance unique (matériel ou logiciel) :

- Les connexions de gestionnaires de files d'attente susceptibles de subir des interruptions ou arrêts de fonctionnement se limitent aux gestionnaires qui subissent directement la panne (par exemple, en cas de panne d'unité centrale, seules les connexions fournies par les processus LQMA exécutés sur ce processeur doivent normalement échouer).
- Le gestionnaire de files d'attente reste disponible pour de nouvelles tentatives de connexion sans qu'aucune intervention manuelle de l'administrateur système ne soit nécessaire.
- L'accès à un objet de gestionnaire de files d'attente (par exemple une file d'attente locale) ne doit être interdit qu'aux connexions directement affectées par la panne.

En plus de ces fonctionnalités MQSeries spécifiques à Compaq NSK, plusieurs fonctionnalités communes à toutes les plateformes MQSeries Version 5.1 sont disponibles pour étendre la disponibilité de MQSeries pour Compaq NSK. Le niveau de disponibilité peut être influencé de plusieurs façons par les choix effectués pour la configuration de MQSeries :

- Choix de la conception de l'application
- Choix des options de configuration PATHWAY
- Utilisation et configuration des fonctions MQSeries standard (dont les clusters)
- Choix du matériel prenant en charge MQSeries
- Choix des options de configuration des messages non persistants
- Utilisation et configuration de NonStop TM/MP
- Utilisation et configuration de NonStop RDF

Données persistantes et non persistantes

Dans le cadre de l'utilisation de MQSeries, le terme *persistant* implique plusieurs qualités des données :

- La modification de données persistantes résiste au redémarrage d'un gestionnaire de files d'attente
- Les données persistantes sont mises en mémoire sur un support non volatile
- Les données persistantes satisfont aux besoins les plus exigeants en matière d'intégrité des données pour cet environnement d'exploitation particulier
- Le niveau d'intégrité supérieur des opérations utilisant des données persistantes se traduit par une rapidité supérieure et une utilisation plus efficace des ressources.

Le plupart des opérations d'administration reposent sur des données persistantes, puisque les bases de données de configuration de MQSeries doivent disposer du plus haut niveau d'intégrité des données pour réduire le risque de dégradation sérieuse du niveau de disponibilité de MQSeries. La vitesse ou l'utilisation des ressources de la majeure partie des opérations d'administration n'est pas primordiale car elles ne sont réalisées que rarement.

Voici des exemples d'opérations d'administration persistantes :

- Modification des attributs d'une file d'attente
- Modification des attributs d'un canal
- Création d'une nouvelle file d'attente, procédure, liste de noms ou canal.

Données persistantes et non persistantes

Voici des exemples d'opérations d'administration non persistantes :

- Démarrage ou arrêt d'un canal
- Demande d'informations sur les attributs d'un objet gestionnaire de files d'attente

MQSeries pour Compaq NSK propose plusieurs méthodes de stockage des messages, selon le choix de persistance ou non persistance effectué par l'application lors de la mise en file d'attente d'un message et des préférences de l'administrateur système en matière de configuration de la longueur de la file d'attente.

Messages persistants

Les messages persistants sont toujours stockés sur disque. En tant qu'administrateur système, vous avez le choix entre deux techniques de stockage des messages persistants pour chaque file d'attente. Votre choix devra s'effectuer en fonction de la taille des messages puisque l'avantage de disposer de différentes techniques est d'améliorer les performances des opérations de messagerie sur des messages très volumineux.

Tous les messages persistants sont enregistrés une seule fois dans le fichier de file d'attente contrôlé. Cet enregistrement contient les en-têtes et informations de contrôle essentielles du message, ainsi que la plus grande quantité possible de données du message pouvant rentrer dans les 4096 octets de l'enregistrement.

Le mécanisme le plus rapide et le plus efficace pour le stockage des messages persistants de petite à moyenne taille sous TM/MP est de stocker les données en surnombre dans le fichier de dépassement de file en plusieurs enregistrements, à l'aide de la même technique de base que celle qui était utilisée dans MQSeries pour Compaq NSK Version 2.2.0.1. Toutes les données de ces messages sont consignées dans le TM/MP et peuvent ainsi être récupérées si nécessaire par la suite à partir des fichiers d'audit.

Pour les messages volumineux (de plus de 200 Ko de données), la méthode la plus efficace est d'utiliser un fichier de dépassement de message dédié non contrôlé. Les données écrites dans le fichier de dépassement de message ne sont donc pas écrites dans le fichier d'audit TM/MP. Ceci permet d'utiliser moins d'E/S disque et processeur et de transférer les données en blocs de taille plus importante.

Certains aspects relatifs à l'intégrité des données diffèrent légèrement entre ces deux mécanismes. Ils sont décrits dans le tableau 6 :

Tableau 6. Comparaison entre le dépassement de file d'attente et le dépassement de message

Aspect relatif à l'intégrité des données	Méthode de dépassement de file d'attente	Méthode de dépassement de message
Quantité de données contrôlée par TM/MP	Toutes les données des messages sont contrôlées	Seules les données tenant dans l'enregistrement du fichier de file d'attente sont contrôlées
La capacité de récupération des données depuis le fichier d'audit se traduit par une perte de volume totale en cas de pannes multiples.	Le message complet peut être récupéré depuis le fichier d'audit	Seuls l'en-tête de message et la première partie des données peuvent être récupérés à partir du fichier d'audit.

Données persistantes et non persistantes

Tableau 6. Comparaison entre le dépassement de file d'attente et le dépassement de message (suite)

Aspect relatif à l'intégrité des données	Méthode de dépassement de file d'attente	Méthode de dépassement de message
Taille maximale des messages	Limitée par le nombre de verrouillages d'enregistrements par volume et par transaction imposé par ENSCRIBE. Pour un fichier non partitionné, la taille est d'environ 20 Mo dans une configuration ENSCRIBE par défaut. En pratique, la limite peut être moins importante en raison des limites de mémoire physique.	Limité uniquement par l'espace disponible, ou 100 Mo, qui est la taille de message maximale permise pour MQSeries.
Compatibilité avec la fonction de base de données éloignée (RDF)	Compatibilité totale (indiquer le paramètre d'environnement MQRDF).	Non compatible avec RDF. Les données de messages n'étant pas contrôlées, vous ne pouvez pas utiliser RDF pour diffuser les opérations de message à l'aide de fichiers au système de sauvegarde.
Résistance aux incidents matériels de disques	Résiste totalement à tout point de défaillance unique en cas d'utilisation de disques miroirs. Puisque toutes les données sont contrôlées, vous pouvez procéder à la reconstitution des fichiers en cas de panne des disques de la paire en miroir, ou du disque unique en l'absence de miroir.	Résiste totalement à tout point de panne unique si des disques en miroir sont utilisés pour stocker les fichiers de dépassement de message. Les données de messages sont irrécupérables en cas de panne totale de volume.

Pour les messages persistants, les deux techniques de stockage atteignent un très haut niveau d'intégrité des données, avec certaines limites pour les fichiers de dépassement de message car toutes les données de message ne sont pas contrôlées. Normalement, ces limites ne concernent que les messages très volumineux, et peuvent être évitées en ayant recours à la segmentation (pour diviser un message d'application en messages physiques plus petits).

Messages non persistants

Les messages non persistants (NPM) sont normalement stockés en mémoire. Si un serveur de file d'attente gère une grande quantité de données non persistantes et atteint un seuil au-delà duquel il risque d'être à cours de mémoire virtuelle, il imposera le stockage des NPM sur disque. Occasionnellement, le serveur de file d'attente copie les NPM dans des fichiers disque dans le cadre de certaines opérations d'administration lorsque des modifications de configuration sont effectuées en ligne. Lorsque les applications accèdent aux NPM, ceux-ci résident toujours en mémoire.

Le serveur de file d'attente gère le stockage de tous les messages des files d'attente dont il est responsable. Pour chaque file d'attente, il fournit une option de

Données persistantes et non persistantes

configuration permettant de contrôler le niveau d'intégrité des données appliqué au NPM. Le paramètre de configuration vérifie que le serveur de file d'attente exécute des points de contrôle sur les NPM lors de la sauvegarde. Ceci permet aux NPM de résister aussi bien que les messages persistants à une panne du processeur contenant le processus de serveur de file d'attente principal.

L'inconvénient de la définition des points de contrôle NPM est que le serveur de file d'attente utilise le processeur de façon plus intensive (fichier principal et de secours), transfère plus de données vers la sauvegarde lors du contrôle, et utilise plus de mémoire lors de la sauvegarde puisqu'il doit de nouveau stocker le message.

Si l'exécution de points de contrôle NPM n'est pas activée pour une file d'attente et que le processus de serveur de file d'attente principal se termine anormalement (suite à un incident de processus ou du processeur), tout NPM présent sur la file avant l'incident est éliminé puisqu'il était uniquement stocké dans le processus principal.

L'option de messages non persistants tolérants aux pannes n'est disponible sur aucune autre plate-forme MQSeries. Dans un souci de cohérence avec le reste de la gamme, la définition de points de contrôle NPM est activée par défaut (utilisez `altmqfls` pour désactiver cette fonction. Prenez le temps de déterminer si les messages non persistants répondent à vos besoins en termes d'intégrité des données. Si c'est le cas, vous serez en mesure d'optimiser considérablement vos ressources et d'améliorer les performances.

Cohérence de la base de données

Le cohérence des bases de données doit être préservée à la fois en interne par MQSeries et en externe, lorsque l'option de point de synchronisation est utilisée pour des opérations de messagerie. Ces deux éléments sont essentiels à l'intégrité et à la disponibilité des données.

Cohérence des bases de données internes

Tous les fichiers de base de données critiques d'un gestionnaire de files d'attente sont contrôlés par TM/MP. Les processus de gestionnaire de files d'attente doivent par conséquent utiliser des transactions pour les modifier. Toutes les modifications sont consignées dans le fichier d'audit TM/MP et (si utilisé) copiées à l'aide de RDF sur un ou plusieurs systèmes de reprise après incident. Ceci permet donc d'assurer le plus haut niveau d'intégrité des bases de données internes.

L'utilisation de TM/MP pour protéger les bases de données internes permet de veiller à ce que l'intégrité des bases de données critiques ne soit pas compromise en cas de panne système ou logicielle. Les chances de réussite du démarrage (automatique ou manuel) des processus ou services utilisant ces bases de données sont donc plus importantes, et permettent de bénéficier d'une meilleure disponibilité.

Cohérence des bases de données externes

La coordination des modifications apportées aux bases de données externes et leur cohérence avec les bases de données MQSeries est possible grâce au fait que le TM/MP et les applications utilisent MQSeries. Les opérations de messagerie MQSeries peuvent être réalisées sous le contrôle d'un point de synchronisation, ce qui nécessite qu'une transaction soit active et en cours dans l'application (qu'elle soit héritée d'un autre processus ou démarrée à l'aide du service TM/MP

BEGINTRANSACTION). Cette transaction est héritée par le gestionnaire de files d'attente, et toute mise à jour de base de données critique ou de stockage nécessaire est effectuée à l'aide de cette transaction. Le TM/MP s'assure que les entrées de fichiers d'audit appropriées soient enregistrées pour toutes les E/S de disque de la transaction, quel que soit le processus à l'origine de la mise à jour.

Le TM/MP gère également une vue cohérente des mises à jour réalisées mais non encore validées en posant des verrouillage sur les enregistrements concernés dans tous les fichiers de base de données. Par exemple, une ligne d'un tableau SQL insérée sous le contrôle TM/MP ne peut être mise à jour ou supprimée jusqu'à validation de la transaction.

Lorsque le gestionnaire de files d'attente répond au processus d'application (et revient de l'interface MQI), l'application peut continuer à fonctionner sous cette transaction—en utilisant MQSeries pour ajouter ou retirer des messages de files d'attente, ou procéder seule à des mises à jour de bases de données à l'aide de ENSCRIBE ou NonStop SQL. Lorsque l'application est prête à modifier les files d'attente MQSeries et autres éléments permanents de bases de données, le service ENDTRANSACTION de TM/MP est appelé, ce qui a pour effet de valider simultanément les modifications apportées à toutes les bases de données du système.

Si l'application détermine qu'une erreur s'est produite pendant le traitement de la transaction et que des mises à jour de bases de données ont eu lieu, elle doit alors appeler ABORTTRANSACTION afin que le TM/MP annule simultanément les modifications apportées à toutes les bases de données du système. Il se peut donc qu'un message soit repositionné dans une file après en avoir été retiré par une opération MQGET de point de synchronisation, ou qu'un élément précédemment ajouté à un tableau de base de données SQL en soit retiré. Si les applications ont été conçues soigneusement, ces erreurs peuvent être gérées de façon à maintenir la cohérence et à améliorer l'intégrité et la disponibilité des données.

Lorsque MQSeries exécute une opération MQPUT ou MQGET de point de synchronisation, il adapte la longueur de la file d'attente au moment de l'opération, en supposant que la transaction pourra par la suite être validée. La longueur de file d'attente comprend donc le nombre de messages validés et non validés de la file. Si la transaction est abandonnée (volontairement ou suite à un incident), MQSeries adapte la longueur de la file d'attente de façon à ce que la valeur proposée soit toujours correcte. Ce point constitue une amélioration par rapport aux versions V2.2 antérieures de MQSeries pour Compaq NSK, dans lesquelles le gestionnaire de files d'attente ne pouvait pas anticiper l'état futur des transactions et maintenir une longueur adéquate de file d'attente quelle que soit la situation.

La section «OpenTMF» décrit ce mécanisme.

OpenTMF

OpenTMF est le nom informel de la nouvelle interface interne du produit NonStop TM/MP. Avec l'aide de Compaq, MQSeries a pu l'utiliser pour déterminer l'état final d'une transaction utilisée par le gestionnaire de files d'attente pour exécuter des opérations de messagerie de point d'accès.

Cette nouvelle caractéristique de TM/MP est à la base de la nouvelle fonction de traitement des transactions hétérogènes de NonStop TM/MP, introduite avec D42. OpenTMF permet à MQSeries de s'enregistrer en tant que participant dans toute

Cohérence de la base de données

transaction héritée des applications. Lorsque la transaction est terminée, TM/MP envoie une notification à MQSeries lui indiquant si elle a réussi ou a été annulée. Cette nouvelle notification envoyée par TM/MP permet à MQSeries d'exécuter les opérations ci-dessous :

- Maintenir une longueur de file d'attente correcte dans tous les cas de figure
- Faire en sorte que les autres informations d'état internes sur les files d'attente locales soient toujours correctes
- Contrôler la disponibilité des messages non persistants impliqués dans des opérations de point de synchronisation
- Améliorer l'efficacité et le délai de réponse des opérations MQGET attendues

Du point de vue de l'administrateur système, l'utilisation de OpenTMF est visible d'une seule façon : les processus MSeries sont visibles dans une liste de gestionnaires de ressource produits à l'aide de la commande STATUS RESOURCEMANAGER de l'utilitaire TMFCOM. Dans cette liste, tous les processus MQSeries qui utilisent OpenTMF apparaissent comme des gestionnaires de ressources VOLATILES, nommés automatiquement par TM/MP.

Aucune opération d'administration particulière n'est nécessaire pour cette nouvelle utilisation de TM/MP—MQSeries l'utilise et la gère automatiquement. Vous devez vous assurer que le paramètre de configuration RMOPENPERCPU (nombre maximum de gestionnaires de ressources VOLATILES et RECUPERABLES par processeur) de TM/MP est défini sur une valeur supérieure au nombre maximum de serveurs de file d'attente et de statut exécutables sur un processeur du système. Notez que vous devez autoriser les sauvegardes périodiques puisque ces serveurs sont des paires de processus NonStop. La valeur par défaut est de 128 : elle convient à la plupart des installations. Le manuel *Compaq NSK NonStop TM/MP Configuration and Planning Guide* décrit le rôle des gestionnaires de ressources et du traitement de transactions hétérogènes.

NonStop Tuxedo

MQSeries peut coordonner les opérations de messagerie des applications OSS à l'aide de NonStop Tuxedo, car ce produit fonctionne sous NonStop TM/MP et utilise les mêmes fonctions de traitement des transactions hétérogènes que MQSeries.

Le manuel *System Application Development Guide* de Compaq NSK NonStop Tuxedo fournit des informations sur l'utilisation de l'environnement de transaction NonStop Tuxedo et sur ses interactions avec TM/MP.

Transactions d'applications imbriquées

Grâce à MQSeries pour Compaq NSK, les applications bénéficient d'un environnement de transaction spécifique à cette plateforme. En général, sous Compaq NSK, un processus peut gérer plusieurs transactions simultanément. Une mise à jour de base de données surveillance est toujours réalisée sous le contrôle de la transaction en cours, et l'application peut basculer sur n'importe quelle transaction active avant de la valider. Une application peut ainsi traiter simultanément plusieurs opérations de messagerie de point de synchronisation MQSeries.

Fichiers de bases de données critiques de MQSeries

Les fichiers de base de données contrôlés critiques pour MQSeries sont décrits dans le tableau 7 à la page 239.

Cohérence de la base de données

Tableau 7. Fichiers de base de données contrôlée critiques

Nom descriptif	Emplacement/Nom	Utilisation
Catalogue des objets	Sous-volume de données/OBJCAT et ABJCAT	Contient les attributs de chaque objet de file d'attente, processus, et liste de noms, ainsi que les attributs du gestionnaire de files d'attente lui-même.
Définition de canal non client	Sous-volume de données/CHDEFS	Contient les attributs de chaque canal non client (types de canaux EMETTEUR, RECEPTEUR, SERVEUR, DEMANDEUR et CLUSTER)
Définition de canal client	Sous-volume de données/CCHDEFS	Contient les attributs de chaque canal client (types de canaux SVRCONN)
base de données OAM	Sous-volume de données/OAMDB	Contient les autorisations (droits d'accès) de chaque objet et principal OAM autorisé à accéder au gestionnaire de files d'attente.
Base de données des principaux	Sous-volume de données/PRIDB et PRIDBA	Contient les noms de tous les Principaux OAM et ID utilisateur Guardian autorisés auxquels correspond le principal.
Fichier de contrôle EC	Sous-volume de données/RUNTIME	Contient les informations utilisées par l'EC Boss et les EC pour coordonner les opérations de démarrage et de reprise.
Définition de listes de noms	Sous-volume de données/Lxxxxxxx	Contient les éléments de chaque objet de liste de noms défini.
Fichier de files d'attente	Sous-volume de messages/Qxxxxxxx	Contient les informations d'en-tête ainsi qu'une partie des données de chaque message persistant d'une file d'attente locale.
Fichier de dépassement de file d'attente	Sous-volume de messages/Oxxxxxxx	Contient les données de tout message persistant de taille moyenne à importante d'une file d'attente locale.
Fichier d'horodatage d'objet	Sous-volumes de données et de messages/Txxxxxxx	Utilisé pour détecter les modifications d'administration apportées aux attributs des objets.
Fichier de synchronisation de canal	Sous-volume de synchronisation/Sxxxxxxx	Contient les informations de synchronisation de canal d'une instance de canal.

Processus critiques

Le tableau 8 décrit les processus critiques d'un gestionnaire de files d'attente, et montre de quelle façon MQSeries est protégé en cas de panne de l'un d'entre-eux, qu'elle soit d'origine logicielle ou système. Il décrit également comment se déroule la récupération. Toute référence à un incident grave indique que plusieurs incidents système se sont produits simultanément, ou désigne un échec total du système.

Tableau 8. Méthodes de protection utilisées pour les processus critiques

Processus	Méthodes de protection utilisées	Gestion de la reprise
Serveur de file d'attente	Paire de processus NonStop. Conserve un état actualisé des files d'attentes locales et messages en permanence sauf en cas d'incident grave.	Se réinitialise depuis les bases de données contrôlées après un incident grave. Aucune autre procédure de reprise n'est nécessaire. Le NPM sera perdu en cas d'incident grave.
Serveur de statut	Paire de processus NonStop. Conserve un état actualisé des objets de files d'attentes non locales et des canaux en permanence sauf en cas d'incident grave.	Se réinitialise depuis les bases de données contrôlées après un incident grave. Aucune autre procédure de reprise n'est nécessaire.
Agent LQMA (Local Queue Manager Agent)	<p>La connexion est indiquée comme étant interrompue pour l'application (2009).</p> <p>L'opération de récupération de place du gestionnaire de référentiels apure les zones d'enregistrement du cache de cluster, laissées par les processus ayant échoué.</p> <p>L'EC, l'EC Boss, le serveur de file d'attente et le serveur de statut identifient immédiatement l'échec grâce à une connexion IPC NSK et corrigent/adaptent les données de statut en conséquence.</p> <p>TM/MP abandonne toute transaction qui était active et en cours d'utilisation par le processus au moment de l'incident.</p>	Aucune. Un agent LQMA met en service une connexion. La connexion est interrompue en cas d'incident, et une nouvelle connexion est ouverte par l'application.

Tableau 8. Méthodes de protection utilisées pour les processus critiques (suite)

Processus	Méthodes de protection utilisées	Gestion de la reprise
Agent MCA	<p>Le Serveur de statut identifie immédiatement l'incident via la connexion IPC NSK et met à jour le statut du canal de façon appropriée.</p> <p>Le serveur de statut lance la reprise des canaux sortants ayant échoué.</p> <p>Il est possible d'utiliser la fonction MCA pour redémarrer le canal ayant échoué.</p> <p>Les données de synchronisation de canal sont contrôlées par TM/MP et utilisées par MQSeries pour préserver l'intégrité du canal. Dans de rares cas d'attente de validation ne pouvant être résolus, des fonctions d'administration standard permettent de procéder à la reprise.</p> <p>Toutes les autres méthodes de protection du processus LQMA.</p>	<p>Aucune. Un agent MCA met un canal en service. Le canal s'arrête en cas d'incident, et des fonctions automatisées de MQSeries permettent de redémarrer le canal.</p> <p>MQSeries pour Compaq NSK V5.1 propose également de nouvelles fonctionnalités telles que les signaux de présence de canaux et la mise en cluster, qui permettent d'accroître la disponibilité des canaux.</p>
Initialisateur de canal	<p>Il est possible de configurer plusieurs initialisateurs de façon à bénéficier d'une disponibilité supérieure en répartissant les files d'attente d'initialisation de canal sur plusieurs initialisateurs.</p> <p>L'Initialisateur de canal est normalement exécuté en tant que classe de serveur PATHWAY, et configuré pour effectuer plusieurs AUTOSTART après incident (10 par défaut) dans un délai de 10 minutes.</p> <p>En cas d'incident d'un processeur, il est possible d'utiliser les options de configuration PATHWAY standard pour en configurer d'autres.</p>	<p>Aucune. Le serveur de statut par défaut maintient un statut de canal précis dans tous les cas de figure, et procède à la reprise des canaux. L'initialisateur de canal utilise les fonctions de déclenchement de MQSeries pour effectuer l'initialisation.</p>

Processus critiques

Tableau 8. Méthodes de protection utilisées pour les processus critiques (suite)

Processus	Méthodes de protection utilisées	Gestion de la reprise
Serveur de commandes	<p>Le serveur de commandes exécute les commandes d'administration de façon synchronisée afin de maintenir la cohérence des données.</p> <p>Fonctions PATHWAY standard identiques à celles de l'initialisateur de canal.</p>	Aucune. Le redémarrage s'accompagne d'une nouvelle connexion au gestionnaire de files d'attente.
EC Boss	Fonctions de protection PATHWAY standard décrites ci-dessus.	L'EC Boss coordonne la reprise avec les EC à l'aide du fichier RUNTIME contrôlé. Les EC s'enregistrent de nouveau auprès de l'EC Boss et continuent le traitement.
EC	Fonctions de protection PATHWAY standard décrites ci-dessus.	L'EC coordonne la reprise avec l'EC Boss à l'aide du fichier RUNTIME contrôlé. L'EC s'enregistre de nouveau auprès de l'EC Boss et continue le traitement.
Programme d'écoute TCP/IP	<p>Plusieurs programmes d'écoute TCP/IP peuvent être configurés de façon à procurer une disponibilité supérieure en répartissant les canaux sur plusieurs ports, adresses IP ou processus de serveur TCP/IP.</p> <p>Fonctions de protection PATHWAY standard décrites ci-dessus.</p>	Aucune. Le programme d'écoute TCP/IP tente de se connecter à n'importe quel port disponible configuré pour le gestionnaire de files d'attente au redémarrage.

Tableau 8. Méthodes de protection utilisées pour les processus critiques (suite)

Processus	Méthodes de protection utilisées	Gestion de la reprise
Gestionnaire de référentiels ou gestionnaire de cache de référentiel	<p>En cas d'incident d'un gestionnaire de référentiels ou gestionnaire de cache de référentiel, les utilisateurs (qu'ils soient nouveaux ou non) du cache d'un même processeur ne subissent aucune interruption d'accès au cache ou des opérations de cluster.</p> <p>Le gestionnaire de référentiels maintient à tout moment une version cohérente sauvegardée sur disque dur du cache de cluster sur la file d'attente de référentiel. Lorsque des modifications sont effectuées, elles sont synchronisées avec le MQGET du message de la file d'attente de commande de cluster ayant provoqué la modification. La cohérence est donc maintenue même si le gestionnaire de référentiel échoue pendant l'enregistrement sur disque dur.</p> <p>Fonctions de protection PATHWAY standard décrites ci-dessus.</p>	<p>Lors d'une reprise, un processus de référentiel devient le gestionnaire de référentiels s'il n'en existe aucun dans le gestionnaire de files d'attente. Dans le cas contraire, il jouera le rôle de serveur de cache de référentiel pour le processeur dans lequel il est exécuté.</p> <p>Un gestionnaire de référentiels ou gestionnaire de cache de référentiel se relie au cache s'il existe toujours en mémoire. Sinon, le cache est rechargé à partir du disque, s'il est présent. Si le fichier de disque n'existe pas, un nouveau cache est créé dans l'unité d'exécution et initialisé à partir de la file d'attente de référentiel.</p>
Serveur de gestionnaire de files d'attente	<p>Le serveur de gestionnaire de files d'attente récupère les messages expirés et génère des rapports d'expiration synchronisés de façon à ce qu'un incident de serveur ne provoque pas d'incohérence.</p> <p>Fonctions de protection PATHWAY standard décrites ci-dessus.</p>	<p>Un serveur de gestionnaire de files d'attente redémarré se resynchronisera automatiquement avec les serveurs de file d'attente. Cette opération se déroulera à chaque fois qu'une opération de gestion interne sera nécessaire, au moment où les serveurs de file d'attente indiqueront dans un rapport un ensemble de messages expirés.</p>

Processus critiques

Tableau 8. Méthodes de protection utilisées pour les processus critiques (suite)

Processus	Méthodes de protection utilisées	Gestion de la reprise
Applications – liées par par STANDARD	<p>Les incidents rencontrés par MQSeries sont détectés par des connexions interrompues ou erreurs inattendues retournées par les appels MQI. L'application appellera MQDISC puis de nouveau MQCONN, pour établir de nouveau la connexion avec le gestionnaire de files d'attente.</p> <p>Le gestionnaire de files d'attente détecte immédiatement les incidents de processus applicatifs. Le gestionnaire de files d'attente exécute un MQDISC implicite à la place de l'application provoquant la fermeture ou la libération de toutes les ressources utilisées.</p> <p>Toutes les opérations de points de synchronisation en cours seront abandonnées par TM/MP et MQSeries devra alors effectuer les adaptations appropriées aux files d'attente locales.</p> <p>Le code d'application contenant les erreurs ne peut pas altérer les bases de données critiques ou ressources partagées d'un gestionnaire de files d'attente puisque la seule mémoire partagée située dans l'espace d'adresse de l'application est en lecture seule, et qu'aucun fichier de bases de données critiques n'est directement accessible.</p>	Aucune. Les applications se reconnectent de nouveau au gestionnaire de files d'attente.

Tableau 8. Méthodes de protection utilisées pour les processus critiques (suite)

Processus	Méthodes de protection utilisées	Gestion de la reprise
Applications – liées par FASTPATH	<p>Les incidents rencontrés par MQSeries sont détectés par des erreurs inattendues retournées par les appels MQI. L'application appellera MQDISC puis de nouveau MQCONN, pour établir de nouveau la connexion avec le gestionnaire de files d'attente.</p> <p>Le gestionnaire de files d'attente détecte immédiatement les incidents de processus applicatifs. Le gestionnaire de files d'attente exécute un MQDISC implicite à la place de l'application provoquant la fermeture ou la libération de toutes les ressources utilisées.</p> <p>Toutes les opérations de points de synchronisation en cours seront abandonnées par TM/MP et MQSeries devra alors effectuer les adaptations appropriées aux files d'attente locales.</p> <p>Un code d'application contenant des erreurs peut altérer les bases de données critiques et ressources partagées d'un gestionnaire de files d'attente puisqu'elles accèdent au cache de référentiel en lecture/écriture et aux structures internes du gestionnaire de files d'attente qui peuvent altérer le catalogue des objets. Notez que les structures de files d'attente et messages eux-mêmes sont protégés puisque seuls les serveurs de files d'attente y ont accès.</p>	Aucune. Les applications redémarrées se connectent à nouveau.

Clusters

Le rôle des clusters MQSeries pour Compaq NSK V5.1 est de réduire les besoins d'administration d'un réseau MQSeries, et d'étendre la disponibilité et l'évolutivité globales de MQSeries en tant que service réparti.

Les gestionnaires de files d'attente appartenant à un cluster peuvent procéder à des insertions MQPUT dans les files d'attente perçues comme locales par le cluster. La

Clusters

fonction d'utilisation de cluster de MQSeries permet l'administration et la gestion de tous les canaux et définitions requis pour transférer le message à la file d'attente de destination.

Les files d'attente de cluster peuvent être définies sur plus d'un gestionnaire de files d'attente d'un cluster. Plusieurs instances d'une file d'attente sont ainsi créées dans le cluster. Une insertion réalisée par une application ne concerne qu'une seule instance de la file d'attente, selon la configuration du gestionnaire de pondération de charge de cluster (CWLM), composant de MQSeries. Ce choix peut être effectué lors de l'ouverture de la file d'attente, ou dynamiquement pour chaque insertion. Le CWLM peut déterminer l'instance d'une file d'attente de cluster la plus appropriée, selon que le canal de l'instance fonctionne ou non, et selon d'autres facteurs tels que la priorité réseau, et les préférences de l'application. Cette opération se fait par l'exit de pondération de charge du cluster.

Les clusters peuvent par conséquent bénéficier d'une meilleure disponibilité du réseau MQSeries. MQSeries sous Compaq NSK est un excellent choix si l'on souhaite lui voir jouer un rôle de Référentiel global pour cluster, en raison de la fiabilité et de l'évolutivité de son fonctionnement.

Considérations sur la configuration pour la disponibilité

Cette section résume les options de configuration permettant d'étendre la disponibilité de MQSeries pour Compaq NSK V5.1 et de ses applications sous Compaq NSK :

- Configurez PATHWAY avec d'autres processeurs pour toutes les classes de serveur de façon à prévenir les incidents liés aux processeurs
- Envisagez l'utilisation du cluster pour étendre la disponibilité des ressources réseau de MQSeries
- Envisagez l'utilisation de messages non persistants lorsque la définition de points de contrôle est activée, de façon à obtenir de meilleures performances et une haute disponibilité pour les types de messages concernés
- Veillez à ce que votre configuration TM/MP soit dimensionnée pour les accroissements de demande prévisibles de MQSeries et de ses applications
- Envisagez d'utiliser des fichiers de dépassement de message pour réduire les besoins de fichier d'audit pour les messages très volumineux
- Dans l'éventualité d'une reprise après incident grave, envisagez d'utiliser RDF pour créer et mettre à jour un site de sauvegarde pour MQSeries
- Veillez à ce que les processeurs aient à leur disposition une quantité de mémoire physique suffisante pour faire face aux accroissements de demandes de MQSeries et de ses applications.
- Veillez à ce qu'une quantité d'espace de permutation suffisante soit mise à la disposition des processeurs utilisant MQSeries et ses applications.

Configuration favorable à l'intégrité des données

Cette section résume les options de configuration permettant d'étendre l'intégrité de MQSeries pour Compaq NSK V5.1 et de ses applications sous Compaq NSK :

- Déterminez quel niveau d'intégrité des données, apporté par le caractère persistant ou non des messages, correspond à chaque type de message et de transaction pris en charge par MQSeries
- Déterminez si les fichiers de dépassement de message sont adaptés au stockage de messages très volumineux dont vous pourriez avoir l'utilité

Configuration favorable à l'intégrité des données

- Déterminez si les messages non persistants nécessitent la définition de points de contrôle ou si c'est seulement le cas pour certains d'entre eux. Les différents types de messages doivent être placés dans différentes files d'attente afin de pouvoir spécifier différentes options de points de contrôle
- Veillez à ce que les opérations de synchronisation utilisant des messages persistants soient utilisées lorsque le plus haut niveau d'intégrité des données est requis
- Dans l'éventualité d'une reprise après incident grave, configurez et utilisez RDF pour créer et mettre à jour une copie de la sauvegarde de l'environnement MQSeries.

Configuration favorable à l'intégrité des données

Partie 2. Section de référence

Chapitre 17. Commandes de contrôle MQSeries	251
Résumé des Commandes de contrôle	251
Noms d'objets MQSeries	251
Lecture des diagrammes de syntaxe	252
altmqfls (modification des attributs d'un fichier de file d'attente)	254
altmqusr (modification des informations sur les utilisateurs MQSeries)	259
cleanrdf (exécution de la gestion interne de la RDF)	261
cnvclchl (conversion des définitions de canaux clients)	263
crtmqcvx (conversion de données)	265
crtmqm (création d'un gestionnaire de files d'attente)	267
dltmqm (suppression d'un gestionnaire de files d'attente)	271
dspmqaout (affichage des droits d'accès)	273
dspmqcsv (affichage du serveur de commandes)	277
dspmqls (affichage des fichiers MQSeries)	278
dspmqrtrc (affichage de la sortie de trace formatée MQSeries)	281
dspmqrusr (affichage des informations utilisateur MQSeries)	282
endmqcsv (arrêt d'un serveur de commandes)	284
endmqm (arrêt d'un gestionnaire de files d'attente)	286
endmqtrc (arrêt de la trace MQSeries)	288
instmqm (installation de MQSeries pour Compaq NSK)	289
runmqchi (exécution de l'initialisateur de canal)	290
runmqchl (démarrage d'un canal)	291
runmqdlq (lancement du gestionnaire de files d'attente de rebut)	292
runmqslr (exécution du programme d'écoute)	294
runmqsc (exécution des commandes MQSeries)	296
runmqtrm (lancement du moniteur de déclenchement)	299
setmqaut (définition ou réinitialisation des droits d'accès)	300
strmqcsv (lancement d'un serveur de commandes)	306
strmqm (lancement d'un gestionnaire de files d'attente)	307
strmqtrc (lancement de la trace MQSeries)	309
upgmqm (Mise à jour du gestionnaire de files d'attente V2.2.0.1)	312

Chapitre 17. Commandes de contrôle MQSeries

Ce chapitre contient des informations de référence sur les commandes de contrôle prises en charge par les produits MQSeries pour Compaq NSK.

Résumé des Commandes de contrôle

Les commandes de contrôle suivantes sont supportées par MQSeries pour Compaq NSK grâce aux macros TACL et aux programmes compilés :

- `altnqfls` (modification des attributs d'un fichier de file d'attente)
- `altnqusr` (modification des informations sur les utilisateurs MQSeries)
- `cleanrdf` (exécution de la gestion interne de la RDF)
- `cnvclchl` (conversion des définitions de canaux clients)
- `crtmqcvx` (conversion de données)
- `crtmqm` (création d'un gestionnaire de files d'attente)
- `dltmqm` (suppression d'un gestionnaire de files d'attente)
- `dspmqaout` (affichage des droits d'accès)
- `dspmqcsv` (affichage du serveur de commandes)
- `dspmqls` (affichage des fichiers MQSeries)
- `dspmqrtrc` (affichage de la sortie de trace formatée MQSeries)
- `dspmqrusr` (affichage des informations utilisateur MQSeries)
- `endmqcsvg` (arrêt d'un serveur de commandes)
- `endmqm` (arrêt d'un gestionnaire de files d'attente)
- `endmqtrc` (arrêt de la trace MQSeries)
- `instmqm` (installation de MQSeries pour Compaq NSK)
- `runmqchi` (exécution de l'initialisateur de canal)
- `runmqchl` (démarrage d'un canal)
- `runmqdlq` (lancement du gestionnaire de files d'attente de rebut)
- `runmqslr` (exécution du programme d'écoute)
- `runmqsc` (exécution des commandes MQSeries)
- `runmqtrm` (lancement du moniteur de déclenchement)
- `setmqaut` (définition ou réinitialisation des droits d'accès)
- `strmqcsvg` (lancement d'un serveur de commandes)
- `strmqm` (lancement d'un gestionnaire de files d'attente)
- `strmqtrc` (lancement de la trace MQSeries)
- `upgmqm` (mise à jour du gestionnaire de files d'attente V2.2.0.1)

Les descriptions détaillées de ces commandes sont présentées dans ce chapitre.

Remarques :

1. Les indicateurs, qui sont des indentificateurs à un caractère précédés d'un tiret (par exemple `-v` pour la commande `runmqsc`), doivent être spécifiés en minuscule.
2. Les messages d'utilisation sont affichés si les appels de commandes de contrôle contiennent `?`, `?` ou aucun paramètre lorsqu'un paramètre est prévu.

Noms d'objets MQSeries

Les noms des objets MQSeries suivants comportent au maximum 48 caractères :

- Gestionnaires de files d'attente
- Files d'attente
- Définitions de processus

Noms

La longueur maximale des noms de canaux est de 20 caractères.

Les caractères pouvant être utilisés pour tous les noms MQSeries sont :

- Les majuscules A à Z
- Les minuscules a à z
- Les numériques 0 à 9
- Le point (.)
- Le souligné (_)
- La barre oblique (/)
- Le signe de pourcentage (%)

Remarques :

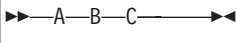

1. Les caractères barre oblique et pourcentage sont des caractères spéciaux. Si vous les utilisez dans un nom, celui-ci doit toujours être indiqué entre guillemets.
2. Les espaces ne sont pas autorisés au début ou au milieu des noms.
3. Les caractères nationaux ne sont pas autorisés.
4. Les noms peuvent être placés entre guillemets, mais ce n'est obligatoire que s'ils comprennent des caractères spéciaux.

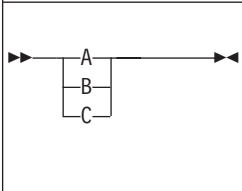
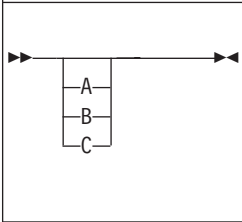
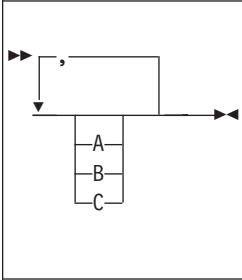
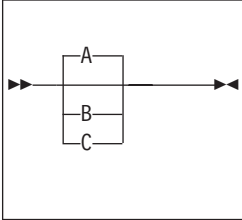
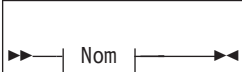
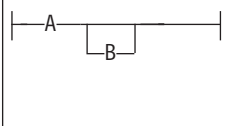
Lecture des diagrammes de syntaxe

Ce chapitre contient des diagrammes de syntaxe (parfois appelés diagrammes «linéaires»).

Chaque diagramme de syntaxe commence par une flèche double vers la droite et se termine par deux flèches, l'une vers la droite, l'autre vers la gauche. Une ligne commençant par une flèche simple vers la droite est la suite de la ligne précédente. Un diagramme de syntaxe se lit de gauche à droite et de haut en bas, en suivant la direction des flèches.

Les autres conventions utilisées dans les diagrammes de syntaxe sont les suivantes :

Convention	Signification
	Vous devez préciser les valeurs A, B et C. Les valeurs requises sont présentées sur la ligne principale du diagramme.
	Vous pouvez indiquer la valeur A. Les valeurs facultatives sont présentées sous la ligne principale du diagramme.

Convention	Signification
	<p>Vous avez le choix entre les trois valeurs A, B ou C, et vous devez en préciser une.</p>
	<p>Les trois valeurs A, B et C sont facultatives, vous pouvez en préciser une.</p>
	<p>Vous pouvez préciser une ou plusieurs des trois valeurs facultatives A, B et C. Tout séparateur requis pour les valeurs multiples ou répétées (dans cet exemple, la virgule (,)) est indiqué sur la flèche.</p>
	<p>Les trois valeurs A, B et C sont facultatives, vous pouvez en préciser une. Si vous n'en précisez aucune, la valeur par défaut A (la valeur présentée au-dessus de la ligne principale) est utilisée.</p>
<p data-bbox="459 1115 699 1186">  </p> <p data-bbox="459 1228 699 1270">Nom :</p> <p data-bbox="459 1291 699 1415">  </p>	<p data-bbox="699 1115 1453 1186">Le fragment de syntaxe Nom est à l'écart du diagramme de syntaxe principal.</p>

altmqfls (modification des attributs d'un fichier de file d'attente)

Objet

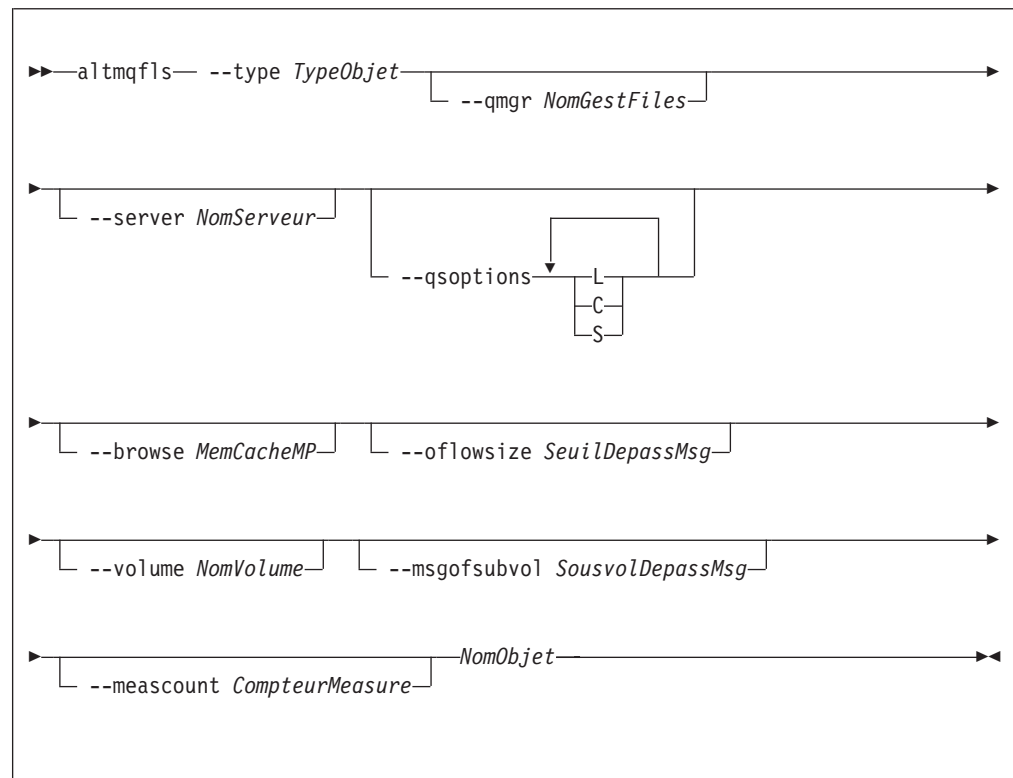
La commande **altmqfls** sert à modifier les attributs du fichier de file d'attente. Une même commande **altmqfls** ne peut effectuer qu'une seule des trois fonctions suivantes en même temps :

- Déplacer vers un autre volume les fichiers de messages qui appartiennent à une file d'attente locale donnée, afin de répartir la charge d'entrées-sorties entre les volumes ;
- Modifier la taille de la file d'attente et des fichiers de dépassement associés à une file d'attente locale. (cette opération est impossible si la file d'attente est ouverte) ;
- Modifier les options du serveur de file d'attente associé à une file d'attente locale. Ces options vous permettent d'optimiser la façon dont le serveur de file d'attente gère le stockage associé à la file d'attente, et contrôle la définition des points de contrôle des messages non persistants. Vous pouvez également modifier le serveur de file d'attente associé à une file d'attente locale le serveur de statut associé à un alias, une file d'attente modèle ou éloignée, ou un processus ; et associer un compteur à une file d'attente.

La commande **altmqfls** ne permet pas de recharger la file d'attente si elle est en cours d'utilisation.

Remarque : L'interface utilisateur et les options de ligne de commande pour la commande **altmqfls** ont été notablement modifiées par rapport à la Version 2.2.0.1. Ces modifications sont le reflet des nouvelles fonctionnalités et utilisent notamment des noms plus parlants pour les options de ligne de commande. De plus, tous les paramètres commencent par '--', afin de se conformer à la syntaxe de POSIX. MQSeries Version 5.1 rejettera toutes les options de ligne de commande de la Version 2.2.0.1, il vous faudra donc revoir tous les scripts qui font appel à la commande **altmqfls**.

Syntaxe



Paramètres obligatoires

NomObjet

Le nom de la file d'attente locale permanente dont les fichiers de messages doivent être déplacés. La file d'attente ne doit pas être ouverte, ni contenir de message non validé.

--type *TypeObjet*

Identifie une file d'attente permanente. Vous devez préciser une valeur pour *TypeObjet*, parmi l'une des suivantes :

- ql** ou **qlocal** Une file d'attente locale
- qa** ou **qalias** Une file d'attente alias
- qr** ou **qremote** Une file d'attente éloignée
- qm** ou **qmodel** Une file d'attente modèle
- proc** ou **process** Un processus

Paramètres facultatifs

--qmgr *NomGestFile*

Indique le nom du gestionnaire de files d'attente à qui la file d'attente locale appartient. Le gestionnaire doit être actif. Si aucun nom n'est indiqué, le gestionnaire par défaut est utilisé.

--server *NomServeur*

Indique le nom d'un processus de serveur de statut ou de serveur de file d'attente qui est responsable des données de statut de l'objet. Si l'objet est une file d'attente locale ou une file d'attente modèle, il convient d'utiliser un nom

de serveur de file d'attente. Si l'objet n'est pas une file d'attente locale ni une file d'attente modèle, c'est le nom du serveur de statut qui doit être utilisé.

Lors de la création d'un objet, la valeur de *NomServeur* est DEFAULT, pour indiquer que c'est le serveur par défaut qui est responsable de la file d'attente. Lors du changement du serveur responsable d'une file d'attente, celle-ci ne doit pas être en cours d'utilisation, et tous les messages non persistants sont éliminés pendant le changement.

--volume *NomVolume*

Il s'agit d'un nom de volume Compaq NSK (par exemple, \$DEV). Ce paramètre doit être précisé si vous utilisez **altnmqfls** pour déplacer des fichiers de messages vers un autre volume. Ce paramètre n'est autorisé qu'avec le nom et le type de l'objet. Son utilisation est interdite avec les autres options.

--qsoptions LSC

Les options L, S et C s'utilisent séparément ou en association, pour ajuster les réglages de fiabilité et de récupération de la file d'attente.

Remarque : *Toutes* les --qsoptions SLC sont modifiées à chaque utilisation de la commande. Par exemple, --qsoptions S désactivera les options L et C. L'option --qsoptions ne peut être utilisée qu'une fois par ligne de commande. Vous devez indiquer au moins une option, mais vous pouvez en indiquer plusieurs.

C Indique que les messages non persistants sont contrôlés par le serveur de file d'attente de secours. Ce système assure la tolérance de panne, au prix d'une certaine charge du processeur pour la gestion des points de contrôle supplémentaires, des messages IPC et de la mémoire nécessaire pour stocker les messages.

Vous utiliserez cette option pour réaliser un degré élevé de récupération des messages non-persistants. Lors de la création d'une file d'attente, ce paramètre est activé par défaut (c'est-à-dire que les points de contrôle sont activés).

L Indique que le serveur de file d'attente verrouille en mémoire les structures de données et chaînes associées à une file d'attente. Normalement, l'espace de stockage associé à une file d'attente est susceptible d'être retiré de l'espace d'adressage du serveur de file d'attente, une fois qu'il n'est plus accédé. Cette option accélère l'accès aux structures des données de mémoire d'une file d'attente, au détriment possible des autres files d'attente.

Lors de la création d'une file d'attente, ce paramètre est désactivé par défaut (c'est-à-dire que la mémoire n'est pas verrouillée).

S Indique que le serveur de file d'attente charge en cache la file d'attente locale depuis le disque, lors du démarrage du gestionnaire de files d'attente. Normalement, les messages d'une file d'attente sont chargés lorsqu'ils sont utilisés la première fois par une application. Si cette option est activée, la file d'attente est chargée dès le démarrage du gestionnaire de files d'attente. Lors de la création d'une file d'attente, ce paramètre est désactivé par défaut (la file d'attente n'est pas chargée au démarrage).

Cette option permet de réduire la charge du processeur et l'activité disque lors de la première ouverture d'une file d'attente, mais

augmentera en contrepartie l'activité du serveur de file d'attente (processeur et disque) lors du démarrage du gestionnaire de files d'attente.

None Indique qu'aucune option n'est activée.

--browse *MemCacheMP*

Indique le nombre maximal d'octets de données de chaque message persistant qui seront conservés dans le cache du serveur de file d'attente (et aussi sur le disque). Lors de la consultation d'un message persistant, le gestionnaire de files d'attente va normalement lire les données d'un message depuis le disque, puis les envoi à l'application. Si la valeur de ce paramètre est non nulle, le nombre d'octets de données précisé sera également conservé en mémoire, et ces données lues et envoyées à l'application sans qu'il y ait d'accès disque. L'utilisation de ce paramètre accroîtra les ressources mémoire nécessaires au serveur de file d'attente. La valeur minimale de ce paramètre est zéro (0) octet, la valeur maximale 25 000 octets. Lors de la création d'une file d'attente, la valeur par défaut de ce paramètre est zéro.

--oflowsize *SeuilDepassMsg*

Indique la taille minimale de message, lors de l'utilisation d'un fichier de dépassement de capacité de message pour stocker les données du message. Les messages persistants de taille inférieure à ce seuil sont enregistrés dans le fichier de dépassement de capacité de la file d'attente. Les données non formatées des messages persistants de taille supérieure ou égale à ce seuil seront enregistrées dans un fichier de dépassement de capacité spécifique. Lors de la création d'une file d'attente, la valeur par défaut de ce paramètre est de 200 000 octets.

--msgofsubvol *SousvolDepassMsg*

Indique un sous-volume du volume sur lequel se trouve la file d'attente, où le serveur de file d'attente crée les nouveaux fichiers de dépassement de capacité de message. Par défaut, toutes les files d'attente utilisent le sous-volume de leur gestionnaire de files d'attente.

--meascount *CompteurMeasure*

Indique le nom d'un compteur MEASURE. S'il fait partie d'un système de mesure activé, ce compteur est initialisé à la profondeur en cours, puis incrémenté et décrémenté par le serveur responsable de la file d'attente, au fur et à mesure que les messages sont ajoutés ou retirés.

Codes retour

- 0 La commande a abouti.
- 10 La commande n'a abouti que partiellement.
- 20 Une erreur s'est produite pendant le traitement.

Exemples

1. Dans l'exemple qui suit, les fichiers de message appartenant à la file d'attente locale `fileattente.silex`, détenue par le gestionnaire de files d'attente `gest.files.cible`, sont déplacés vers le volume `$DATA3`.

```
altnmqfls --qmgr gest.files.cible --type ql --volume $DATA3 file.silex
```

altmqfls

2. Dans l'exemple qui suit, le nom de processus du serveur de file d'attente d'un objet est changé :

```
altmqfls --qmgr gest.files.cible --type ql --server $TQS2 file.silex
```

Cette commande entraîne le déplacement de la file d'attente locale file silex, qui appartient au gest.files.cibler, vers un serveur de file d'attente dont le nom de processus est \$TQS2.

Commandes connexes

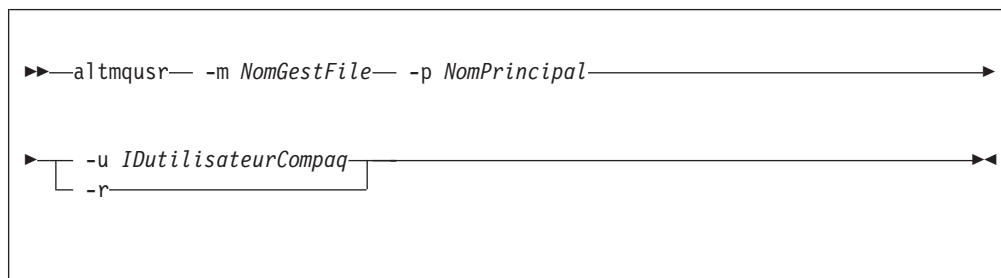
dspmqfls Afficher les fichiers MQSeries

altmqusr (modification des informations sur les utilisateurs MQSeries)

Objet

Utilisez la commande **altmqusr** pour définir ou supprimer un principal correspondant à un ID utilisateur Compaq NSK qui aura accès à MQSeries.

Syntaxe



Description

Vous pouvez utiliser cette commande pour :

- créer un principal (c'est à dire accorder à un ID utilisateur Compaq NSK l'accès à un gestionnaire de files d'attente).
- supprimer un principal (c'est à dire supprimer à un ID utilisateur Compaq son accès à un gestionnaire de files d'attente).
- modifier la définition d'un principal.

Lorsque `-u IDutilisateurCompaq` est défini, **altmqusr** crée un principal, s'il n'en existe pas déjà, ou modifie la définition existante. `IDutilisateurCompaq` peut être défini comme ID utilisateur d'administration Compaq, ou, si SAFEGUARD est en cours de fonctionnement, comme alias SAFEGUARD.

Lorsque `-r` est spécifié, le principal est effacé de la base de données des principaux.

Il faut spécifier soit `-u IDutilisateurCompaq`, soit `-r`.

Paramètres obligatoires

-m *NomGestFiles*

Nom du gestionnaire de files d'attente auquel appartient le principal.

-p *NomPrincipal*

Nom du principal à créer, modifier ou supprimer.

-u *IDutilisateurCompaq*

ID utilisateur d'administration Compaq NSK ou alias SAFEGUARD à associer à la définition du principal.

-r Indique que la définition du principal doit être supprimée du gestionnaire de files d'attente.

Codes retour

- 0 Opération réussie.
- 36 Arguments incorrects fournis.
- 69 La mémoire n'est pas disponible.

altnqusr

71 Erreur inattendue.

Exemples

Pour ajouter un principal mquser1 correspondant à l'ID utilisateur Compaq mqtest.fred :

```
altnqusr -m MT02 -p mquser1 -u mqtest.fred
```

Pour ajouter un principal mquser2 correspondant au groupe group.user01 :

```
altnqusr -m MT02 -p mquser2 -u group.user01
```

Pour visualiser les résultats de ces commandes, utilisez la commande **dspmqusr**, comme décrit dans la section «dspmqusr (affichage des informations utilisateur MQSeries)» à la page 282.

Pour supprimer le principal mquser1 :

```
altnqusr -m MT02 -p mquser1 -r
```

Commandes connexes

dspmqusr Affichage des informations sur les utilisateurs MQSeries

cleanrdf (exécution de la gestion interne de la RDF)

Objet

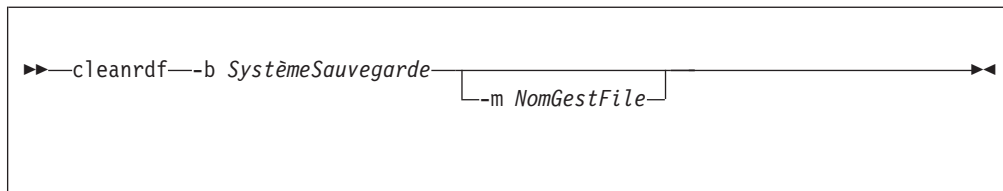
Utilisez la fonction **cleanrdf** pour exécuter la gestion interne de routine sur le gestionnaire de files d'attente du système primaire dans un environnement RDF. La fonction **cleanrdf** permet de retirer les fichiers qui ont été logiquement supprimés sur le système primaire et le système de sauvegarde. Cette fonction copie également des bases de données non validées à leur emplacement normal sur le système de sauvegarde.

Notez que l'utilitaire appelé par **cleanrdf** inspecte l'ensemble du catalogue de l'objet et la base de données de messages si bien que les performances risquent de baisser durant l'utilisation de cette fonctionnalité.

Les fichiers de base de données non validées copiés dans le système de sauvegarde par cet utilitaire sont :

- MQERRLG1
- MQSINI
- QMINI
- PATHCTL
- SHUTDOWN
- SYNCHIDX
- TRACEOPT
- UMQSINI

Syntaxe



Paramètres obligatoires

-b *SystèmeSauvegarde*

Indique le nom Compaq NSK du système où se trouve le site de sauvegarde de la RDF concernant le gestionnaire de files d'attente. *SystèmeSauvegarde* est spécifié sous la forme *\nom* (la forme standard dans un environnement Compaq NSK).

Paramètres facultatifs

-m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente pour lequel **cleanrdf** doit être exécuté. Si aucun nom n'est indiqué, la commande s'exécute sur le gestionnaire par défaut.

Codes retour

- 0 La commande a abouti.
- 20 Une erreur s'est produite pendant le traitement.

cleanrdf

Exemples

Dans l'exemple suivant, **cleanrdf** s'exécute sur le gestionnaire de files d'attente `test.queue.mgr`. Le noeud Compaq NSK \HAWK est configuré comme étant le site de sauvegarde de la RDF pour ce gestionnaire.

```
cleanrdf -b \HAWK -m test.queue.mgr
```

cnvclchl (conversion des définitions de canaux clients)

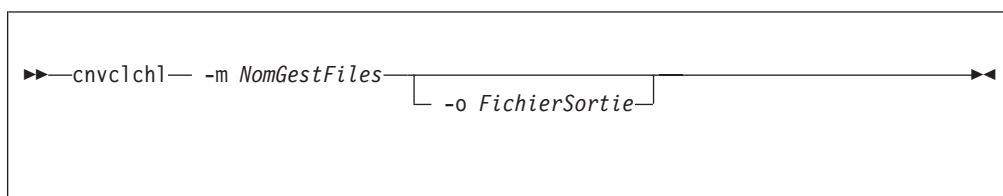
Objet

Utilisez la commande **cnvclchl** pour convertir le fichier de définition de canal client (créé par MQSC pour les canaux CLNTCONN) d'un format de fichier structuré Compaq en un format non structuré accepté par les clients MQSeries.

Remarque :

Les fichiers de définition de la Version 2.2.0.1 ne fonctionnent pas avec les clients de la Version 5.1 car le format des enregistrements a changé. Une fois MQSeries pour Compaq NSK V5.1 installé, vous pouvez réexécuter la commande **cnvclchl** de façon à créer des fichiers de définition utilisables par les clients équipés de la Version 5.1.

Syntaxe



Paramètres obligatoires

-m *NomGestFiles*

Désigne le gestionnaire de files d'attente propriétaire du fichier de définition de canaux (CCHDEFS) à convertir. Cette valeur est obligatoire.

Paramètres facultatifs

-o *FichierSortie*

Désigne le fichier contenant les définitions converties. Le nom de fichier par défaut est AMQCLCHL.

cnvclchl

Exemples

La commande suivante convertit le fichier de définition de canal client structuré fourni par Compaq pour le gestionnaire de files d'attente MT01 en un fichier non structuré. Les deux définitions de canal client figurent dans le fichier de sortie AMQCLCHL, SYSTEM.DEF.CLNTCONN et SOLARIS_TO_Cmpaq :

```
$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 (conversion de données)

Objet

Utilisez la commande **crtmqcvx** pour créer un fragment de code qui effectuera une conversion de données sur des structures de type de données. Cette commande génère une fonction C qui peut servir dans un exit pour convertir des structures.

Elle lit dans un fichier d'entrée les structures à convertir, puis écrit un fichier de sortie contenant les fragments de code destinés à la conversion des structures.

Pour plus d'informations sur cette commande et sur son utilisation, reportez-vous au manuel *MQSeries - Guide de programmation d'applications*.

Syntaxe

```
►►—crtmqcvx—FichierSource—FichierCible—►►
```

Paramètres obligatoires

FichierSource

Indique le fichier d'entrée contenant les structures C à convertir.

FichierCible

Indique le fichier de sortie contenant les fragments de code générés pour la conversion des structures.

Codes retour

- 0 La commande a abouti.
- 10 La commande a abouti avec des résultats imprévus.
- 20 Une erreur s'est produite pendant le traitement.

Exemples

L'exemple suivant montre les résultats de l'utilisation de la commande de conversion de données sur une structure source C. La commande émise est :

```
crtmqcvx FichierSource FichierCible
```

crtmqcvx

Le fichier d'entrée, source se présente de la façon suivante :

```
/* Voici une structure C de test qui peut être convertie par */
/* l'utilitaire crtmqcvx                                     */

struct my_structure
{
    int    code;
    MQLONG value;
};
```

Le fichier de sortie, cible, produit par la commande est présenté ci-dessous. Vous pouvez utiliser ces fragments de code dans vos applications pour convertir des structures de données. Mais, le cas échéant, tenez compte du fait que le fragment se sert des macros fournies dans le fichier MQSVMHHTH d'en-tête MQSeries situé dans le sous-volume ZMQSLIB.

```
MQLONG Convertmy_structure(
    PMQBYTE *in_cursor,
    PMQBYTE *out_cursor,
    PMQBYTE in_lastbyte,
    PMQBYTE out_lastbyte,
    MQHCONN hConn,
    MQLONG  opts,
    MQLONG  MsgEncoding,
    MQLONG  ReqEncoding,
    MQLONG  MsgCCSID,
    MQLONG  ReqCCSID,
    MQLONG  CompCode,
    MQLONG  Reason)
{
    MQLONG ReturnCode = MQRC_NONE;

    ConvertLong(1); /* code */

    AlignLong();
    ConvertLong(1); /* value */

Fail:
    return(ReturnCode);
}
```

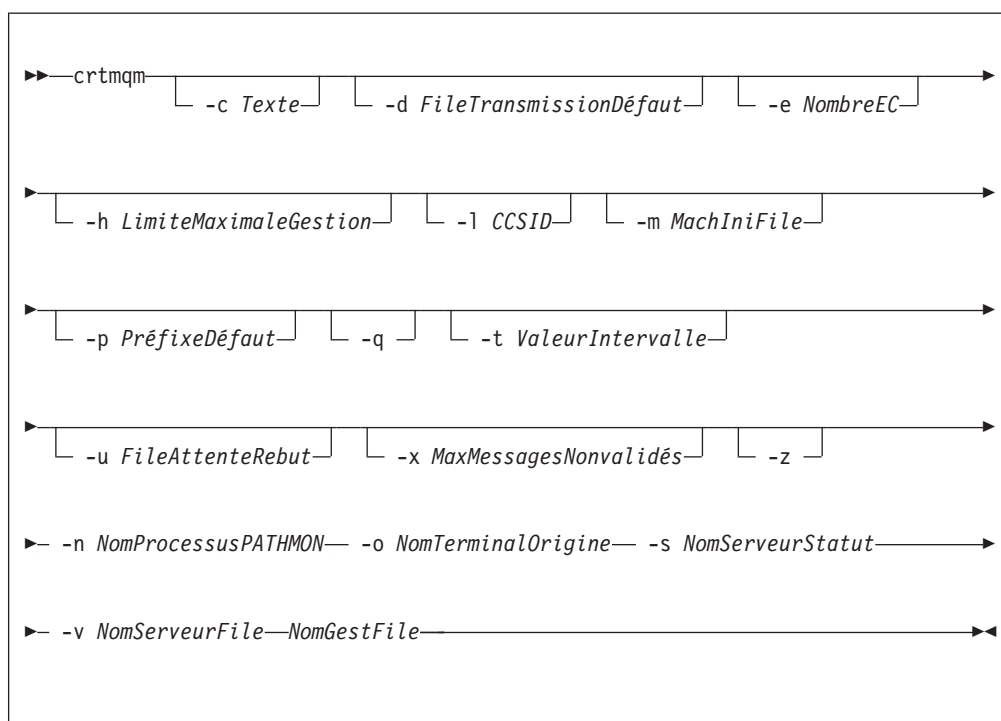
crtmqm (création d'un gestionnaire de files d'attente)

Objet

Utilisez la commande **crtmqm** pour créer un gestionnaire de files d'attente local. Une fois le gestionnaire de files d'attente créé, utilisez la commande **strmqm** pour le démarrer.

La création d'un gestionnaire de files d'attente engendre automatiquement la création des objets par défaut et objets systèmes correspondant. Reportez-vous à l'«Annexe B. Objets système et objets par défaut» à la page 321 pour obtenir une liste des objets créés automatiquement.

Syntaxe



Paramètres obligatoires

- n *NomProcessusPATHMON*
Nom du processus PATHMON TS/MP du gestionnaire. Ce nom de processus doit être unique dans tout le système.
- o *NomTerminalOrigine*
Nom du terminal d'origine (\$DDDD.#SS). Par exemple, \$TRM1.#A.
- s *NomServeurStatut*
Nom de processus à donner au serveur de statut par défaut du gestionnaire de files d'attente. Ce nom de processus doit être unique sur le système.
- v *NomServeurFile*
Spécifie un unique nom de processus à donner au processus de serveur de files d'attente.

NomGestFiles

Nom du gestionnaire de files d'attente à créer. Ce nom peut comporter jusqu'à 48 caractères. Il doit être le dernier élément de la commande.

Paramètres facultatifs

-c *Texte*

Texte (jusqu'à 64 caractères) décrivant ce gestionnaire. Par défaut, le texte reste blanc.

Si des caractères spéciaux sont nécessaires, la description doit être placée entre guillemets.

-d *FileTransmissionDéfaut*

Nom de la file de transmission locale dans laquelle sont insérés les messages éloignés, si une file de transmission n'est pas explicitement définie pour leur destination. Il n'y a pas de valeur par défaut.

-e *NombreEC*

Nombre de processus EC dans le gestionnaire. Valeur par défaut : 1.

-h *LimiteMaximaleGestion*

Dans MQSeries pour Compaq NSK, ce paramètre est ignoré.

Nombre maximal d'indicateurs qu'une application peut ouvrir en même temps. Précisez une valeur comprise entre 1 et 999 999 999. La valeur par défaut est 256.

-l *CCSID*

GestFile CCSID. La valeur par défaut est 819.

-m *MachIniFile*

Remplace l'adresse de fichier MQSINI par défaut et celle qui est spécifiée dans la variable d'environnement MQMACHINEFILE.

-p *DefaultPrefix*

Volume du gestionnaire de files d'attente. Remplace la valeur de QMDefaultVolume dans le fichier MQSINI.

-q Indique que ce gestionnaire de files d'attente sera pris en compte par défaut. Le nouveau gestionnaire remplace tout gestionnaire précédemment utilisé par défaut.

Si vous utilisez accidentellement cet indicateur et désirez revenir au gestionnaire de files précédemment pris en compte comme gestionnaire par défaut, vous pouvez éditer la strophe DefaultQueueManager dans le fichier de configuration MQSeries. Reportez-vous au «Chapitre 13. Fichiers de configuration» à la page 189 pour plus de détails sur les fichiers de configuration.

-t *ValeurIntervalle*

Intervalle de déclenchement en millisecondes pour toutes les files contrôlées par ce gestionnaire. Cette valeur indique le délai pendant lequel le déclenchement est suspendu après la réception d'un message générant un déclenchement. En d'autres termes, si l'arrivée d'un message dans une file provoque l'insertion d'un message de déclenchement dans la file d'attente d'initialisation, tout message arrivant dans la même file pendant le laps de temps spécifié ne génère pas d'autre message de déclenchement.

Vous pouvez utiliser cet intervalle de temps pour garantir que l'application disposera du temps nécessaire au traitement de la condition de déclenchement

avant d'être alertée pour en traiter une autre. De même, si vous désirez suivre tous les événements de déclenchement qui se produisent, indiquez une valeur faible ou égale à zéro dans cette zone.

Précisez une valeur comprise entre 0 et 999 999 999. La valeur par défaut est 999 999 999 millisecondes, soit plus de 11 jours. L'utilisation effective de la valeur par défaut revient à désactiver le déclenchement après le premier message. Le déclenchement peut cependant être réactivé par une application de service de file d'attente à l'aide d'une commande de modification de file d'attente permettant de redéfinir l'attribut de déclenchement.

-u *FileAttenteRebut*

Nom de la file d'attente locale à utiliser comme file d'attente de rebut (messages non transmis). Les messages sont insérés dans cette file s'ils ne peuvent pas être transmis à leur destination correcte.

Aucune file d'attente de rebut n'est créée par défaut.

-x *MaximumMessagesAttente*

Dans MQSeries pour Compaq NSK, ce paramètre est ignoré.

Nombre maximal de messages non validés lors d'un point de synchronisation. Il s'agit de la somme :

- du nombre de messages pouvant être extraits des files d'attente ;
- du nombre de messages pouvant être insérés dans les files d'attente ;
- des messages de déclenchement générés dans cette unité de travail.

Cette limite ne s'applique pas aux messages extraits ou insérés en file d'attente en-dehors d'un point de synchronisation.

Précisez une valeur comprise entre 1 et 10 000. La valeur par défaut est de 1000 messages non validés.

-z Supprime les messages d'erreur.

Cet indicateur est normalement utilisé dans MQSeries pour supprimer des messages d'erreur indésirables. L'utilisation de cet indicateur pouvant entraîner la perte d'informations, il est recommandé de ne pas l'utiliser lors de la saisie de commandes sur une ligne de commande.

Codes retour

- | | |
|-----|--|
| 0 | Le gestionnaire de files d'attente est créé. |
| 8 | Le gestionnaire de files d'attente existe déjà. |
| 49 | Le gestionnaire de files d'attente est en cours d'arrêt. |
| 69 | La mémoire n'est pas disponible. |
| 70 | Pas de place disponible dans la file d'attente. |
| 71 | Erreur inattendue. |
| 72 | Le nom du gestionnaire de files d'attente est erroné. |
| 111 | Le gestionnaire de files d'attente a été créé. Toutefois, il y a eu un incident lors du traitement de la définition du gestionnaire par défaut dans le fichier de configuration du produit. La spécification de gestionnaire de files d'attente par défaut peut être incorrecte. |

crtmqm

Exemples

1. Cette commande crée un gestionnaire de files d'attente par défaut appelé Paint.gest.files, auquel est attribuée la description Paint Shop :

```
crtmqm -c "Paint Shop" -n $PANT -o $TRM1.#A -s $PNT1  
-v $PQS1 Paint.gest.files
```

2. Dans cet exemple, un autre gestionnaire de file d'attente, voyage, est créé. L'intervalle de déclenchement est défini à 5000 millisecondes (soit 5 secondes) et sa file d'attente de rebut est spécifiée comme SYSTEM.FILE.ATTENTE.REBUT

```
crtmqm -t 5000 -u SYSTEM.FILE.ATTENTE.REBUT -n  
$TRAV -o $TRM1.#A -s $TRV1 -v $TQS1 voyage
```

Une fois l'événement de déclenchement généré, les autres événements de déclenchement sont désactivés pendant cinq secondes.

Commandes connexes

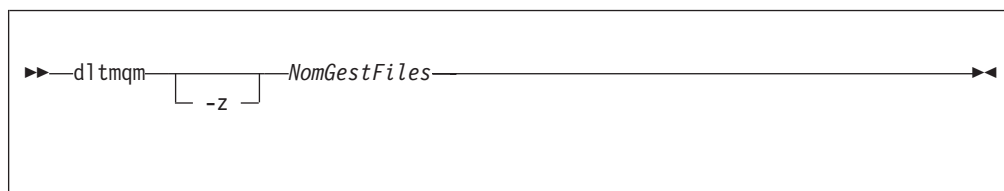
strmqm	lancement d'un gestionnaire de files d'attente
endmqm	arrêt d'un gestionnaire de files d'attente
dltmqm	suppression d'un gestionnaire de files d'attente

dltmqm (suppression d'un gestionnaire de files d'attente)

Objet

La commande **dltmqm** permet de supprimer un gestionnaire de files d'attente spécifié, ainsi que tous les objets qui lui sont associés. Avant de supprimer un gestionnaire de files d'attente, vous devez l'arrêter avec la commande **endmqm**.

Syntaxe



Paramètres obligatoires

NomGestFiles

Indique le nom du gestionnaire de files d'attente à supprimer.

Paramètres facultatifs

-z Supprime les messages d'erreur.

Codes retour

- 0 Le gestionnaire de files d'attente a été supprimé.
- 5 Le gestionnaire de files d'attente est en cours d'exécution.
- 16 Le gestionnaire de files d'attente n'existe pas.
- 69 La mémoire n'est pas disponible.
- 71 Erreur inattendue.
- 72 Le nom du gestionnaire de files d'attente est erroné.
- 112 Le gestionnaire de files d'attente a été supprimé. Toutefois, il y a eu un incident lors du traitement de la définition du gestionnaire par défaut dans le fichier de configuration du produit. La spécification de gestionnaire de files d'attente par défaut peut être incorrecte.

Exemples

1. La commande suivante supprime le gestionnaire de files d'attente saturne.gest.files :

```
dltmqm saturne.gest files
```

2. La commande suivante supprime le gestionnaire de files d'attente voyage ainsi que tous les messages générés par la commande :

```
dltmqm -z "voyage"
```

dltmqm

Commandes connexes

crtmqm	création d'un gestionnaire de files d'attente
strmqm	lancement d'un gestionnaire de files d'attente
endmqm	arrêt d'un gestionnaire de files d'attente

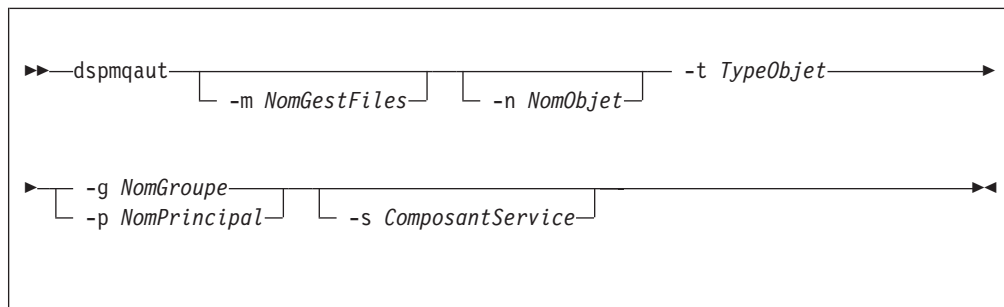
dspmqaout (affichage des droits d'accès)

Objet

La commande **dspmqaout** permet d'afficher les droits d'accès à un objet spécifique. Vous ne pouvez spécifier qu'un seul groupe.

Si un ID utilisateur est membre de plusieurs groupes, ses droits sont constitués par l'ensemble des droits de chaque groupe auquel il appartient.

Syntaxe



Paramètres obligatoires

-g *NomGroupe*

Indique le nom du groupe d'utilisateurs sur lequel porte l'interrogation. Vous ne pouvez spécifier qu'*un seul* nom, qui doit être celui d'un groupe d'utilisateurs existant. Vous devez spécifier soit **-g** *NomGroupe* soit **-p** *NomPrincipal*.

-p *NomPrincipal*

Indique le nom d'un utilisateur dont les droits d'accès à l'objet indiqué doivent être affichés. Vous devez spécifier soit **-g** *NomGroupe* soit **-p** *NomPrincipal*.

-t *TypeObjet*

Indique le type de l'objet sur lequel porte l'interrogation. Les valeurs possibles sont :

queue ou q	Une ou plusieurs files correspondant au paramètre <i>TypeObjet</i>
qmgr	Objet gestionnaire de files d'attente
process ou prcs	Processus
namelist ou nl	Liste de noms

Paramètres facultatifs

-m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente sur lequel porte l'interrogation.

-n *NomObjet*

Indique le nom de l'objet sur lequel porte l'interrogation.

Ce paramètre est obligatoire *sauf* s'il s'agit du gestionnaire de files d'attente lui-même (dans ce cas, le paramètre doit être omis).

dspmqaout

Vous devez préciser le nom du gestionnaire de files d'attente, de la file d'attente ou de la définition de procédure.

-s *ComposantService*

Ce paramètre ne s'applique que si vous utilisez des fonctions de droits d'accès optionnelles.

Si les services d'autorisation optionnels sont pris en charge, ce paramètre indique le nom du service concerné par les autorisations. Ce paramètre est facultatif ; en son absence, la demande de vérification des droits est adressée au premier composant optionnel de la fonction.

Paramètres renvoyés

Cette commande renvoie une liste de droits d'accès qui peut comporter un nombre quelconque de valeurs (y compris aucune). Chaque valeur renvoyée signifie que tout ID utilisateur du groupe indiqué dispose des droits d'accès nécessaires à l'exécution de l'opération définie par cette valeur.

Le tableau 9 présente les droits qui peuvent être accordés suivant les différents types d'objets.

Tableau 9. Droits d'accès de sécurité avec la commande dspmqaout

Droits d'accès	File d'attente	Processus	GestFiles	Liste de noms
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
chgaut	✓	✓	✓	
clr	✓			
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

La liste suivante définit les droits d'accès associés à chaque valeur :

- all** Utilisation de toutes les opérations concernant cet objet.
- alladm** Exécution de toutes les opérations d'administration concernant cet objet.
- allmqi** Utilisation de tous les appels MQI concernant cet objet.

altusr	Indication d'un ID utilisateur secondaire pour un appel MQI.
browse	Extraction d'un message d'une file par l'émission d'un appel MQGET avec l'option BROWSE.
chg	Modification des attributs de l'objet indiqué à l'aide du jeu de commandes approprié.
chgaut	Désignation des droits d'accès à l'objet pour d'autres groupes d'utilisateurs, à l'aide de la commande setmqaut .
clr	Effacement d'une file (commande PCF Clear queue seulement).
connect	Connexion de l'application au gestionnaire de files d'attente indiqué par l'émission d'un appel MQCONN.
crt	Création des objets du type indiqué à l'aide du jeu de commandes approprié.
dlt	Suppression de l'objet indiqué à l'aide du jeu de commandes approprié.
dsp	Affichage des attributs de l'objet indiqué à l'aide du jeu de commandes approprié.
get	Extraction d'un message d'une file par l'émission d'un appel MQGET.
inq	Interrogation sur une file donnée par l'émission d'un appel MQINQ.
passall	Transmission de tout le contexte.
passid	Transmission du contexte d'identité.
put	Insertion d'un message dans une file d'attente donnée par l'émission d'un appel MQPUT.
set	Définition des attributs d'une file à partir de MQI par l'émission d'un appel MQSET.
setall	Définition de tout le contexte d'une file d'attente.
setid	Définition du contexte d'identité d'une file d'attente.

Les droits d'accès aux opérations administratives, si elles sont prises en charge, s'appliquent aux jeux de commandes suivants :

- Commandes de contrôle
- Commandes MQSC
- Commandes PCF

Codes retour

0	Opération réussie.
36	Arguments incorrects fournis.
40	Le gestionnaire de files d'attente n'est pas disponible.
49	Le gestionnaire de files d'attente est en cours d'arrêt.
69	La mémoire n'est pas disponible.
71	Erreur inattendue.
72	Le nom du gestionnaire de files d'attente est erroné.
133	Nom d'objet inconnu.
145	Nom d'objet imprévu.
146	Nom d'objet absent.
147	Type d'objet absent.
148	Type d'objet incorrect.
149	Nom d'entité absent.

dspmqaout

Exemples

L'exemple suivant présente une commande permettant d'afficher les droits d'accès au gestionnaire de files d'attente saturne.gest.files qui est associé au groupe d'utilisateurs personnel :

```
dspmqaout -m "saturne.gest.files" -t qmgr -g personnel
```

Les résultats de cette commande sont les suivants :

```
Entity staff has the following authorizations for object :
  get
  browse
  put
  inq
  set
  connect
  altusr
  passid
  passall
  setid
```

Commandes connexes

setmqaut Définition ou réinitialisation des droits d'accès

dspmqsrv (affichage du serveur de commandes)

Objet

Utilisez la commande **dspmqsrv** pour afficher l'état du serveur de commandes pour un gestionnaire de files d'attente donné.

L'état peut être l'un des suivants :

- Démarrage
- Fonctionnement
- Fonctionnement avec SYSTEM.ADMIN.COMMAND.QUEUE non activée pour les extractions
- Arrêt en cours
- Arrêté

Syntaxe

```
▶▶ dspmqsrv —QMgrName—————▶▶
```

Paramètres obligatoires

NomGestFiles

Indique le nom du gestionnaire de files d'attente local pour lequel l'état du serveur de commandes a été demandé.

Codes retour

- 0 La commande a abouti.
- 10 La commande a abouti avec des résultats imprévus.
- 20 Une erreur s'est produite pendant le traitement.

Exemples

La commande suivante présente l'état du serveur de commandes associé à `gest.files.vénus` :

```
dspmqsrv venus.q.mgr
```

Commandes connexes

- strmqcvs** lancement d'un serveur de commandes
- endmqcvs** arrêt d'un serveur de commandes

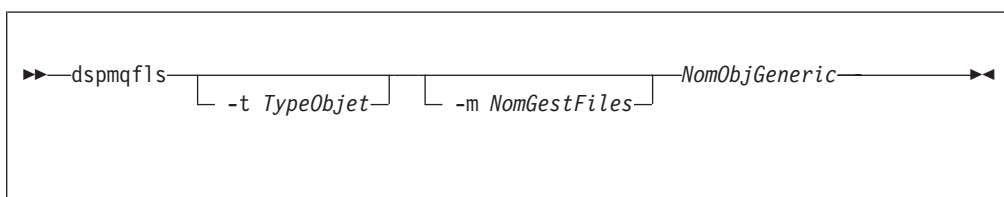
dspmqls (affichage des fichiers MQSeries)

Objet

Utilisez la commande **dspmqls** pour afficher le nom réel du système de fichiers pour tous les objets MQSeries correspondant à un critère donné. Vous pouvez utiliser cette commande pour identifier les fichiers associés à un objet MQSeries déterminé. Cela peut être utile pour sauvegarder des objets spécifiques. Pour plus d'informations sur la transformation des noms, reportez-vous à la section «Structure du volume» à la page 62.

Vous pouvez utiliser la commande **dspmqls** pour afficher les valeurs actuelles du nom du processus du serveur de files d'attente et les options d'un objet.

Syntaxe



Paramètres obligatoires

NomObjGeneric

Indique le nom de l'objet MQSeries. Le nom est une chaîne de caractères sans indicateur. C'est un paramètre obligatoire. En l'absence de nom, une erreur est renvoyée.

Ce paramètre accepte le caractère générique * à la fin de la chaîne.

Paramètres facultatifs

-m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente dont les fichiers doivent être examinés. Si aucun nom n'est indiqué, la commande s'applique au gestionnaire de files d'attente par défaut.

-t *TypeObj*

Indique le type d'objet MQSeries. La liste suivante présente les types d'objets corrects. Le nom abrégé est présenté en premier, suivi du nom complet.

- * ou all** Tous les types d'objets ; il s'agit de la valeur par défaut
- q ou queue** Une ou plusieurs files d'attente correspondant au paramètre NomObjet
- ql ou qlocal** File d'attente locale
- qa ou qalias** File d'attente alias
- qr ou qremote** File d'attente éloignée
- qm ou qmodel** File d'attente modèle
- qmgr** Objet gestionnaire de files d'attente
- prcs ou process** Processus
- ctlg ou catalog** Catalogue d'objets

nl ou namelist

Liste de noms

Remarque : La commande **dspmqls** affiche les noms de tous les fichiers de la file d'attente.

Codes retour

- 0 La commande a abouti.
- 10 La commande n'a abouti que partiellement.
- 20 Une erreur s'est produite pendant le traitement.

Exemples

1. La commande suivante affiche les détails de tous les objets ayant un nom commençant par SYSTEM.ADMIN définis sur le gestionnaire de files d'attente par défaut :

```
dspmqls SYSTEM.ADMIN*
```

2. La commande suivante affiche les détails des fichiers de tous les processus ayant des noms commençant par PROC définis sur le gestionnaire de files RAYON :

```
dspmqls -m RAYON -t prcs PROC*
```

3. La commande suivante affiche des informations concernant les fichiers de MY.LOCAL.QUEUE :

```
dspmqls -m MT02 -t q MY.LOCAL.QUEUE

MQSeries Display MQ Files

CONNECTING.
QLOCAL MY.LOCAL.QUEUE
 $DATA0.MT02M.QMYXLOCA
 $DATA0.MT02M.OMYXLOCA
 $DATA0.MT02M.TMYXLOCA

Queue/Status Server:          DEFAULT
Persistent message browse buffer: 0
Message overflow threshold:    200000
Queue Server Options:         None
Message overflow subvolume:    QMGR0010
Measure Counter:
```

- QMYXLOCA est le fichier de file d'attente, OMYXLOCA est le fichier de dépassement de la file d'attente, et TMYXLOCA est le fichier d'horodatage (touch file) dans le répertoire de données du gestionnaire.

dspmqfls

4. L'exemple suivant montre une commande **altmqfls** qui définit la charge au démarrage et les options de point de contrôle du serveur de file d'attente NPM. La commande **dspmqfls** affiche les résultats de la commande **altmqfls**.

```
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:
```

Commandes connexes

altmqfls Modification du volume de file d'attente

dspmqrtrc (affichage de la sortie de trace formatée MQSeries)

Objet

La commande **dspmqrtrc** permet d'afficher une sortie de trace formatée MQSeries. Pour plus d'informations sur le traçage de MQSeries, reportez-vous à la section «Traçage de MQSeries» à la page 215.

Syntaxe

```
►—dspmqrtrc— -t ModèleFormat —NomFichEntrée—◄
```

Paramètres obligatoires

NomFichEntrée

Indique le nom du fichier contenant les traces non formatées. Par exemple \$DATA.MQTRACE.AMQ12345.

-t *ModèleFormat*

Indique le nom du fichier modèle contenant les détails d'affichage de la trace. Un fichier modèle de format de trace, AMQTRC, se trouve dans le sous-volume ZMQSSMPL.

Commandes connexes

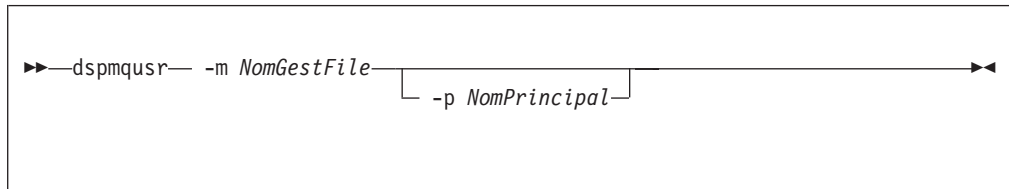
endmqrtrc	Arrêt de la trace MQSeries
strmqrtrc	Lancement de la trace MQSeries

dspmqsqr (affichage des informations utilisateur MQSeries)

Objet

La commande **dspmqsqr** affiche les informations concernant un principal donné d'un gestionnaire de files d'attente, ou tous ses principaux.

Syntaxe



Description

Cette commande permet de :

- Afficher tous les principaux (ou un principal donné) définis pour un gestionnaire de files d'attente.
- Afficher les groupes de partage de fichiers SAFEGUARD et Administrative de Compaq NSK, correspondant à l'ID utilisateur associé à chaque principal de Compaq NSK.

Paramètres obligatoires

-m *NomGestFiles*
 Nom du gestionnaire de files d'attente propriétaire des principaux

Paramètres facultatifs

-p *NomPrincipal*
 Le nom du principal à afficher.

Codes retour

- 0 Opération réussie.
- 36 Arguments incorrects fournis.
- 69 La mémoire n'est pas disponible.
- 71 Erreur inattendue.

Exemples

1. Cet exemple montre l'affichage obtenu par une commande **dspmqsqr** et concernant un gestionnaire de files d'attente qui vient d'être créé :

```

dspmqsqr -m MT02

```

Principal	Userid	Username	Alias	GroupName	GroupType
	0.1				
NOBODY	0.0				
mqm	20.255	MQM.MANAGER	n	MQM	a

La base de données des principaux contient le principal mqm, lequel renvoie vers le nom de l'utilisateur qui a créé le gestionnaire de files d'attente.

2. Cette exemple montre le résultat d'une commande **dspmqusr**, une fois que d'autres principaux ont été ajoutés par la commande **altmqusr**:

```
dspmqusr -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

Le principal mquser1 a été ajouté. Il renvoie au nom d'utilisateur Compaq MQTEST.FRED. FRED est un membre du groupe MQTEST et un membre du groupe MQM qui utilise le système d'alias SAFEGUARD.

Le principal mquser2 renvoie au nom d'utilisateur GROUP.USER01.

Commandes connexes

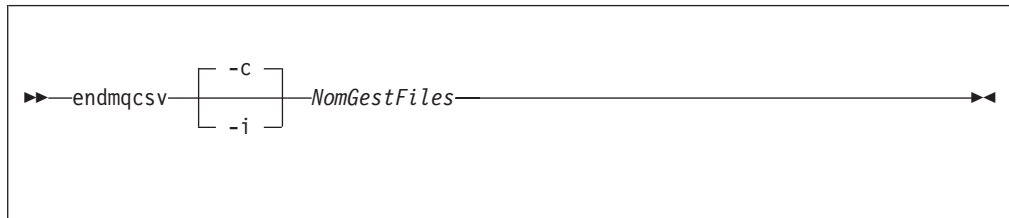
altmqusr Modification des informations utilisateur MQSeries

endmqcsv (arrêt d'un serveur de commandes)

Objet

Utilisez la commande **endmqcsv** pour arrêter le serveur de commandes du gestionnaire de files d'attente indiqué.

Syntaxe



Paramètres obligatoires

NomGestFiles

Indique le nom du gestionnaire de files d'attente dont le serveur de commandes doit être arrêté.

Paramètres facultatifs

-c Indique que le serveur de commandes doit être arrêté de façon contrôlée. Le serveur de commandes est autorisé à terminer le traitement du message de commandes en cours. Aucun nouveau message n'est lu dans la file d'attente de commandes.

Il s'agit du mode par défaut.

-i Indique que le serveur de commandes doit être arrêté immédiatement. Les actions associées à un message de commandes en cours d'exécution peuvent ne pas être terminées.

Codes retour

- 0 La commande a abouti.
- 10 La commande a abouti avec des résultats imprévus.
- 20 Une erreur s'est produite pendant le traitement.

Exemples

1. La commande suivante arrête le serveur de commandes du gestionnaire de files d'attente `saturne.gest.files` :

```
endmqcsv -c saturne.gest files
```

Avant de s'arrêter, le serveur de commandes peut terminer le traitement de toute commande préalablement lancée. Toute nouvelle commande reçue restera dans la file de commandes, sans être traitée, jusqu'au redémarrage du serveur de commandes.

2. La commande suivante arrête immédiatement le serveur de commandes du gestionnaire de files d'attente pluton :

```
endmqcsv -i "pluton"
```

Commandes connexes

strmqcsv	lancement d'un serveur de commandes
dspmqcsv	affichage du serveur de commandes

endmqm (arrêt d'un gestionnaire de files d'attente)

Objet

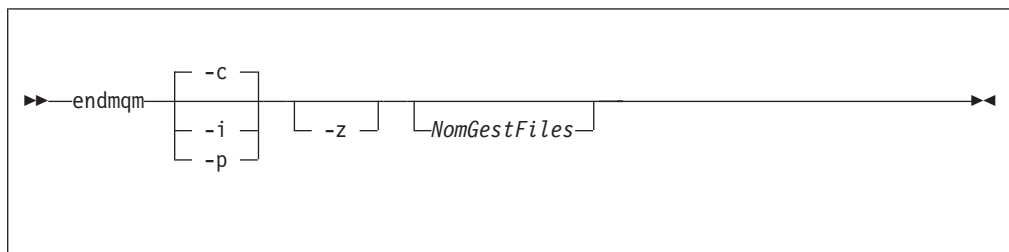
Utilisez la commande **endmqm** pour arrêter un gestionnaire local de files d'attente. Cette commande s'exécute selon l'un des trois modes suivants :

- Arrêt normal ou mise au repos
- Arrêt immédiat
- Arrêt forcé

Les attributs du gestionnaire de files d'attente et les objets associés ne sont concernés par l'arrêt. Vous pouvez redémarrer le gestionnaire de files d'attente avec la commande **strmqm** (lancement d'un gestionnaire de files d'attente).

Pour supprimer un gestionnaire de files d'attente, vous devez l'arrêter puis utiliser la commande **dltmqm** (suppression d'un gestionnaire de files d'attente).

Syntaxe



Paramètres facultatifs

NomGestFiles

Indique le nom du gestionnaire de files d'attente de messages à arrêter. Si aucun nom n'est indiqué, le gestionnaire par défaut est arrêté.

- c Arrêt contrôlé (ou mise au repos). Le gestionnaire de files d'attente ne s'arrête que lorsque toutes les applications sont déconnectées. Tous les appels MQI en cours de traitement sont achevés. Il s'agit du type d'arrêt par défaut.
- i Arrêt immédiat. Le gestionnaire de files d'attente s'arrête lorsque tous les appels MQI en cours d'exécution sont terminés. Toute demande MQI suivant l'émission de la commande échoue. Toutes les unités de travail incomplètes sont invalidées quand le gestionnaire de files d'attente redémarre.
- p Arrêt forcé.

N'utilisez ce type d'arrêt que dans des circonstances exceptionnelles (par exemple, quand un gestionnaire de files d'attente ne s'arrête pas à la suite d'une commande normale **endmqm**).

Le gestionnaire de files d'attente s'arrête sans attendre la déconnexion des applications ou la fin des appels MQI. Cette méthode peut avoir des conséquences imprévisibles sur les applications MQI. Tous les processus du gestionnaire de files d'attente qui n'ont pas réussi à s'arrêter sont clos dans les 30 secondes suivant l'émission de la commande.

- z Supprime les messages d'erreur de la commande.

Codes retour

- 0 Le gestionnaire de files d'attente est arrêté.
- 16 Le gestionnaire de files d'attente n'existe pas.
- 36 Arguments incorrects
- 40 Le gestionnaire de files d'attente n'est pas disponible.
- 69 La mémoire n'est pas disponible.
- 71 Erreur inattendue.
- 72 Le nom du gestionnaire de files d'attente est erroné.

Exemples

Les exemples suivants présentent des commandes qui permettent d'arrêter les gestionnaires de files d'attente indiqués.

1. Cette commande effectue un arrêt progressif du gestionnaire de files d'attente. Toutes les applications actuellement connectées ont la possibilité de se déconnecter.

```
endmqm
```

2. Cette commande arrête immédiatement le gestionnaire de files d'attente appelé `saturne.gest.files`. Tous les appels MQI en cours sont terminés mais aucun autre n'est autorisé.

```
endmqm -i saturne.gest files
```

Commandes connexes

- | | |
|---------------|---|
| crtmqm | Création d'un gestionnaire de files d'attente. |
| strmqm | Lancement d'un gestionnaire de files d'attente. |
| dltmqm | Suppression d'un gestionnaire de files d'attente. |

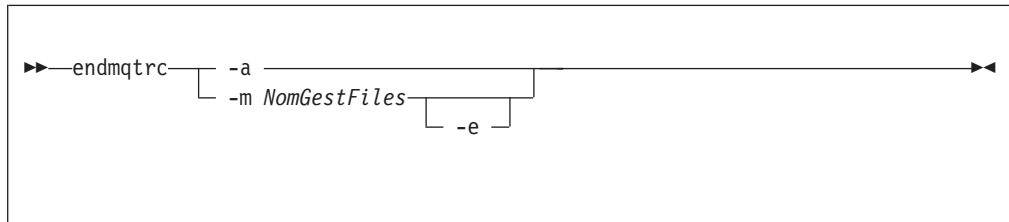
endmqtrc (arrêt de la trace MQSeries)

Objet

La commande **endmqtrc** permet d'arrêter le traçage pour le ou les éléments spécifiés.

Pour plus d'informations concernant l'utilisation de la trace MQSeries, reportez-vous à la section «Traçage de MQSeries» à la page 215.

Syntaxe



Paramètres obligatoires

-m *NomGestFiles*

Nom du gestionnaire de files d'attente dont la trace doit être arrêtée.

Un nom de gestionnaire peut être spécifié dans la même commande que l'indicateur **-e**.

-a Lorsque cet indicateur est spécifié, toute trace s'arrête.

Cet indicateur *doit* être spécifié seul.

Paramètres facultatifs

-e Lorsque cet indicateur est spécifié, la trace initiale s'arrête sur le gestionnaire spécifié.

Codes retour

AMQ5611 Ce message est généré si des arguments incorrects sont fournis dans la commande.

Exemples

Cette commande met fin au traçage des données pour un gestionnaire de files d'attente QM1.

```
endmqtrc -m GFA1
```

Commandes connexes

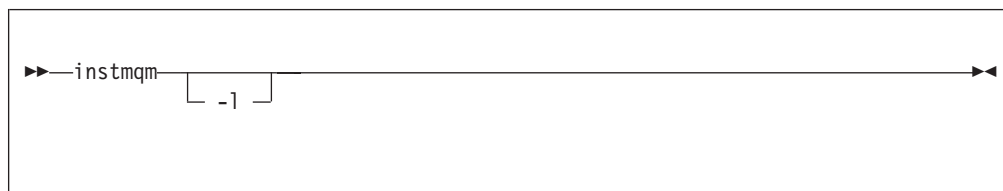
dspmqrtrc Affichage d'une sortie de trace formatée.
strmqtrc Lancement de la trace MQSeries

instmqm (installation de MQSeries pour Compaq NSK)

Objet

Utilisez la commande **instmqm** pour installer MQSeries pour Compaq NSK ou mettre à jour les informations concernant les licences.

Syntaxe



Paramètres facultatifs

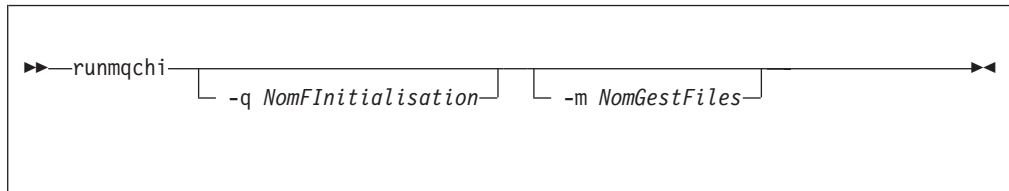
- l Appel de **instmqm** pour la mise à jour des informations concernant les licences.

runmqchi (exécution de l'initialisateur de canal)

Objet

Utilisez la commande **runmqchi** pour exécuter un processus d'initialisation de canal. Pour plus d'informations sur l'utilisation de cette commande, reportez-vous au manuel *MQSeries Intercommunication*.

Syntaxe



Paramètres facultatifs

-q *NomFinitialisation*

Indique le nom de la file d'initialisation devant être traitée par cet initiateur de canal. Par défaut, SYSTEM.CHANNEL.INITQ est utilisée.

-m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente propriétaire de la file d'initialisation. Si le nom est omis, le gestionnaire par défaut est utilisé.

Codes retour

- 0 La commande a abouti.
- 10 La commande a abouti avec des résultats imprévus.
- 20 Une erreur s'est produite pendant le traitement.

Si des erreurs se produisent avec le code retour 10 ou 20, vous devez consulter les messages du journal des erreurs du gestionnaire de files d'attente auquel le canal est associé. Vous devez également examiner le journal d'erreur \$SYSTEM car c'est là que sont consignés les incidents survenant avant que le canal soit associé au gestionnaire de files d'attente. Pour plus d'informations sur les journaux d'erreurs, reportez-vous au chapitre «Journaux d'erreurs» à la page 212.

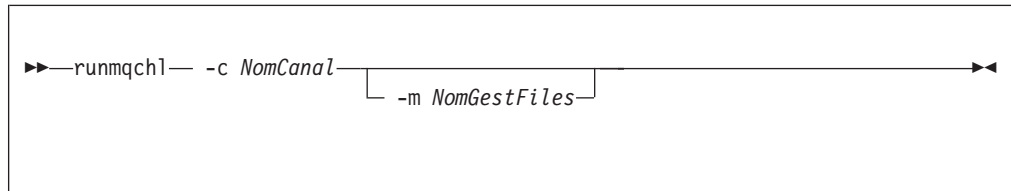
runmqchl (démarrage d'un canal)

Objet

Utilisez la commande **runmqchl** pour démarrer un canal émetteur (SDR), demandeur (RQSTR) ou complet.

Le canal fonctionne en mode asynchrone. Pour l'arrêter, envoyez la commande MQSC STOP CHANNEL.

Syntaxe



Paramètres obligatoires

-c *NomCanal*

Indique le nom du canal à démarrer.

Paramètres facultatifs

-m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente auquel ce canal est associé. Si aucun nom n'est indiqué, le gestionnaire par défaut est utilisé.

Codes retour

- 0 La commande a abouti.
- 10 La commande a abouti avec des résultats imprévus.
- 20 Une erreur s'est produite pendant le traitement.

Si le code retour 10 ou 20 est généré, consultez les messages du journal des erreurs du gestionnaire de files d'attente associé. Vous devez également examiner le journal d'erreur \$SYSTEM (situé dans ZMQSSYS) car c'est là que sont consignés les incidents survenant avant que le canal soit associé au gestionnaire de files d'attente.

runmqdlq (lancement du gestionnaire de files d'attente de rebut)

Objet

La commande **runmqdlq** sert à lancer le gestionnaire de files d'attente de rebut (DLQ). Cet utilitaire permet de gérer et de contrôler les messages d'une file d'attente de rebut.

Syntaxe



Paramètres facultatifs

NomF

Indique le nom de la file d'attente de rebut à traiter.

Si vous précisez un paramètre *NomQ*, il remplace toute valeur INPUTQ contenue dans la table de règles. Si aucun nom n'est précisé, ni en entrée de **runmqdlq** ni dans la table de règles (ou s'il s'agit d'un nom vide), la file d'attente de rebut qui est traitée est celle qui est associée au gestionnaire de files d'attente dont le nom est donné par le paramètre *QMgrName*.

NomGestFiles

Nom du gestionnaire de files d'attente propriétaire de la file à traiter.

Si vous précisez un paramètre *NomQ*, il remplace toute valeur INPUTQM contenue dans la table de règles. Si aucun nom n'est précisé, ni en entrée de **runmqdlq** ni dans la table de règles (ou s'il s'agit d'un nom vide), il est supposé que la file d'attente de rebut appartient au gestionnaire de files d'attente par défaut.

TableRègles

Le nom du fichier contenant la table de règles, laquelle doit comporter au moins une règle.

Par défaut, la commande **runmqdlq** reçoit ses données du fichier standard IN. Une fois les commandes traitées, les résultats et un résumé sont envoyés au fichier OUT standard. En redirigeant l'entrée depuis un fichier, vous pouvez appliquer une table de règles à une file d'attente donnée.

Si vous ne précisez aucune table de règles en entrée de la commande **runmqdlq**, les règles et les actions devront être précisées de façon interactive.

Dans ce cas, le gestionnaire DLQ :

- lit ses entrées à partir du clavier,
- Ne commence pas à traiter la file d'attente indiquée avant de recevoir un caractère de fin de fichier end_of_file (Ctrl-Y).

Les règles MQSC concernant les lignes de commentaire et les caractères de suite s'appliquent aussi aux paramètres d'entrée du gestionnaire DLQ.

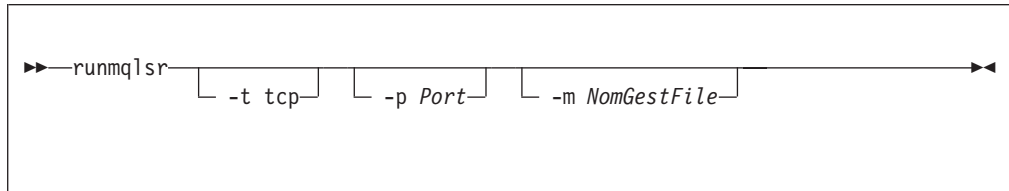
Pour plus de détails sur les tables de règles et leur élaboration, reportez-vous à la section «Table de règles du gestionnaire DLQ» à la page 162.

runmqsr (exécution du programme d'écoute)

Objet

La commande **runmqsr** lance un processus d'écoute TCP/IP.

Syntaxe



Description

Lorsque la commande **runmqsr** est lancée depuis l'invite TACL, elle ne lui rend pas le contrôle avant la fin du programme d'écoute. On dit que la commande **runmqsr** est suspendue à l'exécution (run waited).

Le retour à l'invite TACL ne se fait qu'en cas d'échec, ou si le programme d'écoute s'arrête. Si l'on arrête le terminal (TACL) avant le retour de la commande **runmqsr**, le programme d'écoute ne peut plus accéder à son terminal d'origine ou à son fichier de sortie. Il faut définir tous les PARAM (comme MQEMSEVENTS) avant de lancer une commande **runmqsr**.

Pour toutes ces raisons, il est recommandé de démarrer et arrêter le programme d'écoute depuis l'interface PATHWAY du gestionnaire de files d'attente, car elle offre davantage de contrôle.

Paramètres facultatifs

-p *Port*

Numéro de port TCP/IP. Si aucune valeur n'est indiquée, le numéro de port utilisé est celui qui est indiqué par l'entrée TCPListenerPort de la strophe TCPConfig du fichier QMINI. Sa valeur par défaut est : 1414. Si plusieurs ports d'écoute sont définis dans QMINI, c'est le premier port disponible suivant qui est utilisé.

Si le PARAM MPORTNUMBER a été défini dans l'environnement TACL, ou si une classe de serveur PATHWAY a été définie pour le programme, la commande **runmqsr** écoute sur le port indiqué dans ces définitions, au lieu de celui précisé dans le fichier QMINI.

Si aucun des ports n'est disponible (qu'il soit indiqué dans le QMINI ou par la commande **runmqsr**), la commande échoue.

-m *NomGestFiles*

Définit le nom du gestionnaire de files d'attente. Si aucun nom n'est indiqué, le gestionnaire de files d'attente par défaut de l'installation est utilisé.

-t *tcp*

Définit TCP/IP comme protocole de transmission.

Si l'instruction DEFINE =TCPIP^PROCESS^NAME a été définie dans l'environnement TACL, ou si une classe de serveur PATHWAY a été définie

runmqtsr

pour le programme, la commande **runmqtsr** utilise le processus serveur TCP/IP Guardian au lieu de celui par défaut.

Si le PARAM MQPORTNUMBER a été défini dans l'environnement TAQL, ou si une classe de serveur PATHWAY a été définie pour le programme, la commande **runmqtsr** écoute sur le port indiqué dans ces définitions, au lieu de celui précisé dans le fichier QMINI.

Il s'agit de la seule valeur possible (et celle par défaut) pour MQSeries pour Compaq NSK.

Codes retour

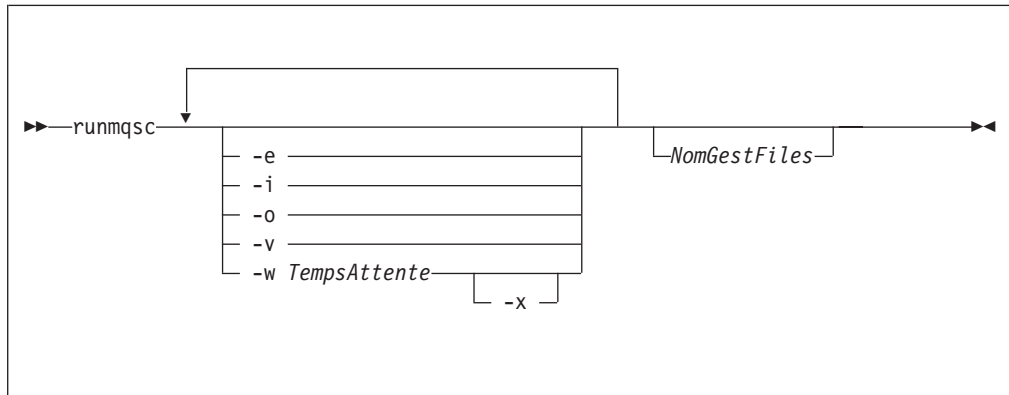
- 0** La commande a abouti.
- 10** La commande a abouti avec des résultats imprévus.
- 20** Une erreur s'est produite pendant le traitement.

runmqsc (exécution des commandes MQSeries)

Objet

Utilisez la commande **runmqsc** pour émettre des commandes MQSC vers un gestionnaire de files d'attente. Les commandes MQSC permettent d'effectuer des tâches d'administration telles que la définition, la modification ou la suppression d'un objet dans une file d'attente locale. Les commandes MQSC et leur syntaxe sont décrites dans le manuel *MQSeries - Guide de référence des commandes MQSC*.

Syntaxe



Description

Vous pouvez lancer la commande **runmqsc** selon trois modes :

mode vérification

Les commandes MQSC sont vérifiées mais pas exécutées. Un état est généré en sortie, indiquant la réussite ou l'échec de chaque commande. Ce mode n'est disponible que sur un gestionnaire de files d'attente local.

mode direct

Les commandes MQSC sont envoyées directement à un gestionnaire de files d'attente local.

mode indirect

Les commandes MQSC sont exécutées sur un gestionnaire de files d'attente éloigné. Ces commandes sont insérées dans la file d'attente des commandes d'un gestionnaire de files d'attente éloigné et exécutées dans l'ordre dans lequel elles y ont été insérées. Les états concernant les commandes sont renvoyés au gestionnaire de files d'attente local.

La commande **runmqsc** prend ses données d'entrée dans le fichier IN standard. Une fois les commandes traitées, les résultats et un résumé sont placés dans un rapport envoyé vers le fichier OUT standard.

En saisissant le fichier IN standard à partir du clavier, vous pouvez entrer les commandes MQSC de façon interactive. Pour plus de détails sur l'utilisation de la commande Fix, reportez-vous à la section «Utilisation de Compaq NSK Fix Command» à la page 30.

En redirigeant les entrées depuis un fichier, vous pouvez exécuter une séquence de commandes fréquemment utilisées contenue dans le fichier. Vous pouvez aussi rediriger l'état de sortie vers un fichier.

Remarque : Pour exécuter cette commande, votre ID utilisateur doit appartenir au groupe d'utilisateurs MQM.

Paramètres facultatifs

- e Empêche le texte source des commandes MQSC d'être copié dans un état. Cela peut être utile quand vous saisissez des commandes en mode interactif.
- i Nom du fichier d'entrée
- o Nom du fichier de sortie
- v Indique le mode vérification (vérification des commandes indiquées sans exécution). Ce mode n'est disponible qu'en local. Les indicateurs -w et -x sont ignorés s'ils sont indiqués en même temps que l'indicateur -v.

-w *TempsAttente*

Indique le mode indirect, c'est-à-dire que les commandes MQSC doivent être exécutées sur un autre gestionnaire de files d'attente. Le canal et les files d'attente de transmission nécessaires doivent être définis en conséquence. Consultez la section «Préparation des canaux et des files d'attente de transmission pour l'administration éloignée» à la page 126 pour plus d'informations.

TempsAttente

Indique la durée, en secondes, pendant laquelle **runmqsc** attend une réponse. Toute réponse reçue ultérieurement est supprimée ; néanmoins, les commandes MQSC sont toujours exécutées. Indiquez un temps compris entre 1 et 999 999 secondes.

Chaque commande est envoyée sous forme de PCF d'échappement à la file d'attente de commandes (SYSTEM.ADMIN.COMMAND.QUEUE) du gestionnaire de files d'attente cible.

Les réponses sont reçues dans la file SYSTEM.MQSC.REPLY.QUEUE et les sorties sont ajoutées à l'état. Il peut s'agir d'une file locale ou modèle.

Le fonctionnement en mode indirect est réalisé via le gestionnaire de files d'attente par défaut.

Cet indicateur est ignoré si l'indicateur -v est indiqué.

- x Indique que le gestionnaire de files d'attente cible fonctionne sous MVS/ESA. Cet indicateur ne s'applique qu'en mode indirect. L'indicateur -w doit aussi être indiqué. En mode indirect, les commandes MQSC sont écrites sous une forme compatible avec la file d'attente de commandes de MQSeries pour MVS/ESA.

NomGestFiles

Indique le nom du gestionnaire de files d'attente cible sur lequel les commandes MQSC doivent s'exécuter. Si aucun nom n'est indiqué, les commandes s'exécutent sur le gestionnaire de files d'attente par défaut.

Codes retour

- 00 Le traitement du fichier de commandes MQSC a abouti.
- 10 Le moniteur de déclenchement client a été interrompu par une erreur.
- 20 Erreur- le moniteur de déclenchement client ne s'exécute pas.

runmqsc

Exemples

1. Entrez cette commande à l'invite TACL :

```
runmqsc
```

Vous pouvez maintenant entrer des commandes MQSC directement. En l'absence d'indication de gestionnaire de files d'attente, les commandes MQSC sont traitées sur le gestionnaire par défaut.

2. L'exemple suivant montre comment indiquer que les commandes MQSC doivent seulement être vérifiées :

```
runmqsc -i $SYSTEM.CONFIG.MQSCIN -v BANK
```

Cette commande vérifie le fichier de commandes MQSC \$SYSTEM.CONFIG.MQSCIN. Le nom du gestionnaire de files d'attente est BANK. La sortie s'affiche dans la fenêtre en cours.

3. Cette commande exécute un fichier de commandes MQSC sur le gestionnaire de files d'attente appelé BANK :

```
runmqsc -i MQSCFILE -o $TEST.MQ.MQSCOUT BANK
```

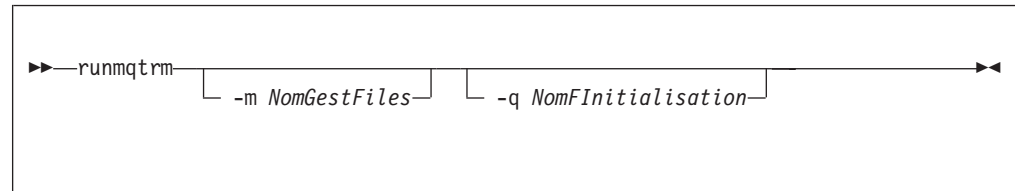
Dans cet exemple, la sortie est dirigée vers le fichier \$TEST.MQ.MQSCOUT. Le fichier d'entrée est MQSCFILE dans le sous-volume actuel.

runmqtrm (lancement du moniteur de déclenchement)

Objet

Utilisez la commande **runmqtrm** pour appeler un moniteur de déclenchement. Pour plus d'informations sur les moniteurs de déclenchement, reportez-vous au manuel *MQSeries - Guide de programmation d'applications*.

Syntaxe



Paramètres facultatifs

-m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente sur lequel doit fonctionner le moniteur de déclenchement. Si le nom est omis, le moniteur de déclenchement fonctionne sur le gestionnaire de files d'attente par défaut.

-q *NomFInitialisation*

Indique le nom de la file d'initialisation à traiter. Si le nom est omis, la file SYSTEM.DEFAULT.INITIATION.QUEUE est utilisée.

Codes retour

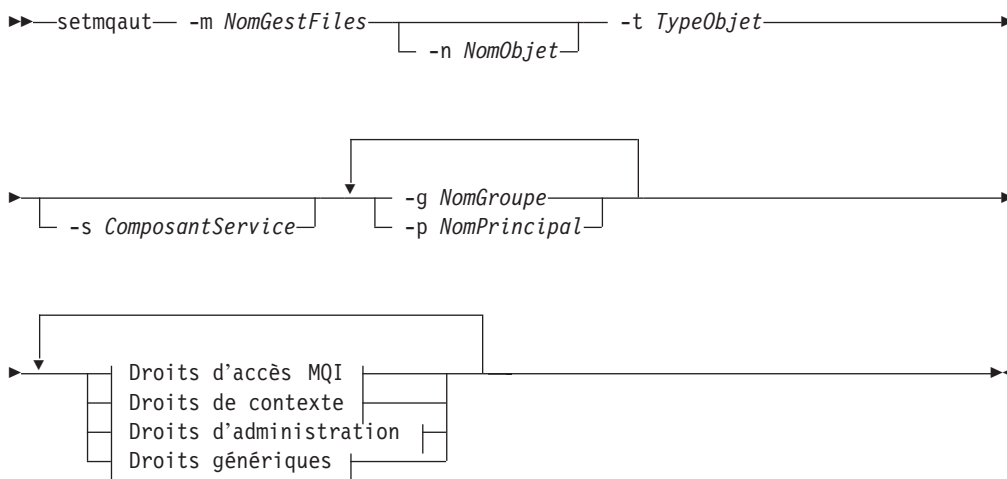
- 10 Le moniteur de déclenchement a été interrompu par une erreur.
- 20 Erreur— le moniteur de déclenchement ne s'exécute pas.

setmqaut (définition ou réinitialisation des droits d'accès)

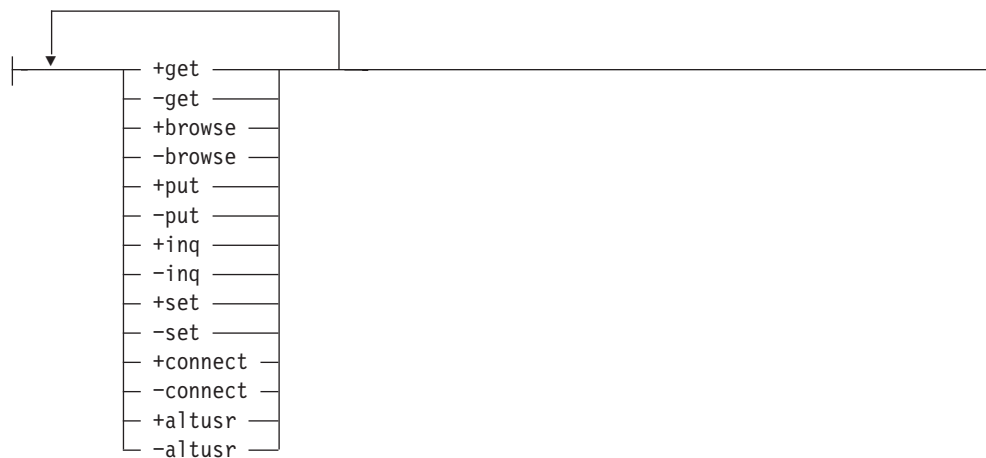
Objet

Utilisez la commande **setmqaut** pour modifier les droits d'accès à un objet ou à une classe d'objets. Les droits d'accès peuvent être accordés ou annulés pour tout nombre de principaux ou de groupes.

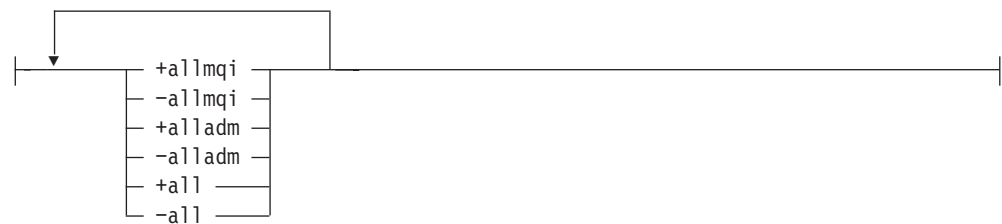
Syntaxe



Droits d'accès MQI :



Droits de contexte :

**Droits d'administration :****Droits génériques :****Description**

Vous pouvez utiliser cette commande à la fois pour *définir* un droit d'accès, c'est-à-dire donner à un groupe d'utilisateurs ou à un principal la permission d'effectuer une opération, et pour *réinitialiser* un droit d'accès, c'est-à-dire retirer la permission d'effectuer une opération. Vous devez indiquer les groupes d'utilisateurs et les principaux auxquels s'appliquent les droits d'accès ainsi que le gestionnaire de files d'attente, le type et le nom de l'objet. Vous pouvez indiquer plusieurs groupes dans une seule commande.

Les droits d'accès qui peuvent être accordés sont répartis dans les catégories suivantes :

- Droits d'émettre des appels MQI
- Droits de contexte MQI
- Droits d'émettre des commandes pour des tâches d'administration
- Droits génériques

setmqaut

Chaque droit d'accès à modifier est indiqué dans une liste de droits faisant partie de la commande. Chaque élément de la liste est une chaîne de caractères précédée de '+' ou '-'. Par exemple, si vous incluez +put dans la liste d'autorisations, vous donnez le droit d'émettre des appels MQPUT pour une file d'attente. A l'opposé, si vous incluez -put dans la liste d'autorisations, vous supprimez le droit d'émettre des appels MQPUT.

Les droits d'accès peuvent être indiqués dans n'importe quel ordre pourvu qu'il n'y ait pas de conflit. Par exemple, si vous indiquez allmqi avec set vous provoquez un conflit.

Vous pouvez indiquer autant de groupes ou de droits que nécessaire dans une seule commande.

Si un ID utilisateur est membre de plusieurs groupes, ses droits sont constitués par l'ensemble des droits de chaque groupe auquel il appartient.

Paramètres obligatoires

-g *NomGroupe*

Indique le nom du gestionnaire de files d'attente de l'objet dont les droits d'accès doivent être modifiés. Vous pouvez indiquer plusieurs noms de groupes mais chaque nom doit être précédé par l'indicateur -g.

Vous devez indiquer au moins un principal ou un groupe.

-m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente de l'objet dont les droits d'accès doivent être modifiés. Ce nom peut comporter jusqu'à 48 caractères.

-p *NomPrincipal*

Indique le nom du principal dont les droits d'accès doivent être modifiés. Vous pouvez indiquer plusieurs noms d'identificateurs de droits mais chaque nom doit être précédé de l'indicateur -p.

Vous devez indiquer au moins un principal ou un groupe.

-t *TypeObjet*

Indique le type des objets dont les droits d'accès doivent être modifiés.

Les valeurs possibles sont :

- **q** ou **queue**
- **prcs** ou **process**
- **qmgr**
- **nl** ou **namelist**

Paramètres facultatifs

-n *NomObjet*

Indique le nom de l'objet dont les droits d'accès doivent être modifiés.

Ce paramètre est obligatoire *sauf* s'il s'agit du gestionnaire de files d'attente lui-même. Vous devez préciser le nom du gestionnaire de files d'attente, de la file d'attente ou du processus sans utiliser de nom générique.

-s *ComposantService*

Ce paramètre ne s'applique que si vous utilisez des fonctions de droits d'accès optionnelles.

Si les services d'autorisation optionnels sont pris en charge, ce paramètre indique le nom du service concerné par les autorisations. Ce paramètre est

facultatif ; en son absence, la demande de vérification des droits est adressée au premier composant optionnel de la fonction.

Droits d'accès

Indique les droits à accorder ou à retirer. Chaque élément de la liste est précédé d'un '+' indiquant que le droit est accordé ou d'un '-', indiquant qu'il est retiré. Par exemple, pour accorder le droit d'émettre un appel MQPUT à partir de MQI, indiquez +put dans la liste. Pour supprimer le droit d'émettre un appel MQPUT, indiquez -put.

Le tableau 10 présente les droits qui peuvent être accordés suivant les différents types d'objets.

Tableau 10. Indication des droits d'accès pour différents types d'objets

Droits d'accès	File d'attente	Processus	GestFiles	Liste de noms
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
clr	✓			
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
put	✓			
inq	✓	✓	✓	✓
get	✓			
passall	✓			
passid	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

Droits d'accès MQI

altusr Utilisation d'un autre ID utilisateur dans un message.

Reportez-vous au manuel *MQSeries - Guide de programmation d'applications* pour plus d'informations sur les ID utilisateurs secondaires.

browse

Extraction d'un message d'une file par l'émission d'un appel MQGET avec l'option BROWSE.

connect

Connexion de l'application au gestionnaire de files d'attente indiqué par l'émission d'un appel MQCONN.

get Extraction d'un message d'une file par l'émission d'un appel MQGET.

setmqaut

- inq** Interrogation sur une file donnée par l'émission d'un appel MQINQ.
- put** Insertion d'un message dans une file d'attente donnée par l'émission d'un appel MQPUT.
- set** Définition des attributs d'une file à partir de MQI par l'émission d'un appel MQSET.

Remarque : Si vous ouvrez une file d'attente avec des options multiples, vous devez avoir un droit d'accès à chacune.

Droits de contexte

- passall** Transmission de tout le contexte dans la file indiquée. Toutes les zones de contexte sont copiées à partir de la demande d'origine.
- passid** Transmission du contexte d'identité sur la file d'attente indiquée. Le contexte d'identité est le même que celui de la demande.
- setall** Définition de tout le contexte sur la file indiquée. Utilisé par des utilitaires système spéciaux.
- setid** Définition du contexte d'identité sur la file indiquée. Utilisé par des utilitaires système spéciaux.

Droits pour les commandes

- chg** Modification des attributs de l'objet indiqué.
- clr** Effacement de la file indiquée (seulement pour la commande PCF Clear queue).
- crt** Création d'objets du type indiqué.
- dlt** Suppression de l'objet indiqué.
- dsp** Affichage des attributs de l'objet indiqué.

Droits génériques

- all** Utilisation de toutes les opérations applicables à cet objet.
- alladm** Exécution de toutes les opérations administratives applicables à cet objet.
- allmqi** Utilisation de tous les appels MQI applicables à cet objet.

Codes retour

- 0 Opération réussie.
- 36 Arguments incorrects fournis.
- 40 Le gestionnaire de files d'attente n'est pas disponible.
- 49 Le gestionnaire de files d'attente est en cours d'arrêt.
- 69 La mémoire n'est pas disponible.
- 71 Erreur inattendue.
- 72 Le nom du gestionnaire de files d'attente est erroné.
- 133 Nom d'objet inconnu.
- 145 Nom d'objet imprévu.
- 146 Nom d'objet absent.
- 147 Type d'objet absent.
- 148 Type d'objet incorrect.
- 149 Nom d'entité absent.
- 150 Spécification de droits d'accès absente.

151 Spécification de droits d'accès incorrecte.

Exemples

1. L'exemple ci-dessous présente une commande indiquant que l'objet auquel des droits d'accès ont été accordés est la file d'attente `file.orange` du gestionnaire de files d'attente `saturne.gest.files`.

```
setmqaut -m "saturne.gest.files" -n "file.orange" -t queue -g "tango"
+inq +alladm
```

Les droits d'accès ont été accordés au groupe d'utilisateurs `tango` et la liste de droits associée indique que le groupe d'utilisateurs `tango` :

- peut émettre des appels MQINQ.
 - a le droit d'effectuer toutes les opérations administratives sur cet objet.
2. Dans cet exemple, la liste des droits d'accès indique que le groupe `foxy` :
 - ne peut pas émettre d'appels à partir de MQI vers la file indiquée.
 - a le droit d'effectuer toutes les opérations administratives sur la file d'attente spécifiée.

```
setmqaut -m "saturne.gest.files" -n "file.orange" -t queue -g "foxy" -allmqi +alladm
```

Commandes connexes

`dspmqaut` Affichage des droits d'accès

strmqcsv (lancement d'un serveur de commandes)

Objet

Utilisez la commande **strmqcsv** pour lancer le serveur de commandes pour le gestionnaire de files d'attente indiqué. Cela permet à MQSeries de traiter les commandes envoyées à la file d'attente de commandes.

Syntaxe

```
▶▶—strmqcsv—NomGestFiles—————▶▶
```

Paramètres obligatoires

NomGestFiles

Indique le nom du gestionnaire de files d'attente pour lequel le serveur de commandes doit être lancé.

Codes retour

- 0 La commande a abouti.
- 10 La commande a abouti avec des résultats imprévus.
- 20 Une erreur s'est produite pendant le traitement.

Exemples

La commande suivante lance un serveur de commandes pour le gestionnaire de files d'attente terre :

```
strmqcsv "terre"
```

Commandes connexes

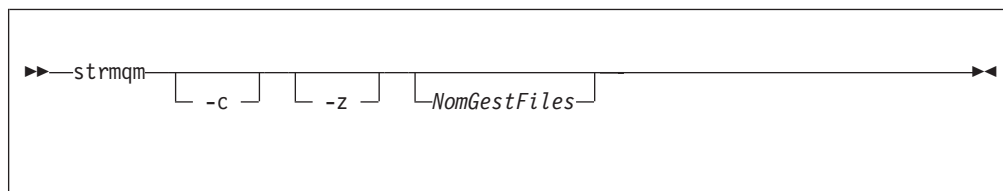
- endmqcsv** arrêt d'un serveur de commandes
- dspmqcsv** affichage du serveur de commandes

strmqm (lancement d'un gestionnaire de files d'attente)

Objet

Utilisez la commande **strmqm** pour lancer un gestionnaire de files d'attente local. Ce n'est qu'une fois que le gestionnaire de files d'attente sera disponible pour traiter des connexions ou autres requêtes que la commande **strmqm** retournera à la ligne de commande.

Syntaxe



Paramètres facultatifs

- c Démarre le gestionnaire de files d'attente, redéfinit les objets par défaut et objets système, puis arrête le gestionnaire de files d'attente. (Les objets par défaut et objets système du gestionnaire sont initialement créés à l'aide de la commande `crtmqm`). Si vous spécifiez cet indicateur, tous les objets système et objets par défaut appartenant au gestionnaire de files d'attente seront remplacés.

NomGestFiles

Indique le nom du gestionnaire de files d'attente local à lancer. Si aucun nom n'est indiqué, le gestionnaire de files d'attente par défaut est lancé.

- z Supprime les messages d'erreur.

Cet indicateur est utilisé dans MQSeries pour supprimer les messages d'erreur indésirables. Son utilisation pouvant provoquer une perte d'informations, vous ne devez pas l'utiliser dans la saisie de commandes sur une ligne de commande.

Codes retour

- 0 Le gestionnaire de files d'attente est lancé.
- 3 Le gestionnaire de files d'attente est en cours de création.
- 5 Le gestionnaire de files d'attente est en cours d'exécution.
- 16 Le gestionnaire de files d'attente n'existe pas.
- 49 Le gestionnaire de files d'attente est en cours d'arrêt.
- 69 La mémoire n'est pas disponible.
- 71 Erreur inattendue.
- 72 Le nom du gestionnaire de files d'attente est erroné.

Exemples

La commande suivante lance le gestionnaire de files d'attente `comptes` :

```
strmqm "comptes"
```

strmqm

Commandes connexes

crtmqm	Création d'un gestionnaire de files d'attente.
dltmqm	Suppression d'un gestionnaire de files d'attente.
endmqm	Arrêt d'un gestionnaire de files d'attente.

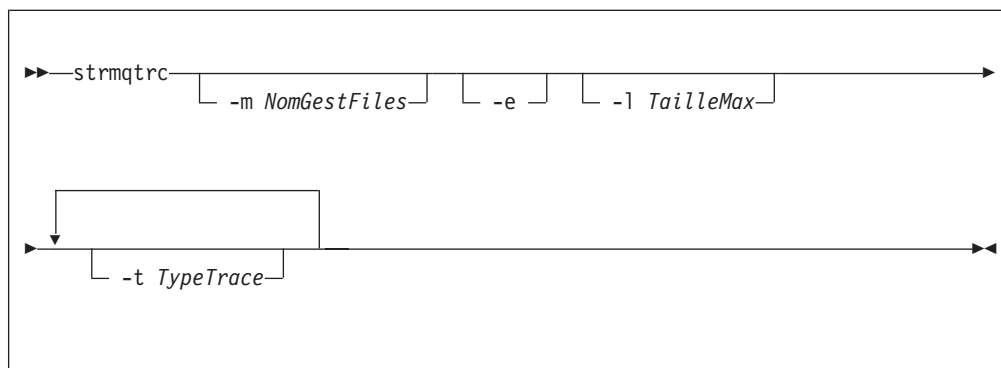
strmqtrc (lancement de la trace MQSeries)

Objet

Utilisez la commande **strmqtrc** pour activer le traçage. Elle peut s'exécuter que le traçage soit actif ou non. Si le traçage est déjà activé, les options de trace en cours deviennent celles indiquées lors du dernier appel de la commande.

Pour plus d'informations sur le traçage de MQSeries, reportez-vous à la section «Traçage de MQSeries» à la page 215.

Syntaxe



Paramètres facultatifs

-m *NomGestFiles*

Nom du gestionnaire de files d'attente dont la trace doit être exécutée. Si aucun nom n'est indiqué, le gestionnaire par défaut est utilisé.

Le gestionnaire de files d'attente indiqué ne doit pas nécessairement être en cours de fonctionnement ou même exister. Il est donc possible de tracer la création ou le démarrage d'un gestionnaire de files d'attente.

Un nom de gestionnaire peut être spécifié dans la même commande que l'indicateur **-e**. Lorsque plusieurs spécifications de trace s'appliquent à une même entité, la trace réelle comprend toutes les options spécifiées.

- e** Lorsque cet indicateur est spécifié, la trace initiale est demandée. Ceci implique que les informations de trace soient écrites avant que les processus ne sachent à quel composant MQSeries elles appartiennent. Tout processus appartenant à un composant de gestionnaire de files d'attente trace son traitement initial si cet indicateur est spécifié. Lorsque l'indicateur n'est pas précisé, par défaut, le traçage initial n'est pas effectué.

-l *TailleMax*

La valeur de *TailleMax* indique la taille maximale d'un fichier de trace (AMQnnnn.TRC) en millions d'octets. Par exemple, si *TailleMax* a pour valeur 1, la taille du fichier de trace est limitée à 1 millions d'octets.

Lorsqu'un fichier de trace atteint sa taille maximale, son nom AMQnnnn.TRC devient AMQnnnn.TRS et un nouveau fichier AMQnnnn.TRC est démarré. Tous les fichiers de trace sont relancés lorsqu'ils atteignent leur taille maximale. Si une copie antérieure d'un fichier AMQnnnn.TRS existe, elle est détruite.

strmqtrc

-t *TypeTrace*

Définit les points à tracer lors du traitement. Vous pouvez également fournir une ou plusieurs options de la liste suivante.

all	Données de sortie de chaque point de trace du système. Il s'agit également de la valeur par défaut si l'indicateur -t n'est pas précisé.
api	Données de sortie des points de trace associés au MQI et aux composants principaux du gestionnaire de files d'attente.
comms	Données de sortie des points de trace associés aux données transmises par des réseaux de communication.
csflows	Données de sortie des points de trace associés au flot de traitement des fonctions communes.
lqmfloWS	Données de sortie des points de trace associés au flot de traitement du gestionnaire de files d'attente local.
remotefloWS	Données de sortie des points de trace associés au flot de traitement du composant de communication.
otherfloWS	Données de sortie des points de trace associés au flot de traitement d'autres composants.
csdata	Données de sortie des points de trace associés aux tampons de données internes des fonctions communes.
lqmdata	Données de sortie des points de trace associés aux tampons de données internes du gestionnaire de files d'attente local.
remotedata	Données de sortie des points de trace associés aux tampons de données internes du composant de communication.
otherdata	Données de sortie des points de trace associés aux tampons de données internes d'autres composants.
versiondata	Données de sortie des points de trace associés à la version de MQSeries en cours d'exécution.
commentary	Données de sortie des points de trace associés aux commentaires des composants MQSeries.

Si cet indicateur est omis, tous les points de trace sont activés et la trace complète est générée.

Remarque : Si plusieurs types de trace sont fournis, chaque type *doit* avoir son propre indicateur -t. Vous pouvez spécifier autant d'indicateurs -t que vous le souhaitez à condition qu'un type de trace correct soit affecté à chacun.

Vous pouvez indiquer le même type de trace pour plusieurs indicateurs -t.

Codes retour

AMQ7024

Ce message est généré si des arguments incorrects sont fournis dans la commande.

AMQ8304

Neuf traces simultanées (nombre maximum de traces admis) sont déjà en cours d'exécution.

Exemples

La commande ci-dessous active le traçage des données des fonctions communes et du gestionnaire de files d'attente local, pour un gestionnaire de files d'attente appelé QM1.

```
strmqtrc -m QM1 -t csdata -t lqmdata
```

Commandes connexes

dspmqtrc	Affichage d'une sortie de trace formatée.
endmqtrc	Arrêt de la trace MQSeries

upgmqm (Mise à jour du gestionnaire de files d'attente V2.2.0.1)

Objet

Cette commande permet la mise à jour d'un gestionnaire de files d'attente Version 2.2.0.1 pour être utilisé avec MQSeries pour Compaq NSK V5.1. L'utilitaire appelé par **upgmqm** envoie des messages pour indiquer la progression au terminal d'où il a été appelé. Lorsque l'utilitaire a accompli sa tâche, le gestionnaire de files d'attente nommé est prêt à être utilisé avec MQSeries pour Compaq NSK V5.1. Les nouveaux attributs de gestionnaire de files d'attente de la Version 5.1 sont réglés sur leurs valeurs par défaut. Vous pouvez les modifier par la manipulation habituelle.

Comme la fonctionnalité du nouveau serveur de statut de la Version 5.1 diffère du serveur MQSS Version 2.2.0.1, cette mise à jour efface toutes les classes de serveurs MQS-Status nm de votre configuration de PATHWAY existante, et les remplace par une classe de serveur MQS-Status00 par défaut et une classe de serveur MQS-Queue00 par défaut. Cela signifie qu'après la mise à jour de votre gestionnaire de files d'attente, tous les objets assignés à vos processus de serveur MQSS actuels devront être réassignés à vos nouveaux processus de serveur de statut ou de file d'attente selon le cas. (Dans la Version 5.1, seules les files d'attente locales sont assignées aux serveurs de file d'attente. Tous les autres objets sont assignés aux serveurs de statut.) Il faudra peut-être aussi modifier les scripts existant faisant référence à vos classes de serveurs de statut dans le PATHWAY.

Si vous avez choisi de ne pas nettoyer au cours de la mise à jour, vous pouvez effacer les fichiers suivants à votre convenance :

Tableau 11. Exemples de fichiers pouvant être effacés après une mise à jour

Emplacement	FichiersFiles	Exemple
Le sous-volume indiqué par l'option -p sur upgmqm	Tous les fichiers	\$VOL.scratch
Sous-volume 'S' gestionnaires de file d'attente	Z*	\$VOL.myv2201S

Dans ces exemples, le gestionnaire de files d'attente se trouve sur les sous-volumes \$VOL.myv2201?.

Remarque : Si un gestionnaire de files d'attente Version 2.2.0.1 n'est pas mis à jour à l'aide de **upgmqm**, aucune commande de contrôle ne fonctionnera avec ce gestionnaire. C'est le cas de **dltmqm**, qui échouera avec FFST en essayant d'ouvrir la base de données de principaux. Un gestionnaire de files d'attente Version 2.2.0.1 devenu inutile sous la Version 5.1 doit être supprimé avant la mise à jour du code, ou mis à jour à l'aide de **upgmqm**, puis retiré.

Syntaxe

```

▶ upgmqm -m NomGestFile -v ServeurFileDéfaut -p CheminSousvol
▶ -s ServeurEtatDéfaut

```

Paramètres obligatoires

- m *NomGestFiles*
Indique le nom du gestionnaire de files d'attente pour lequel la fonction **upgmqm** doit être appliquée.
- v *ServeurFileDéfaut*
Nom de processus unique du serveur de file d'attente par défaut pour le gestionnaire de files d'attente
- p *CheminSousvol*
Chemin d'accès au sous-volume (\$VOL.SOUSVOL) pouvant être utilisé par l'utilitaire de mise à jour pour les fichiers d'exécution. Ce sous-volume **doit** être sur le même volume que le gestionnaire de files d'attente. N'indiquez que la partie du sous-volume du chemin d'accès, pas le nom du volume. La commande **upgmqm** vérifie que le sous-volume n'existe pas déjà avant de le valider.
- s *ServeurEtat Défaut*
Nom de processus unique du serveur de statut par défaut pour le gestionnaire de files d'attente.

Exemples

Cet exemple met à jour un gestionnaire de files d'attente Version 2.2.0.1 queue manager Myv2201qm avec un nom de serveur de file d'attente par défaut de \$MYQS, un nom de serveur de statut par défaut de \$MYSS, et utilise le sous-volume \$VOL.scratch pour les fichiers d'exécution (où \$VOL est le volume sur lequel se trouve le gestionnaire):

```
upgmqm -m Myv2201qm -p scratch -s $MYSS -v $MYQS
```

Partie 3. Annexes

Annexe A. MQSeries pour Compaq NSK, généralités

Programme et référence

- 5724-A39 MQSeries pour Compaq NSK, Version.1, référence 0791003

Caractéristiques matérielles requises

Vous devez au minimum disposer du matériel suivant :

- toute machine de la gamme Compaq NSK supportée par Guardian D45 ou D4x ultérieur, G06 ou G0x ultérieur,
- un matériel spécifique pour prendre en charge les protocoles de transport réseau choisis par l'utilisateur.

Il est aussi recommandé de disposer d'au moins un disque de données miroir avec l'espace requis spécifié pour l'espace d'audit TMF et la base de données MQSeries.

Caractéristiques logicielles requises

Vous devez au minimum disposer des logiciels suivants :

- Systèmes d'exploitation Compaq NSK Guardian D45 ou D4x ultérieur (matériel de la série K) et G06 ou G0x ultérieur (matériel de la série S) comprenant TM/MP (TMF), ENSCRIBE, et EMS.
- TS/MP (PATHWAY) correspondant au système d'exploitation.
- SCF pour la configuration, la commande, et le contrôle des transferts par réseaux SNA et TCP.

Connectivité SNA :

- SNAX/APC et SNAX/XF ou SNAX/APN correspondant au système d'exploitation

ou

- Insession ICE Version 3.2 ou ultérieure

Connectivité TCP/IP :

- TCP/IP correspondant au système d'exploitation.

Pour utiliser les fonctions MQSeries basées sur OSS (liaisons MQI, applications OSS, liaisons Java), il vous faut la version du produit OSS compatible avec votre système d'exploitation.

La journalisation des opérations est conservée avec le produit Compaq TM/MP (TMF).

Sécurité

MQSeries pour Compaq NSK utilise les dispositifs de sécurité du système de fichiers NSK, qui offre un contrôle d'accès au niveau du fichier à l'UTILISATEUR et au GROUPE pour les opérations de lecture, d'écriture, d'exécution et de suppression. SAFEGUARD n'est pas nécessaire pour utiliser MQSeries pour Compaq NSK; cependant, ce produit est compatible avec un environnement SAFEGUARD.

Toutes les ressources MQSeries sont détenues par un seul ID utilisateur du groupe MQM. Pour administrer MQSeries avec les menus SCOBOL ou **runmqsc**, vous devez être connecté avec un ID utilisateur assigné ou associé au groupe MQM.

Fonctions de maintenance

MQSeries fonctionne avec :

- la fonction de gestion des files d'attente des messages (MQM) en utilisant les écrans de configuration du demandeur SCOBOL dans un environnement PATHWAY,
- l'interface de ligne de commande **runmqsc**,
- l'utilitaire SCF pour la configuration, la commande et le contrôle afin de maintenir les environnements TCP/IP et SNA pour les offres de protocole de réseau Compaq,
- les utilitaires ICE fournis avec ce produit pour contrôler l'interface ICE LU 6.2.
- MQSeries Explorer (non fourni avec MQSeries pour Compaq NSK).
- Tout autre produit ou utilitaire utilisant des commandes PCF standards pour l'administration à distance.

Compatibilité

MQI pour MQSeries pour Compaq NSK V5.1 est compatible avec les applications qui fonctionnent sous la Version 2.2.0.1 de MQSeries for Tandem NonStop Kernel, en version de révision PTF U473441.

Compilateurs pris en charge

MQSeries pour Compaq NSK V5.1 est construit à l'aide de l'environnement CRE afin faire le link de tous les objets. Cette méthode impose les éléments suivants aux utilisateurs d'une version MQI antérieure à la version 2.2.0.1 :

1. Tout COBOL et objet C antérieur à D45 doit être recompilé avec le compilateur D45 ou ultérieur pour intégrer le lien CRE.
2. Tout code objet antérieur à D45 doit être recompilé avec le compilateur D45 ou ultérieur et vous devez vous assurer que le programme TAL est conforme aux considérations concernant les programmations spéciales, développées dans le manuel *Common Run-time Environment Programmer's Guide*. Pour plus d'informations sur ces considérations sur la programmation, référez-vous au manuel *TAL Programmer's Guide*.
3. Un bind séparé est fourni pour les codes objet produits par les compilateurs natifs sur D45.
4. Les programmes C doivent utiliser le modèle de mémoire WIDE (entiers sur 32 bits).
5. Les programmes COBOL doivent être conformes aux exigences du CRE.

6. Dans les programmes TAL, tous les entiers passés aux fonctions MQI doivent être sur 32 bits (ou forcés sur 32 bits avec la macro \$INT32()).

Les programmes MQSeries sont eux-même compilés et liés à l'aide des outils de mode natif pour Guardian NSK. Les applications de mode natif font normalement le link du gestionnaire de files d'attente SRL directement, à moins que l'application n'utilise déjà une SRL privée. Dans ce cas, étant donné que les applications ne peuvent utiliser qu'une seule SRL privée, l'application doit faire le link de la bibliothèque de liaison MQI statique, ou le code se trouvant dans la SRL privée de l'application doit être associé à la SRL MQSeries dans une nouvelle SRL privée.

Licences

Entrez le type de système pour définir l'habilitation du programme. Ce paramètre peut être entré lors de l'installation ou plus tard en cas d'achat d'une mise à jour de licence. Au démarrage, cette valeur est vérifiée par rapport à la configuration du poste physique Compaq. En cas d'insuffisance de la licence ou de l'habilitation du programme, un message d'avertissement s'affiche.

Choix de la langue

Un fichier texte des messages est codé dans le jeu de caractères à 7 bits du système d'exploitation Compaq NSK. MQSeries pour Compaq NSK permet de choisir le langage une fois le produit installé. La valeur par défaut est l'anglais américain.

Compatibilité internationale

Avec MQSeries pour Compaq NSK, il est possible de préciser le CCSID lors de la création du gestionnaire de files d'attente (bien que le CCSID puisse aussi être modifié après la création du gestionnaire). La valeur par défaut du CCSID du gestionnaire de files d'attente est 819. MQSeries pour Compaq NSK assure la conversion des jeux de caractères, vers le CCSID du gestionnaire de files d'attente qui a été configuré. Le manuel *MQSeries Application Programming Reference* contient davantage d'informations concernant les CCSID qui peuvent être spécifiés pour un gestionnaire de files d'attente MQSeries pour Compaq NSK, y compris ceux qui supportent le caractère euro.

Annexe B. Objets système et objets par défaut

Lors de la création d'un gestionnaire de files d'attente à l'aide de la commande de contrôle **crtmqm**, les objets système et objets par défaut sont créés automatiquement.

- Les objets système sont les objets MQSeries nécessaires au fonctionnement d'un gestionnaire de files d'attente ou d'un canal.
- Les objets par défaut définissent tous les attributs d'un objet. Lorsque vous créez un objet, tel qu'une file d'attente locale, les attributs que vous ne spécifiez pas explicitement prennent la valeur des attributs correspondants de l'objet par défaut.

Tableau 12. Objets système et objets par défaut - Files d'attente

Nom d'objet	Description
SYSTEM.DEFAULT.ALIAS.QUEUE	File d'attente alias par défaut.
SYSTEM.DEFAULT.LOCAL.QUEUE	File d'attente locale par défaut.
SYSTEM.DEFAULT.MODEL.QUEUE	File d'attente modèle par défaut.
SYSTEM.DEFAULT.REMOTE.QUEUE	File d'attente éloignée par défaut.
SYSTEM.DEAD.LETTER.QUEUE	File d'attente de rebut (des messages non livrés).
SYSTEM.DEFAULT.INITIATION.QUEUE	File d'attente d'initialisation par défaut.
SYSTEM.CICS.INITIATION.QUEUE	File d'attente d'initialisation CICS* par défaut.
SYSTEM.ADMIN.COMMAND.QUEUE	File d'attente de commandes d'administration. Utilisée pour les commandes MQSC éloignées et les commandes PCF.
SYSTEM.MQSC.REPLY.QUEUE	File d'attente de réponses MQSC. Il s'agit d'une file d'attente modèle qui crée une file d'attente dynamique temporaire pour les réponses aux commandes MQSC éloignées.
SYSTEM.ADMIN.QMGR.EVENT	File d'attente des événements Gestionnaire de files d'attente.
SYSTEM.ADMIN.PERFM.EVENT	File d'attente des événements Performances.
SYSTEM.ADMIN.CHANNEL.EVENT	File d'attente des événements Performances.
SYSTEM.CHANNEL.INITQ	File d'attente d'initialisation du canal.
SYSTEM.CHANNEL.SYNCQ	File d'attente contenant les données de synchronisation des canaux. (Cet objet est créé mais n'est pas utilisé dans MQSeries pour Compaq NSK. Les informations de synchronisation des canaux sont stockées dans des bases de données protégées TM/MP).
SYSTEM.CLUSTER.COMMAND.QUEUE	File d'attente servant à transmettre les messages au gestionnaire de files d'attente de référentiel.
SYSTEM.CLUSTER.REPOSITORY.QUEUE	File d'attente qui contient toutes les données du référentiel.

Objets système et objets par défaut

Tableau 12. Objets système et objets par défaut - Files d'attente (suite)

Nom d'objet	Description
SYSTEM.CLUSTER.TRANSMIT.QUEUE	File d'attente de transmission de tous les messages destinés aux clusters.

Tableau 13. Objets système et objets par défaut - Canaux

Nom d'objet	Description
SYSTEM.DEF.SENDER	Canal émetteur par défaut.
SYSTEM.DEF.SERVER	Canal serveur par défaut.
SYSTEM.DEF.RECEIVER	Canal récepteur par défaut.
SYSTEM.DEF.REQUESTER	Canal demandeur par défaut.
SYSTEM.DEF.SVRCONN	Canal de connexion serveur par défaut.
SYSTEM.DEF.CLNTCONN	Canal de connexion client par défaut.
SYSTEM.AUTO.RECEIVER	Canal récepteur dynamique.
SYSTEM.AUTO.SVRCONN	Canal de connexion serveur dynamique.
SYSTEM.DEF.CLUSRCVR	Canal récepteur par défaut qui fournit les valeurs par défaut pour les attributs non définis lors de la création d'un canal CLUSRCVR sur l'un des gestionnaires de files d'attente du cluster.
SYSTEM.DEF.CLUSSDR	Canal émetteur par défaut qui fournit les valeurs par défaut pour les attributs non définis lors de la création d'un canal CLUSSDR sur l'un des gestionnaires de files d'attente du cluster.

Tableau 14. Objets système et objets par défaut - Listes de noms

Nom d'objet	Description
SYSTEM.DEFAULT.NAMELIST	Liste de noms par défaut

Tableau 15. Objets système et objets par défaut - Processus

Nom d'objet	Description
SYSTEM.DEFAULT.PROCESS	Définition de processus par défaut.

Annexe C. Définition de variables d'environnement TACL pour MQSeries pour Compaq NSK

MQSeries crée et utilise un certain nombre de variables d'environnement ou PARAMs Compaq NSK. En définissant ces PARAMs, pensez aux points suivants :

- Le MQDEFAULTPREFIX PARAM doit être présent dans l'environnement de tous les programmes. La configuration TS/MP (Pathway) faite automatiquement par la commande **crtmqm** garantit que ces PARAMs soient correctement définis pour tous processus de serveur de gestionnaire de files d'attente. Les utilisateurs des applications MQSeries et des commandes de contrôle doivent s'assurer que les configurations TACL et TS/MP indiquent aussi ces variables.
- Il est recommandé d'inclure les instructions PARAM dans les fichiers de telle sorte que, lors de la connexion, ces PARAMs soient créés correctement et que tout programme exécuté depuis TACL hérite des bonnes valeurs. Les variables d'environnement suivantes sont également à modifier pour valider l'emplacement des exécutables MQSeries :

```
#SET #PMSEARCH $SYSTEM.ZMQSEXE [#PMSEARCH]
```

MQCONNECTTYPE

Ce PARAM, s'il est présent, sert à désactiver la fonction permettant aux applications d'utiliser les connexions FASTPATH. S'il est réglé sur STANDARD, les applications ne peuvent utiliser que les connexions STANDARD, même si elles demandent un FASTPATH. Toute autre valeur est traitée comme si PARAM n'était pas spécifié (les applications ont donc la possibilité d'utiliser les connexions FASTPATH).

MQDEFAULTPREFIX

Nom du volume contenant le sous-volume ZMQSSYS. Ce PARAM doit être correctement défini dans tous les environnements.

Par exemple :

```
PARAM MQDEFAULTPREFIX $data00
```

MQEMSEVENTS

Ce PARAM active les événements EMS MQSeries. Par exemple, pour activer tous les événements EMS pour MQSeries, réglez les PARAM MQEMSEVENTS comme suit :

```
PARAM MQEMSEVENTS 127
```

MQMACHINIFILE

Emplacement du fichier MQSINI pour l'installation. La valeur par défaut est *MQDEFAULTPREFIX.ZMQSSYS.MQSINI*. Ce PARAM n'est nécessaire que si un emplacement personnalisé est demandé.

Variables d'environnement TACL

Par exemple :

```
PARAM MQMACHINIFILE $data00.altinst.mqsini
```

MQRDF

Si ce PARAM est activé, MQSeries modifie l'exécution de l'opération suppression pour travailler avec le RDF pour les fichiers contrôlés. Si ce PARAM n'est pas défini ou est réglé sur une valeur autre que ON, la fonction suppression de MQSeries fonctionne normalement. S'il est utilisé, ce PARAM doit être réglé dans l'environnement TACL de tout utilisateur exécutant des programmes d'administration, et dans l'environnement de toutes les classes de serveurs TS/MP configurées dans le gestionnaire de files d'attente.

MQRDFUPPROCESSNAME

Ce PARAM n'est interprété que par l'utilitaire **cleanrdf**. Il est utilisé pour définir un nom de processus Guardian qui sera assigné au processus de serveur FUP créé par l'utilitaire **cleanrdf**. Si ce PARAM n'est pas défini, le nom de processus du serveur FUP est assigné par le système d'exploitation.

MQRDFUPPROGNAME

Ce PARAM n'est interprété que par l'utilitaire **cleanrdf**. Il sert à indiquer le nom complet du fichier exécutable FUP à utiliser par l'utilitaire. La valeur par défaut est <nom système défaut>.\$SYSTEM.SYS00.FUP.

MQSNOAUT

Si ce PARAM est réglé sur 1 durant l'exécution de **crtmqm**, le nouveau gestionnaire de files d'attente est créé avec l'OAM désactivé.

Par exemple :

```
PARAM MQSNOAUT 1
```

MQLISTENPORTNUM

Si ce PARAM est configuré, le processus de programme d'écoute TCP/IP l'utilise pour déterminer sur quel port écouter. Dans le cas contraire, le port est déterminé à partir de la strophe TCP/IP du fichier QMINI.

SAVE-ENVIRONMENT ON

Nécessaire lors de l'exécution de programmes d'application pour garantir que l'environnement CRE passe des PARAM de l'environnement au programme d'application.

Par exemple :

```
PARAM SAVE-ENVIRONMENT ON
```

Si ce PARAM n'est pas configuré, les applications reçoivent des codes de retour 2058, indiquant une erreur de nom de gestionnaire de files d'attente.

Ce PARAM n'est nécessaire que pour les applications TAL ou COBOL exécutées comme programmes non natifs.

Paramètres d'adaptation du serveur de file d'attente

Les PARAMs suivants, s'ils sont définis dans l'environnement TACL d'un serveur de file d'attente, peuvent être utilisés pour remplacer les valeurs par défaut d'origine du serveur de file d'attente pour diverses opérations de gestion interne :

MQQSHKEEPINT

Si ce PARAM est configuré, une valeur en secondes peut être définie pour remplacer l'intervalle de gestion interne par défaut (60s) du serveur de file d'attente. L'intervalle de gestion interne contrôle la fréquence à laquelle le serveur de file d'attente inspecte les files d'attente pour repérer les messages expirés, et examine son utilisation de mémoire pour optimiser les opérations.

MQQSSIGTIMEOUT

Si ce PARAM est configuré, une valeur en secondes peut être définie pour remplacer le délai d'expiration par défaut (60s) de l'émission d'un signal IPC vers une application qui a lancé un MQGET à l'aide de l'option MQGMO_SET_SIGNAL. Si un serveur de file d'attente ne parvient pas à émettre un signal avant l'expiration de ce délai (une fois les conditions d'émission du message réunies), le serveur répertorie cet échec et annule le signal.

MQQSMAXBATCHEXP

Si ce PARAM est configuré, une valeur peut être définie pour remplacer le nombre maximum de messages persistants expirés (100) qui seront éliminés au cours d'une opération de gestion interne par le serveur de file d'attente. Lors de l'expiration de messages persistants, ils doivent être supprimés physiquement de la base de données de file d'attente, ce qui nécessite une transaction TM/MP interne. Ce PARAM permet de contrôler le nombre maximum de messages effacés au cours d'une transaction TM/MP.

MQQSMAXMSGSEXPIRE

Si ce PARAM est configuré, une valeur peut être définie pour remplacer le nombre maximum par défaut de messages expirés (300) qui seront détectés lors d'une opération de gestion interne du serveur de file d'attente.

Variables d'environnement TACL

Annexe D. Comparaison des jeux de commandes

Les tableaux suivants comparent les fonctions disponibles depuis les différents jeux de commandes d'administration :

Remarque : Seules les commandes MQSC applicables à MQSeries pour Compaq NSK sont indiquées.

Tableau 16. Commandes d'administration des gestionnaires de files d'attente

PCF	MQSC	Commande de contrôle
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
Remarque : * Commande non disponible sous PCF.		

Tableau 17. Commandes d'administration des serveurs de commandes

Description	Commande de contrôle
Affichage du serveur de commandes	dspmqcsv
Arrêt du serveur de commandes	endmqcsv
Lancement du serveur de commandes	strmqcsv
Remarque : Au lieu des commandes de contrôle, vous pouvez utiliser les commandes PATHCOM décrites dans la section «Administration TS/MP (PATHWAY)» à la page 32. Il n'existe pas de commandes MQSC ou PCF équivalentes.	

Comparaison des jeux de commandes

Tableau 18. Commandes d'administration des files d'attente

PCF	MQSC
Change Queue	ALTER QLOCAL ALTER QALIAS ALTER QMODEL ALTER QREMOTE
Clear Queue	CLEAR QLOCAL
Copy Queue	DEFINE QLOCAL(x) LIKE(y) DEFINE QALIAS(x) LIKE(y) DEFINE QMODEL(x) LIKE(y) DEFINE QREMOTE(x) LIKE(y)
Create Queue	DEFINE QLOCAL DEFINE QALIAS DEFINE QMODEL DEFINE QREMOTE
Delete Queue	DELETE QLOCAL DELETE QALIAS DELETE QMODEL DELETE QREMOTE
Inquire Queue	DISPLAY QUEUE
Inquire Queue Names	DISPLAY QUEUE
Remarque : Il n'existe pas de commandes de contrôle équivalentes dans ce groupe.	

Tableau 19. Commandes d'administration des processus

PCF	MQSC
Change Process	ALTER PROCESS
Copy Process	DEFINE PROCESS(x) LIKE(y)
Create Process	DEFINE PROCESS
Delete Process	DELETE PROCESS
Inquire Process	DISPLAY PROCESS
Inquire Process Names	DISPLAY PROCESS
Remarque : Il n'existe pas de commandes de contrôle équivalentes dans ce groupe.	

Comparaison des jeux de commandes

Tableau 20. Commandes d'administration des canaux

PCF	MQSC	Commande de contrôle
Change Channel	ALTER CHANNEL	–
Copy Channel	DEFINE CHANNEL(x) LIKE(y)	–
Create Channel	DEFINE CHANNEL	–
Delete Channel	DELETE CHANNEL	–
Inquire Channel	DISPLAY CHANNEL	–
Inquire Channel Names	DISPLAY CHANNEL	–
Inquire Channel Status	DISPLAY CHSTATUS	–
Ping Channel	PING CHANNEL	–
Reset Channel	RESET CHANNEL	–
Resolve Channel	RESOLVE CHANNEL	–
Start Channel	START CHANNEL	runmqchl
Start Channel Initiator	–	runmqchi
Start Channel Listener	–	runmqslr
Stop Channel	STOP CHANNEL	–
<p>Remarque : Sous MQSeries pour Compaq NSK, utilisez TS/MP ou la commande de contrôle runmqslr pour démarrer les programmes d'écoute de canal TCP/IP. Pour plus d'informations, reportez vous aux sections «Détermination et contrôle des programmes d'écoute TCP/IP» à la page 32 et «runmqslr (exécution du programme d'écoute)» à la page 294.</p>		

Tableau 21. Autres commandes de contrôle

Description	Commande de contrôle
Modifier le volume de file d'attente et les options de stockage de messages de serveur de file d'attente	altnmqfls
Ajouter, supprimer ou modifier des principaux MQSeries	altnmqsr
Utilitaire de gestion interne RDF	cleanrdf
Conversion de la table de définition du canal client	cvclchl
Création de l'exit de conversion MQSeries	crtmqcvx
Affichage des droits d'accès	dspmquat
Affichage des fichiers utilisés par les objets ; options de stockage de messages de serveur de file configurées pour un objet	dspmqlfs
Affichage d'une sortie de trace MQSeries formatée	dspmqtrc
Affichage des principaux de MQSeries	dspmqusr
Arrêt de la trace MQSeries	endmqtrc
Installation de MQSeries pour Compaq NSK	instmqm
Exécution du gestionnaire de file d'attente de rebut	runmqdlq
Exécution de commandes MQSC	runmqsc
Lancement du moniteur de déclenchement	runmqtrm
Définition ou réinitialisation des droits d'accès	setmqaut
Lancement de la trace MQSeries	strmqtrc

Comparaison des jeux de commandes

Tableau 21. Autres commandes de contrôle (suite)

Description	Commande de contrôle
Mise à jour du gestionnaire de file d'attente V2201	upgmqm
Remarque : Au lieu de la commande de contrôle runmqtrm , vous pouvez utiliser les commandes PATHCOM décrites dans la section «Administration TS/MP (PATHWAY)» à la page 32. Il n'existe pas de commande MQSC ou PCF équivalente.	

Annexe E. Arrêt et suppression manuels de gestionnaires de file d'attente

Si les méthodes habituelles d'arrêt et de suppression de gestionnaire de file d'attente ne fonctionnent pas, utilisez les méthodes décrites ci-dessous.

Arrêt manuel d'un gestionnaire de files d'attente

La méthode habituelle de fermeture des gestionnaires de file d'attente, à l'aide de la commande **endmqm**, devrait fonctionner même en cas d'échecs au sein du gestionnaire. Exceptionnellement, en cas d'échec de cette méthode, utilisez la procédure suivante pour l'arrêter manuellement :

1. Recherchez les ID processus des programmes des gestionnaires de files d'attente qui sont encore en fonctionnement.
2. FUP LISTOPENS sur le fichier TRACEOPT dans le sous-volume de données du gestionnaire de files d'attente fournit les numéros de processeur et les PIN des processus appartenant au gestionnaire de files d'attente.
3. Fermez les processus des gestionnaires de files d'attente encore en fonctionnement. Utilisez la commande **STOP**, ainsi que les ID processus découverts au cours de l'étape précédente.

Fermez les processus dans l'ordre suivant :

- a. MQECBOSS – EC Boss
- b. MQEC – ECs
- c. Tous autres processus encore ouverts

Remarque : L'arrêt manuel du gestionnaire de files d'attente risque d'entraîner le déclenchement de FFST et la production de fichiers FD. Ce n'est pas un défaut du gestionnaire de files d'attente.

Le gestionnaire de files d'attente devrait redémarrer normalement, même après un arrêt manuel.

Pour supprimer le gestionnaire de files d'attente après un arrêt manuel, utilisez normalement la commande **dltmqm**. Si jamais cette commande ne parvient pas à supprimer le gestionnaire de files d'attente, vous pouvez utiliser la procédure manuelle expliquée dans la section «Suppression manuelle de gestionnaires de file d'attente».

Suppression manuelle de gestionnaires de file d'attente

Pour supprimer les gestionnaires de file d'attente manuellement :

1. Assurez-vous qu'aucun processus de gestionnaire de files d'attente n'est en cours de fonctionnement pour le gestionnaire que vous souhaitez supprimer.
2. Ouvrez le fichier MQSINI et supprimez la strophe du gestionnaire de files d'attente et si nécessaire, modifiez la strophe du gestionnaire de files d'attente par défaut. Notez l'emplacement des fichiers du gestionnaire de files d'attente avant de supprimer la strophe.
3. Supprimez tous les fichiers de tous les sous-volumes du gestionnaire de files d'attente à l'aide de la commande FUP PURGE. Par exemple, FUP PURGE \$VOL.QMSVOL*.*.

Suppression manuelle de gestionnaires de file d'attente

Annexe F. MQSeries et Compaq NonStop Server pour Java

MQSeries pour Compaq NSK est compatible avec Compaq NonStop Server pour Java, Version 1.5 et plus. Le produit prend en charge l'ensemble complet de classes Java MQSeries. Le manuel *MQSeries - Utilisation deJava* décrit ces fonctions en détail.

MQSeries peut fonctionner de deux façons avec NonStop Server pour Java :

- En tant que Servlet exécuté dans le contexte de Compaq iTP Webserver
- En tant qu'application Java, exécutée directement depuis la ligne de commande

Le manuel *Compaq NonStop Server for Java 1.5 Reference* donne des informations sur l'utilisation de Servlets.

Pour accéder à MQSeries depuis Java, en utilisant des Servlets ou applications, il est nécessaire de créer une version personnalisée de NonStop JVM en faisant un link avec les bibliothèques de produits MQSeries suivantes :

- La Product SRL MQSRLLIB, du sous-volume Guardian ZMQSEXEXE
- L'archive de liaison Java libMQSESSION.a dans /opt/mqm/lib

La documentation Java de Compaq fournit des informations sur l'édition de liens de JVM. Le répertoire d'exemples MQSeries (/opt/mqm/samp) contient deux exemples de fichiers make. :

MakeJVM.smp

Il s'agit d'un exemple de fichier make indiquant comment reconstruire le JVM NonStop pour accéder à MQSeries. Pour utiliser ce fichier make, modifiez-le de façon à ce qu'il indique les emplacements d'installation de vos bibliothèques de produits MQSeries.

MakeJava.smp

Exemple de fichier make servant à assembler une application Java.

Prise en charge des transactions

Le langage Java ne permet pas un accès direct à l'interface TM/MP. MQSeries prend en charge les transactions en java par l'intermédiaire de JTS *Current Class*. Le tableau ci-dessous décrit l'interface *Current*.

Tableau 22. Interface Java

Fonction	Description
begin()	Démarrer une nouvelle transaction et lui associer l'unité d'exécution appelante.
commit(boolean)	Valider la transaction associée à l'unité d'exécution appelante.
get_control()	Obtenir un objet de contrôle représentant la transaction associée à l'unité d'exécution appelante.
get_status()	Connaître le statut de la transaction associée à l'unité d'exécution appelante.
get_transaction_name()	Connaître le nom descriptif de la transaction associée à l'unité d'exécution appelante.

Tableau 22. Interface Java (suite)

Fonction	Description
resume(ControlRef)	Définir ou reprendre l'association d'une transaction à l'unité d'exécution appelante.
rollback()	Annuler la transaction associée à l'unité d'exécution appelante.
suspend()	Interrompre l'association de l'unité d'exécution appelante et du contexte d'une transaction.

Pour plus d'informations sur JTS, consultez la documentation Compaq NSK NonStop Java ou la documentation JDK.

Annexe G. Commandes MQSC prises en charge par MQSeries pour Compaq NSK

Cette annexe présente les commandes MQSeries (MQSC) prises en charge par MQSeries pour Compaq NSK. Vous trouverez davantage d'informations sur la syntaxe de ces commandes dans le manuel *MQSeries - Guide de référence des commandes MQSC*.

Tableau 23. Commandes MQSC prises en charge par MQSeries pour Compaq NSK

Commande	Description	Par rapport au manuel <i>Guide de référence des commandes MQSeries</i>
ALTER CHANNEL	Modifie les attributs de canal.	Avec les exceptions décrites à la section «CONNAMES» à la page 338.
ALTER NAMELIST	Modifie une liste de noms.	Oui
ALTER PROCESS	Modifie les attributs de processus.	Oui
ALTER QALIAS	Modifie les attributs d'une file d'attente alias.	Oui
ALTER QLOCAL	Modifie les attributs d'une file d'attente locale.	Avec les exceptions décrites à la section «HARDENBO et NOHARDENBO» à la page 338.
ALTER QMGR	Modifie les attributs de gestionnaire de files d'attente.	Avec les exceptions décrites à la section «MAXUMSGS et MAXHANDS» à la page 338.
ALTER QMODEL	Modifie les attributs d'une file d'attente modèle.	Avec les exceptions décrites à la section «HARDENBO et NOHARDENBO» à la page 338.
ALTER QREMOTE	Modifie les attributs de la définition locale d'une file d'attente éloignée, d'un alias de gestionnaire de files d'attente ou d'un alias de file d'attente de réponse.	Oui
CLEAR QLOCAL	Supprime les messages d'une file d'attente locale.	Oui
DEFINE CHANNEL	Crée une définition de canal.	Avec les exceptions décrites à la section «CONNAMES» à la page 338.
DEFINE NAMELIST	Définit une liste de noms.	Oui
DEFINE PROCESS	Crée une définition de processus.	Oui
DEFINE QALIAS	Crée une définition de file d'attente alias.	Oui

Commandes MQSeries

Tableau 23. Commandes MQSC prises en charge par MQSeries pour Compaq NSK (suite)

Commande	Description	Par rapport au manuel <i>Guide de référence des commandes MQSeries</i>
DEFINE QLOCAL	Crée une définition de file d'attente locale.	Avec les exceptions décrites à la section «HARDENBO et NOHARDENBO» à la page 338.
DEFINE QMODEL	Crée une définition de file d'attente modèle.	Avec les exceptions décrites à la section «HARDENBO et NOHARDENBO» à la page 338.
DEFINE QREMOTE	Crée une définition locale d'une file d'attente éloignée, d'un alias de gestionnaire de files d'attente ou d'un alias de file d'attente de réponses.	Oui
DELETE CHANNEL	Supprime une définition de canal.	Oui
DELETE NAMELIST	Supprime une liste de noms.	Oui
DELETE PROCESS	Supprime une définition de processus.	Oui
DELETE QALIAS	Supprime une définition de file d'attente alias.	Oui
DELETE QLOCAL	Supprime une définition de file d'attente locale.	Oui
DELETE QMODEL	Supprime une définition de file d'attente modèle.	Oui
DELETE QREMOTE	Supprime une définition de file d'attente éloignée.	Oui
DISPLAY CHANNEL	Affiche une définition de canal.	Avec les exceptions décrites à la section «CONNNAME» à la page 338.
DISPLAY CHSTATUS	Affiche le statut d'un ou plusieurs canaux.	Avec les exceptions décrites aux sections «Informations d'état de canal (DISPLAY CHSTATUS)» à la page 337 et «CONNNAME» à la page 338.
DISPLAY CLUSQMGR	Affiche le statut d'un ou plusieurs canaux.	Avec les exceptions décrites à la section «CONNNAME» à la page 338.
DISPLAY NAMELIST	Affiche une liste de noms.	Oui
DISPLAY PROCESS	Affiche une définition de processus.	Oui
DISPLAY QMGR	Affiche les attributs du gestionnaire de files d'attente.	Avec les exceptions décrites à la section «MAXUMSGS et MAXHANDS» à la page 338.
DISPLAY QUEUE	Affiche les attributs de la file d'attente.	Avec les exceptions décrites à la section «HARDENBO et NOHARDENBO» à la page 338.
PING CHANNEL	Teste un canal.	Oui

Tableau 23. Commandes MQSC prises en charge par MQSeries pour Compaq NSK (suite)

Commande	Description	Par rapport au manuel <i>Guide de référence des commandes MQSeries</i>
PING QMGR	Teste si le gestionnaire de files d'attente réagit aux commandes.	Oui
REFRESH CLUSTER	Elimine toutes les informations détenues localement par le cluster, et impose leur reconstruction.	Oui
RESET CHANNEL	Remet à zéro le numéro de séquence de message pour un canal.	Oui
RESET CLUSTER	Effectue des opérations spéciales sur les clusters.	Oui
RESOLVE CHANNEL	Gère les messages en attente de validation sur un canal serveur ou émetteur.	Oui
RESUME QMGR	Informe les autres gestionnaires de file d'attente d'un cluster que le gestionnaire de file d'attente local est à nouveau disponible et peut recevoir des messages.	Oui
START CHANNEL	Démarre un canal.	Oui
STOP CHANNEL	Arrête un canal.	Oui
SUSPEND QMGR	Informe les autres gestionnaires de file d'attente d'un cluster que le gestionnaire de file d'attente local n'est plus disponible et ne peut recevoir de messages.	Oui

Si vous intégrez des commandes MQSC dans un script, vous devez limiter le nombre de caractères par ligne à 72.

Attributs de MQSC

Cette section présente des informations sur les attributs de MQSC spécifiques à MQSeries pour Compaq NSK.

Informations d'état de canal (DISPLAY CHSTATUS)

La commande DISPLAY CHSTATUS est implémentée de façon identique à la description du manuel *MQSeries - Guide de référence des commandes MQSC*, excepté que le statut de canal n'est pas mis à jour pendant un traitement par lots. Ces informations ne sont pas mises à jour à chaque transfert de message à cause de l'impact possible sur les performances des canaux. Ce qui signifie que les valeurs des données de statut communes sont identiques pour les ensembles courants et enregistrés.

DISPLAY CHSTATUS

MAXUMSGS et MAXHANDS

Les attributs d'objet MAXUMSGS et MAXHANDS sont ignorés. Ce qui a un impact sur les commandes suivantes :

```
ALTER QMGR  
DISPLAY QMGR
```

HARDENBO et NOHARDENBO

Dans MQSeries pour Compaq NSK, les attributs de files d'attente modèles ou locales HARDENBO et NOHARDENBO sont ignorés. La variable *Backoutcount* d'un message est toujours sauvegardée pour les messages persistants, jamais pour les messages non persistants. Ce qui a un impact sur les commandes suivantes :

```
ALTER QLOCAL  
ALTER QMODEL  
DEFINE QLOCAL  
DEFINE QMODEL  
DISPLAY QUEUE
```

CONNAME

L'attribut CONNAME des canaux TCP peut recevoir un champ supplémentaire optionnel au début de sa valeur, précisant le nom du processus de Serveur TCP/IP Guardian utilisé pour ce canal. Ce qui a un impact sur les commandes suivantes :

```
ALTER CHANNEL  
DEFINE CHANNEL  
DISPLAY CHANNEL  
DISPLAY CHSTATUS  
DISPLAY CLUSQMGR
```

Attribut USERDATA des applications déclenchées

Les données envoyées au moniteur de déclenchement via l'attribut USERDATA des commandes MQSC DEFINE PROCESS ou ALTER PROCESS doivent être entre guillemets s'il s'agit d'une chaîne contenant des espaces. Par exemple, si l'attribut USERDATA -o \$DISK.VOLUME.PROGRAM -d 1 doit être passé au moniteur de déclenchement, il doit être envoyé à l'entrée de MQSC entre guillemets, comme ci-dessous :

```
' -o $DISK.VOLUME.PROGRAM -d 1 '
```

Si vous affichez la définition de processus via MQSC, elle se présente ainsi :

```
"-o $DISK.VOLUME.PROGRAM -d 1"
```

Utilisation de noms d'exit comme attributs d'objets

Chaque fois que les noms d'exit sont précisés dans les attributs d'un objet, ils doivent être dans le format particulier à MQSeries pour Compaq NSK.

Annexe H. Guide de référence de programmation des applications

Les sections qui suivent sont nouvelles pour MQSeries pour Compaq NSK :
utilisez-les avec le manuel *MQSeries Application Programming Reference*.

Types de données de structure

Cette section décrit les modifications apportées aux types de données.

Type de données de structure	Pris en charge dans la V2.x	Pris en charge dans la V5.1.x	Fonctionne de la façon décrite dans le <i>Guide de référence de programmation de l'application MQSeries</i>
MQBO – Options de mise en route	Non	Non	
MQCH – En-tête de pont CICS	Non	Oui	Oui
MQCNO – Options de connexion	Non	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQCNO – Options de connexion» à la page 340.
MQDH – En-tête de diffusion	Non	Oui	Oui
MQDLH – En-tête de rebut	Oui	Oui	Oui
MQGMO – Options Get Message	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQGMO – Options de Get Message» à la page 341.
MQIH – En-tête de pont IMS	Oui	Oui	Oui
MQMD – Descripteur de message	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQMD – Descripteur de message» à la page 341.
MQMDE – Extension de descripteur de message	Non	Oui	Oui
MQOD – Descripteur d'objet	Oui	Oui	Oui
MQOR – Enregistrement d'objet	Non	Oui	Oui

Types de données de structure

Type de données de structure	Pris en charge dans la V2.x	Pris en charge dans la V5.1.x	Fonctionne de la façon décrite dans le <i>Guide de référence de programmation de l'application MQSeries</i>
MQPMO – Options Put Message	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQPMO – Options de Put Message» à la page 342.
MQPMR – Enregistrement de Put Message	Non	Oui	Oui
MQRMH – En-tête de référence de message	Non	Oui	Oui
MQRR – Enregistrement de réponse	Non	Oui	Oui
MQTM – Message de déclenchement	Oui	Oui	Oui
MQTMC2 – Format des caractères des messages de déclenchement	Oui	Oui	Oui
MQWIH – En-tête d'informations de charge de travail	Non	Oui	Oui
MQXQH – En-tête de file d'attente de transmission	Oui	Oui	Oui

Cette section décrit les types de données de structure MQSeries :

MQCNO – Options de connexion

La structure des données MQCNO est indiquée dans le manuel *MQSeries Application Programming Reference* avec les notes suivantes :

- L'unité d'exécution est définie en tant que processus.
- MQCNO_FASTPATH_BINDING peut seulement être utilisé dans un processus doté d'une connexion unique à un gestionnaire de files d'attente
- MQCNO_FASTPATH_BINDING nécessite d'exécuter l'application à l'aide d'un ID utilisateur faisant partie du MQM Administrative User Group ayant créé le gestionnaire de files d'attente.
- Le paramètre Guardian MQCONNECTTYPE peut être utilisé avec le type de définition d'accès indiqué dans la zone *Options*, pour contrôler le type de liaisons utilisé. Si ce paramètre est indiqué, il doit avoir la valeur **FASTPATH** ou **STANDARD**; toute autre valeur sera ignorée. La valeur de ce paramètre respecte les majuscules.
- MQSeries pour Compaq NSK prend en charge MCNO_VERSION_2 et MQCNO_VERSION_1, mais les zones *ClientConnOffset* et *ClientConnPtr* sont ignorées.

MQGMO – Options de Get Message

La structure MQGMO est un paramètre d'entrée-sortie de l'appel MGET. Notez les informations suivantes relatives aux options MQGMO_SET_SIGNAL, MQGMO_WAIT, MQGMO_SYNCPOINT, et MQGMO_NO_SYNCPOINT de MQSeries pour Compaq NSK:

- Si vous souhaitez que l'application continue de travailler en attendant l'arrivée d'un message, utilisez l'option de signal MQGMO_SET_SIGNAL au lieu de MQGMO_WAIT. Toutefois, l'option de signal est spécifique à l'environnement et ne doit pas être utilisée par des applications susceptibles de changer d'environnement.
- Si plus d'un appel MQGET est en attente pour le même message, avec diverses options de signal et d'attente, chaque appel en attente est traité de la même façon. N'indiquez pas MQGMO_SET_SIGNAL avec MQGMO_WAIT. Il serait aussi incorrect d'indiquer cette option avec un descripteur de file d'attente pour lequel un signal est en attente.
- Si une application indique MQGET avec MQGMO_SET_SIGNAL et un WaitInterval de 0, l'option MQGMO_SET_SIGNAL sera ignorée et traitée comme un appel MQGET avec MQGMO_NO_WAIT.

Cela signifie qu'une application doit être préparée de façon à recevoir MQRC_NO_MSG_AVAILABLE sur un appel MQGET avec MQGMO_SET_SIGNAL si le WaitInterval est susceptible d'être égal à zéro. Les applications reçoivent un IPC de signal uniquement dans les conditions suivantes :

- l'application reçoit un MQRC_SIGNAL_REQUEST_ACCEPTED de MQGET (ce qui indique qu'un signal a été *posté*)
- l'application a pu traiter le message `file_open_system` et accepter l'IPC de signal pendant le délai de livraison de signal autorisé par le serveur de files d'attente. Par défaut, ce délai est de 60 secondes. Cette valeur peut être remplacée pour un serveur de files d'attente en indiquant le PARAM MQQSSIGTIMEOUT dans l'environnement du serveur de files d'attente.

Le gestionnaire de files d'attente consigne l'impossibilité de livrer un message IPC à une application, s'il n'est pas parvenu à ouvrir le processus et à envoyer l'IPC avant la fin du délai. Le gestionnaire de files d'attente ne fera alors plus aucune tentative de livraison. Il se peut que certaines applications n'attendent pas indéfiniment un signal IPC.

- MQGMO_SYNCPOINT_IF_PERSISTENT est maintenant pris en charge
- Si aucune des options MQGMO_SYNCPOINT et MQGMO_NO_SYNCPOINT n'est définie, MQSeries pour Compaq NSK applique MQGMO_SYNCPOINT par défaut.
- MQSeries pour Compaq NSK ne prend pas en charge la zone *MsgToken*.

MQMD – Descripteur de message

La structure MQMD contient les informations de contrôle qui décrivent un message. Tenez compte des points suivants :

- *BackoutCount* fonctionne de la façon décrite dans le manuel *MQSeries Application Programming Reference*. Il indique le nombre de fois qu'un message a été précédemment retourné par l'appel MQGET dans le cadre d'une unité de travail, puis annulé. Il aide l'application à détecter les erreurs de traitement basées sur le contenu d'un message. Dans la Version 2.2.0.1, le *BackoutCount* était en fait une estimation.

Types de données de structure

- Sous MQSeries pour Compaq NSK, l'élimination d'un message (et la génération d'un rapport, si nécessaire) a lieu pendant un appel MQGET, sous le contrôle du serveur de file d'attente de façon périodique en fonction des paramètres du gestionnaire de files d'attente.
- La valeur de la zone UserIdentifier, lorsqu'elle est définie par le gestionnaire de files d'attente pendant un appel MQPUT ou MQPUT1 est le nom principal MQSeries trouvé dans la base de données de principaux du gestionnaire de files d'attente, correspondant à l'ID utilisateur effectif de l'application.

MQPMO – Options de Put Message

La structure MQPMO est un paramètre d'entrée/sortie des appels MQPUT et MQPUT1. Notez les informations suivantes relatives à l'option MQPMO_NO_SYNCPOINT sous MQSeries pour Compaq NSK :

- Si aucune des options MQPMO_SYNCPOINT et MQPMO_NO_SYNCPOINT n'est définie, MQSeries pour Compaq NSK applique MQPMO_SYNCPOINT par défaut.

Appels MQI

Cette section décrit les modifications apportées aux appels MQI suivants :

Description de l'appel MQI	Pris en charge dans la V2.x	Pris en charge dans la V5.1.x	Fonctionne de la façon décrite dans le <i>Guide de référence de programmation de l'application MQSeries</i>
MQBACK – Modification des annulations	Retourne une erreur ^{1 à la page 343}	Retourne une erreur ^{1 à la page 343}	Oui
MQBEGIN – Mise en route d'une unité de travail	Non	Retourne une erreur ^{1 à la page 343}	Oui
MQCLOSE – Fermeture d'un objet	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQCLOSE – Fermeture d'un objet» à la page 343.
MQCMIT – Validation des modifications	Non	Retourne une erreur ^{1 à la page 343}	Oui
MQCONN – Connexion d'un gestionnaire de files d'attente	Oui	Oui	Oui
MQCONNX – Connexion d'un gestionnaire de files d'attente (étendue)	Non	Oui	Oui
MQDISC – Déconnexion d'un gestionnaire de files d'attente	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQDISC – Déconnexion d'un gestionnaire de files d'attente» à la page 344.

Description de l'appel MQI	Pris en charge dans la V2.x	Pris en charge dans la V5.1.x	Fonctionne de la façon décrite dans le Guide de référence de programmation de l'application MQSeries
MQGET – Get Message	Oui	Oui	Oui
MQINQ – Demande d'informations sur les attributs d'un objet	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQINQ – Demande d'informations sur les attributs d'un objet» à la page 344.
MQOPEN – Ouverture d'un objet	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQOPEN – Ouverture d'un objet» à la page 344.
MQPUT – Put Message	Oui	Oui	Oui
MQPUT1 – Put Message Un	Oui	Oui	Oui
MQSET – Définition des attributs d'un objet	Oui	Oui	Oui mais avec certaines notes supplémentaires. Pour plus d'informations, consultez la section «MQSET– Définition des attributs d'un objet» à la page 344.
MQSYNC – Synchronisation des mises à jour de statistiques	Retourne une erreur ²	Retourne une erreur ²	Oui

Remarques :

1. L'appel MQI peut être émis par l'application, mais il retourne toujours le code d'achèvement MQCC_FAILED et le code de motif MQRC_ENVIRONMENT_ERROR.
2. Cet appel renvoie toujours le *CompCode* MQCC_OK et le code de motif MQRC_NONE.

MQCLOSE – Fermeture d'un objet

L'appel MQCLOSE, qui est l'inverse de l'appel MQOPEN, abandonne l'accès à un objet.

Sous MQSeries pour Compaq NSK, si une requête MQGET avec l'option MQGMO_SET_SIGNAL est en attente pour le descripteur de file d'attente en cours de fermeture, elle est alors annulée. Les demandes de signaux destinées à une même file mais déposées par des descripteurs différents (*Hobj*) ne sont pas concernées (à moins que ce ne soit une file d'attente dynamique qui fasse l'objet de la suppression, auquel cas, elles sont aussi annulées.)

Si une application FASTPATH ouvre ou ferme une file d'attente dynamique, il se peut que MQSeries démarre ou arrête une transaction TM/MP de façon à mettre à

Appels MQI

jour les bases de données contrôlées. Si l'application a ouvert le fichier T TM/MP (si elle a la capacité d'initier plusieurs transactions), alors ENDTRANSACTION est une opération sans délai d'attente, et l'application reçoit un appel d'arrêt pour la transaction démarrée par MQSeries. La conception des applications doit être contrôlée afin de déterminer si c'est le cas et vérifier que les achèvements logiques de descripteurs sont en mesure de faire face aux appels ENDTRANSACTION provoqués par MQSeries.

MQDISC – Déconnexion d'un gestionnaire de files d'attente

L'appel MQDISC, à l'inverse de MQCONN, rompt la connexion entre le gestionnaire de files d'attente MQSeries et le programme d'application.

La Note d'utilisation 3 du manuel *MQSeries Application Programming Reference* est incorrecte. Sous MQSeries pour Compaq NSK aucun point de synchronisation implicite ne s'effectue si une unité de travail coordonnée par un gestionnaire de files d'attente est en cours lorsque MQDISC est appelé. Ceci est dû au fait qu'un gestionnaire de files d'attente NSK ne peut pas être coordinateur d'une unité de travail. La coordination est réalisée par le sous-système TM/MP.

MQINQ – Demande d'informations sur les attributs d'un objet

L'appel MQPUT retourne une matrice de nombres entiers et un ensemble de chaînes de caractères contenant les attributs d'un objet.

MQOPEN – Ouverture d'un objet

L'appel MQOPEN accorde l'accès à un objet. Sous MQSeries pour Compaq NSK, l'attribut *MaxHandles* du gestionnaire de files d'attente est ignoré.

Si une application FASTPATH ouvre ou ferme une file d'attente dynamique, il se peut que MQSeries démarre ou arrête une transaction TM/MP de façon à mettre à jour les bases de données contrôlées. Si l'application a ouvert le fichier T TM/MP (si elle a la capacité d'initier plusieurs transactions), alors ENDTRANSACTION est une opération sans délai d'attente, et l'application reçoit un appel d'arrêt pour la transaction démarrée par MQSeries. La conception des applications doit être contrôlée afin de déterminer si c'est le cas et vérifier que les achèvements logiques de descripteurs sont en mesure de faire face aux appels ENDTRANSACTION provoqués par MQSeries.

MQSET– Définition des attributs d'un objet

L'appel MQSET modifie les attributs d'un objet représenté par un descripteur. Il doit s'agir d'une file d'attente. Sous MQSeries pour Compaq NSK, le sélecteur MQIA_DIST_LISTS est pris en charge.

Si une application FASTPATH modifie les attributs d'un objet à l'aide de MQSET, MQSeries démarrera ou arrêtera une transaction TM/MP de façon à mettre à jour les bases de données contrôlées. Si l'application a ouvert le fichier T TM/MP (si elle a la capacité d'initier plusieurs transactions), alors ENDTRANSACTION est une opération sans délai d'attente, et l'application reçoit un appel d'arrêt pour la transaction démarrée par MQSeries. La conception des applications doit être contrôlée afin de déterminer si c'est le cas et vérifier que les achèvements logiques de descripteurs sont en mesure de faire face aux appels ENDTRANSACTION provoqués par MQSeries.

Attributs des objets MQSeries

Pour MQSeries pour Compaq NSK, les attributs de tous les objets sont décrits dans le manuel *MQSeries Application Programming Reference*, avec les ajouts et exceptions suivants :

Attributs de toutes les files d'attente

Pour MQSeries pour Compaq NSK, les attributs de toutes les files d'attente sont décrits dans le manuel *MQSeries Application Programming Reference*, avec les ajouts et exceptions suivantes :

Les attributs *AlterationDate* et *AlterationTime* sont mis à jour uniquement lorsque des modifications d'administration sont apportées aux attributs d'un objet. Les attributs *CurrentQDepth*, *OpenInputCount* et *OpenOutputCount* peuvent uniquement être modifiés dynamiquement. *QDepthHighCount*, *QDepthLowEvent*, *QDepthMaxEvent* et *QServiceIntervalEvent* peuvent être modifiés dynamiquement ou lors d'une opération d'administration, mais seules les modifications d'administration (telles que celles réalisées à l'aide des commandes MQSC ou via MQSET) engendreront des modifications dans les attributs *AlterationDate* et *AlterationTimes*.

Attributs des files d'attentes locales et modèles

Sous MQSeries pour Compaq NSK,

- l'attribut *Archive* est ignoré.
- l'attribut *HardenGetBackout* est ignoré car le nombre d'annulation n'est pas enregistré sur le disque dur. Aucune capacité d'archivage des messages n'est disponible.
- Pour les messages persistants, l'attribut *BackoutCount* est toujours sauvegardé sur disque dur. Ce n'est pas le cas pour les messages non persistants. Si toutefois l'attribut option C de serveur -q de la file d'attente locale est défini, *BackoutCount* fera l'objet d'un point de contrôle sur le serveur de file d'attente de secours. Les messages contrôlés de cette façon résistent aux pannes de serveur de file d'attente. Dans un souci de compatibilité avec les autres plateformes MQSeries, l'attribut peut être interrogé par l'appel MQINQ à l'aide du sélecteur MQIA_HARDEN_GET_BACKOUT.

Attributs des gestionnaires de files d'attente

- *MaxMsgLength* est égal à 100 Mo.
- *CommandLevel* est MQCMDL_LEVEL_510.
- *SyncPoint* est MQSP_AVAILABLE.
- La valeur de *CodedCharSetId* est celle qui a été indiquée lors de la création de l'instance du gestionnaire de files d'attente.
- L'attribut *MaxHandles* est ignoré. Il est impossible d'indiquer un nombre maximum d'identificateurs d'ouverture pour MQSeries pour Compaq NSK. La valeur maximale sera déterminée par les contraintes ressources du système.
- L'attribut *MaxUncommittedMsgs* est ignoré. Il est impossible d'indiquer un nombre maximal de messages autorisés pour une seule unité de travail. La valeur maximale sera déterminée par les contraintes ressources.
- Le CCSID peut être altéré.

Conversion de données

Consultez l'«Annexe L. Exits utilisateur» à la page 365 pour connaître le schéma permettant de supporter tous les exits sous MQSeries pour Compaq NSK V5.1. Le mécanisme est différent de celui des versions précédentes de façon à prendre en charge une mise en oeuvre plus cohérente et adaptable des exits.

Annexe I. Construction et exécution d'applications

Les exemples de programmes et de scripts de compilation et de liaison qui figurent dans le sous-volume ZMQSSMPL, illustrent les principales fonctionnalités de l'interface MQI de MQSeries pour Compaq NSK. Ils expliquent également comment compiler et lier une application.

Ecriture des application

Cette section présente les informations les plus récentes concernant les quelques différences (minimes) entre l'interface standard de MQI Version 5.1, telle qu'elle est décrite dans le manuel *MQSeries - Guide de programmation d'applications*, et l'interface MQI pour MQSeries pour Compaq NSK. Cette section vient mettre à jour le manuel *MQSeries - Guide de programmation d'applications* pour MQSeries pour Compaq NSK V5.1.

Utilisation du délai d'attente WAIT MQGET, et des commandes de canaux DISCINT et HBINT

Lors de l'exécution d'une commande MQGET avec l'option MQGMO_NO_SYNCPOINT, une transaction TM/MP est démarrée, puis arrêtée par MQSeries, uniquement si un message persistant qui satisfait au critère de recherche est disponible. Les applications n'ont pas à s'inquiéter de la valeur du délai d'attente WAIT, quelle que soit l'opération de point de synchronisation.

Lors de l'exécution d'une commande MQGET avec l'option MQGMO_SYNCPOINT ou MQGMO_SYNCPOINT_IF_PERSISTENT, la transaction TM/MP est sous le contrôle de l'application qui a envoyé la BEGINTRANSACTION. Le délai d'attente WAIT ne doit pas dépasser le délai d'expiration TMF Autoabort. Dans l'idéal, sa valeur doit être faible, pour éviter de pénaliser le fichier d'audit TM/MP (en général, la valeur est inférieure à une minute). Le fait de fixer un délai trop long sur une file d'attente peu active, ou qui va rester inactive toute une nuit, peut entraîner le dépassement du délai Autoabort, et l'envoi d'une condition MQRC_UOW_CANCELLED (2297) à la commande MQGET lorsqu'un message devient disponible dans la file d'attente. Un délai d'attente WAIT élevé ou infini peut pénaliser les fichiers d'audit TMF, voire pousser le sous-système TM/MP à désactiver les transactions sur l'ensemble du système si le problème n'est pas corrigé.

De même, les valeurs DISCINT et HBINT d'envoi de canaux contrôlent la durée d'une transaction TM/MP. Les canaux peuvent reboucler les transactions en cas d'inactivité, et donc supportent des intervalles prolongés de déconnexion, sans pénaliser les fichiers d'audit TM/MP.

Le paramètre MQTRANSACTIONLIFE, peut servir à contrôler la régénération d'une transaction TM/MP, pour les intervalles de déconnexion de canal et les signaux de présence de valeur nulle. C'est une fonction utile si l'on souhaite une durée de vie de transaction TM/MP plus courte ou plus longue, comme pour modifier le niveau de l'activité TM/MP généré par un canal en veille. Une valeur élevée réduira l'activité, et inversement.

Ecriture des applications

Il convient d'ajouter à chaque classe de serveur PATHWAY de MQSeries MQS-ECxx :

```
PARAM MQTRANSACTIONLIFE <nombre>
```

où <nombre> est une valeur telle que 100.

Ce paramètre remplace la valeur par défaut de l'intervalle de régénération des transactions TM/MP (qui est de 10 secondes). Par exemple, une canal avec un intervalle de déconnexion non limité et sans le paramètre, déclenchera la régénération environ une fois toutes les 10 secondes, en attendant l'arrivée d'un message.

Gestion des unités de travail (transactions)

La gestion des transactions se fait sous le contrôle du produit TM/MP de Compaq, et non de MQSeries lui-même.

Les effets de cette différence sont :

- Par défaut, la valeur de l'option SYNCPOINT pour les appels à MQPUT et MQGET est SYNCPOINT au lieu de NO_SYNCPOINT.
- Pour utiliser cette option par défaut (SYNCPOINT) pour les opérations MQPUT, MQGET et MQPUT1, l'application doit avoir une Transaction TM/MP active, qui définit l'unité de travail à valider. Une application lance une transaction TM/MP en appelant la fonction BEGINTRANSACTION(). Toutes les opérations MQPUT, MQPUT1 et MQGET réalisées par l'application pendant que cette transaction est active, se font au sein de la même unité de travail (transaction). Toute autre opération de base de donnée effectuée par cette application se fait également à l'intérieur de cette unité de travail (UOW). Veuillez noter que le système impose des limites sur le nombre et la taille des messages qui peuvent être écrits ou effacés au cours d'une même transaction TM/MP. Lorsque l'application a terminé l'unité de travail, la transaction TM/MP est arrêtée (l'unité de travail est validée) par la fonction ENDTRANSACTION(). En cas d'erreur, l'application peut annuler la transaction TM/MP (on parle d'annulation) par la fonction ABORTTRANSACTION(). Par conséquent, les fonctions MQCMIT(), MQBACK() et MQBEGIN(), qui sont standard à la Version 5, ne sont pas prises en charge par ce produit. Un appel à ces fonctions se traduit par le retour d'un code d'erreur.
- Si une application utilise l'option NO_SYNCPOINT pour les fonctions MQPUT, MQGET ou MQPUT1, MQSeries lance lui-même une transaction TM/MP, effectue l'opération de mise en file et valide la transaction avant de rendre la main à l'application. Chaque opération est donc entièrement effectuée dans sa propre unité de travail, et lorsqu'elle est terminée, ne peut être annulée par l'application avec TM/MP.
- MQSeries ne doit lancer lui-même une transaction TM/MP (pour une opération NO_SYNCPOINT) que si le message est persistant et requiert par conséquent d'être mis à jour dans un fichier de file d'attente TM/MP protégé.
- Une transaction TM/MP n'a pas besoin d'être active pour les appels MQI autres que MQGET, MQPUT et MQPUT1.
- Le code anomalie MQRC_SYNCPOINT_LIMIT_REACHED est utilisé par MQSeries pour Compaq NSK pour informer une application que la limite du nombre d'opérations d'entrées-sorties permises par le système pour une même transaction TM/MP a été atteinte. Si l'application utilise l'option SYNCPOINT,

elle doit annuler la transaction (backout) et essayer à nouveau avec un plus petit nombre d'opérations pour cette unité de travail.

- Le code anomalie MQRC_UOW_CANCELED informe l'application que la transaction TM/MP (UOW) a été annulée, soit par le système (TM/MP impose des seuils d'utilisation sur l'ensemble du système qui peuvent entraîner cette annulation), soit par une intervention de l'utilisateur ou par l'application elle-même.
- Le code anomalie MQRC_BACKED_OUT informe l'application que MQSeries a du annuler l'UOW à cause d'une erreur ou à cause d'une défaillance du serveur de file d'attente primaire. L'application doit alors appeler la fonction ABORTTRANSACTION (si l'opération était en mode syncpoint) et faire une nouvelle tentative.
- Le code anomalie MQRC_SYNCPOINT_NOT_AVAILABLE informe l'application que MQSeries n'a pas pu démarrer ou utiliser une transaction TM/MP qui était nécessaire pour terminer l'opération. Ce code anomalie indique généralement un problème avec TM/MP, et des informations complémentaires peuvent être disponibles dans le journal des erreurs ou les FFST générés par le gestionnaire de files d'attente.

Considérations générales sur le développement d'applications

Veillez noter que :

- La bibliothèque MQI (liée au processus d'application) n'ouvre pas elle-même \$RECEIVE ni \$TMP (pseudo-fichier de transactions TM/MP). Vous devez par conséquent programmer votre application pour utiliser ces fonctions.
- La bibliothèque MQI utilise un appel à la fonction SERVERCLASS_SEND() pour ses communications initiales avec le gestionnaire de files d'attente. Une fois en connexion, elle maintient au minimum deux fichiers processus ouverts (avec le processus LINKMON et avec un agent local LQMA du gestionnaire de file d'attente), ainsi qu'un petit nombre de fichiers ouverts sur le disque (moins de 10). Les ouvertures de processus se font également pour tout serveur de file d'attente supportant des files d'attente locales qui sont ouvertes en entrée, sortie ou navigation, suite à un appel MQOPEN.
- Avant d'appeler MQCONN, vous devez vous assurer qu'il n'y a aucune entrée-sortie PATHSEND en cours avec l'option NOWAIT. En effet, MQCONN effectue toutes ses entrées-sorties PATHSEND avec l'option NOWAIT, et risque d'interférer avec les entrées-sorties de l'application, et de causer des erreurs.
- Les application liées par FASTPATH ont des particularités, si elles sont également demandeurs TM/MP et gèrent plusieurs unités d'exécution (consultez plus haut les descriptions de MQOPEN, MQCLOSE et MQSET).

Interface XA

Aucune interface XA n'est fournie pour la coordination des unités de travail. Cette coordination est faite par TM/MP.

MQGMO_BROWSE_* avec l'option MQGMO_LOCK

MQGMO_BROWSE_* avec l'option MQGMO_LOCK est dorénavant supportée. Consultez le manuel *MQSeries Application Programming Reference*.

Applications déclenchées

Sous Compaq NSK, les applications MQSeries déclenchées reçoivent les données utilisateur par des variables d'environnement qui sont définies dans le processus

Ecriture des applications

TACL actif. En effet, il y a une limite à la longueur de la liste des arguments qui peuvent être passés à un processus Compaq en C.

Pour pouvoir accéder à ces informations, les applications déclenchées doivent comporter un code semblable à l'exemple ci-dessous (vous pouvez également consulter l'exemple amqsinqa pour plus de détails) :

```
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;
```

Environnements et langages pris en charge

MQSeries pour Compaq NSK V5.1 supporte les langages et environnements précisés dans le tableau 24. Ce tableau indique également si l'application peut ou non utiliser le link FASTPATH ou STANDARD.

Tableau 24. Liste des langages et environnements supportés

Langage	Compatible Guardian ¹	Compatible OSS	Link STANDARD	Link FASTPATH ¹
C natif	Oui	Oui	Oui	Oui
C non-natif	Oui	Non	Oui	Non
COBOL natif	Oui	Oui	Oui	Oui
COBOL non-natif	Oui	Non	Oui	Non
C++ natif	Oui	Oui	Oui	Oui
TAL non-natif	Oui	Non	Oui	Non
Java NonStop ²	Non	Oui	Oui	Non

Remarques :

1. L'environnement Guardian et les applications sous OSS liées avec FASTPATH ne peuvent pas utiliser d'unités d'exécution. Seules les applications Java, et celles sous OSS qui utilisent le link STANDARD, peuvent utiliser des unités d'exécution. Vous trouverez davantage d'informations sur l'utilisation des unités d'exécution dans votre application à la section «Création d'applications utilisant des unités d'exécution».
2. Les applications Java NonStop utilisent JTS (Java Transaction Services) pour leurs transactions.
3. Une application native en link FASTPATH ne pourra résoudre les références MQI que par l'intermédiaire de la bibliothèque partagée (Shared Resource Library) de MQSeries pour Compaq NSK. Une application native en link STANDARD pourra résoudre les références MQI avec la bibliothèque partagée de MQSeries pour Compaq NSK, comme avec une bibliothèque statique native MQI. Pour plus d'informations sur le link en mode FASTPATH ou STANDARD, consultez la section «Comparaison des options de bind FASTPATH et STANDARD» à la page 352.

Création d'applications utilisant des unités d'exécution

Les applications Guardian n'acceptent pas les unités d'exécution. Elles doivent implémenter leur propre mécanisme de coopération d'unités d'exécution. Cependant, les règles d'utilisation de MQI dans un environnement Guardian doivent être respectées par le processus qui l'utilise.

Dans un environnement OSS, il existe un logiciel d'émulation d'unités d'exécution, basé sur les unités d'exécution POSIX. Cette émulation utilise un mécanisme de coopération où le code d'une unité d'exécution doit rendre la main avant qu'une autre unité d'exécution puisse s'exécuter. Les applications peuvent utiliser ce module pour organiser leurs processus en unités d'exécution, mais doivent respecter les contraintes suivantes :

- L'interface MQI n'autorise pas la programmation coopérative entre unités d'exécution. Par conséquent, si la MQI est appelée depuis une unité d'exécution, aucune autre ne peut obtenir le contrôle, quelle que soit la durée d'exécution de la première. Par exemple, si une unité d'exécution appelle la fonction MQGET

Ecriture des applications

avec l'option 'wait indefinitely' aucune autre unité d'exécution ne pourra s'exécuter dans ce processus d'application tant que MQGET n'aura pas rendu la main.

- Les applications sous OSS, liées avec FASTPATH, ne peuvent pas utiliser les unités d'exécution.
- Le gestionnaire de files d'attente n'accepte pas d'agents à plusieurs unités d'exécution (processus MQLQMAG ou LQMA -Local Queue Manager Agents), ni d'Agent MCA (Message Channel Agents).

Compilation et lien des applications

L'interface MQI de MQSeries pour Compaq NSK est implémentée dans le mode mémoire Compaq wide (le type de données int est sur 4 octets), sous l'environnement CRE (Common Runtime Environment). Les applications doivent donc être compatibles avec cet environnement pour fonctionner convenablement. Veuillez consulter les fichiers build donnés en exemple pour déterminer les options nécessaires à chaque compilateur pour assurer la compatibilité.

En particulier, les applications en TAL et en COBOL doivent suivre les règles de compatibilité avec CRE, décrites dans les manuels Compaq correspondants.

Veuillez noter que, pour utiliser la fonction MQGMO_SET_SIGNAL de MQGET, l'attribut HIGHREQUESTERS du code objet des applications en COBOL ou TAL doit être sur ON.

MQSeries pour Compaq NSK est livré avec trois versions de la bibliothèque MQI, placées dans ZMQSLIB. Assurez-vous d'utiliser celle qui convient, conformément au tableau 25.

Tableau 25. Correspondance entre le langage et la version de la bibliothèque MQI

Langage de programmation	Non natif	Natif/Statique	Natif/Dynamique
TAL	MQMLIB	Non applicable	Non applicable
COBOL	MQMLIB	MQMLIBN	MQSRLLIB
C	MQMLIB	MQMLIBN	MQSRLLIB
C++	Non applicable	MQMLIBN	MQSRLLIB

Comparaison des options de bind FASTPATH et STANDARD

MQSeries pour Compaq NSK V5.1 supporte les deux modes de bind, FASTPATH et STANDARD. Le tableau 25 précise la compatibilité entre les langages et chaque méthode.

Mode bind STANDARD

Lors de l'utilisation du mode bind STANDARD dans une application, veuillez considérer les points suivants :

- Ce mode de bind peut être utilisé par les applications natives ou non.
- Une application native liée en mode STANDARD pourra résoudre les références MQI avec soit :
 - La bibliothèque de ressources partagée de MQSeries pour Compaq NSK (Shared Resource Library)
 - Une bibliothèque statique MQI native. Ce qui permet de supporter les applications qui utilisent déjà une bibliothèque de ressources partagée.

Mode bind FASTPATH

Lors de l'utilisation du mode bind FASTPATH dans une application, veuillez considérer les points suivants :

- Seules les applications natives peuvent utiliser le mode bind FASTPATH.
- Une application native liée avec FASTPATH ne pourra résoudre les références MQI que par l'intermédiaire de la librairie partagée (Shared Resource Library) de MQSeries pour Compaq NSK.
- Les applications sous OSS en mode bind FASTPATH, ne peuvent pas utiliser les unités d'exécution.
- Les applications liées avec FASTPATH doivent tourner dans le groupe d'utilisateurs Administrateurs de Compaq NSK MQM, sous l'ID utilisateur qui a créé le gestionnaire de files d'attente.

Exécution d'applications

Pour qu'une application puisse se connecter à un gestionnaire de files d'attente, son environnement doit être correctement défini :

- Le PARAM MQDEFAULTPREFIX est obligatoire pour l'environnement de toute application.
- Si vous avez précisé un autre emplacement (que celui par défaut) pour le fichier MQSINI, l'application ne pourra pas se connecter au gestionnaire de files d'attente si la valeur du PARAM MQMACHINIFILE n'est pas définie convenablement.
- Pour les applications en TAL ou en COBOL, le PARAM SAVE-ENVIRONMENT doit être défini sur ON, sinon elles ne pourront pas se connecter au gestionnaire de files d'attente.

Une application peut fonctionner avec un numéro d'identification de processus (PIN) inférieur à 256 (low-pin) ou supérieur (high-pin). Les exécutables MQSeries sont configurés pour fonctionner en mode high-pin.

Les applications MQSeries sont prises en charge par les environnements Guardian et OSS.

Une application MQSeries peut fonctionner sous PATHWAY, depuis TAACL, ou comme processus-fils d'un autre processus. Il est même possible d'ajouter des applications à la configuration PATHWAY du gestionnaire de files d'attente, pourvu qu'elles réagissent convenablement lors de l'arrêt du gestionnaire de files d'attente.

Exécution d'applications

Annexe J. Interface d'administration MQSeries (MQAI)

MQSeries pour Compaq NSK V5.1 prend en charge l'interface MQAI.

MQAI est l'interface de programmation en langage C vers MQSeries. A l'aide de jeux de données, elle procède aux tâches d'administration sur un gestionnaire de files d'attente MQSeries. Les jeux de données permettent, plus facilement que les commandes PCF, de manipuler les propriétés (ou paramètres) des objets.

MQAI permet de manipuler plus simplement les commandes PCF, que les appels MQGET et MQPUT. Vous pouvez utiliser MQAI pour :

- mettre en oeuvre des applications d'administration automatique et des outils d'administration.
- simplifier l'utilisation des messages PCF. L'interface MQAI est un outil simple permettant de gérer MQSeries. Comme il n'est pas nécessaire d'écrire vos propres messages PCF, vous évitez les problèmes liés aux structures de données complexes.
- faciliter la gestion des incidents. Il est difficile d'obtenir les codes retour des commandes MQSC, mais MQAI rend la gestion des erreurs plus aisée pour le programme.

Remarque : MQSeries pour Compaq NSK V5.1 fournit uniquement des fichiers d'en-tête C. Les fichiers d'en-tête Visual Basic n'y figurent pas.

Pour plus d'informations sur MQAI, consultez le manuel *MQSeries Administration Interface Programming Guide and Reference*

Annexe K. Programmes exemples MQSeries pour Compaq NSK

Les programmes exemples en C et COBOL suivants sont fournis avec MQSeries pour Compaq NSK V5.1 :

Description	Source C	Exécutable C	Source COBOL85	Exécutable COBOL85
Lire et sortir le descripteur de message et le contexte de chaque message de la file d'attente	amqsbcg0	amqsbcg	Pas d'exemple	Pas d'exemple
Répercuter un message d'une file d'attente de messages à une file d'attente de réponses.	amqsecha	amqsech	amq0ech0	amq0ech
Inscrire les messages d'une file d'attente dans stdout, en laissant les messages dans la file (Browse).	amqsgbr0	amqsgbr	amq0gbr0	amq0gbr
Supprimer les messages de la file d'attente nommée et les écrire dans stdout	amqsget0	amqsget	amq0get0	amq0get
Lire la file d'attente de déclenchement, répondre avec les informations sur la file d'attente	amqsinqa	amqsinq	Pas d'exemple	Pas d'exemple
Utiliser une file d'attente d'entrée partagée	Pas d'exemple	Pas d'exemple	amq0inq0	amq0inq
Copier stdin dans un message et placer celui-ci dans une file d'attente précise queue	amqsput0	amqsput	amq0put0	amq0put
Placer un message de demande dans une file d'attente précise et afficher les réponses	amqsreq0	amqsreq	amq0req0	amq0req
Empêcher l'enregistrement sur une file d'attente nommée et répondre en indiquant le résultat (pour la fonction de déclenchement).	amqsseta	amqsset	amq0set0	amq0set
Moniteur de déclenchement	amqstrg0	amqstrg	Pas d'exemple	Pas d'exemple
Exemple de structure d'exit de conversion de données	amqsvfcn	Pas d'exemple	Pas d'exemple	Pas d'exemple
Exemple de structure d'exit de canal	amqsvchn	Pas d'exemple	Pas d'exemple	Pas d'exemple
Exemple de structure d'exit de pondération de charge de cluster	amqswlm0	Pas d'exemple	Pas d'exemple	Pas d'exemple
Exemple de structure pour MQLOADEXIT	amqslxp0	Pas d'exemple	Pas d'exemple	Pas d'exemple

Programmes exemples

Les programmes exemples TAL suivants sont fournis avec MQSeries pour Compaq NSK V5.1 :

Description	Source TAL	Exécutable TAL
Lire <i>n</i> messages d'une file d'attente	zmqreadt	zmqread
Ecrire <i>n</i> messages de longueur <i>n</i> dans une file d'attente	zmqwritt	zmqwrit

Construction de programmes exemples en C

Non natif (à l'aide de la bibliothèque statique non native MQMLIB)

Le sous-volume ZMQSSMPL contient les fichiers de macro TACL suivants, à utiliser pour construire des exemples d'applications non natives en C :

CSAMP

Utilisation : CSAMP *nom-fichier-code-source*

Macro de base servant à compiler un fichier source en C à l'aide des fichiers d'inclusion du sous-volume ZMQSLIB. Par exemple, pour compiler l'exemple AMQSBCG0, utilisez CSAMP AMQSBCG0. Si la compilation réussit, la macro produit un fichier objet dont le dernier caractère du nom est remplacé par la lettre O ; par exemple, AMQSBCGO.

BSAMP

Utilisation : BSAMP *nom-fichier-exe*

Macro de base utilisée pour lier un fichier objet à la bibliothèque utilisateur MQMLIB de ZMQSLIB. Par exemple, pour lier l'exemple compilé AMQSBCG0, utilisez BSAMP AMQSBCG. La macro produit un fichier exécutable nommé *nom-fichier-exeE* ; par exemple, AMQSBCGE.

COMPALL

Utilisation : COMPALL

Macro TACL permettant de compiler chacun des fichiers exemples de code source à l'aide de la macro CSAMP.

BINDALL

Utilisation : BINDALL

Macro TACL permettant de lier chacun des exemples de fichiers objets en exécutables à l'aide de la macro BSAMP.

BUILD C

Utilisation : BUILD C

Macro TACL permettant de compiler et lier tous les exemples de fichiers en C à l'aide des macros COMPALL et BINDALL.

Natif (à l'aide de la bibliothèque statique native MQMLIBN)

Dans le cadre d'une installation native, utilisez les fichiers de macro TACL pour construire des exemples d'applications MQI :

NMCALL	Utilisation : NMCALL Macro permettant de compiler tous les exemples en natif à l'aide de NMCSAMP.
NMCSAMP	Utilisation : NMCSAMP <i>nom-fichier-code-source</i> Macro de base servant à compiler un fichier source en C à l'aide des fichiers d'inclusion du sous-volume ZMQSLIB. Par exemple, pour compiler l'exemple AMQSBCG0, utilisez NMCSAMP AMQSBCG0. Si la compilation réussit, la macro produit un fichier objet dont le dernier caractère du nom est remplacé par la lettre O ; par exemple, AMQSBCGO.
NMLDSAMP	Utilisation : NMLDSAMP <i>nom-fichier-exe</i> Macro de base permettant de lier un fichier objet à la bibliothèque MQI native MQMLIBN de ZMQSLIB.
NMLDALL	Utilisation : NMLDALL Macro TACL permettant de lier chacun des exemples de fichiers objets en exécutables à l'aide de la macro NMLDSAMP.
NMBUILD	Utilisation : NMBUILD Macro TACL permettant de compiler et lier tous les exemples de fichiers natifs en C à l'aide des macros NMCALL et NMLDALL.

Natif (à l'aide de SRL MQMLIBN)

NMLDSSMP	Utilisation : NMLDSSMP <i>nom-fichier-exe</i> Macro de base permettant de lier un fichier objet à la MQ SRL MQSRLLIB native de ZMQSLIB.
NMLDSALL	Utilisation : NMLDSALL Macro TACL permettant de lier chacun des exemples de fichiers objets en exécutables à l'aide de la macro NMLDSSMP.
NMBULDSC	Utilisation : NMBULDSC Macro TACL permettant de compiler et lier tous les exemples de fichiers natifs en C à l'aide des macros NMCALL et NMLDSALL.
NMLDPSRL	Utilisation : NMLDPSRL <i>nom-fichier-exe</i> Macro de base permettant de lier un fichier objet à la SRL privée MQSeries de ZMQSLIB.

Construction d'exemples en C

NMCPSRL

Utilisation : NMCPSRL *nom-fichier-code-source*

Macro permettant de compiler le code utilisateur en vue d'une inclusion dans la PSRL MQSeries.

NMLDUSRL

Utilisation : NMLDUSRL *fichier-entrée-objet*, où *fichier-entrée-objet* est un fichier contenant la liste des objets à lier.

Macro de base permettant de relier le code utilisateur à une bibliothèque fiable.

Remarque : Les applications non natives peuvent se connecter à des gestionnaires de files d'attente natifs et vice versa. Toutes les combinaisons d'opérations natives et non natives sont valides et prises en charge.

Construction de programmes exemples en C++

Natif (à l'aide de la bibliothèque statique native MQMLIBN)

NMCPALL

Utilisation : NMCPALL

Macro permettant de compiler tous les exemples en natif à l'aide de NMCCPP.

NMCCPP

Utilisation : NMCCPP *nom-fichier-code-source*

Macro de base servant à compiler un fichier source en C++ à l'aide des fichiers d'inclusion du sous-volume ZMQSLIB. Par exemple, pour compiler l'exemple IMQSGETP, utilisez NMCCPP IMQSGETP. Si la compilation réussit, la macro produit un fichier objet dont le dernier caractère du nom est remplacé par la lettre O ; par exemple, IMQSGETO.

NMLDCPP

Utilisation : NMLDCPP *nom-fichier-exe*

Macro de base permettant de lier un fichier objet à la bibliothèque MQI native statique MQMLIBN de ZMQSLIB.

NMLDCPPA

Utilisation : NMLDCPPA

Macro TACL permettant de lier chacun des exemples de fichiers objets en exécutables à l'aide de la macro NMLDSAMP.

NMBLDCPP

Utilisation : NMBUILDC

Macro TACL permettant de compiler et lier tous les exemples de fichiers natifs en C++ à l'aide des macros NMCPALL et NMLDCPPA.

Natif (à l'aide de SRL MQMLIBN)

NMLDCPPS

Utilisation : NMLDCPPS *nom-fichier-exe*

Macro de base permettant de lier un fichier objet à la MQ SRL MQSRLLIB native de ZMQSLIB.

NMLDCPSA

Utilisation : NMLDCPSA

Macro TACL permettant de lier chacun des exemples de fichiers objets en exécutables à l'aide de la macro NMLDCPPS.

NMBLDSCP

Utilisation : NMBLDSCP

Macro TACL permettant de compiler et lier tous les exemples de fichiers natifs en C à l'aide des macros NMCPPALL et NMLDCPSA.

Construction de programmes exemples en COBOL

Non natif (à l'aide de la bibliothèque statique non native MQMLIB)

Le sous-volume ZMQSSMPL contient les fichiers suivants, à utiliser pour construire des exemples d'applications en COBOL :

COBSAMP

Utilisation : COBSAMP *nom-fichier-code-source*

Macro de base servant à compiler un fichier source en COBOL à l'aide des fichiers de définition du sous-volume ZMQSLIB. Par exemple, pour compiler le programme AMQ0GBR0, utilisez COBSAMP AMQ0GBR0. Si la compilation réussit, la macro produit un fichier objet dont le dernier caractère du nom est remplacé par la lettre O ; par exemple, AMQ0GBRO.

BCOBSAMP

Utilisation : BCOBSAMP *nom-fichier-exe*

Macro de base utilisée pour lier un objet aux bibliothèques utilisateurs de ZMQSLIB. Par exemple, pour lier l'exemple compilé AMQ0GBRO, utilisez BCOBSAMP AMQ0GBR. La macro produit un exécutable nommé *nom-fichier-exe* AMQ0GBR.

CCBSMPLS

Utilisation : CCBSMPLS

Macro TACL permettant de compiler chacun des fichiers exemples de code source COBOL.

BCBSMPLS

Utilisation : BIND /IN BCBSMPLS/

Ce fichier d'entrée de liaison lie chacun des exemples de fichiers objets COBOL en exécutables.

BUILDCOB

Utilisation : BUILDCOB

Macro TACL permettant de compiler et lier tous les exemples de fichiers en COBOL à l'aide des macros CCBSMPLS et BCBSMPLS.

Natif (à l'aide de la bibliothèque statique native MQMLIBN)

NMCOBSMP

Utilisation : NMCOBSMP *nom-fichier-code-source*

Macro permettant la compilation le COBOL natif 'NMCOBOL' à l'aide de la bibliothèque native MQSeries MQMLIBN de ZMQSLIB. Par exemple, pour compiler le programme AMQ0GBR0, utilisez

Construction d'exemples en COBOL

NMCOBSMP AMQ0GBR0. Si la compilation réussit, la macro produit un fichier objet dont le dernier caractère du nom est remplacé par la lettre O ; par exemple, AMQ0GBRO.

NMLDCOB

Utilisation : NMLDCOB *nom-fichier-exe*

Macro permettant de lier un objet à la bibliothèque native MQSeries MQMLIBN de ZMQSLIB. Par exemple, pour lier l'exemple compilé AMQ0GBRO, utilisez NMLDCOB AMQ0GBR. La macro produit un exécutable nommé *nom-fichier-exe* AMQ0GBR.

NMCOBALL

Utilisation : NMCOBALL

Macro TACL permettant de compiler chacun des fichiers exemples de code source COBOL à l'aide de NMCOBSMP.

NMLDACOB

Utilisation : NMLDACOB

Ce fichier d'entrée de liaison lie chacun des exemples de fichiers objets NMLDCOB en exécutables.

NMBLDCOB

Utilisation : NMBLDCOB

Macro TACL permettant de compiler et lier tous les exemples de fichiers en COBOL à l'aide des macros NMCOBALL et NMLDACOB.

Natif (à l'aide de SRL MQMLIBN)

NMCBSSMP

Utilisation : NMCBSSMP *nom-fichier-code-source*

Macro permettant la compilation du COBOL natif 'NMCOBOL' à l'aide de la SLR MQSRLLIB MQSeries de ZMQSLIB. Par exemple, pour compiler le programme AMQ0GBR0, utilisez NMCBSSMP AMQ0GBR0. Si la compilation réussit, la macro produit un fichier objet dont le dernier caractère du nom est remplacé par la lettre O ; par exemple, AMQ0GBRO.

NMLDSCOB

Utilisation : NMLDSCOB *nom-fichier-exe*

Macro permettant de lier un objet à la SRL MQSRLLIB MQSeries de ZMQSLIB. Par exemple, pour lier l'exemple compilé AMQ0GBRO, utilisez NMLDSCOB AMQ0GBR. La macro produit un exécutable nommé *nom-fichier-exe* AMQ0GBR.

NMCBSALL

Utilisation : NMCBSALL

Macro TACL permettant de compiler chacun des fichiers exemples de code source COBOL à l'aide de NMCBSSMP.

NMLDSCOB

Utilisation : NMLDSCOB

Ce fichier d'entrée de liaison lie chacun des exemples de fichiers objets NMLDSCOB en exécutables.

NMBLDSCB

Utilisation : NMBLDSCB

Macro TACL permettant de compiler et lier tous les exemples de fichiers en COBOL à l'aide des macros NMCBSALL et NMLDSCOB.

Construction de programmes exemples en TAL

Non natif (à l'aide de la bibliothèque statique non native MQMLIB)

Le sous-volume ZMQSSMPL contient les fichiers suivants, à utiliser pour construire des exemples de programmes en TAL.

TALSAMP

Utilisation : TALSAMP *nom-fichier-code-source*
Macro de base servant à compiler un fichier source TAL à les des fichiers de définition du sous-volume Appendix I. MQSeries for Compaq NSK sample programs 335.ZMQSLIB. Par exemple, pour compiler le programme ZMQWRITT, utilisez TALSAMP ZMQWRITT. Si la compilation réussit, la macro produit un fichier objet dont le dernier caractère du nom est remplacé par la lettre O ; par exemple, ZMQWRITO.

BTALSAMP

Utilisation : BTALSAMP *nom-fichier-exe*
Macro de base utilisée pour lier un objet aux bibliothèques utilisateurs de ZMQSLIB. Par exemple, pour lier l'exemple compilé ZMQWRITO, utilisez BTALSAMP ZMQWRIT.

CTLSMPLS

Utilisation: CTLSMPLS
Macro TACL permettant de compiler chacun des fichiers exemples de code source COBOL.

BTLSTMPLS

Utilisation : BIND /IN BTLSTMPLS/
Ce fichier d'entrée de liaison lie chacun des exemples de fichiers objets TAL en exécutable.

BUILDTAL

Utilisation : BUILDTAL
Macro TACL permettant de compiler et lier tous les exemples de fichiers en TAL à l'aide des macros CTLSMPLS et BTLSTMPLS.

Constructions de programmes exemples OSS (mode natif uniquement)

Le répertoire - /opt/mqm/samp contient les programmes exemples pour MQSeries et le fichier make MQMAKE.SMP. Ce MakeFile contient toutes les cibles nécessaires à la construction de tous les exemples en C, C++ et NMCOBOL.

Le répertoire - /opt/mqm/inc contient tous les copylibs et fichiers d'en-tête nécessaires à la construction de programmes OSS.

Remarque : Le fichier SRL MQSeries et la librairie MQI native —MQMLIBN existent uniquement dans le système de fichiers Guardian. Vous

Construction d'exemples OSS

devrez modifier vos fichiers build et make de façon à pointer vers eux si nécessaire. Par exemple, /G/system/zmqslib/mqsrllib.

Annexe L. Exits utilisateur

MQSeries pour Compaq NSK V5.1 prend en charge les programmes d'exit de canal et de conversion de données, le programme d'exit CLWL (Cluster Workload Management), ainsi qu'un programme d'exit de charge spécifique à Compaq NSK. Pour plus d'informations sur les exits de canal, consultez le manuel *MQSeries Intercommunication*. Pour des détails complémentaires sur les exits de conversion des données, reportez-vous aux manuels *MQSeries - Guide de programmation d'applications* et *MQSeries Application Programming Reference*. Pour plus d'informations sur les exits de pondération de charge de travail de cluster, consultez le manuel *MQSeries Queue Manager Clusters*.

Cet annexe contient des informations spécifiques à l'utilisation des programmes d'exit sous MQSeries pour Compaq NSK.

Exits utilisateur pris en charge

Le tableau 26 présente les caractéristiques de chaque type d'exit utilisateur pris en charge par MQSeries pour Compaq NSK.

Tableau 26. Exits utilisateur pris en charge par MQSeries pour Compaq NSK

Type d'exit utilisateur	Longueur maximale du nom d'exit	Longueur maximale des données d'exit	Activé dans	Chaîné ?
Exit de canal MSG	32	32	DEFINE CHANNEL	Oui
Exit de canal SEND	32	32	DEFINE CHANNEL	Oui
Exit de canal RECEIVE	32	32	DEFINE CHANNEL	Oui
Exit de canal SECURITY	32	32	DEFINE CHANNEL	
Exit de canal MSGRETRY	32	32	DEFINE CHANNEL	
Exit d'auto-définition de canal	32	32	ALTER QMGR	
Exit de pondération de charge de cluster	32	32	ALTER QMGR	
Exit de conversion de données	8	Non applicable	Nom de FORMAT inconnu	
MQ_LOAD_ENTRY_POINT_EXIT	Nom imposé	Non applicable	Appelé lorsque l'un des exits ci-dessus est nécessaire ou activé.	

MQ_LOAD_ENTRY_POINT_EXIT est le seul exit utilisateur spécifique à Compaq NSK.

Format de nom d'exit

Un nom d'exit (à l'exception de MQ_LOAD_ENTRY_POINT_EXIT) est une chaîne alphanumérique d'au maximum 32 caractères. La longueur du champ de données associé à certains exits est au maximum de 32 caractères.

MQ_LOAD_ENTRY_POINT_EXIT - Chargement des exits utilisateur

Tous les programmes d'exit utilisateur doivent être liés avec la SRL de MQSeries ou avec une bibliothèque statique. Tout programme d'exit utilisateur doit contenir au moins une fonction externe (un symbole), qui puisse être appelé par MQSeries en cas de besoin.

Avant d'activer les autres exits utilisateur de MQSeries, vous devez installer un programme MQ_LOAD_ENTRY_POINT_EXIT, afin de mapper vos noms d'exit aux adresses d'entrée. Votre programme MQ_LOAD_ENTRY_POINT_EXIT doit être lié avec la SRL de MQSeries ou une bibliothèque statique. Il est appelé chaque fois que l'un des autres exits utilisateur est activé.

Le nom du programme MQ_LOAD_ENTRY_POINT_EXIT est imposé, ce qui signifie que son nom externe de fonction doit rester MQ_LOAD_ENTRY_POINT.

MQ_LOAD_ENTRY_POINT_EXIT a ensuite toute liberté pour mapper un nom d'exit à un quelconque point d'entrée, ou pour mapper plusieurs points d'entrée sur la même adresse de point d'entrée.

MQSeries fournit un squelette de fonction MQ_LOAD_ENTRY_POINT_EXIT qui renvoie systématiquement MQXCC_FAILED (nom d'exit introuvable) à chaque appel. Avant d'activer les autres exits utilisateur, vous devez remplacer cette fonction par la vôtre.

MQ_LOAD_ENTRY_POINT_EXIT (*ExitParms*)

Paramètres :

Exitparms (PMQLXP) – input/output
LoadExit Parameter Block

Notes : la fonction réalisée par le programme MQ_LOAD_ENTRY_POINT_EXIT est définie par celui qui fournit l'exit.

La figure 46 à la page 369 montre un exemple de programme MQ_LOAD_ENTRY_POINT_EXIT, qui mappe trois noms d'exit aux adresses de points d'entrée.

MQLXP - Structure des paramètres de MQ_LOAD_ENTRY_POINT_EXIT

La structure MQLXP décrit les informations qui sont passées à l'exit de charge.

Cette structure est spécifique à Compaq NSK.

Champs

StrucId (MQCHAR4)

Identificateur de structure.

Programmes d'exit pris en charge

Sa valeur est : MQLXP_STRUC_ID.

Identificateur de la structure de paramètres d'exit de charge.

Pour le langage C, la constante MQLXP_STRUC_ID_ARRAY est également définie. Sa valeur est identique à celle de MQLXP_STRUC_ID, mais il s'agit d'un tableau de caractères au lieu d'une chaîne.

Pour l'exit, il s'agit d'un champ en entrée.

Version (MQLONG)

Numéro de version de structure.

Sa valeur est : MQLXP_VERSION_1.

Structure de paramètre d'exit de charge, Version-1.

Le numéro de la version courante est spécifié par la constante :
MQLXP_CURRENT_VERSION

Version courante de structure du paramètre d'exit de charge.

Pour l'exit, il s'agit d'un champ en entrée.

NomGestFile (MQCHAR48)

Définit le nom du gestionnaire de files d'attente local.

Il s'agit du nom du gestionnaire de files d'attente qui a appelé l'exit de charge. Le nom est complété avec des caractères blancs jusqu'à la longueur du champ.

Pour l'exit, il s'agit d'un champ en entrée.

NomPointEntrée (MQCHAR32)

Définit le nom du point d'entrée demandé.

Il s'agit du nom dont a besoin l'exit de charge, afin de le convertir en une adresse de point d'entrée. Le nom est complété avec des caractères blancs jusqu'à la longueur du champ.

Pour l'exit, il s'agit d'un champ en entrée.

AdresseEntrée (PMQFUNC)

L'adresse d'appel retournée.

Il s'agit de l'adresse du point d'entrée demandé.

Pour l'exit, il s'agit d'un champ en sortie.

RéponseExit (MQLONG)

La réponse de l'exit.

Cette valeur est modifiée par l'exit, pour indiquer s'il a réussi à convertir l'adresse du point d'entrée. Sa valeur doit être l'une des suivantes :

MQXCC_OK
Success.

Cette valeur indique que le traitement de l'exit a réussi à convertir en une adresse d'appel le nom indiqué dans le champ NomPointEntrée de la structure ExitParms. Cette adresse est retournée dans le champ AdresseEntrée de la structure MQLXP.

MQXCC_FAILED
Failed.

Programmes d'exit pris en charge

Cette valeur indique que le traitement de l'exit n'a pas pu convertir en une adresse d'appel le nom indiqué dans le champ NomPointEntrée de la structure ExitParms.

Toute autre valeur retournée dans le champ RéponseExit a la même signification que MQXCC_FAILED.

Pour l'exit, il s'agit d'un champ de sortie.

Exemple de programme MQ_LOAD_ENTRY_POINT_EXIT

La figure 46 à la page 369 montre un exemple de programme MQ_LOAD_ENTRY_POINT_EXIT fonctionnel, qui mappe trois noms d'exit (deux canaux d'exit et un exit de conversion de données) vers leurs adresses de point d'entrée. Le code source du programme d'exemple MQ_LOAD_ENTRY_POINT_EXIT se trouve dans le sous-volume d'exemples (AMQSLXP0).


```

/*****/
/*
/* Program name: AMQSLXP0          (Compaq NSK only)          */
/*
/* Description: Sample C skeleton of a Load Exit function    */
/*
/* Statement:      Licensed Materials - Property of IBM      */
/*
/*                  (C) Copyright IBM Corp. 1993, 2001      */
/*
/*****/
/*
/* Function:
/*
/* AMQSLXP0 is a sample C skeleton of a Load Exit function  */
/*
/* The function resolves EntryNames to callable addresses    */
/*
/* Once complete the code should be compiled into a loadable */
/* object, the name of the object should be the name of the  */
/* format to be converted. Instructions on how to do this are */
/* contained in the README file in this directory.            */
/*
/*****/
/*
/* AMQSLXP0 takes the parameters defined for a Load Exit    */
/* routine in the CMQXC.H header file.                       */
/*
/*****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>

#include <cmqc.h>
#include <cmqxc.h>

/*****/
/* Load Exit
/*
/*
/*****/
void
MQENTRY MQ_LOAD_ENTRY_POINT(
    PMQLXP pExitParms          /* exit Parameter */
)
{
    /* No loadable entry points are defined */
    pExitParms->ExitResponse = MQXCC_FAILED;

    return;
}
/*****/
/* End of AMQSLXP0
/*****/

```

Figure 46. Exemple de programme MQLOADEXIT

Programmes d'exit pris en charge

Avec le programme MQ_LOAD_ENTRY_POINT_EXIT ci-dessus, et les exits de conversion de données et de canal installés, vous pouvez activer vos exits d'entrée et de sortie de canal par la commande MQSC :

```
ALTER CHANNEL(CHAN) CHLTYPE(SDR) SENDEXIT(MY_CHANNEL_SEND_EXIT)
ALTER CHANNEL(CHAN) CHLTYPE(SDR) RCVEXIT(MY_CHANNEL_RCV_EXIT)
```

L'exit de conversion de données sera appelé par MQSeries lorsqu'une fonction MQGET sera effectuée avec le mode de conversion activé (MQGMO_CONVERT) et que le format de message sera MY_FORMAT.

Installation des exits utilisateur

Tous les exits utilisateur que vous créez devront être installés dans la SRL privée de MQSeries. Certains types d'exit doivent également être installés dans toute bibliothèque statique MQSeries utilisée par les applications.

Type d'exit utilisateur	Doit être installé dans
Exit de canal MSG	SRL privée
Exit de canal SEND	SRL privée
Exit de canal RECEIVE	SRL privée
Exit de canal SECURITY	SRL privée
Exit de canal MSGRETRY	SRL privée
Exit d'auto-définition de canal	SRL privée
Exit de pondération de charge de cluster	SRL privée
Exit de conversion de données	SRL privée (et aussi dans les bibliothèques statiques, natives ou non, utilisées par les applications)
MQ_LOAD_ENTRY_POINT_EXIT	SRL privée (et aussi dans les bibliothèques statiques, natives ou non, utilisées par les applications)

Installation d'un exit dans la SRL privée de MQSeries

Pour installer un programme d'exit dans la SRL privée de MQSeries, il faut créer une nouvelle version de cette SRL, contenant le nouvel exit :

1. Compiler la fonction d'exit. Par exemple :

```
NMCPSRL AMQSVFCN
```

2. Compiler la fonction de point d'entrée MQLOAD. Par exemple :

```
NMCPSRL AMQSLXP0
```

Programmes d'exit pris en charge

3. Faire le link des objets exit et point d'entrée pour construire une bibliothèque qui puisse subir un nouveau link et être utilisée dans la SRL. Par exemple :

```
NMLDUSRL OBJECTS EXITS
```

4. Créer une nouvelle version de la SRL privée de MQSeries en faisant un link de cet objet de conversion de données avec la SRL privée de MQSeries (la version capable de subir un nouveau link). Par exemple :

```
MAKEPSRL EXITS $VOL.ZMQSLIB NEWMQSRL
```

5. Arrêter tous les gestionnaires de file d'attente et applications qui utilisent la SRL privée de MQSeries.
6. Effectuer à nouveau le link de toutes les applications MQSeries avec la nouvelle SRL. Par exemple :

```
NMLDEXES $VOL.ZMQSLIB.NEWMQSRL $VOL.ZMQSEXE
```

7. Compiler l'application Get. Par exemple :

```
NMCSAMP AMQSGET0
```

8. Effectuer à nouveau le link de toutes les applications utilisateur avec la nouvelle SRL. Par exemple :

```
AMQSGET NMLDPSRL AMQSGET
```

9. Redémarrer MQSeries et toutes les applications MQI.

Les étapes 6 et 8 sont assez rapides, mais peuvent être omises si la nouvelle SRL privée est placée dans ZMQSLIB et nommée MQSRLLIB. Ces étapes peuvent être répétées s'il faut effectuer un link avec différentes SRL privées de MQSeries.

Installation d'un exit dans la bibliothèque statique native de MQSeries

la macro NMLDEXIT de TACL permet de faire un link entre un exit, l'application voulue et la bibliothèque MQI. Par exemple :

```
NMLDEXIT Fichier-Objet Fichier-Objet-Exit
```

Installation d'un exit dans la bibliothèque statique non-native de MQSeries

La macro BEXITE de TACL permet de faire un link entre un exit et l'exécutable (ou la bibliothèque) choisi.

Programmes d'exit pris en charge

Remarque : Cette procédure modifie l'exécutable. Il est par conséquent recommandé de faire une copie de l'exécutable ou de la bibliothèque cible avant de lancer la macro.

Les fonctions d'exit, une fois compilées, doivent être liées avec l'exécutable ou la bibliothèque cible pour pouvoir être utilisées par MQSeries. C'est le rôle de la macro BEXITE. Par exemple :

```
BEXITE Exécutable-Ou-Bibliothèque-Cible  
FichierExit-Ou-Bibliothèque-Source
```

Pour faire le link de l'exit de conversion de données fourni en exemple avec l'exemple MQSGETA, procédez aux étapes suivantes :

1. Compiler la fonction d'exit. Par exemple :

```
CSAMP AMQSVFCN
```

2. Compiler la fonction de point d'entrée MQLOAD. Par exemple :

```
CSAMP AMQSLXP0
```

3. Compiler l'application Get. Par exemple :

```
CSAMP AMQSGET0
```

4. Faire le bind de l'application Get. Par exemple :

```
BSAMP AMQSGET
```

5. Faire le bind de la fonction d'exit avec l'application Get. Par exemple :

```
BEXITE AMQSGET AMQSVFC0
```

6. Faire le bind de la fonction de point d'entrée avec l'application Get. Par exemple :

```
BEXITE AMQSGET AMQSLXP0
```

Programmes d'exit pris en charge

Voici une autre possibilité, si toutes les applications doivent disposer du même exit de conversion de données. Les étapes suivantes créeront à la fois une bibliothèque utilisateur et une application avec l'exit lié :

1. Compiler la fonction d'exit. Par exemple :

```
CSAMP AMQSVFCN
```

2. Compiler la fonction de point d'entrée MQLOAD. Par exemple :

```
CSAMP AMQSLXPO
```

3. Compiler l'application Get. Par exemple :

```
CSAMP AMQSGET0
```

4. Faire le bind de la fonction d'exit avec la bibliothèque utilisateur. Par exemple :

```
BEXITE ZMQSLIB.MQMLIB AMQSVFCO
```

5. Faire le bind de la fonction de point d'entrée avec la bibliothèque utilisateur.
Par exemple :

```
BEXITE ZMQSLIB.MQMLIB AMQSLXPO
```

6. Faire le bind de l'application Get avec la bibliothèque modifiée. Par exemple :

```
BSAMP AMQSGET
```

Programmes d'exit pris en charge

Annexe M. Configuration des communications

Cette annexe décrit comment configurer les communications pour MQSeries pour Compaq NSK à l'aide des protocoles SNA et TCP/IP. Les exemples suivants sont proposés :

- «Exemple de communications SNAX» à la page 382
- «Exemple de communications ICE» à la page 388
- «Exemple de communications TCP/IP» à la page 391

Canaux SNA

Les attributs de canaux suivants sont nécessaires pour les canaux SNA sous MQSeries pour Compaq NSK V5.1 :

CONNNAME

La valeur de CONNNAME dépend du protocole (SNAX ou ICE) utilisé :

Si SNAX est utilisé :

CONNNAME('\$PPPP.LOCALLU.REMOTELU')

S'applique à l'émetteur, au demandeur et aux canaux de serveurs remplissant toutes les conditions, où :

\$PPPP est le nom du processus SNAX/APC.
LOCALLU est le nom de la LU locale.
REMOTELU est le nom de la LU partenaire sur la machine éloignée.

Par exemple :

```
CONNNAME('$BP01.IYAHT080.IYCNVM03')
```

Si ICE est utilisé :

CONNNAME('\$PPPP.#OPEN.LOCALLU.REMOTELU')

S'applique à l'émetteur, au demandeur et aux canaux de serveurs remplissant toutes les conditions, où :

\$PPPP est le nom du processus ICE.
#OPEN est le nom ouvert ICE.
LOCALLU est le nom de la LU locale.
REMOTELU est le nom de la LU partenaire sur la machine éloignée.

Par exemple :

```
CONNNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+
```

MODENAME

est le nom de mode SNA. Par exemple, MODENAME(LU62PS).

TPNAME('LOCALTP[.REMOTETP]')

est le nom du traitement de transaction (TP).

LOCALTP est le nom de la classe de serveurs (habituellement

Configuration des communications

MQSeries) dans le PATHWAY utilisé pour les communications SNA. Si SNAX est utilisé, la classe de serveurs doit exister dans le même PATHWAY que le répartiteur SNAX et la classe de serveur du processus APC. Si le protocole ICE est utilisé, la classe de serveurs doit figurer dans le PATHWAY déclaré dans la commande NOF - ADD TP.

REMOTETP est le nom du TP de la machine éloignée. Cette valeur est facultative. Si elle n'est pas précisée et si le canal entame une conversation (s'il s'agit d'un canal émetteur, demandeur ou remplissant toutes les conditions requises), le nom LOCALTP est utilisé.

Les valeurs LOCALTP et REMOTETP peuvent comprendre jusqu'à 16 caractères.

Processus de canaux répondeurs LU 6.2

Dans MQSeries pour Compaq NSK V5.1, un processus de programme d'écoute SNA est nécessaire pour écouter les demandes de connexions entrantes émises par les canaux de gestionnaires de files d'attente éloignés.

Les programmes d'écoute SNA MQSeries doivent être répartis par le produit SNA lorsqu'une demande de connexion entrante est émise par un canal éloigné.

Le fait d'utiliser le répartiteur APC SNAX permet à SNAX d'accepter les demandes de connexions entrantes émises par les programmes transactionnels partenaires. Pour cela, les conditions suivantes doivent être réunies :

- le répartiteur SNAX est exécuté dans un PATHWAY différent de celui des gestionnaires de files d'attente
- le processus APC est exécuté en tant que classe de serveur depuis le même PATHWAY

L'utilisation du protocole ICE de Insession nécessite :

- Qu'un programme de transaction (classe de serveur) soit défini en tant que PATHWAY
- Qu'un programme de transaction soit ajouté dans la fonction NOF, pointant sur cette classe de serveurs.

Il n'existe aucun répartiteur ICE autonome, le processus ICE gère lui-même les demandes de connexions entrantes.

Répartition des programmes de transaction (TP) SNAX

Le produit Compaq SNAX SNA permet le lancement de programmes de transaction (TP) APPC lorsqu'une connexion APPC émise par un programme de transaction partenaire arrive. Un répartiteur SNAX distribue ces demandes au processus SNAX \$APC qui lui est associé, par l'intermédiaire d'une classe de serveurs de programme de transaction locale.

L'utilisation du répartiteur SNAX pour MQSeries nécessite que les conditions suivantes soient réunies :

- Le répartiteur SNAX doit être exécuté sur le même PATHWAY que le processus \$APC qui lui est associé.
- Chaque TPNAME entrant doit être défini en tant que classe de serveurs (habituellement nommée MQSERIES) dans le même PATHWAY que le répartiteur et le processus \$APC.

Configuration des communications

- Le nom de programme de la classe de serveurs est le programme **runmqlsr** existant dans le sous-volume des exécutable MQSeries (habituellement ZMQSEXE).
- Le processus de répartiteur lance la classe de serveurs et transmet les informations appropriées (\$APC Process, LUName, TPName) à cette classe de serveurs (TP) dans une demande IPC DISPATCH-TP.

Répartition des programmes de transaction (TP) ICE

ICE Version 3.2 met en oeuvre son gestionnaire de connexion SNA de la même façon que SNAX, dans le sens où son TP est une classe de serveurs PATHWAY. Le processus ICE accepte une demande de connexion et est lui-même le répartiteur. Toutefois, ce processus ICE n'a pas besoin de s'exécuter dans le même PATHWAY que le TP (ServerClass). Dans ce cas, les conditions requises sont les suivantes :

- Un processus ICE actif doit être en cours.
- Un TP de répartition doit être ajouté dans la NOF. Par exemple :

```
ADD TP <tpname>, PROCESS <$process>, SERVERCLASS <nom Serverclass>
```

Le nom *Serverclass* est habituellement MQSeries.

- Chaque TPNAME entrant doit être défini en tant que classe de serveurs (habituellement nommée MQSERIES) dans le PATHWAY<\$process>.
- Le nom de programme de la classe de serveurs est le programme RUNMQLSR existant dans le sous-volume des exécutable MQSeries (habituellement ZMQSEXE)
- Le processus ICE lance la classe de serveurs et transmet les informations appropriées (\$ICE Process, ApplName, TPName) à cette classe de serveurs (TP) dans une demande IPC DISPATCH-TP.

Exemple de configuration en environnement SNA

Les éléments ci-dessous sont des exemples de configuration des environnements SNA.

Utilisation de APC SNAX : Si vous utilisez APC SNAX :

- Créez un PATHWAY qui sera utilisé exclusivement par ce programme d'écoute
- Exécutez une serverclass de processus APC depuis ce PATHWAY.

Entrez la chaîne suivante à l'invite TACL :

```
TACL> Pathmon /name $PMAP, nowait, out $vhs, cpu 3/4
TACL> Pathcom $PMAP
= 0 LU62SCFG
```

Configuration des communications

où LU62SCFG est un fichier d'édition contenant le texte suivant :

```
[ 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
```

Figure 47. Exemple de fichier de configuration SNAX MQSeries (Numéro 1 de 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
```

Figure 47. Exemple de fichier de configuration SNAX MQSeries (Numéro 2 de 3)

Configuration des communications

```
[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
```

Figure 47. Exemple de fichier de configuration SNAX MQSeries (Numéro 3 de 3)

Remarque : La classe de serveurs de programmes d'écoute est identique à la classe de serveurs MQS-TCPLIS00 du PATHWAY des gestionnaires de files d'attente, hormis la présence d'un paramètre de démarrage : SET SERVER STARTUP "-t LU62"

Utilisation de Insession ICE

Si vous utilisez Insession ICE, créez un PATHWAY qui sera utilisé exclusivement par ce programme d'écoute. Le processus ICE n'est pas exécuté depuis ce PATHWAY.

1. Ajoutez le TP à la fonction NOF de la façon suivante :

```
ADD TP <tpname>, PROCESS <processus>, SERVERCLASS <server> [, <option> ...]
```

Où :

- | | |
|------------------|--|
| processus | est le nom du processus PATHMON chargé de la gestion du TP |
| serveur | est le nom du PATHMON SERVERCLASS auquel appartient le TP |
| option | peut être |
- [ATTACHTIMER n] -- le temps (en centièmes de seconde) pendant lequel ICE attendra une connexion après avoir distribué une nouvelle unité d'exécution de TP. La valeur par défaut est de 6000 (60 secondes).
 - [MAXDISPATCHTHREADS n] -- nombre maximum de TP répartis simultanément VALEUR PAR DEFAULT : 0 (le nombre de TP répartis simultanément est illimité)
 - [TIMEOUT n] -- détermine de quelle façon ICE répondra à une connexion si MAXDISPATCHTHREADS a été atteint sur

Configuration des communications

un TP TIMEOUT -1 = la CONNEXION est mise en file d'attente indéfiniment
TIMEOUT >0 = la CONNEXION est mise en file d'attente pendant n/100 secondes
VALEUR PAR DEFAUT : 0 (la CONNEXION est rejetée immédiatement)

- Il est encore nécessaire d'ajouter la classe de serveurs au PATHWAY. A l'invite TACL, entrez :

```
TACL> Pathmon /name $PMAP, nowait, out $vhs, cpu 3/4
TACL> Pathcom $PMAP
= 0 LU62ICFG
```

où LU62ICFG est un fichier d'édition contenant le texte suivant :

```
[
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
[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
```

Figure 48. Exemple de fichier de configuration SNA MQSeries pour ICE

Configuration des communications

Remarque : La classe de serveurs de programmes d'écoute est identique à la classe de serveurs MQS-TCPLIS00 du PATHWAY des gestionnaires de files d'attente, hormis la présence d'un paramètre de démarrage : SET SERVER STARTUP "-t LU62"

Canaux TCP/IP

MQSeries pour Compaq NSK permet d'utiliser plusieurs processus TCP/IP dans un environnement utilisant un seul gestionnaire de files d'attente MQSeries. Vous pouvez donc sélectionner des processus TCP/IP utilisés par un gestionnaire de files d'attente en associant le processus TCP/IP souhaité à un canal donné. Les canaux sortants (émetteur, serveur et demandeur) peuvent préciser le nom du processus TCP/IP dans la zone CONNAME de la définition de canal.

A l'aide de **runmqsc**:

```
alter channel ... conname ('$ZTC1.123.456.789.012(1415)')
alter channel ... conname ('$ZTC1.nomhotedns(1415)')
```

A l'aide des écrans MQMC :

```
TCPIP/SNA Process: $ZTC1
```

A l'aide des commandes PCF :

```
strncpy( pPCFString->String, ('$ZTC1.123.456.789.012(1415)', len );
```

Pour reconfigurer la résolution DNS d'un convertisseur personnalisé, ajoutez la commande suivante à toutes les classes de serveurs PATHWAY ECnn :

```
DEFINE =TCPIP^RESOLVER^NAME, FILE nomfichier
```

où nomfichier est l'emplacement du fichier convertisseur.

Si vous utilisez un fichier hôte, ajoutez la commande suivante à toutes les classes de serveurs PATHWAY ECnn :

```
DEFINE =TCPIP^HOST^FILE, FILE nomfichier
```

où nomfichier est l'emplacement du fichier hôte.

Configuration des communications

Les canaux entrants utilisent des variables d'environnement pour savoir quels processus TCP/IP utiliser. Les programmes d'écoute TCP/IP transmettent la valeur de ce processus à leurs EC et processus de canaux répondeurs TCP/IP respectifs par l'intermédiaire de leurs agents. Pour définir le processus TCP/IP de canal entrant :

A l'aide de TAACL :

```
ADD DEFINE =TCPIP^PROCESS^NAME, FILE nomprocessus
```

où nomprocessus est le nom du processus TCP/IP.

A l'aide de PATHWAY, pour les classes de serveurs MQS-TCPLIS nn , où nn est le numéro de classe de serveurs de programmes d'écoute :

```
DEFINE =TCPIP^PROCESS^NAME, FILE \HAWK.$ZTC1  
PARAM MQLISTENPORTNUM "1415"
```

Pour des informations sur l'utilisation d'un processus TCP/IP personnalisé dans le cadre de communications TCP/IP, consultez la section «Reconfiguration de la classe de serveur MQS-TCPLIS nn pour un port et un processus TCP/IP personnalisés» à la page 56. Pour des détails supplémentaires sur les ports TCP/IP écoutés par un gestionnaire de files d'attente, reportez-vous à la section «Ports TCP/IP écoutés par le gestionnaire de files d'attente» à la page 55.

Exemples de communication

Cette section propose des exemples de configurations de communications pour SNA (SNAX et ICE) et TCP/IP.

Exemple de communications SNAX

Cette section fournit :

- Un exemple de fichier de configuration SCF pour la ligne SNA
- Des exemples de paramètres SYSGEN permettant de prendre en charge la ligne
- Un exemple de fichier de configuration SCF pour la définition de processus SNA
- Des exemples de définitions de canaux MQSC

Fichier de configuration de ligne SCF SNA

Un exemple de fichier de configuration SCF est présenté ci-dessous :

```
==  
== 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
```

Configuration des communications

```
==
== 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

== 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
```

Configuration des communications

== 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
```

Paramètres SYSGEN

La sortie ci-dessous est constituée des entrées d'un fichier CONFTEXT, qui permettent à un SYSGEN de prendre en charge les lignes SNA et en anneau à jeton :

```
!*****  
!                               LAN MACRO  
!*****  
! This macro is used for all 361x LAN controllers  
! REQUIRES T9375 SOFTWARE PACKAGE  
  
C3613^MLAM          = MLAM  
                    TYPE 56,          SUBTYPE 0,  
                    PROGRAM          C9376P00,  
                    INTERRUPT        IOP^INTERRUPT^HANDLER,  
                    MAXREQUESTSIZE   32000,  
                    RSIZE            32000,  
                    BURSTSIZE        16,  
                    LINEBUFFERSIZE   32,  
                    STARTDOWN #;  
!*****  
!                               SNAX macro for Token ring lines  
!*****  
TOKEN^RING^SNAX^MACRO = SNATS  
                    TYPE 58,  
                    SUBTYPE 4,  
                    RSIZE 1024,  
                    SUBTYPE 4,  
                    FRAMESIZE 1036 # ;  
  
!*****  
!                               SNAX MANAGER  
!*****  
SSCP^MACRO          = SNASVM  
                    TYPE 13,          SUBTYPE 5,  
                    RSIZE            256 #;  
  
!*****  
!                               LAN CONTROLLER  
!*****  
LAN1      3616    0,1    %130    ;  
  
!*****      Service manager  
SNAX      6999    0,1    %370    ;  
  
!*****      SNAX/Token Ring Pseudocontroller  
RING      6997    0,1    %360    ;  
  
!*****      Token Ring Line  
$CHAMB    LAN1.0, LAN1.1    C3613^MLAM, NAME #LAN1;  
  
!*****      Configure the SSCP  
$SSCP     SNAX.0, SNAX.1 SSCP^MACRO;
```


Configuration des communications

```
!***** 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;
```

Configuration de processus SNAX/APC

Les définitions suivantes configurent l'exemple de processus APC (nom de processus \$BP01) par l'intermédiaire de SCF, pour la ligne SNA.

Remarque : Le processus APC \$BP01 est défini comme un processus de classe de serveurs exécuté sur le même PATHWAY que le répartiteur SNAX APC.

```
==
== 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, SNANAME GBIBMIYA.IYAHT080, SNAXFILENAME $SNA02.#ZNTLU1, &
    MAXSESSION 256, AUTOSTART YES

== TPnames for MQSeries

ADD TPN IYAHT080.MQSeries

=== Spinach (HP) Partner LU

ADD PTNR-LU IYAHT080.IYABT0F0, SNANAME GBIBMIYA.IYABT0F0, &
    PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYABT0F0.LU62PS, MODENAME LU62PS, &
    DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
    DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
    DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
    SENDWINDOW 4

==
== Winas18 (AS400) Partner LU
==

ADD PTNR-LU IYAHT080.IYAFT110, SNANAME GBIBMIYA.IYAFT110, &
    PERIPHERAL-NODE NO, PARALLEL-SESSION-LU YES

ADD PTNR-MODE IYAHT080.IYAFT110.LU62PS, MODENAME LU62PS, &
    DEFAULTMAXSESSION 8, DEFAULTMINCONWINNER 4, &
    DEFAULTMINCONLOSER 3, MAXAUTOACT 1, RCVWINDOW 4, &
    DEFAULTMAXINRUSIZE 1024, DEFAULTMAXOUTRUSIZE 1024, &
    SENDWINDOW 4

==
== Stingray (AIX) Partner LU
```

Configuration des communications

```
==  
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, &
```

Configuration des communications

```
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 *
```

PourMQSeries, la valeur Maxapplio, qui permet de contrôler la taille maximale des transferts de données interprocessus entre MQSeries et le processus de serveur de communications, doit indiquer 32000 (c'est à dire plus que la valeur par défaut).

Définitions de canaux

Voici des exemples de définitions de canaux MQSeries, prenant en charge la configuration SNAX :

- Un canal émetteur vers MQSeries sur MVS/ESA (mécanisme de déplacement non-CICS) :

```
DEFINE CHANNEL(MT01.VM03.SDRC.0002) CHLTYPE(SDR) +
TRPTYPE(LU62) +
SEQWRAP(9999999) MAXMSGL(2048) +
XMITQ('VM03NCM.TQ.SDRC.0001') +
CONNAME('$BP01.IYAHT080.IYCNVM03') +
MODENAME('LU62PS') TPNAME(MQSERIES)
```

- Un canal récepteur depuis MQSeries sur MVS/ESA :

```
DEFINE CHANNEL(VM03.MT01.SDRC.0002) CHLTYPE(RCVR) +
TRPTYPE(LU62) REPLACE DESCR('Receiver channel from VM03NCM') +
SEQWRAP(9999999) +
MAXMSGL(2048)
```

Configuration des communications

- Un canal de serveur vers MQSeries sur MVS/ESA, capable d'entamer une conversation, ou d'être lancé depuis un canal demandeur éloigné :

```
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)
```

où MQSeries est le TPNAME écouté par le gestionnaire de files d'attente MVS™.

Exemple de communications ICE

La configuration de ICE pour MQSeries s'effectue en deux étapes :

1. Le processus ICE lui-même doit être configuré.
2. Les informations de ligne (\$ICE01, dans l'exemple suivant) et SNA doivent être entrées dans le processus ICE.

Configuration du processus ICE

Un exemple de configuration de processus ICE est présenté ci-dessous. Cette configuration figure par défaut dans un fichier nommé GOICE:

```
?tacl macro
clear all
param backupcpu 1
param cinittimer 120
param collector $0
param config icectl
param idblk 05d
param idnum 312FF
param cpname IYHR00C
param datapages 64
param dynamicrly yes
param genesis $gen
param maxrcv 32000
param loglevel info
param netname GBIBMIYA
param password xxxxxxxxxxxxxxxxxxxx
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/
```

Remarques :

1. Le mot de passe PARAM a été remplacé par xxxxxxxxxxxxxxxxxxxx.
2. Pour MQSeries, le PARAM maxrcv, qui permet de contrôler la taille maximale des transferts de données interprocessus entre MQSeries et le processus de serveur de communications, doit indiquer 32000 (c'est à dire plus que la valeur par défaut).

Définition des informations de ligne et APC

Une fois le processus ICE démarré avec cette configuration, les informations suivantes sont entrées dans le processus ICE à l'aide de la fonction NOF** (Node Operator Facility). Cet exemple définit une ligne nommée \$ICE01, exécutée sur le port en anneau à jeton \$CHAMB.#ICE :

```
==
== ICE definitions for PU IYHR00C.
== Local LU for this PU is IYAHT0C0.
==

ALLOW ERRORS

OPEN $ICE

ABORT LINE $ICE01, SUB ALL

DELETE LINE $ICE01, SUB ALL

==
== ADD TOKEN RING LINE
==

ADD LINE $ICE01, TNDM $CHAMB.#ICE, &
      IDBLK %H05D, &
      PROTOCOL TOKENRING, WRITEBUFFERSIZE 8192

==
== ADD PU OBJECT
==

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 IYAHT0C0
ADD APPL IYAHT0C0, ALIAS IYAHT0C0, PROTOCOL CPIC, &
      OPENNAME #IYAHT0C

==
== Add Mode LU62PS
==

DELETE MODE LU62PS
ADD MODE LU62PS, MAXSESS 8, MINCONWIN 4, MINCONLOS 3

==
== Add Partner LU Objects
==

== spinach (HP)

ABORT RLU IYABT0F0
DELETE RLU IYABT0F0
ADD RLU IYABT0F0, MODE LU62PS, PARSESS YES

== stingray (AIX)

ABORT RLU IYA3T995
DELETE RLU IYA3T995
ADD RLU IYA3T995, MODE LU62PS, PARSESS YES
```

Configuration des communications

```
== 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
```

Remarque : Pour que cette configuration puisse fonctionner, le port #ICE doit avoir été défini sur la ligne en anneau à jeton.

Par exemple, les commandes suivantes pourraient être saisies dans SCF :

```
add port $chamb.#ice, type tr8025, address %H08
start port $chamb.#ice
```

où \$chamb est le contrôleur de l'anneau à jeton, et où le SAP du port est %08.

Définitions de canaux pour ICE

Les définitions de canaux MQSeries suivantes pourraient prendre en charge cette configuration :

- Un canal émetteur vers MQSeries sur MVS/ESA (mécanisme de déplacement non-CICS) :

```
DEFINE CHANNEL(MT01.VM03.SDRC.ICE) CHLTYPE(SDR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.SDRC.ICE') +
  CONNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+
  MODENAME('LU62PS') TPNAME(MQSERIES)
```

- Un canal récepteur depuis MQSeries sur MVS/ESA :

```
DEFINE CHANNEL(VM03.MT01.SDRC.ICE) CHLTYPE(RCVR) +
  TRPTYPE(LU62) REPLACE DESCR('Receiver channel from VM03NCM') +
  SEQWRAP(9999999) +
  MAXMSGL(2048) +
  TPNAME(VM03NCMSDRCRCVR)
```

- Un canal de serveur vers MQSeries sur MVS/ESA, capable d'entamer une conversation, ou d'être lancé depuis un canal demandeur éloigné :

```
DEFINE CHANNEL(MT01.VM03.RQSV.ICE) CHLTYPE(SVR) +
  TRPTYPE(LU62) +
  SEQWRAP(9999999) MAXMSGL(2048) +
  XMITQ('VM03NCM.TQ.RQSV.ICE') +
  CONNAME('$ICE.#IYAHT0C.IYAHT0C0.IYCNVM03')+
  MODENAME('LU62PS') TPNAME(MQSERIES) +
```

où MQSeries est le TPNAME écouté par le gestionnaire de files d'attente MVS.

Exemple de communications TCP/IP

Cet exemple indique comment établir des communications avec un système MQSeries éloigné sur TCP/IP.

Strophe TCPConfig de QMINI

Le fichier QMINI doit contenir une strophe TCPConfig appropriée. Par exemple :

```
TCPConfig:
  TCPPort=1414
  TCPNumListenerPorts=1
  TCPListenerPort=1996
  TCPKeepAlive=1
```

La valeur TCPPort est le port de sortie par défaut pour les canaux dont la zone CONNAME n'est pas renseignée. TCPListenerPort identifie le port par défaut utilisé si l'option -p n'est pas indiquée lors de l'utilisation de la ligne de commande **runmqtsr**.

Configuration des communications

Définition d'un canal émetteur TCP/IP

Un canal émetteur TCP/IP doit être défini. Dans cet exemple, le gestionnaire de files d'attente est MH01 sur un hôte nommé SPINACH:

```
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)')
```

Ce canal tenterait de définir une liaison avec un port TCP/IP numéro 2000 sur l'hôte SPINACH.

L'exemple qui suit montre une définition de canal émetteur TCP/IP pour un gestionnaire de files d'attente MH01 de l'hôte SPINACH à l'aide du port TCP/IP sortant *par défaut* :

```
DEFINE CHANNEL(MT01_MH01_SDR_0001) CHLTYPE(SDR) +
  TRPTYPE(TCP) +
  SEQWRAP(9999999) MAXMSGL(4194304) +
  XMITQ('MH01_TQ_SDR_0001') +
  CONNAME('SPINACH.HURSLEY.IBM.COM')
```

Aucun numéro de port n'est indiqué dans le CONNAME. Par conséquent, la valeur indiquée dans l'entrée TCPPort du fichier QMINI (1414) est utilisée.

Définition d'un canal récepteur TCP/IP

Exemple de canal récepteur TCP/IP :

```
DEFINE CHANNEL(MH01_MT01_SDR_0001) CHLTYPE(RCVR) +
  TRPTYPE(TCP)
```

Un canal récepteur TCP/IP ne requiert aucune valeur CONNAME, mais un programme d'écoute TCP/IP doit être en cours d'exécution. Un programme d'écoute TCP/IP peut être démarré de deux façons. Soit :

1. En allant dans le PATHWAY du gestionnaire de files d'attente à l'aide de PATHCOM, et en entrant :

```
start server mqs-tcplis00
```

ou

en saisissant la commande suivante à l'invite TACL :

```
runmq1sr -m NomGestFiles
```

Un programme d'écoute TCP/IP, qui écouterait le port défini dans le fichier QMINI (1996 dans cet exemple) est démarré.

Configuration des communications

Remarque : Ce numéro de port peut être remplacé par l'indicateur *-p Port* dans la commande `runmqtsr`.

Définition d'un canal émetteur TCP/IP sur le système éloigné

La définition du canal émetteur du système éloigné, à connecter à ce canal récepteur pourrait se présenter comme suit :

```
DEFINE CHANNEL(MH01_MT01_SDR0001) CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  XMITQ('MT01_TQ_SDR0001') +  
  CONNAME('Compaq.ISC.UK.IBM.COM(1996)')
```

Configuration de QMINI pour prendre en charge plusieurs programmes d'écoute TCP/IP

Pour permettre à un gestionnaire de files d'attente de prendre en charge plusieurs programmes d'écoute TCP/IP, vous devez créer une nouvelle classe de serveurs `PATHWAY` pour chaque programme d'écoute supplémentaire, basé sur `MQS-TCPLIS00`.

De plus, chaque programme d'écoute TCP/IP doit disposer de sa propre entrée de port d'écoute dans la strophe `TCPCConfig` du fichier `QMINI`.

Par exemple :

```
TCPCConfig:  
  TCPPort=1414  
  TCPNumListenerPorts=3  
  TCPLListenerPort=1996  
  TCPLListenerPort=1997  
  TCPLListenerPort=1998  
  TCPKeepAlive=1
```

`TCPNumListenerPorts` doit correspondre au nombre d'entrées `TCPLListenerPort` (trois dans cet exemple). Ce fichier `QMINI` est capable d'assurer le support de trois programmes d'écoute TCP/IP pour les ports 1996, 1997 et 1998. Les classes de serveurs prenant en charge ces trois ports sont habituellement nommées `MQS-TCPLIS00`, `MQS-TCPLIS01`, et `MQS-TCPLIS02`.

Pour plus d'informations sur l'ajout de classes de serveurs, consultez la section «Administration TS/MP (PATHWAY)» à la page 32.

Annexe N. Clients MQSeries

Un client MQSeries est un système MQSeries qui n'intègre aucun gestionnaire de files d'attente. Le code client MQSeries dirige les appels MQI émis par les applications exécutées sur le système client vers un gestionnaire de files d'attente situé sur un serveur MQSeries auquel il est connecté.

Cette annexe fournit des informations sur les clients MQSeries, spécifiques à MQSeries pour Compaq NSK V5.1. Utilisez-la avec le manuel *MQSeries - Clients*.

Prise en charge des clients

MQSeries pour Compaq NSK peut jouer le rôle de serveur MQSeries pour tous les clients MQSeries capables de se connecter au serveur sous le protocole TCP/IP ou SNA LU 6.2. Toutefois, il n'existe pas de client MQSeries pour Compaq NSK.

Lorsqu'un client MQSeries se connecte à un gestionnaire de files d'attente dans MQSeries pour Compaq NSK :

- Tout appel MQGET, MQPUT, ou MQPUT1 avec option MQ*_SYNCPOINT émet une transaction Compaq, si aucune autre n'est déjà associée au descripteur de connexion.
- Tout appel MQGET, MQPUT, ou MQPUT1 sans option MQ*_SYNCPOINT ou MQ*_NO_SYNCPOINT émet une transaction Compaq, si aucune autre n'est déjà associée au descripteur de connexion.
- L'appel MQCMIT valide une transaction Compaq, si aucune autre n'a été associée au descripteur de connexion. L'appel MQBACK annule la transaction Compaq, si une transaction est associée au descripteur de connexion.

Dans tous les cas, si BEGINTRANSACTION échoue, le *CompCode* MQCC_FAILED, et la *Reason* MQRC_SYNCPOINT_NOT_AVAILABLE sont retournés au demandeur.

Gestion de la sécurité

MQSeries pour Compaq NSK assure le support des exits de sécurité des canaux pour la validation des clients. Ce support s'effectue comme suit :

- Une fois la connexion établie entre le client MQSeries et le serveur, le client appelle l'exit de sécurité du serveur avant de procéder au retour de l'appel MQCONN.
- L'exit de sécurité du serveur peut retourner les informations à l'exit de sécurité du client.

Ce dialogue permet par exemple la communication de données confidentielles entre le serveur et le client. Si le client n'a défini aucun exit de sécurité, les valeurs des variables d'environnement MQ_USER_ID et MQ_PASSWORD sont transmises au serveur par l'intermédiaire des attributs de canaux. Ces attributs sont à la disposition de l'exit de sécurité du serveur pour validation.

Clients MQSeries

Annexe O. Programmation de la gestion du système

MQSeries pour Compaq NSK supporte les fonctions suivantes de gestion de système de MQSeries :

- Evénements outils
- Commandes programmables PCF
- Services optionnels.

Cet annexe présente un résumé de ces fonctions, incluses dans MQSeries pour Compaq NSK. Pour plus d'informations, reportez-vous au manuel *MQSeries Programmable System Management*.

Evénements outils

MQSeries pour Compaq NSK assure le support des événements outil standard de MQSeries, qui entraînent l'insertion d'un message d'événement dans une file d'attente des événements.

Pour activer ou désactiver des événements, vous définissez les valeurs des attributs du gestionnaire de files d'attente et de la file d'attente, avec :

- MQSC, comme il est indiqué dans le manuel *MQSeries - Guide de référence des commandes MQSC*
- Des commandes PCF, comme il est indiqué dans le manuel *MQSeries Programmable System Management*
- MQM (Message Queue Management), comme il est indiqué au «Chapitre 4. Gestion des gestionnaires de files d'attente» à la page 45

Types d'événements acceptés par MQSeries pour Compaq NSK

MQSeries pour Compaq NSK accepte les types d'événements suivants :

Tableau 27. Types d'événements acceptés par MQSeries pour Compaq NSK

Type d'événement	Nom de l'événement
Evénements Droits d'accès	Non autorisé (Not Authorized) type 1
Evénements Canal	Canal activé (Channel Activated) Erreur de conversion de canal (Channel Conversion Error) Canal non activé (Channel Not Activated) Canal démarré (Channel Started) Canal arrêté (Channel Stopped)
Evénements Interdiction	Get interdit (Get Inhibited) Put interdit (Put Inhibited)
Evénements Locaux	Erreur de type de file d'attente de base Erreur de type de file d'attente File d'attente de base inconnue Nom d'objet inconnu

événements

Tableau 27. Types d'événements acceptés par MQSeries pour Compaq NSK (suite)

Type d'événement	Nom de l'événement
Événements Performances	Longueur de file d'attente trop élevée Longueur de file d'attente trop basse File d'attente saturée Intervalle de maintenance de file d'attente trop élevé Intervalle de maintenance de file d'attente correct
Événements éloignés	Erreur de type de file d'attente de transmission par défaut Erreur d'utilisation de file d'attente de transmission par défaut Erreur de type de file d'attente Erreur de nom de file d'attente éloignée Erreur d'utilisation de file d'attente de transmission File d'attente de transmission par défaut inconnue Gestionnaire de file d'attente éloigné inconnu File d'attente de transmission inconnue
Événements Start et Stop	Gestionnaire de file d'attente actif Gestionnaire de file d'attente inactif

Format de message d'événement

MQSeries pour Compaq NSK supporte le format standard des messages d'événement de MQSeries. Le message d'événement est en deux parties, le *descripteur de message* (MQMD) et les *données du message*. La partie données comporte un en-tête d'événement et des données particulières au type d'événement.

La structure d'un message d'événement (MQMD) est résumée dans la section «MQMD – Descripteur de message» à la page 341. La structure de l'en-tête (MQCFH) est résumée dans le tableau 29 à la page 399.

Tableau 28. Structure d'un message d'événement (MQMD)

Paramètre	Type	Valeur
<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	Codage employé par le gestionnaire de files d'attente qui a généré l'événement.
<i>CodedCharSetId</i>	MQLONG	CCSID (ID de jeu de caractères) utilisé par le gestionnaire de files d'attente qui a généré l'événement.
<i>Format</i>	MQCHAR8	MQFMT_EVENT
<i>Priority</i>	MQLONG	Priorité par défaut de la file d'attente des événements (s'il s'agit d'une file d'attente locale) ou sa définition locale dans le gestionnaire de files d'attente qui a généré l'événement.

Tableau 28. Structure d'un message d'événement (MQMD) (suite)

Paramètre	Type	Valeur
<i>Persistence</i>	MQLONG	Persistance par défaut de la file d'attente des événements (s'il s'agit d'une file d'attente locale) ou sa définition locale dans le gestionnaire de files d'attente qui a généré l'événement.
<i>MsgId</i>	MQBYTE24	Cette valeur est générée par le gestionnaire de files d'attente, elle est unique.
<i>CorrelId</i>	MQBYTE24	MQCI_NONE
<i>BackoutCount</i>	MQLONG	Cette valeur est toujours égale à 0.
<i>ReplyToQ</i>	MQCHAR48	Toujours vide.
<i>ReplyToQMgr</i>	MQCHAR48	Le nom du gestionnaire de files d'attente du système d'origine.
<i>UserIdentifier</i>	MQCHAR12	Toujours vide.
<i>AccountingToken</i>	MQBYTE32	MQACT_NONE
<i>ApplIdentityData</i>	MQCHAR32	Toujours vide.
<i>PutApplType</i>	MQLONG	Le type de l'application qui insère le message.
<i>PutApplName</i>	MQCHAR28	Le nom de l'application qui insère le message.
<i>PutDate</i>	MQCHAR8	La date d'insertion du message, générée par le gestionnaire de files d'attente.
<i>PutTime</i>	MQCHAR8	L'heure d'insertion du message, générée par le gestionnaire de files d'attente.
<i>ApplOriginData</i>	MQCHAR4	Toujours vide.

Tableau 29. Structure de l'en-tête (MQCFH)

Paramètre	Type	Valeur
<i>Type</i>	MQLONG	MQCFT_EVENT
<i>StrucLength</i>	MQLONG	MQCFH_STRUC_LENGTH
<i>Version</i>	MQLONG	MQCFH_VERSION_1
<i>Command</i>	MQLONG	MQCMD_Q_MGR_EVENT MQCMD_PERFM_EVENT MQCMD_CHANNEL_EVENT
<i>MsgSeqNumber</i>	MQLONG	Toujours égal à 1.
<i>Control</i>	MQLONG	MQCFC_LAST
<i>CompCode</i>	MQLONG	MQCC_OK MQCC_WARNING
<i>Reason</i>	MQLONG	Code anomalie identifiant l'événement.
<i>ParameterCount</i>	MQLONG	Le nombre de structures de paramètres qui suivent la structure MQCFH.

Commandes programmables PCF

MQSeries pour Compaq NSK supporte les fonctions PCF standard, telles qu'elles sont décrites dans le manuel *MQSeries Programmable System Management*. Un message PCF est en deux parties, le *descripteur de message* (MQMD) et les *données du message*. La partie données comporte un en-tête PCF (MQCFH) et des paramètres PCF, définis par les structures MQCFIN, MQCFIL, MQCFST et MQCFSL.

La structure du descripteur de message PCF (MQMD) est résumée dans le tableau 30. La structure de l'en-tête PCF (MQCFH) est résumée dans le tableau 31. Les structures des paramètres PCF sont résumées du tableau 32 à la page 401 au tableau 35 à la page 402.

Descripteur de message PCF

Le descripteur standard de messages PCF s'applique à MQSeries pour Compaq NSK. Il contient les champs suivants :

Tableau 30. Descripteur de message PCF

Champ	Valeur
<i>Report</i>	Toute valeur valide
<i>MsgType</i>	MQMT_REQUEST
<i>Expiry</i>	Toute valeur valide
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	Méthode de codage utilisée pour les données du message ; si nécessaire, une conversion sera effectuée.
<i>CodedCharSetId</i>	CCSID utilisé pour les données du message ; si nécessaire, une conversion sera effectuée.
<i>Format</i>	MQFMT_ADMIN MQFMT_PCF (pour les données utilisateur)
<i>Priority</i>	Toute valeur valide
<i>Persistence</i>	Toute valeur valide
<i>MsgId</i>	Toute valeur valide, y compris MQMI_NONE
<i>CorrelId</i>	Toute valeur valide, y compris MQMI_NONE
<i>ReplyToQ</i>	Nom de file d'attente
<i>ReplyToQMgr</i>	Nom du gestionnaire de files d'attente
Champs de contexte de message	Toute valeur valide, y compris MQPMO_DEFAULT_CONTEXT

En-tête PCF (MQCFH)

Le descripteur standard de l'en-tête PCF s'applique à MQSeries pour Compaq NSK. Il contient les champs suivants :

Tableau 31. En-tête PCF

Champ	Type	Valeur
<i>Type</i>	MLONG	MQCFT_COMMAND MQCFT_RESPONSE MQCFT_EVENT
<i>StrucLength</i>	MLONG	MQCFH_STRUC_LENGTH
<i>Version</i>	MLONG	MQCFH_VERSION_1

Tableau 31. En-tête PCF (suite)

Champ	Type	Valeur
<i>Command</i>	MQLONG	Identificateur de commande valide.
<i>MsgSeqNumber</i>	MQLONG	Numéro de séquence du message.
<i>Control</i>	MQLONG	MQCFC_LAST MQCFC_NOT_LAST
<i>CompCode</i>	MQLONG	MQCC_OK MQCC_WARNING MQCC_FAILED MQCC_UNKNOWN
<i>Reason</i>	MQLONG	Code anomalie identifiant le code d'achèvement.
<i>ParameterCount</i>	MQLONG	Nombre de structures de paramètre.

Paramètre chaîne PCF (MQCFST)

La structure standard de chaîne PCF (MQCFST) s'applique à MQSeries pour Compaq NSK. Elle contient les champs suivants :

Tableau 32. Paramètre PCF chaîne

Champ	Type	Valeur
<i>Type</i>	MQLONG	MQCFT_STRING
<i>StrucLength</i>	MQLONG	Longueur de la structure MQCFST, en octets.
<i>Parameter</i>	MQLONG	Identificateur de paramètre.
<i>CodedCharSetId</i>	MQLONG	Identificateur de jeu de caractères codés (CCSID).
<i>StringLength</i>	MQLONG	Longueur des données dans le champ <i>String</i> , en octets.
<i>String</i>	MQCHAR × <i>StringLength</i>	Contenu de la chaîne.

Paramètre PCF liste d'entiers (MQCFIL)

La structure standard de liste d'entiers PCF (MQCFIL) s'applique à MQSeries pour Compaq NSK. Elle contient les champs suivants :

Tableau 33. Liste d'entiers PCF

Champ	Type	Valeur
<i>Type</i>	MQLONG	MQCFT_INTEGER_LIST
<i>StrucLength</i>	MQLONG	Longueur de la structure MQCFIL, en octets.
<i>Parameter</i>	MQLONG	Identificateur de paramètre.
<i>Count</i>	MQLONG	Nombre d'éléments du tableau <i>Values</i> .
<i>Values</i>	MQLONG × <i>Count</i>	Tableau d'entiers.

Entier PCF (MQCFIN)

La structure standard d'entier PCF (MQCFIN) s'applique à MQSeries pour Compaq NSK. Elle contient les champs suivants :

Tableau 34. Entier PCF

Champ	Type	Valeur
<i>Type</i>	MQLONG	MQCFT_INTEGER

Tableau 34. Entier PCF (suite)

Champ	Type	Valeur
<i>StrucLength</i>	MLONG	MQCFIN_STRUC_LENGTH
<i>Parameter</i>	MLONG	Identificateur de paramètre
<i>Value</i>	MLONG	Valeur du paramètre

Liste de chaînes PCF (MQCFSL)

La structure standard de liste de chaînes PCF (MQCFSL) s'applique à MQSeries pour Compaq NSK. Elle contient les champs suivants :

Tableau 35. liste de chaînes PCF

Champ	Type	Valeur
<i>Type</i>	MLONG	MQCFT_STRING_LIST
<i>StrucLength</i>	MLONG	Longueur de la structure MQCFSL, en octets.
<i>Parameter</i>	MLONG	Identificateur de paramètre
<i>CodedCharSetId</i>	MLONG	CCSID utilisé pour les données contenues dans le champ <i>Strings</i> .
<i>Count</i>	MLONG	Nombre de chaînes du champ <i>Strings</i> .
<i>StringLength</i>	MLONG	Longueur de chaque chaîne du champ <i>Strings</i> , en octets.
<i>Strings</i>	MQCHAR × <i>StringLength</i> × <i>Count</i>	Liste de chaînes pour le paramètre identifié par le champ <i>Parameter</i> .

Commandes PCF prises en charge par MQSeries pour Compaq NSK

MQSeries pour Compaq NSK prend en charge les commandes PCF MQSeries suivantes. Consultez le manuel *MQSeries Programmable System Management* pour une description complète de ces commandes.

Tableau 36. Commandes PCF prises en charge par MQSeries pour Compaq NSK

Commande	Identificateur
Changement de canal	MQCMD_CHANGE_CHANNEL
Changement de liste de noms	MQCMD_CHANGE_NAMELIST
Changement de processus	MQCMD_CHANGE_PROCESS
Changement de file d'attente	MQCMD_CHANGE_Q
Changement de gestionnaire de file d'attente	MQCMD_CHANGE_Q_MGR
Effacement de la file d'attente	MQCMD_CLEAR_Q
Copie de canal	MQCMD_COPY_CHANNEL
Copie de liste de noms	MQCMD_COPY_NAMELIST
Copie de processus	MQCQ_COPY_PROCESS
Copie de file d'attente	MQCMD_COPY_Q
Création de canal	MQCMD_CREATE_CHANNEL
Création de liste de noms	MQCMD_CREATE_NAMELIST

Tableau 36. Commandes PCF prises en charge par MQSeries pour Compaq NSK (suite)

Commande	Identificateur
Création de processus	MQCMD_CREATE_PROCESS
Création de file d'attente	MQCMD_CREATE_Q
Suppression de canal	MQCMD_DELETE_CHANNEL
Suppression de liste de noms	MQCMD_DELETE_NAMELIST
Suppression de processus	MQCMD_DELETE_PROCESS
Suppression de file d'attente	MQCMD_DELETE_Q
Echappement	MQCMD_ESCAPE
Demande de canal	MQCMD_INQUIRE_CHANNEL
Demande de noms de canal	MQCMD_INQUIRE_CHANNEL_NAMES
Demande de statut de canal	MQCMD_INQUIRE_CHANNEL_STATUS
Demande de gestionnaire de file d'attente de cluster	MQCMD_INQUIRE_CLUSTER_Q_MGR
Demande de liste de noms	MQCMD_INQUIRE_NAMELIST
Demande de processus	MQCMD_INQUIRE_PROCESS
Demande de noms de processus	MQCMD_INQUIRE_PROCESS_NAMES
Demande de file d'attente	MQCMD_INQUIRE_Q
Demande de gestionnaire de file d'attente	MQCMD_INQUIRE_Q_MGR
Demande de noms de file d'attente	MQCMD_INQUIRE_Q_NAMES
Ping de canal	MQCMD_PING_CHANNEL
Ping de gestionnaire de file d'attente	MQCMD_PING_Q_MGR
Régénération de cluster	MQCMD_REFRESH_CLUSTER
Remise à zéro de canal	MQCMD_RESET_CHANNEL
Remise à zéro des statistiques de file d'attente	MQCMD_RESET_Q_STATS
Conversion de canal	MQCMD_RESOLVE_CHANNEL
Reprise de gestionnaire de file d'attente de cluster	MQCMD_RESUME_Q_MGR_CLUSTER
Démarrage de canal	MQCMD_START_CHANNEL
Arrêt de canal	MQCMD_STOP_CHANNEL
Interruption de gestionnaire de file d'attente de cluster	MQCMD_SUSPEND_Q_MGR_CLUSTER

Remarque : MQSeries pour Compaq NSK ne prend pas en charge les commandes Démarrage d'initialisateur de canal (Start Channel Initiator) et Démarrage de programme d'écoute de canal (Start Channel Listener).

Réponses des commandes PCF

Dans MQSeries pour Compaq NSK, le serveur de commandes génère des messages standard de réponse pour chaque commande PCF. Elles sont de trois types :

- OK
- Erreur
- Données

Pour plus d'informations, reportez-vous au manuel *MQSeries Programmable System Management*.

Services optionnels

MQSeries pour Compaq NSK assure les service d'autorisation et d'annuaire.

Interface du service d'autorisation

Les gestionnaires de file d'attente utiliseront le service d'autorisation pour effectuer les fonctions d'autorisation. Par exemple, pour vérifier qu'un ID utilisateur donné a le droit d'ouvrir une file d'attente.

MQSeries pour Compaq NSK est livré avec un composant d'autorisation. Ce composant est l'OAM (Object Authority Manager). L'OAM est activé par défaut, et s'utilise par les commandes de contrôle **dspmqa**t (affichage des droits d'accès) et **setmqa**t (définition des droits d'accès).

Vous pouvez compléter ou remplacer l'OAM avec votre propre composant de service d'autorisation, ainsi que l'explique le manuel *MQSeries Programmable System Management*.

Interface de service annuaire

Le service annuaire sert à convertir le nom du gestionnaire de files d'attente propriétaire d'une file d'attente.

L'interface de service annuaire, décrite dans le manuel *MQSeries Programmable System Management*, est prise en charge par MQSeries pour Compaq NSK.

Annexe P. Modèle de fichier d'événement EMS utilisé par MQSeries pour Compaq NSK

Le fichier modèle EMS (SMQSTMPL) contient le code source pour la définition des événements EMS de MQSeries. Ces définitions contrôlent la façon dont les informations contenues dans les messages d'événements EMS sont affichées. Elles expliquent également le type et la signification des données contenues dans chaque message d'événement EMS.

Les types d'événements suivants sont générés :

ZMQS-VAL-EVT-ERROR

Un FFST (problème de ressource système, problème logiciel ou incident matériel).

ZMQS-VAL-EVT-ERR

Une erreur de MQSeries, faisant référence à une événement FFST et à des données enregistrées sur le disque dur.

ZMQS-VAL-EVT-MSG

Un message MQSeries, par exemple le démarrage d'un gestionnaire de files d'attente ou d'un canal. Tous ces événements correspondent à un message de journalisation de MQSeries, AMQxxxxx, et contiennent les mêmes informations et texte. Les données variables de chaque message sont contenues par le message d'événement, dans des éléments individuels. Pour plus de détails sur les messages AMQxxxxx, consultez le manuel *MQSeries - Guide des messages*.

ZMQS-VAL-EVT-QMGR

Un événement du gestionnaire de files d'attente, de type droits (authority), désactivation (inhibit), local, éloigné (remote), démarrage et arrêt. Ces événements EMS ont en réalité le même contenu que les messages d'événement PCF correspondants, qui sont décrits dans le manuel *MQSeries Programmable System Management*. Les données variables sont contenues par des éléments individuels de chaque message.

ZMQS-VAL-EVT-PERF

Un événement de performances, en correspondance avec les événements standard du même type de MQSeries. Ces événements apportent des statistiques concernant les files d'un gestionnaire de files d'attente. Les données variables de chaque événement de performances sont contenues par le message d'événement, dans des éléments individuels.

ZMQS-VAL-EVT-CHNL

Un événement de canal, en correspondance avec les événements standard du même type de MQSeries. Ces événements Canal indiquent des modifications du statut des canaux, ou des problèmes de communication entre les gestionnaires de file d'attente. Tout comme pour les autres types de message d'événement, les données variables sont contenues dans des éléments individuels de chaque message.

Voici une partie du fichier modèle EMS :

```
VERSION: "IBM.MQS - 10JAN97"  
SSID: ZMQS-VAL-SSID  
SSNAME: "MQSeries", "MQS"
```

Modèle de fichier d'événement EMS

```

==
== 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 :
      " <*CR> <2>
      "
      "<*CR> ACTION :
      " <*CR> <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>
      "<*IF 13><*CR> Object QMgr : <14>
      "<*IF 9><*CR> Options : <10>
      "<*IF 11><*CR> Command : <12>
      "<*IF 15><*CR> Queue Name : <16>
      "<*IF 17><*CR> Queue Type : <18>

```

Modèle de fichier d'événement EMS

```
"<*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>                           "
"<*CR> Auxiliary rc 1 : <15>                         "
"<*CR> Auxiliary rc 2 : <17>                         "
"<*CR> CCSID 1 : <19>                                "
"<*CR> Auxiliary string 1 : <21>                    "
"<*CR> CCSID 2 : <23>                                "
```

Modèle de fichier d'événement EMS

```
"<*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
```

Annexe Q. Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing
IBM Europe Middle-East Africa
Tour Descartes
La Défense 5
2, avenue Gambetta
92066 - Paris-La Défense CEDEX
France

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LE PRESENT DOCUMENT EST LIVRE EN L'ETAT. IBM DECLINE TOUTE RESPONSABILITE, EXPLICITE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE VALEUR MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les produits et logiciels décrits dans ce document.

Remarques

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation IBM.

Marques

Les termes qui suivent sont des marques d'IBM Corporation dans certains pays :

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 et NonStop sont des marques de Compaq Computer Corporation.

Intel est une marque d'Intel.

Microsoft, Windows et Windows NT sont des marques de Microsoft Corporation.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

UNIX est une marque enregistrée dans certains pays, dont seule la société X/Open Limited Company peut concéder la licence.

D'autres sociétés sont propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

Bibliographie

Cette section décrit la documentation disponible pour toute la gamme des produits MQSeries.

Publications multi-plateformes MQSeries

La plupart de ces publications, parfois appelées manuels de la MQSeries «gamme» concernent l'ensemble des produits MQSeries. Les derniers produits MQSeries sont les suivants :

- MQSeries pour AIX, version 5.2
- MQSeries for AS/400, version 5.2
- MQSeries pour AT&T GIS UNIX, version 2.2
- MQSeries pour Compaq OpenVMS Alpha, V5.1
- MQSeries pour Compaq Tru64 UNIX, version 5.1
- MQSeries pour HP-UX, version 5.2
- MQSeries pour Linux V5.2
- MQSeries pour OS/2 Warp, version 5.1
- MQSeries pour OS/390, version 5.2
- MQSeries pour SINIX et DC/OSx, version 2.2
- MQSeries pour Sun Solaris, version 5.2
- MQSeries pour Sun Solaris, Edition plateforme Intel, version 5.1
- MQSeries pour Compaq NSK, V5.1
- MQSeries pour VSE/ESA, V2.1.1
- MQSeries pour Windows, version 2.0
- MQSeries pour Windows, version 2.1
- MQSeries pour Windows NT et Windows 2000, version 5.2

Les publications multi-plateformes MQSeries sont les suivantes :

- *Brochure MQSeries*, G511-1908
- *An Introduction to Messaging and Queuing*, GC11-1091
- *MQSeries Intercommunication*, SC33-1872
- *MQSeries Queue Manager Clusters*, SC34-5349
- *MQSeries - Clients*, GC11-1099
- *MQSeries - Administration du système*, GC11-1219
- *MQSeries - Guide de référence des commandes MQSC*, GC11-1526
- *MQSeries Event Monitoring*, SC34-5760
- *MQSeries Programmable System Management*, SC33-1482
- *MQSeries Administration Interface Programming Guide and Reference*, SC34-5390
- *MQSeries - Guide des messages*, GC11-1218

- *MQSeries - Guide de programmation d'applications*, SC11-1087
- *MQSeries Application Programming Reference*, SC33-1673
- *MQSeries Programming Interfaces Reference Summary*, SX33-6095
- *MQSeries - Utilisation de C++*, SC33-1877
- *MQSeries - Utilisation de Java*, SC11-1511
- *MQSeries Application Messaging Interface*, SC34-5604

Publications MQSeries spécifiques d'une plateforme

Outre les manuels de la gamme MQSeries, il existe au moins une publication spécifique à chaque plateforme pour chaque produit MQSeries.

MQSeries pour AIX, version 5.2

MQSeries pour AIX Mise en route, GC11-1213

MQSeries for AS/400, version 5.2

MQSeries pour AS/400 Mise en route, GC11-1588

MQSeries for AS/400 System Administration, SC34-5558

MQSeries for AS/400 Application Programming Reference (RPG), SC34-5559

MQSeries pour AT&T GIS UNIX, version 2.2

MQSeries for AT&T GIS UNIX System Management Guide, SC33-1642

MQSeries pour Compaq OpenVMS Alpha, V5.1

MQSeries pour Compaq OpenVMS Alpha - Mise en route, GC11-1797

MQSeries pour Compaq OpenVMS Alpha - Administration du système, SC11-1798

MQSeries pour Compaq NSK, V5.1

MQSeries pour Compaq NSK - Mise en route, GC11-1795

MQSeries pour Compaq NSK - Administration du système, SC11-1796

MQSeries pour Compaq Tru64 UNIX, version 5.1

Bibliographie

*MQSeries pour Compaq Tru64 UNIX
Mise en route, GC11-1697*

MQSeries pour HP-UX, version 5.2

*MQSeries pour HP-UX Mise en route,
GC11-1215*

MQSeries pour Linux V5.2

*MQSeries pour Linux - Mise en route,
GC11-1722*

MQSeries pour OS/2 Warp, version 5.1

*MQSeries pour OS/2 Warp Mise en route,
GC11-1214*

MQSeries pour OS/390, version 5.2

*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 pour R/3, Version 1.2

*MQSeries link for R/3 User's Guide,
GC11-1445*

MQSeries pour SINIX et DC/OSx, version 2.2

*MQSeries for SINIX and DC/OSx System
Management Guide, GC33-1768*

MQSeries pour Sun Solaris, version 5.2

*MQSeries pour Sun Solaris Mise en
route, GC11-1216*

MQSeries pour Sun Solaris, Edition plateforme Intel, version 5.1

*MQSeries for Sun Solaris, Intel Platform
Edition Quick Beginnings, GC34-5851*

MQSeries pour VSE/ESA, V2.1.1

*MQSeries for VSE/ESA Licensed Program
Specifications, GC34-5365*

*MQSeries for VSE/ESA System
Management Guide, GC34-5364*

MQSeries pour Windows, version 2.0

*MQSeries for Windows User's Guide,
GC11-1170*

MQSeries pour Windows, version 2.1

*MQSeries for Windows User's Guide,
GC11-1289*

MQSeries pour Windows NT et Windows 2000, version 5.2

*MQSeries pour Windows NT et Windows
2000 Mise en route, GC11-1509*

*MQSeries pour Windows NT Utilisation
de l'interface COM (Component Object
Model), SC11-1510*

*MQSeries LotusScript Extension,
SC34-5404*

Manuel sous forme électronique

La plupart des manuels MQSeries sont proposés à la fois en version papier et sous forme électronique.

format HTML

La documentation MQSeries est livrée au format HTML avec les produits MQSeries suivants :

- MQSeries pour AIX, version 5.2
- MQSeries for AS/400, version 5.2
- MQSeries pour Compaq OpenVMS Alpha, V5.1
- MQSeries pour Compaq Tru64 UNIX, version 5.1
- MQSeries pour HP-UX, version 5.2
- MQSeries pour Linux V5.2
- MQSeries pour OS/2 Warp, version 5.1
- MQSeries pour OS/390, version 5.2
- MQSeries pour Sun Solaris, version 5.2
- MQSeries pour Sun Solaris, Edition plateforme Intel, version 5.1
- MQSeries pour Windows NT et Windows 2000, version 5.2 (HTML compilé)
- MQSeries link pour R/3, V1.2

Les manuels MQSeries sont aussi disponibles au format HTML sur le site Web consacré à la gamme de produits MQSeries :

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

Format PDF (Portable Document Format)

Les fichiers de type PDF peuvent être visualisés et imprimés à l'aide de Adobe Acrobat Reader.

Pour vous procurer Adobe Acrobat Reader ou obtenir les informations les plus récentes sur les plateformes prenant en charge ce produit, accédez à l'adresse suivante sur le Web :

<http://www.adobe.com/>

Les manuels MQSeries sont livrés au format PDF pour les produits MQSeries suivants :

- MQSeries pour AIX, version 5.2
- MQSeries for AS/400, version 5.2
- MQSeries pour Compaq OpenVMS Alpha, V5.1
- MQSeries pour Compaq Tru64 UNIX, version 5.1
- MQSeries pour HP-UX, version 5.2
- MQSeries pour Linux V5.2
- MQSeries pour OS/2 Warp, version 5.1
- MQSeries pour OS/390, version 5.2
- MQSeries pour Sun Solaris, version 5.2
- MQSeries pour Sun Solaris, Edition plateforme Intel, version 5.1
- MQSeries pour Windows NT et Windows 2000, version 5.2
- MQSeries link pour R/3, V1.2

La totalité de la documentation MQSeries est également disponible au format PDF sur le site Web consacré à la gamme des produits MQSeries :

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

Format BookManager

La bibliothèque MQSeries est fournie au format IBM BookManager dans une série de bibliothèques en ligne, y compris la série *Transaction Processing and Data*, SK2T-0730. Pour visualiser la documentation en ligne dans un format IBM BookManager utilisez les logiciels IBM suivants :

- BookManager READ/2
- BookManager READ/6000
- BookManager READ/DOS
- BookManager READ/MVS
- BookManager READ/VM
- BookManager READ pour Windows

Format PostScript

La bibliothèque MQSeries est livrée au format PostScript (.PS) avec de nombreux produits MQSeries Version 2. Les manuels PostScript peuvent être imprimés sur une imprimante PostScript ou visualisés à l'aide d'un programme approprié.

Format d'aide Windows

Le manuel *MQSeries pour Windows - Guide de l'utilisateur* est fourni sous forme d'aide Windows pour les versions MQSeries pour Windows, Version 2.0 et MQSeries pour Windows, Version 2.1.

Informations sur MQSeries disponibles sur Internet

Le site Web consacré à la gamme de produits MQSeries se trouve à l'adresse suivante :
<http://www.ibm.com/software/mqseries/>

A partir de ce site, vous pouvez :

- obtenir les informations les plus récentes sur la gamme des produits MQSeries.
- accéder aux manuels MQSeries au format HTML ou PDF.
- télécharger un MQSeries SupportPac.

Publications connexes

- *SNAX/APC Planning and Configuration Manual*, (Compaq Part No. 098289)
SNAX/APC supporte LU 6.2 pour la mise en oeuvre du SNA. Ce manuel fournit des instructions d'installation et de configuration pour SNAX/APC.
- *SCF Reference Manual for SNAX/APC*, (Compaq Part No. 064525)
SNAX/APC supporte LU 6.2 pour la mise en oeuvre du SNA. Ce guide explique le fonctionnement de l'interface interactive SCF qui permet aux opérateurs et aux administrateurs du réseau de configurer et contrôler SNAX/APC.
- *Pathway System Management Guide*, (Compaq Part No. 096881)
Ce guide fournit des instructions pour la configuration et le contrôle des systèmes de traitement des transactions de chemin d'accès.
- *Introduction to NonStop Transaction Manager/MP (TM/MP)*, (Compaq Part No. 085812)
Ce guide explique comment utiliser le sous-système TMF pour protéger vos transactions commerciales et garantir l'intégrité de vos bases de données.
- *Introduction to Compaq Networking and Data Communications*, (Compaq Part No. 093148)
Ce manuel présente les manuels, produits, tâches et concepts de communication de données et de mise en réseau de Compaq.
- *Intersystem Communications Environment (ICE) Installation Guide*, (Version 3 Release 2, or later edition)
Ce guide explique comment installer l'ICE et comment configurer les paramètres de

Publications connexes

lancement de l'ICE. (ICE supporte LU 6.2 pour la mise en oeuvre de SNA par Insessions.)

- *Intersystem Communications Environment (ICE) Administrator's Guide*, (Version 3 Release 2, or later edition)

Ce guide explique comment configurer et utiliser l'ICE, ses interfaces, et ses utilitaires.

Glossaire

Ce glossaire définit les termes et les abréviations MQSeries utilisés dans ce manuel. Si vous ne trouvez pas le terme que vous recherchez, reportez-vous à l'index ou à la *Terminologie du traitement de l'information*, GCF2-0076, publiée par IBM France, ou encore à un dictionnaire d'informatique générale.

Certains termes et leur définition ont été repris et adaptés, pour la traduction française, de l'ouvrage *American National Dictionary for Information Systems*, ANSI X3.172-1990-1990, copyright 1990, American National Standards Institute (ANSI). Vous pouvez vous le procurer auprès de l'American National Standards Institute, 11 West 42 Street, New York, New York 10036. Les définitions sont suivies de la mention (A).

A

abandon : Synonyme de *annulation*.

agent MCA : Programme qui transmet des messages d'une file d'attente de transmission vers une liaison ou d'une liaison vers une file d'attente cible. Voir aussi *interface de file d'attente de messages*.

alerte : Message envoyé à un point focal des services de gestion d'un réseau pour identifier un incident survenu ou sur le point de survenir.

annulation : Opération qui annule toutes les modifications effectuées durant l'unité de reprise ou de travail en cours. Une fois l'opération terminée, une nouvelle unité de reprise ou de travail s'amorce.

annulation monophasé : Méthode dans laquelle aucune tâche en cours ne peut être terminée et toutes les modifications en cours doivent être annulées.

APAR : Sigle de Authorized Program Analysis Report (voir *rapport officiel d'analyse de programme*).

APPC : Communication évoluée de programme à programme.

application client : Application s'exécutant sur un poste de travail, liée à un client, permettant l'accès aux fonctions de mise en file d'attente sur un serveur.

application répartie : Dans le cadre des files d'attente de messages, jeu de programmes d'application pouvant être connectés séparément à des gestionnaires de files d'attente différents, tout en ne formant collectivement qu'une seule application.

arrêt : Voir *arrêt immédiat*, *arrêt préemptif*, et *arrêt avec mise au repos*.

arrêt avec mise au repos : Dans MQSeries, arrêt d'un gestionnaire de files d'attente permettant la déconnexion de toutes les applications connectées. S'oppose à la fois à *arrêt immédiat* et à *arrêt préemptif*.

arrêt contrôlé : Voir *arrêt avec mise au repos*.

arrêt immédiat : Dans MQSeries, arrêt d'un gestionnaire de files d'attente sans attendre que les applications se déconnectent de celui-ci. Les appels MQI en cours sont autorisés à se terminer, mais les nouveaux appels MQI échouent une fois l'arrêt immédiat demandé. S'oppose à *arrêt avec mise au repos* et *arrêt préemptif*.

arrêt préemptif : Dans MQSeries, arrêt d'un gestionnaire de files d'attente sans attendre que les applications se déconnectent de celui-ci ni même que les appels MQI en cours se terminent. S'oppose à la fois à *arrêt immédiat* et à *arrêt avec mise au repos*.

attribut : Élément définissant les caractéristiques d'un objet MQSeries.

B

BMS : Sigle de Basic Mapping Support (voir *module de structuration des entrées-sorties*).

browse : Dans le cadre des files d'attente de messages, utiliser l'appel MQGET pour copier un message sans le supprimer de la file. Voir aussi *recupérer*.

C

canal : Voir *canal de communication*.

canal de communication : Dans le domaine de la gestion répartie des files d'attente de messages, mécanisme permettant de déplacer les messages d'un gestionnaire de files d'attente à un autre. Un canal de communication se compose de deux agents (émetteur à une extrémité, récepteur à l'autre) et d'une liaison de données. Egalement appelé *canal de transmission*. S'oppose à *canal MQI*.

canal demandeur : Dans le cadre de la mise en file d'attente des messages, canal pouvant être démarré à distance par un canal émetteur. Le canal demandeur accepte les messages du canal émetteur via une liaison et place ces messages dans la file d'attente locale désignée dans le message. Voir aussi *canal serveur*.

canal émetteur : Dans le domaine de la mise en file d'attente de messages, canal qui prend l'initiative de la transmission, supprime des messages d'une file d'attente de transmission et les place sur un canal demandeur ou récepteur via une liaison de données.

canal MQI : Relie un client MQSeries à un gestionnaire de files d'attente sur un serveur et transmet uniquement les appels et les réponses MQI en mode bidirectionnel. S'oppose à *canal de communication*.

canal récepteur : Dans le cadre des files d'attente de messages, canal qui répond à un canal émetteur, reçoit des messages d'une liaison de données et les place dans une file d'attente locale.

canal répondeur : Dans le cadre de la gestion répartie des files d'attente, programme qui répond aux demandes de connexion réseau d'un autre système.

canal serveur : Dans le domaine de la mise en file d'attente de messages, canal qui répond à un canal demandeur, supprime des messages d'une file d'attente de transmission et les place sur le canal demandeur via une liaison de données.

caractère null : Caractère dont le code est X'00'.

CCF : Sigle de Channel Control Function (voir *fonction de contrôle des canaux*).

CCSID : Identificateur de jeu de caractères codés.

CDF : Sigle de Channel Definition File (voir *fichier de définition de canal*).

chaîne symptôme : Informations de diagnostic affichées dans un format structuré et servant à effectuer une recherche dans la base de données de support logiciel d'IBM.

chemin de résolution : Ensemble des files d'attente ouvertes lorsqu'une application spécifie un alias ou une file d'attente éloignée en entrée d'un appel MQOPEN.

CICS : Sigle de Customer Information Control System (système de contrôle d'informations clients).

client : Composant d'exécution permettant aux applications utilisateur locales d'accéder aux fonctions de mise en file d'attente sur un serveur. Les files d'attente utilisées par les applications résident sur le serveur. Voir aussi *client MQI*.

client MQI : Partie de MQSeries pouvant être installée sur un système sans installer la totalité du gestionnaire de files d'attente. Le MQI client accepte des appels MQI des applications et communique avec un gestionnaire de files d'attente sur un système serveur.

cluster : Réseau de gestionnaires de files d'attente associés de façon logique.

code achèvement : Code retour indiquant la façon dont un appel MQI s'est terminé.

code anomalie : Code retour qui décrit la raison de l'échec ou de la réussite partielle d'un appel MQI.

codes retour : Nom générique des codes achèvement et anomalie.

commande : Sous MQSeries, instruction d'administration pouvant être exécutée par le gestionnaire de files d'attente.

commande PCF : Voir *commande programmable PCF*.

commande PING : Dans le cadre de la mise en file d'attente répartie, aide au diagnostic utilisant l'échange de messages de test pour confirmer le fonctionnement d'un canal.

commande programmable (PCF) : Type de message MQSeries utilisé par :

- les applications d'administration, pour placer les commandes PCF dans la file d'attente d'entrée des commandes système d'un gestionnaire donné ;
- les applications d'administration, pour récupérer les résultats d'une commande PCF à partir d'un gestionnaire de files d'attente donné ;
- un gestionnaire de files d'attente (comme notification de la survenue d'un événement).

S'oppose à *MQSC*.

commandes d'administration : Commandes MQSeries permettant de gérer les objets MQSeries (tels que des files d'attente, des processus et des listes de noms).

commandes de contrôle du système : Commandes utilisées pour traiter les entités spécifiques à certaines plateformes telles que les pools mémoire, les classes d'archivage et les ensembles de pages.

commandes MQSeries (MQSC) : Commandes lisibles par l'utilisateur, identiques sur toutes les plateformes, permettant de manipuler les objets MQSeries. S'oppose à *format de commande programmable (PCF)*.

communications évoluées de programme à programme : Fonction générale qui caractérise l'architecture LU 6.2 et ses différentes applications pratiques.

consignation linéaire : Sous MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, consignation des données de redémarrage dans une chaîne de fichiers. De nouveaux fichiers sont ajoutés à la chaîne en fonction des besoins. L'espace dans lequel les données sont inscrites n'est pas réutilisé tant que le gestionnaire de files d'attente n'a pas été redémarré. S'oppose à *consignation circulaire*.

contexte : Informations concernant l'origine d'un message.

contrôle d'autorisation : Contrôles de sécurité effectués lorsque vous tentez d'ouvrir un objet MQSeries.

contrôle d'autorisation : Contrôles de sécurité effectués lorsqu'un utilisateur essaie d'appliquer des commandes d'administration à un objet, par exemple pour ouvrir une file d'attente ou se connecter à un gestionnaire de files d'attente.

CRE : Sigle de Common Run-Time Environment.

CRE, environnement d'exécution commun (Common Run-Time Environment) : Un ensemble de services qui permet aux programmeurs d'écrire des applications en plusieurs langages. Ces services partagés de runtime peuvent être utilisés par les programmes en C, COBOL85, FORTRAN, Pascal et TAL.

curseur de défilement : Dans le cadre des files d'attente de messages, indicateur utilisé lors de la consultation d'une file d'attente pour identifier le message suivant.

D

datagramme : Forme la plus simple des messages pris en charge par MQSeries. Ce type de message ne demande pas de réponse.

DCE : Sigle de Distributed Computing Environment.

DCI : Sigle de Data Conversion Interface (voir *interface de conversion de données*).

déclenchement : Sous MQSeries, procédure permettant à un gestionnaire de files d'attente de démarrer automatiquement une application lorsque certaines conditions prédéfinies sont réunies.

définition locale : Objet MQSeries appartenant à un gestionnaire de files d'attente local.

définition locale d'une file d'attente éloignée : Objet MQSeries appartenant à un gestionnaire de files d'attente local. Cet objet définit les attributs d'une file d'attente appartenant à un autre gestionnaire. En outre, il est utilisé comme gestionnaire de files d'attente alias et file d'attente de réponses alias.

descripteur de message : Informations décrivant le format du message et sa présentation, faisant partie intégrante d'un message MQSeries. Le format du descripteur est défini par la structure MQMD.

descripteur d'objet : Structure de données identifiant un objet MQSeries donné. Le descripteur comprend le nom de l'objet et son type.

données d'événement : Dans un message d'événement, partie contenant les informations relatives à l'événement (par exemple, nom du gestionnaire de

files d'attente, application ayant généré l'événement). Voir aussi *en-tête d'événement*.

E

EC Boss (contrôleur d'exécution principal) : Le contrôleur d'exécution principal est le principal processus de contrôle du gestionnaire de files d'attente.

EC (contrôleur d'exécution) : Un contrôleur d'exécution est un processus de contrôle du gestionnaire de files d'attente. Il est responsable d'un ensemble d'agents.

élément de données : Dans MQAI, l'un des éléments contenus dans un jeu de données. Il peut s'agir d'un entier, d'une chaîne de caractères, d'un élément utilisateur ou système.

EMS : Observateur d'événements (Event Monitoring System).

en-tête d'événement : Dans un message d'événement, partie des données identifiant le type du code anomalie de l'événement.

environnement DCE : Logiciel intermédiaire (middleware) offrant des fonctions de base et facilitant le développement d'applications réparties. DCE est défini par l'Open Software Foundation (OSF).

Environnement de communication intersystème (ICE) : L'environnement de communication intersystème est une famille de logiciels Compaq qui donne accès à un grand nombre d'applications sur les ordinateurs Compaq.

événement : Voir *événement Canal*, *événement Outil*, *événement Performances* et *événement Gestionnaire de files d'attente*.

événement Canal : Événement indiquant la disponibilité ou l'indisponibilité d'une instance de canal. Ce type d'événement est généré par le gestionnaire de files d'attente aux deux extrémités du canal.

événement Déclenchement : Événement (par exemple, message arrivant dans une file d'attente) pour lequel le gestionnaire de files d'attente crée un message de déclenchement dans la file d'attente d'initialisation.

événement d'intervalle de service : Événement lié à l'intervalle de service.

événement du gestionnaire de files d'attente : Événement indiquant :

- Un cas d'erreur lié aux ressources utilisées par un gestionnaire de file d'attente. Par exemple un cas d'erreur causé par une file d'attente non disponible.
- Un changement significatif au niveau du gestionnaire de files d'attente. Exemple : le gestionnaire vient de s'arrêter ou de démarrer.

événement en instance : Événement non planifié qui est le résultat d'une demande de connexion par un adaptateur CICS.

événement Outil : Outil permettant de gérer le fonctionnement des gestionnaires de files d'attente dans un réseau MQSeries. Sous MQSeries, les événements outils servent à la gestion des définitions de ressources des gestionnaires de files d'attente, des performances et des canaux. Ils peuvent être utilisés par un programme de reporting dans le cadre d'une application d'administration qui affiche les événements sur l'écran de l'opérateur système. Ils permettent également aux applications qui jouent le rôle d'agents pour les autres réseaux d'administration, de gérer les rapports et de générer les alertes nécessaires.

événement Performances : Catégorie d'événement indiquant qu'une condition limite est atteinte.

F

FFST : Abréviation de First Failure Support Technology.

fichier d'autorisation : Fichier contenant les définitions de sécurité d'un objet, d'une classe d'objets ou de toutes les classes d'objets.

fichier de configuration (également appelé fichier ini file) : Fichier contenant les informations nécessaires à la configuration, par exemple, des journaux, des communications ou des fonctions optionnelles. Voir aussi *strophe*.

fichier de définition de canal (CDF) : Sous MQSeries, fichier contenant les définitions des canaux de communication qui associent les files d'attente de transmission aux liaisons.

fichier ini : Voir *fichier de configuration*.

file d'attente : Objet MQSeries. Certaines applications placent des messages en file d'attente et les en retirent. Une file d'attente appartient toujours à un gestionnaire de files d'attente qui en assure la maintenance. Les files d'attente locales contiennent une liste de messages en attente de traitement. Les autres types de files d'attente ne contiennent aucun message, elles désignent d'autres files ou peuvent être utilisées comme modèles pour les files d'attente dynamiques.

file d'attente d'application : File d'attente utilisée par une application.

file d'attente de messages : Synonyme de *file d'attente*.

file d'attente de rebut (DLQ) : File d'attente dans laquelle un gestionnaire ou une application place les messages qu'elle ne peut acheminer vers la destination correcte.

file d'attente de rebut (DLQ) : File d'attente dans laquelle un gestionnaire ou une application place les messages qu'elle ne peut acheminer vers la destination correcte.

file d'attente de réponse : Nom de la file d'attente dans laquelle le programme qui a émis un appel MQPUT souhaite recevoir un message de réponse ou de rapport.

file d'attente des messages non transmis : Voir *file d'attente de rebut*.

file d'attente de transmission : File d'attente locale dans laquelle sont stockés temporairement les messages destinés à un gestionnaire de files d'attente éloigné.

file d'attente d'événement : File d'attente dans laquelle le gestionnaire insère un message lorsqu'il détecte un événement. Chaque catégorie d'événement (gestionnaire de files d'attente, performances ou canal) possède sa propre file d'attente.

file d'attente d'initialisation : File d'attente locale dans laquelle le gestionnaire de files d'attente place des messages de déclenchement.

file d'attente dynamique : Une file d'attente locale est créée lorsqu'un programme ouvre un objet file d'attente modèle. Voir aussi *file d'attente dynamique permanente* et *file d'attente dynamique temporaire*.

file d'attente dynamique permanente : File d'attente dynamique qui n'est supprimée à sa fermeture que si cette suppression est explicitement demandée. Les files d'attente dynamiques permanentes sont récupérées en cas d'incident affectant le gestionnaire de files d'attente ; elles peuvent donc contenir des messages persistants. S'oppose à *file d'attente dynamique temporaire*.

file d'attente dynamique temporaire : File d'attente dynamique qui est supprimée à sa fermeture. Les files d'attente dynamiques temporaires ne sont pas récupérées en cas d'incident affectant le gestionnaire de files d'attente ; elles ne doivent donc contenir que des messages non persistants. S'oppose à *file d'attente dynamique permanente*.

file d'attente éloignée : File d'attente appartenant à un gestionnaire de files d'attente éloigné. Les programmes peuvent placer des messages en file d'attente éloignée mais ne peuvent les en extraire. S'oppose à *file d'attente locale*.

file d'attente locale : File d'attente appartenant au gestionnaire de files d'attente local. Ce type de file d'attente peut contenir une liste de messages en attente de traitement. S'oppose à *file d'attente éloignée*.

file d'attente system.command.input : File d'attente locale dans laquelle les programmes d'application peuvent placer des commandes MQSeries. Les

commandes sont ensuite récupérées par le serveur qui les valide et les transmet au processeur.

First Failure Support Technology (FFST) : Utilisé par MQSeries pour UNIX, MQSeries pour OS/2 Warp, MQSeries pour Windows NT et Windows 2000 et MQSeries pour AS/400 pour détecter et signaler les incidents logiciels.

fonction de contrôle des canaux (CCF) : Dans MQSeries, programme qui transmet des messages d'une file d'attente de transmission vers une liaison, et d'une liaison vers une file d'attente cible en local. Ce programme est également doté d'un panneau de commande qui permet la configuration et le contrôle des canaux.

fonctions optionnelles : Sous MQSeries pour UNIX, MQSeries pour Compaq, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, fonctions supplémentaires fournies sous forme de plusieurs composants indépendants. L'installation de ces composants est facultative, chacun d'eux pouvant être remplacé par des composants tiers ou internes. Voir aussi *service d'autorisation*, *service annuaire* et *service d'identification*.

G

gestionnaire de file d'attente de rebut : Utilitaire MQSeries qui gère une file d'attente de rebut (DLQ) et traite les messages qu'elle contient suivant une table de règles définies par l'utilisateur.

gestionnaire de files d'attente : Programme système mettant à la disposition des applications des fonctions de mise en file d'attente. Il comporte une interface de programme d'application qui permet aux programmes d'accéder aux messages des files d'attente appartenant au gestionnaire de files d'attente. Voir aussi *gestionnaire de files d'attente local* et *gestionnaire de files d'attente éloigné*. Objet MQSeries qui définit les attributs d'un gestionnaire de files d'attente donné.

gestionnaire de files d'attente de messages (MQM) : Dans MQSeries pour Compaq NSK V2.2, la fonction MQM utilise des formats de commande PCF et des commandes de contrôle. Le MQM fonctionne sous TCP, comme un demandeur PATHWAY SCOBOL, et utilise un serveur SERVERCLASS MQM qui appelle l'API du C pour exécuter des commandes PCF. Il existe une instance MQM séparée pour chaque gestionnaire de files d'attente configuré sur un système puisque chaque gestionnaire est contrôlé sous sa propre configuration PATHWAY. Un MQM est donc limité à la gestion du gestionnaire de files d'attente auquel il appartient.

gestionnaire de files d'attente éloigné : Pour un programme, un gestionnaire de files d'attente est considéré comme éloigné si ce n'est pas celui auquel le programme est relié.

gestionnaire de files d'attente local : Gestionnaire de files d'attente auquel un programme est relié et qui fournit à celui-ci des fonctions de mise en file d'attente. Les gestionnaires auxquels le programme n'est pas relié sont appelés *gestionnaires de files d'attente éloignés*, même s'ils se trouvent sur le même système que le programme.

gestionnaire des droits d'accès aux objets (OAM) : Sous MQSeries pour systèmes UNIX, MQSeries pour Compaq et MQSeries pour Windows NT et Windows 2000, le service d'autorisation par défaut pour la gestion des commandes et des objets. La fonction OAM peut être remplacée par une fonction de sécurité fournie par le client, ou lui être associée.

gestion répartie des files d'attente (DQM) : Dans le cadre des files d'attente de messages, configuration et contrôle de canaux de communication avec des gestionnaires situés sur d'autres systèmes.

get : Dans le cadre des files d'attente de messages, utiliser l'appel MQGET pour retirer un message d'une file d'attente. Voir aussi *parcourir*.

I

identificateur : Voir *identificateur de connexion* et *identificateur d'objet*.

identificateur de connexion : Identificateur ou marque par lequel un programme accède au gestionnaire de files d'attente auquel il est connecté.

identificateur de jeu de caractères codés (CCSID) : Nom d'un jeu de caractères codés et affectations des pages de code correspondantes.

identificateur d'objet : Identificateur ou marque par le biais duquel un programme accède à un objet MQSeries qu'il doit utiliser.

infrastructure : Sous MQSeries, ensemble d'interfaces de programmation permettant aux clients ou aux fournisseurs d'écrire des programmes étendant ou remplaçant certaines fonctions fournies par MQSeries. Ces interfaces sont les suivantes :

- MQSeries DCI (interface de conversion des données)
- MQSeries MCI (interface des canaux de communication)
- MQSeries NSI (interface du service de noms)
- MQSeries SEI (interface d'activation de la sécurité)
- MQSeries TMI (interface du moniteur de déclenchement)

interface d'activation de sécurité (SEI) : Interface MQSeries avec laquelle doivent être compatibles les programmes client ou fournisseur qui contrôlent les droits d'accès, fournissent les ID utilisateur ou effectuent les opérations d'authentification. Partie de l'infrastructure MQSeries.

interface d'administration MQSeries (MQAI) :
Interface de programmation MQSeries.

interface de conversion de données (DCI) : Interface MQSeries avec laquelle doivent être compatibles les programmes client ou fournisseur convertissant les données d'une machine à une autre et d'un CCSID à un autre. Partie de l'infrastructure MQSeries.

interface de file d'attente de messages (MQI) :
Interface de programmation fournie par les gestionnaires de files d'attente MQSeries. Cette interface de programmation permet aux programmes d'application d'accéder aux fonctions de files d'attente de message.

interface de moniteur de déclenchement (TMD) :
Interface MQSeries avec laquelle doivent être compatibles les moniteurs de déclenchement. Partie de l'infrastructure MQSeries.

interface de service annuaire (NSI) : Interface MQSeries avec laquelle doivent être compatibles les programmes de résolution de propriété et de nom de file d'attente. Partie de l'infrastructure MQSeries.

interface MCI : Interface MQSeries avec laquelle doivent être compatibles les programmes qui transmettent les messages entre un gestionnaire de files d'attente MQSeries et un autre système de messagerie. Partie de l'infrastructure MQSeries.

intervalle de service : Intervalle de temps standard auquel le gestionnaire de files d'attente compare le délai séparant un Put ou un Get du Get suivant, ce qui lui permet de déterminer si les conditions d'un événement Intervalle de service sont remplies. L'intervalle de service d'une file d'attente constitue un des attributs de cette dernière.

IP : Protocole Internet

J

jeu : Voir *jeu de données*.

jeu de commandes : Dans MQAI, un type de jeu conçu pour administrer MQSeries. Il ne peut pas modifier l'ordre des données ni créer des listes, à l'intérieur d'un message.

jeu de données : Dans MQAI, un jeu de données permet de décrire les propriétés (ou paramètres) des objets.

jeu de données d'administration : Dans MQAI, un type de jeu de données conçu pour administrer MQSeries. On considère qu'il peut modifier l'ordre des données, créer des listes et vérifier les sélecteurs à l'intérieur d'un message.

jeu système : un type de jeu de données qui est créé par le MQAI.

jeu utilisateur : Dans MQAI, un type de jeu de données qui est créé par l'utilisateur.

journal : Sous MQSeries, fichier dans lequel est enregistrée l'activité des gestionnaires de files d'attente (réception, émission et livraison de messages) pour permettre la reprise en cas d'incident.

L

liste de noms : Objet MQSeries contenant une liste de noms (de files d'attente, par exemple).

M

MCA : Programme qui transmet des messages d'une file d'attente de transmission vers une liaison ou d'une liaison vers une file d'attente cible.

MCI : Interface MQSeries avec laquelle doivent être compatibles les programmes qui transmettent les messages entre un gestionnaire de files d'attente MQSeries et un autre système de messagerie. Partie de l'infrastructure MQSeries.

message : Dans les applications utilisant les fonctions de mise en file d'attente, communication transmise d'un programme à un autre. Voir aussi *message persistant*. En programmation système, information adressée à l'opérateur du terminal ou à l'administrateur système.

message de déclenchement : Message contenant des informations concernant le programme qui doit être lancé par le moniteur de déclenchement.

message de demande : Message utilisé pour demander une réponse à un autre programme.

message de rapport : Type de message fournissant des informations sur un autre message. Un message de rapport peut signaler qu'un message a été délivré, est arrivé à destination, a expiré ou n'a pu être traité pour une raison ou pour une autre.

message de réponse : Message utilisé pour répondre à un message de demande.

message d'événement : Message contenant des informations (catégorie de l'événement, nom de l'application ayant généré l'événement, statistiques concernant le gestionnaire de files d'attente) relatives à l'origine d'un événement outil dans un réseau de système MQSeries.

message non persistant : Message éliminé après le redémarrage du gestionnaire de files d'attente. S'oppose à *message persistant*.

message persistant : Message conservé après le redémarrage du gestionnaire de files d'attente. S'oppose à *message non persistant*.

messagerie : Voir *messagerie synchrone* et *messagerie asynchrone*.

messagerie asynchrone : Méthode de communication entre programmes faisant appel à la fonction de mise en file d'attente de messages. En mode asynchrone, le programme émetteur poursuit son exécution sans attendre une réponse au message qu'il a émis. S'oppose à *messagerie synchrone*.

messagerie sans contrainte de temps : Voir *messagerie asynchrone*.

messagerie synchrone : Méthode de communication entre programmes faisant appel à la fonction de mise en file d'attente de messages. Selon cette méthode, le programme émetteur attend une réponse à son message avant de reprendre son propre traitement. S'oppose à *messagerie asynchrone*.

mise au repos : Dans MQSeries, état d'un gestionnaire de files d'attente juste avant son arrêt. Dans cet état, les programmes peuvent terminer leur traitement, mais aucun nouveau programme n'est autorisé à se lancer.

mise en file d'attente : Voir *mise en file d'attente de messages*.

mise en file d'attente de messages : Technique de programmation dans le cadre de laquelle chaque programme au sein d'une application communique avec les autres programmes par le biais de messages placés en file d'attente.

mise en file d'attente éloignée : Dans le cadre de la mise en file d'attente de messages, utilisation de fonctions permettant aux applications de placer des messages dans des files d'attente appartenant à d'autres gestionnaires de files d'attente.

modification provisoire de logiciel (PTF) : Solution à un problème diagnostiqué par les services techniques IBM, résultant d'un défaut dans un programme.

module de structuration des entrées-sorties (BMS) : Interface entre CICS et les programmes d'application, qui formate les données en entrée et en sortie pour affichage, et qui achemine les messages de plusieurs pages sans considération pour les caractères de contrôle utilisés par différents terminaux.

moniteur de déclenchement : Application à fonctionnement continu tenant lieu de serveur pour une ou plusieurs files d'attente d'initialisation. Lorsqu'un message de déclenchement arrive dans une file d'attente d'initialisation, le moniteur de déclenchement le récupère. Il utilise alors les informations contenues dans le message pour lancer un processus serveur associé à la file d'attente dans laquelle l'événement Déclenchement est survenu.

MQAI : Sigle de MQSeries Administration Interface (voir *interface d'administration MQSeries*).

MQI : Sigle de Message Queue Interface : Interface de programmation fournie par les gestionnaires de files d'attente MQSeries. Cette interface de programmation permet aux programmes d'application d'accéder aux fonctions de files d'attente de message.

MQM : Gestionnaire de files d'attente de messages

MQSC : Commandes MQSeries.

MQSeries : Famille de programmes IBM permettant de gérer des files d'attente de messages.

N

NetBIOS : Sigle de Network Basic Input/Output System. Interface de système d'exploitation pour programmes d'application utilisés sur les ordinateurs personnels IBM connectés au réseau en anneau à jeton IBM.

niveau de priorité du message : Sous MQSeries, attribut d'un message pouvant modifier l'ordre dans lequel sont récupérés les messages d'une file d'attente et décider de la création d'un événement Déclenchement ou pas.

NSI : Sigle de Name Service Interface (voir *interface de service annuaire*).

numérotation en séquence des messages : Technique de programmation dans laquelle les messages se voient attribuer des numéros uniques au cours de leur transmission. Le processus destinataire peut ainsi vérifier s'il a bien reçu tous les messages, placer les messages dans une file d'attente dans leur ordre originel et supprimer les messages reçus en double.

O

objet : Sous MQSeries, on appelle objet un gestionnaire de files d'attente, une file d'attente, une définition de processus, ou un canal.

objet définition de processus : Objet MQSeries contenant la définition d'une application MQSeries. Par exemple, un gestionnaire de files d'attente utilise cette définition lorsqu'il traite des messages de déclenchement.

objet file d'attente alias : Objet MQSeries dont le nom est l'alias d'une file d'attente de base définie pour le gestionnaire de files d'attente local. Lorsqu'une application ou un gestionnaire de files d'attente utilise une file d'attente alias, l'alias est résolu et l'opération demandée est effectuée sur la file d'attente correspondante.

objet file d'attente éloignée : Voir *définition locale d'une file d'attente éloignée*.

objet file d'attente modèle : Série d'attributs de file d'attente servant de modèle lorsqu'un programme crée une file d'attente dynamique.

objet par défaut : Définition d'un objet (par exemple, une file d'attente) dans laquelle tous les attributs sont renseignés. Si un utilisateur définit un objet sans préciser tous les attributs possibles de cet objet, le gestionnaire de file d'attente utilise pour le ou les attributs non précisés les attributs par défaut.

P

paramètre d'entrée : Paramètre d'un appel MQI dans lequel vous placez une valeur au moment de l'appel.

paramètre d'entrée-sortie : Paramètre d'un appel MQI dans lequel vous placez une valeur au moment de l'appel et dont le gestionnaire de files d'attente modifie la valeur suite à la réussite ou à l'échec de l'appel.

paramètre de sortie : Paramètre d'un appel MQI dont le gestionnaire de files d'attente modifie la valeur suite à la réussite ou à l'échec de l'appel.

PCF : Sigle de Programmable Command Format (voir *format de commande programmable*).

plateforme : Sous MQSeries, système d'exploitation sous lequel fonctionne un gestionnaire de files d'attente.

point de contrôle : Moment auquel les informations importantes sont consignées dans le fichier journal. S'oppose à *point de synchronisation*. Sous MQSeries pour UNIX, moment auquel un enregistrement de données figurant dans le journal est identique à l'enregistrement figurant en file d'attente. Les points de contrôle sont générés automatiquement et sont utilisés lors du redémarrage du système.

point de synchronisation : Point intermédiaire ou final durant le traitement d'une transaction, au stade duquel les ressources protégées de la transaction sont synchronisées. Au point de synchronisation, les modifications des ressources peuvent être effectuées en toute sécurité ou peuvent être réinitialisées au point de synchronisation précédent.

premier entré, premier sorti (FIFO) : Technique de mise en file d'attente dans laquelle l'élément suivant à récupérer est celui ayant passé le plus de temps en file d'attente.

premier entré, premier sorti (FIFO) : Technique de mise en file d'attente dans laquelle l'élément suivant à récupérer est celui ayant passé le plus de temps en file d'attente.

processeur de commandes : Composant MQSeries chargé de traiter les commandes.

programme d'écoute : Dans le cadre de la mise en files d'attente réparties dans MQSeries, programme qui gère les informations concernant les connexions réseau entrantes.

programme de transmission : Voir *agent MCA*.

Protocole de datagramme utilisateur (UDP) : Element du protocole TCP/IP. Un protocole de transmission de paquets, établi directement sur la couche IP (Internet Protocol). UDP est un mode de liaison sans connexion, moins fiable que TCP. Il sert à la communication interapplications entre systèmes hôtes TCP/IP.

Protocole Internet (IP) : Protocole utilisé pour diriger les données de leur source à leur destination dans un environnement Internet. Il s'agit de la couche de base sur laquelle s'appuient d'autres couches de protocole telles que TCP et UDP.

Protocole TCP : Element du protocole TCP/IP. Un protocole de communication hôte à hôte, sur les réseaux à commutation de paquets. TCP assure une livraison des données sur la base d'une connexion donnée. La livraison se fait de manière fiable et ordonnée.

Protocole TCP/IP : Un ensemble de protocoles de communication qui assure les fonctions de connexion d'égal à égal, pour les réseaux locaux et étendus.

PTF : Sigle de Program Temporary Fix (voir *modification provisoire du logiciel*).

R

rappel : Dans MQSeries, pour lancer une transmission par un canal de communication émetteur, un canal demandeur appelle d'abord l'émetteur, puis coupe la communication et attend un rappel de l'émetteur.

rapport officiel d'analyse de programme (APAR) : Rapport concernant un incident provoqué par une défaillance du programme.

RBA : Sigle de Relative Byte Address (voir *adresse relative en octets*).

référentiel : Ensemble d'informations concernant les gestionnaires de file d'attente membres d'un cluster. Ces informations comprennent les noms des gestionnaires, leurs emplacements, leurs canaux, les files qu'ils contiennent etc.

remise séquentielle : Dans MQSeries, méthode de transmission des messages avec numéro d'ordre, de telle sorte que le canal récepteur puisse stocker les messages dans le bon ordre. Le recours à cette méthode est obligatoire lorsque les messages doivent être remis une seule et unique fois, et dans le bon ordre.

resynchronisation : Dans MQSeries, option permettant de demander à un canal de démarrer et de convertir tout message de statut en attente de validation sans pour autant relancer la transmission des messages.

S

se connecter : Fournir un identificateur de connexion au gestionnaire de files d'attente, utilisé par une application pour les appels MQI suivants. La connexion est établie par l'appel MQCONN ou automatiquement par l'appel MQOPEN.

sécurité de substitution : Procédure permettant d'étendre les droits d'accès d'un ID utilisateur à un autre utilisateur (par exemple, pour ouvrir un objet MQSeries).

sécurité du contexte : Sous MQSeries, méthode permettant d'afficher le détail de l'origine d'un message dans son descripteur.

SEI : Sigle de Security Enabling Interface (voir *interface d'activation de sécurité*).

serveur : (1) Sous MQSeries, gestionnaire de files d'attente offrant des fonctions de mise en file d'attente à des applications client s'exécutant sur un poste de travail éloigné. (2) Programme qui répond aux demandes d'informations dans le modèle particulier client-serveur. Voir aussi *client*.

serveur de commandes : Composant MQSeries qui lit les commandes à partir de la file d'attente d'entrée des commandes système, les contrôle, puis transmet celles qui sont correctes au processeur.

serveur de file d'attente : Paire de processus NonStop qui traite toutes les opérations de messagerie des files d'attente locales.

Serveur de statut : Supporte toutes les informations concernant le statut de tous les objets à l'exception des files d'attente locales. Le serveur de statut par défaut traite aussi les informations concernant l'état pour le gestionnaire de files d'attente.

serveur MQI : Un serveur MQI est un gestionnaire de files d'attente qui fournit des fonctions de gestion de files à un ou plusieurs clients. Tous les objets MQSeries, par exemple les files d'attente, n'existent que sur le système du gestionnaire de files d'attente, c'est-à-dire sur le serveur MQI. Un serveur peut également prendre en charge les applications MQI locales normales.

SIT : Sigle de System Initialization Table (voir *table d'initialisation du système*).

SNA : Sigle de Systems Network Architecture (architecture unifiée de réseau).

stockage et transmission : Stockage temporaire de paquets, messages ou trames en un point d'un réseau avant leur retransmission vers leur destination.

strophe : Groupe de lignes, dans un fichier de configuration, qui affecte à un paramètre une valeur modifiant le comportement d'un gestionnaire de files d'attente, d'un client ou d'un canal. Dans MQSeries sur des systèmes, un fichier (ini) de configuration peut contenir un nombre donné de strophes.

Système de contrôle d'informations clients (CICS) : Système IBM de gestion des opérations qui offre un accès en ligne aux fichiers de données grâce à des programmes d'application écrits par l'utilisateur. Ce système contient aussi des utilitaires permettant de créer, utiliser et entretenir des bases de données.

T

table de règles : Fichier de commande contenant une ou plusieurs règles que le gestionnaire de file d'attente de rebut applique aux messages de cette file d'attente.

TACL : Sigle de Tandem Advanced Command Language.

TCP : Protocole TCP

TCP/IP : Protocole TCP/IP

TMF : Sigle de Transaction Management Facility.

TMI : Sigle de Trigger Monitor Interface (interface de moniteur de déclenchement).

TM/MP : Sigle de NonStop Transaction Manager/MP.

trace des performances : Option de trace de MQSeries dans le cadre de laquelle les données de trace sont utilisées pour effectuer une analyse et un ajustement des performances.

trandid : Voir *ID de transaction*.

type de canal de connexion client : Type de définition de canal MQI associée à un client MQI. Voir aussi *type de canal de connexion serveur*.

type de canal de connexion serveur : Type de définition de canal MQI associée au serveur sur lequel s'exécute le gestionnaire de files d'attente. Voir aussi *type de canal de connexion client*.

U

UDP : Protocole de datagramme utilisateur.

unité de reprise : Séquence d'opérations, pouvant faire l'objet d'une reprise, effectuées dans un gestionnaire de ressources unique. S'oppose à *unité de travail*.

unité de travail : Séquence d'opérations, pouvant faire l'objet d'une reprise, effectuées par une application entre deux points de contrôle de cohérence. Une unité de travail débute lorsqu'une transaction démarre ou après un point de synchronisation demandé par l'utilisateur. Elle se termine soit à un point de synchronisation demandé par l'utilisateur, soit à la fin d'une transaction. S'oppose à *unité de reprise*.

unité d'exécution : Sous MQSeries, niveau le plus bas d'exécution parallèle disponible sur une plateforme de système d'exploitation.

unité logique de travail (LUW) : Voir *unité de travail*.

utilitaire : Sous MQSeries, jeu de programmes fournissant à l'opérateur système ou à l'administrateur système des fonctions complétant celles fournies par les commandes MQSeries. Certains utilitaires appellent plusieurs fonctions.

V

valeur de bouclage du numéro de séquence : Dans MQSeries, méthode permettant de s'assurer que les deux extrémités d'une liaison réinitialisent leurs numéros de séquence de messages en même temps. La transmission des messages avec numéro de séquence, permet au canal récepteur de stocker les messages dans le bon ordre.

validation : Action de valider une opération pour que les modifications apportées à une base de données soient enregistrées. Les données protégées sont sauvegardées une fois l'opération terminée.

validation à deux phases : Protocole de coordination des modifications apportées aux ressources pouvant faire l'objet d'une reprise lorsque plusieurs gestionnaires de ressources sont utilisés par une seule et même transaction. S'oppose à *validation monophasé*.

validation monophasé : Méthode dans laquelle un programme peut exécuter des mises à jour dans une file d'attente sans les coordonner avec celles qui sont apportées aux ressources contrôlées par un autre gestionnaire de ressource. S'oppose à *validation à deux phases*.

Index

A

- abandon 181
- accès non autorisé (nécessité de protection) 139
- activation
 - événements outil 175
 - sécurité 143
- ADD SERVER (commande PATHCOM) 32
- administration
 - droits d'accès 154
 - éloignée 125
 - canaux 126
 - files d'attente de transmission 126
 - objets 123
 - jeux de commandes 29
 - commandes de contrôle 29
 - commandes MQSeries (MQSC) 30
 - commandes programmables (PCF) 31
 - locale 93
- administration éloignée
 - incidents 131
 - serveur de commandes 121
- administration locale 93
- administration TS/MP 32
- affectation de l'espace de permutation 56
- affichage
 - attributs du gestionnaire de files d'attente 96
 - commande de sortie de trace formatée MQSeries 281
 - commandes de fichiers MQSeries 278
 - définitions de processus 117
 - droit d'accès 273
 - état du serveur de commandes 121
 - serveur de commandes 277
 - statut des canaux 88
- agent MCA
 - fonction 23
 - reprise sur incident 240
- aide Windows 415
- ajout d'un serveur de file d'attente personnalisé 60
- ajout d'un serveur de statut personnalisé 58
- algorithmes d'extraction de messages 10
- alias
 - files d'attente de réponse 135
 - gestionnaire de files d'attente 134
- alias de files d'attente de réponse 135
- ALTER SERVER
 - commande PATHCOM 33
- application
 - asynchrone 7
 - bind FASTPATH 352
 - bind STANDARD 352
 - compilation et lien 352

- application (*suite*)
 - considération sur le développement 207
 - déclenchée 349
 - données 8
 - écriture 347
 - effets de la consultation des messages persistants en file d'attente 225
 - erreurs de programmation (exemples) 203
 - exécution 353
 - gestion des transactions simultanées 238
 - guardian 351
 - liée par FASTPATH
 - reprise sur incident 240
 - liée par STANDARD
 - reprise sur incident 240
 - optimisation 219
 - prise en charge de l'administration MQI 93
 - restrictions d'utilisation de FASTPATH 229
 - sécurisée 228
 - utilisation de OpenTMF 237
- applications asynchrones 7
- applications déclenchées 349
 - transmission de l'attribut USERDATA 338
- arrêt
 - canaux 89
 - gestionnaire de files d'attente 286, 331
 - trace 74
- arrêt (gestionnaire de files d'attente) 68
- arrêt contrôlé 68
- arrêt d'un gestionnaire de files d'attente 68
- arrêt de la trace MQSeries 288
- arrêt forcé d'un canal (arrêt immédiat) 89
- Arrêt forcé du gestionnaire de files d'attente 68
- arrêt progressif 68
- attribut de canal NPMSPPEED 221
- attribut LIKE, DEFINE QLOCAL 105
- attribut REPLACE (commandes DEFINE) 99
- attributs des objets MQSeries
 - attribut ALL 104
 - comparaison entre MQSC et PCF 31
 - files d'attente 12
 - gestionnaire de files d'attente
 - affichage 96
 - modification 98
 - modification, file d'attente locale 106
 - valeur par défaut 104

B

- base de données
 - cohérence 236
 - cohérence externe 236
 - fichiers de base de données contrôlée 238
 - protégée par TM/MP 236
- base de données des principaux 238
- base de données OAM 238
- bases de données conformes à la norme XA 20
- bibliothèque de ressources partagées 23
- bibliothèques
 - MQI 352
- bind FASTPATH 228, 352
 - activation 229
 - réduction de charge 228
 - restrictions 229
- bind STANDARD 352
- BookManager 415

C

- C++ natif
 - langage pris en charge 351
- C natif
 - langage pris en charge 351
- C non-natif
 - langage pris en charge 351
- canal
 - administration éloignée 126
 - canal de transmission de messages 221
 - commande d'exécution de l'initialisateur 290
 - commande de démarrage 291
 - commandes 150
 - configuration 193
 - définition 127
 - définition automatique 129
 - définition d'un canal émetteur TCP/IP 392
 - définition d'un canal récepteur TCP/IP 392
 - description 15, 123
 - droits des commandes d'échappement 155
 - événements 174, 397
 - exits 365
 - ICE 391
 - identification parmi des gestionnaires 12
 - lancement 128
 - mise en file d'attente éloignée 123
 - nombre maximum 54
 - sécurité 151
 - sécurité requise 150
- canal de connexion serveur (définition automatique) 129
- canal de transmission de messages 221

- canal récepteur (définition automatique) 129
- catalogue des objets 238
- CCSID
 - méthode de conversion 52
 - pris en charge par MQSeries pour Compaq NSK 319
- classe de serveur
 - personnalisée 43
- classe de serveur MQS-STATUS00 34
- classe de serveurs
 - convention d'appellation
 - serveur de file d'attente 61
 - serveurs de statut 59
 - MQS-TCPLISnn 56
- classes de serveur personnalisées 43
- classes de serveur TS/MP
 - MQS-CHANINIT00 32
 - MQS-CMDSERV00 32
 - MQS-EC00 34
 - MQS-TCPLIS00 32
 - MQS-TRIGMON00 33
- clients 17, 395
- cluster
 - amélioration de la disponibilité
 - réseau 245
 - de gestionnaires de files d'attente 10
 - description 124
 - exits de pondération de charge 365
 - gestionnaire de files d'attente
 - exit de pondération de charge 18
 - mise en file d'attente éloignée 123
- COBOL natif
 - langage pris en charge 351
- COBOL non-natif
 - langage pris en charge 351
- code anomalie 11
- codes retour 200
 - commande altmqfls 257
 - commande altmqusr 259
 - commande dspmqcsv 277
 - commande dspmqusr 282
 - commande endmqcsv 284
 - commande runmqslr 295
- commande altmqfls 106, 254
 - commandes connexes 258
 - construction de la structure de données au démarrage du serveur de file d'attente 225
 - définition du seuil de consultation 225
 - déplacement des fichiers de file d'attente 223
 - messages non persistants avec point de contrôle 226
 - mesure de la profondeur de la file d'attente 226
 - verrouillage en mémoire du serveur de file d'attente 226
- commande altmqusr 259
 - commandes connexes 260
- commande cleanrdf 261
- commande cnvclchl 263
- commande crtmqcvx 265
- commande crtmqm 267
- commande d'exécution du programme d'écoute (runmqslr) 294
- commande de correction 30
- commande de lancement de la trace MQSeries 309
- commande DISPLAY CHSTATUS 337
- commande dltnmqm 271
- commande dspmqaut 273
 - utilisation 144, 147
- commande dspmqcsv 277
- commande dspmqfls 278
- commande dspmqtrc 281
- commande dspmqusr 282
- commande endmqcsv 284
- commande endmqm 68, 286
- commande endmqtrc 288
- commande instmqm 289
- commande runmqchi 290
- commande runmqchl 291
- commande runmqdlq 161, 292
- commande runmqslr 294
- commande runmqtrm 299
- commande setmqaut 300
 - commandes connexes 305
 - services optionnels 147
 - utilisation 144, 146
- commande strmqcsv 306
 - commandes connexes 306
- commande strmqm 307
 - commandes connexes 308
- commande strmqtrc 309
 - commandes connexes 311
- commande upgmqm 312
- commandes
 - affichage d'une sortie de trace formatée MQSeries (dspmqtrc) 281
 - affichage des droits d'accès (dspmqaut) 273
 - affichage des informations concernant un principal donné (dspmqusr) 282
 - affichage du serveur de commandes (dspmqcsv) 277
 - arrêt d'un gestionnaire de files d'attente (endmqm) 286
 - arrêt d'un serveur de commandes (endmqcsv) 284
 - arrêt de la trace MQSeries (endmqtrc) 288
 - commande programmable (PCF) 31
 - commandes de sécurité
 - dspmqaut 147
 - setmqaut 144
 - comparaison de jeux de commandes 327
 - contrôle 29
 - altmqfls 254
 - altmqusr 259
 - cleanrdf 261
 - cnvclchl 263
 - crtmqcvx 265
 - crtmqm 267
 - dltnmqm 271
 - dspmqaut 273
 - dspmqcsv 277
 - dspmqfls 278
 - dspmqtrc 281
 - dspmqusr 282
 - endmqcsv 284
 - endmqm 286
- commandes (suite)
 - contrôle 29 (suite)
 - endmqtrc 288
 - instmqm 289
 - runmqchi 290
 - runmqchl 291
 - runmqdlq 292
 - runmqslr 294
 - runmqsc 296
 - runmqtrm 299
 - setmqaut 300
 - strmqcsv 306
 - strmqm 307
 - strmqtrc 309
 - upgmqm 312
 - conversion des définitions de canaux clients (cnvclchl) 263
 - création d'un gestionnaire de files d'attente (crtmqm) 267
 - définition d'un principal correspondant à un ID utilisateur Compaq NSK (altmqusr) 259
 - définition/réinitialisation des droits d'accès (setmqaut) 146, 300
 - démarrage d'un canal (runmqchl) 291
 - démarrage du gestionnaire de files d'attente de rebut (runmqdlq) 161
 - exécution de l'initialisateur de canal (runmqchi) 290
 - exécution de la gestion interne de la RDF (cleanrdf) 261
 - exécution du programme d'écoute (runmqslr) 294
 - gestionnaire de files d'attente de rebut 292
 - installation de MQSeries pour Compaq NSK (instmqm) 289
 - lancement d'un gestionnaire de files d'attente (strmqm) 307
 - lancement d'un serveur de commandes (strmqcsv) 306
 - lancement de la trace MQSeries (strmqtrc) 309
 - lancement du moniteur de déclenchement (runmqtrm) 299
 - mise à jour gestionnaire de files d'attente V2.2.0.1 (upgmqm) 312
 - modification des attributs d'un fichier de file d'attente (altmqfls) 254
 - MQSC
 - ALTER QLOCAL 106
 - ALTER QREMOTE 133
 - DEFINE CHANNEL 127
 - DEFINE QALIAS 112
 - DEFINE QLOCAL 105
 - DEFINE QLOCAL LIKE 105
 - DEFINE QLOCAL REPLACE 106
 - DEFINE QMODEL 114
 - DEFINE QREMOTE 131
 - DELETE QLOCAL 108
 - DISPLAY QREMOTE 133
 - fichiers de commandes 99
 - utilisation 31
 - vérification 101
 - runmqsc 95

- commandes (*suite*)
 - suppression d'un gestionnaire de files d'attente (dltnqm) 271
- commandes de contrôle 29
 - altnqfls 254
 - altnqusr 259
 - cleanrdf 261
 - cnvclchl 263
 - crtmqcvx 265
 - crtmqm 267
 - distinction entre majuscules et minuscules 29
 - dltnqm 271
 - dspmqaout 273
 - dspmqcsv 277
 - dspmqlfs 278
 - dspmqrtrc 281
 - dspmqrusr 282
 - endmqcsvg 284
 - endmqm 286
 - endmqtrc 288
 - instmqm 289
 - runmqchi 290
 - runmqchl 291
 - runmqdlq 292
 - runmqslr 294
 - runmqsc 95, 296
 - runmqtrm 299
 - setmqaut 300
 - strmqcsvg 306
 - strmqm 307
 - strmqtrc 309
 - upgmqm 312
- commandes MQSC
 - ALTER QLOCAL 106
 - ALTER QREMOTE 133
 - DEFINE CHANNEL 127
 - DEFINE QALIAS 112
 - DEFINE QLOCAL 105
 - DEFINE QLOCAL LIKE 105
 - DEFINE QLOCAL REPLACE 106
 - DEFINE QMODEL 114
 - DEFINE QREMOTE 131
 - DELETE QLOCAL 108
 - DISPLAY QREMOTE 133
 - distinction entre majuscules et minuscules 95
 - longueur maximale des lignes 99
 - utilisation 31
- commandes opérateur (sans réponse) 204
- commandes PATHCOM
 - ADD SERVER 32
 - ALTER SERVER 33
 - FREEZE SERVER 32
 - START SERVER 32
 - STATUS SERVER 32
 - STOP SERVER 32
 - THAW SERVER 32
- Commandes PCF
 - Attributs dans les commandes MQSC et PCF 120
 - automatisation des tâches d'administration à l'aide de commandes PCF 119
 - commandes PCF d'échappement 120
- Commandes PCF (*suite*)
 - MQAI (utilisation à des fins de simplification) 120
 - commandes PCF d'échappement 31, 120
 - commandes programmables (PCF) 31
 - présentation 31
 - commandes programmables PCF 400
 - Compaq NonStop Server pour Java 333
 - compilation et lien d'applications 352
 - composant de communication
 - inter-processus (IPCC) 26
 - composant service 19
 - compteur Measure 226
 - configuration des communications 375
 - configuration pour l'intégrité des données 246
 - configuration pour la disponibilité 246
 - CONNNAME
 - attribut de processus 338
 - contenu
 - MQSINI 190
 - QMINI 191
 - contrôle
 - statut TMF 183
 - contrôle d'un canal 87
 - contrôle d'une file d'attente 80
 - contrôle des gestionnaires de files d'attente 173
 - contrôleur d'exécution principal (rôle) 49
 - convention d'appellation
 - classes de serveur de files d'attente 61
 - classes de serveur de statut 59
 - conventions de dénomination
 - support de langue nationale 251
 - conversion de données 136, 346
 - commande crtmqcvx 265
 - conversion de données par défaut 136
 - conversion de formats de message définis par l'utilisateur 136
 - EBCDIC 51
 - conversion de données par défaut 136
 - copie d'un canal 92
 - copie d'une file d'attente 78
 - CorrelId
 - effet sur les performances 208
 - création
 - définition de canal 84
 - définitions de processus 116
 - file d'attente 76
 - file d'attente de transmission 133
 - gestionnaire de files d'attente 45, 65
 - principaux 66
 - CURDEPTH 226
- D**
 - débugage
 - contrôles secondaires 204, 207
 - erreurs courantes de programmation 203
 - vérifications préalables 199
 - déclenchement
 - définition 7
 - déclenchement (*suite*)
 - définition d'une file d'attente d'application pour 115
 - événements 174
 - gestion des objets 114
 - messages d'une file d'attente d'initialisation 13
 - moniteur
 - commande de lancement 299
 - configuration à l'aide des commandes PATHCOM 33
 - description 14
 - définition automatique de canaux 129
 - définition des files d'attente 76
 - définition dynamique de canaux 129
 - définition PATHWAY APC (exemple) 385
 - définitions de canaux clients 238
 - définitions de canaux non-clients 238
 - définitions de listes de noms 238
 - définitions de processus
 - affichage 117
 - création 116
 - description 15
 - délai des réponses des commandes MQSC 129
 - démarrage
 - canaux 89
 - gestionnaire de files d'attente 67
 - trace 74
 - Démarrage d'initialisateur de canal (Start Channel Initiator) 403
 - Démarrage de programme d'écoute de canal (Start Channel Listener) 403
 - dépassement de capacité de file d'attente
 - déplacement 223
 - désactivation du gestionnaire des droits d'accès aux objets (OAM) 143
 - diagrammes (lecture) 252
 - diagrammes linéaires (lecture) 252
 - disponibilité
 - configuration 246
 - définition 232
 - distinction entre majuscules et minuscules
 - commandes de contrôle 29
 - commandes MQSC 95
 - données non persistantes 233
 - données persistantes 233
 - droit d'accès
 - administration 154
 - droits d'accès
 - commande de définition ou réinitialisation 300
 - commande dspmqaut 147
 - commande setmqaut 147
 - commandes 147
 - contexte 149
 - événements 397
 - fichiers 157
 - groupes d'utilisateurs 142
 - listes 145
 - MQI 151
 - services optionnels 147
 - utilisateur secondaire 148
 - droits d'accès d'utilisateur secondaire 148
 - droits d'accès MQOPEN 152

droits d'accès MQPUT 152
droits de contexte 149

E

EBCDIC

conversion de données 51

EC

fichier de contrôle 238
fonction 22
reprise sur incident 240

EC Boss

reprise sur incident 240

ECBoss

fonction 22

écrans

MQM 70

éloigné

objet de file d'attente (utilisation) 134

éloignée

administration des objets 123
définition d'une file d'attente
(création) 131
émission de commandes MQSC 129
gestion de la sécurité 150
mise en file d'attente
recommandations 130

éloignées

files d'attente
alias de files d'attente de
réponse 135
droits d'accès aux 148
en tant que gestionnaire de files
d'attente alias 134

Eloignés

événements 397

émission de commandes MQSeries 94

EMSCollectorName 52

en-tête de rebut (MQDLH) 161

entrée ExtPoolSize

fichier QMINI 56

environnement d'exploitation 317

équilibre de la charge 49

équilibre de la charge de travail 219

à l'aide de clusters 10

erreur de syntaxe dans les commandes

MQSC 96

erreurs de programmation

(exemples) 203

événement Interdiction 397

événements

canal 174
déclenchement 174

EMS 405

files d'attente 174

outil

activation 175
définition 173
description 173
message 175
types 173
utilisation 173

support dans MQSeries pour Compaq
NSK 397

types 173

Événements EMS 215

événements EMS 175

événements EMS 175 (*suite*)
autre programme de collecte
(définition) 179

définition du PARAM

MQEMSEVENTS 179

écriture de programmes de
traitement 179

programme de collecte par
défaut 179

Événements EMS (Event Management
Service) 175

événements Locaux 397

événements outil

activation 175

but 173

description 173

messages 175

pris en charge par MQSeries pour

Compaq NSK 397

types 173

événements Performances 174, 397

événements Start et Stop 397

évolutivité 219

exemple de communications ICE 388

exemple de communications

TCP/IP 391

exemple de données de trace 216

exemples

commande almqfls 257

commande almqusr 260

commande cleanrdf 262

commande cnvclchl 264

commande crtmqcvx 265

commande crtmqm 270

commande dlmqm 271

commande dspmqaut 276

commande dspmqcsv 277

commande dspmqfls 279

commande dspmqusr 282

commande endmqcsv 284

commande endmqm 287

commande endmqtrc 288

commande runmqsc 298

commande setmqaut 305

commande strmqcsv 306

commande strmqm 307

commande strmqtrc 311

commande upgmqm 313

configuration des

communications 382

erreurs de programmation 203

journal des erreurs 214

exemples de communication

ICE 388

SNAX 382

TCP/IP 391

exemples de communications SNAX 382

exigences

logiciel 317

matériel 317

exit

exit de canal 18

exit de pondération de charge du

cluster 18

exit utilisateur 18

exits utilisateur 365

format de nom 366

exit (*suite*)

installation 370

MQ_LOAD_ENTRY_POINT_EXIT 366

exit utilisateur

description 18

exit de canal 18

exit de conversion de données 18

pondération de charge du cluster 18

exits de conversion de données 365

exits utilisateur 365

ExpectedNumECs 49

F

FFST

consultation 217

sous-volume 62

fichier

base de données des principaux 238

base de données OAM 238

catalogue des objets 238

configuration

identification des incidents 215

définitions de canaux non-clients 238

définitions de listes de noms 238

dépassement de capacité de file

d'attente 222

emplacement 223

dépassement de capacité de

message 222

emplacement 223

modification du seuil 224

droits d'accès 157

ENSCRIBE 27

fichier d'horodatage 238

emplacement 223

fichier d'horodatage d'objet 238

fichier de contact

pour files d'attente d'alias et

éloignées 28

fichier de contrôle EC 238

fichier de dépassement de file

d'attente 238

fichier de file d'attente 222, 238

emplacement 223

fichiers de base de données

contrôlée 238

horodatage 222

initialisation 25

liste de noms 28

Fichier CCSID 136

fichier d'horodatage 222, 238

déplacement 223

fichier d'horodatage d'objet 238

fichier de configuration SCF

(exemple) 382

fichier de dépassement de capacité de file

d'attente 222

fichier de dépassement de capacité de

message 222

déplacement 223

modification du seuil 224

fichier de dépassement de file

d'attente 238

fichier de dépassement de message

configuration pour l'intégrité des

données 246

- fichier de file d'attente 222, 238
 - déplacement 223
 - partition 224
- fichier de synchronisation de canal 238
- fichier make
 - Java 333
- fichier modèle (événements EMS) 175
- fichier modèle d'événements EMS (ZMQSTMPL) 175
- fichiers
 - configuration
 - gestionnaire de files d'attente (QMINI) 191
 - MQSeries (MQSINI) 189
 - fichiers de commandes 99
 - fichiers de configuration
 - généralités 189
 - gestionnaire de files d'attente (QMINI)
 - contenu 191
 - désactivation du gestionnaire des droits d'accès aux objets 143
 - sauvegarde 46
 - strophes 191
 - modification 198
 - MQSeries (mqs.ini)
 - chemin 101
 - MQSeries (MQSINI) 189
 - contenu 190
 - généralités 189
 - sauvegarde 46
 - fichiers ENSCRIBE 20, 27
 - file d'attente d'initialisation
 - définition 116
 - description 13
 - initialisateur de canal (modification) 33
 - file d'attente de commandes 14
 - file d'attente de rebut
 - définition 46
 - description 14
 - gestionnaire 292
 - file d'attente de réponses 15
 - file d'attente de transmission
 - administration éloignée 126
 - création 133
 - définition 127
 - description 14
 - identification dans un gestionnaire 12
 - valeur par défaut 46, 134
 - file d'attente de transmission de cluster 223
 - file d'attente des événements 15
 - files
 - file d'attente de transmission de cluster 223
 - files d'attente
 - alias 13
 - alias (manipulation) 111
 - application (définition pour le déclenchement) 115
 - attributs 12, 345
 - attributs, modification 106
 - commande 14
 - définition 12
 - description 9
 - droits d'accès aux 148
 - files d'attente (*suite*)
 - dynamiques 9
 - éloignées 11, 12
 - création 131
 - gestionnaire de files d'attente
 - alias 134
 - utilisation 134
 - événement 15, 174
 - file d'attente de réponses 15, 134
 - initialisation
 - définition 116
 - messages de déclenchement 13
 - locales 11, 12
 - copie 105
 - définition 103
 - nettoyage 108
 - suppression 108
 - manipulation 103
 - message non transmis 14, 46
 - modèle 13
 - définition 114
 - manipulation 113
 - objets
 - alias 13
 - éloignées 12
 - locaux 12
 - modèle 13
 - pour applications MQI 93
 - prédéfinies 9
 - profondeur de file 226
 - rebut 14, 46
 - réparties (sortie incorrecte) 210
 - taille des 103
 - taille physique des 103
 - temporaires 9
 - transmission 14
 - administration éloignée 126
 - création 133
 - définition 127
 - valeur par défaut 46, 134
 - visualisation 108
 - files d'attente alias
 - description 13
 - droits d'accès aux 148
 - files d'attente dynamiques 9
 - description 9
 - droits d'accès aux 148
 - files d'attente éloignées 123
 - description 11, 12
 - files d'attente locales
 - attributs 345
 - commande 14
 - copie de définitions 105
 - définition 103
 - description 11, 12
 - initialisation 13
 - message non transmis 14
 - nettoyage 108
 - rebut 14
 - suppression 108
 - transmission 14
 - files d'attente modèles
 - attributs 345
 - définition 114
 - description 13
 - manipulation 113
 - files d'attente permanentes 9
 - files d'attente prédéfinies 9
 - files d'attente temporaires 9
 - fin des commandes interactives sur MQSC 96
 - fonction MQMC 70
 - format de message d'événement 398
 - format PDF (Portable Document Format) 414
 - format PostScript 415
 - formats de messages définis par l'utilisateur 136
 - FREEZE SERVER (commande PATHCOM) 32

G

 - gestion des objets pour le déclenchement 114
 - gestion des unités de travail 348
 - gestionnaire de cache de référentiel
 - reprise sur incident 240
 - gestionnaire de files d'attente
 - administration éloignée 123
 - administration locale 93
 - alias avec file d'attente éloignée 134
 - arrêt 68
 - contrôlé 68
 - manuel 331
 - mise au repos 68
 - arrêt forcé 68
 - arrêt immédiat 68
 - attribut 345
 - commande endmqm 286
 - composants 25
 - contrôle 173
 - création
 - commande crtmqm 267
 - instructions 45
 - valeur par défaut 65
 - démarrage 67
 - description 11
 - disponibilité 233
 - droits d'accès 148
 - événements 173
 - fichier de configuration
 - contenu 191
 - sauvegarde 46
 - gestionnaire des droits d'accès aux objets
 - désactivation 143
 - description 140
 - nom unique 46
 - nombre de 46
 - objets
 - appels MQI 12
 - processus 21
 - propriétés configurables 48
 - redémarrer 69
 - répertoires 148
 - reprise sur incident 240
 - serveur de commandes 121
 - sous MVS/ESA 130
 - sous-volume de fichiers de données 62
 - sous-volume de files d'attente de messages 63

- gestionnaire de files d'attente (*suite*)
 - sous-volume de synchronisation des canaux 64
 - sous-volume des journaux d'erreurs 63
 - sous-volume FFST 62
 - spécification dans runmqsc 97
 - suppression 70, 271
 - suppression manuelle 331
 - valeur par défaut 46
 - modification 69
 - modification par erreur 69
 - suppression accidentelle 268
 - volume origine 48

- gestionnaire de files d'attente MVS/ESA 130

- gestionnaire de référentiels
 - processus 23
 - reprise sur incident 240

- gestionnaire de transactions (NonStop TM/MP) 181

- gestionnaire des droits d'accès aux objets (OAM)
 - commande dspmqaut 147
 - commande setmqaut 144, 146
 - désactivation 143
 - fonctionnement 141
 - groupes 141
 - opérations sensibles 148

- Gestionnaire des droits d'accès aux objets (OAM)
 - introduction à 19

- Gestionnaire DLQ
 - lancement 161
 - table de règles 162

- Glossaire 417

- groupe d'utilisateur MQM 139
 - pour les droits d'accès 142

H

- HARDENBO
 - attribut de processus 338
- HTML (Hypertext Markup Language) 414
- Hypertext Markup Language (HTML) 414

I

- ID d'utilisateur connecté Compaq NSK 148
- ID d'utilisateur Guardian 238
- ID de segment 54
- ID de segment guardian 54
- ID utilisateur
 - droits d'accès 139, 148
 - utilisateur connecté Compaq NSK 148
- identification des incidents
 - commandes sans réponse 204
 - éléments à vérifier en premier lieu 199
 - erreurs de programmation 203
 - fichiers de configuration 215

- identification des incidents (*suite*)
 - sortie incorrecte 210
 - vérifications approfondies 204, 207
- identification des problèmes 184
- informations renvoyées par les commandes MQSC 96
- initialisateur de canal
 - configuration à l'aide des commandes PATHCOM 32
 - reprise sur incident 240
- intégrité des données
 - configuration 246
 - définition 232
- interface de file d'attente de messages (MQI) 7
- interface de service annuaire 404
- interface MQM (Message Queue Management) 70
- interface XA 349

J

- Java
 - compatible 333
 - langage pris en charge 351
- Java NonStop
 - langage pris en charge 351
- Java Transaction Services 351
- jeu de commandes
 - administration 29
 - comparaison 327
- journal, erreur 212, 214
- journal des erreurs 212
 - erreur produite avant sa création 213
 - exemple 214
 - sous-volume 63

L

- lancement
 - canaux 128
- lancement d'un gestionnaire de files d'attente. 307
- langage
 - pris en charge 351
- liaison dynamique 23
- liaison native 23
- liaison non native 23
- liaisons 23
- licences 319
- local queue manager agent (LQMA)
 - reprise sur incident 240
- Local Queue Manager Agent (LQMA)
 - fonction 22
- longueur actuelle de la file d'attente 105
- longueur de file d'attente 105
- longueur maximale des lignes pour les commandes MQSC 99
- LQMA (Local Queue Manager Agent) 228
- LQMAgentPriority 53

M

- manuels sous forme électronique 414
- MaxActiveChannels 54

- Maxapplio (valeur) 387
- MaxChannels 53
- MAXHANDS
 - attribut de processus 338
- MaxIdleAgentReuse 53
- MaxIdleAgents 53
- maximum
 - nombre de messages 9
 - taille de file d'attente 9
- maxrcv PARAM 388
- MAXUMSGS
 - attribut de processus 338
- MCAAgentPriority 53
- menu Channels (MQM) 81
- menu Queue Manager (MQM) 72
- menu Queues (MQM) 75
- message
 - algorithmes d'extraction 10
 - considérations sur les performances 208
 - contenant des informations imprévues 210
 - d'événements outil 175
 - descripteur 8
 - description 8
 - longueur 8, 208
 - longueur variable 208
 - mise en file d'attente 7
 - n'apparaissant pas dans les files 209
 - non transmis 215
 - opérateur 213
 - recherche d'un message particulier 208
 - regroupement 9
 - segment 9
 - taille 222
- Message Queue Management (MQM) 70
- messages
 - non-persistants 220
 - persistants 220
- messages d'erreur 96
- messages non persistants
 - configuration pour l'intégrité des données 246
 - différence avec les persistants 8
 - disponibilité 235
- messages non-persistants
 - journalisation synchronisée 221
 - optimisation 220
 - point de contrôle 226
 - stockage 222
- messages opérateur 213
- messages persistants
 - différence avec les non persistants 8
 - disponibilité 234
 - optimisation 220
 - remplacement du nombre maximum par défaut 325
 - stockage 222
- MinIdleLQMAgents 52
- MinIdleMCAcallers 52
- MinIdleMCALU62Responders 52
- MinIdleMCATCPResponders 52
- mise en file d'attente répartie
 - file d'attente de rebut 14
 - file d'attente des messages non transmis 14

mise en file d'attente répartie (*suite*)
 sortie incorrecte 210

mode file d'attente (runmqsc) 129

Modèle de fichier d'événement EMS
 utilisé par MQSeries 405

modification
 canal 83
 CCSID 51
 file d'attente 79

modification des attributs d'un
 gestionnaire de files d'attente 98

modification des attributs de file
 d'attente 106

mot clé APPLIDAT (table de règles) 164

mot clé APPLNAME (table de
 règles) 164

mot clé APPLTYPE (table de règles) 164

mot clé DESTQ (table de règles) 164

mot clé DESTQM (table de règles) 164

mot clé FEEDBACK (table de
 règles) 164

mot clé FORMAT (table de règles) 165

mot clé FWDQ (table de règles) 166

mot clé FWDQM (table de règles) 166

mot clé HEADER (table de règles) 166

mot clé INPUTQ (table des règles) 162

mot clé INPUTQM (table de règles) 163

mot clé MSGTYPE (table de règles) 165

mot clé PERSIST (table de règles) 165

mot clé PUTAUT (table de règles) 166

mot clé REASON (table de règles) 165

mot clé REPLYQ (table de règles) 165

mot clé REPLYQM (table de règles) 165

mot clé RETRY (table de règles) 166

mot clé RETRYINT (table de règles) 163

mot clé USERID (table de règles) 165

mot clé WAIT (table de règles) 163

mots clés d'action 165

mots clés de modèles (table des
 règles) 164

MQ_LOAD_ENTRY_POINT_EXIT 366

MQAI
 description 355
 Description 120

MQCFH (en-tête PCF) 400

MQCFIL, paramètre PCF liste
 d'entiers 401

MQCFIN, entier PCF 401

MQCFSL, liste de chaînes PCF 402

MQCFST (paramètre PCF chaîne) 401

MQCNO 340

MQCNO_FASTPATH_BINDING 229

MQCONN 228

MQCONNECTTYPE 229

MQCONNX 229

MQDATACONVEXIT 346

MQDEFAULTPREFIX 323

MQDISC 344

MQDLH (en-tête de rebut) 161

MQEMSEVENTS 323

MQGMO 341

MQGMO_BROWSE_* 349

MQI
 appels 342
 appels du gestionnaire de files
 d'attente 12
 bibliothèques 352

MQI (*suite*)
 description 7
 droits d'accès 151
 langage de programmation TAL
 MQCLOSE 343
 MQDISC 344
 MQINQ 344
 MQSET 344
 prise en charge de l'administration
 locale 93

MQINQ 344

MQLISTENPORTNUM 323

MQM
 groupe d'utilisateur 139
 ID utilisateur 139, 148

MQMACHINIFILE 323

MQMACHINIFILE (variable
 d'environnement) 189

MQMD 341

MQMREFRESHINT
 paramètre PATHWAY 35

MQPMO 342

MQPUT et MQPUT1 (considérations de
 performance) 208

MQQSHKEEPINT 325

MQQSMAXBATCHEXPURE 325

MQQSMAXMSGSEXPURE 325

MQQSSIGTIMEOUT 325

MQRC_Q_ESPACE_NON_DISPONIBLE
 code retour 103

MQRDF 323

MQRDFUPPROCESSNAME 323

MQRDFUPPROGNAME 323

MQS-CHANINIT00
 classe de serveur TS/MP 32

MQS-CMDSERV00
 classe de serveur TS/MP 32

MQS-EC00
 classe de serveur TS/MP 34

mqs.ini, fichier de configuration
 chemin vers 101

MQS-TCPLIS00
 classe de serveur TS/MP 32

MQS-TCPLISnn
 reconfiguration 56

MQS-TRIGMON00
 classe de serveur TS/MP 33

MQSC 31
 délai des réponses des
 commandes 129
 émission à distance 129
 émission de commandes 94
 émission de commandes de façon
 interactive 95
 fichiers de commandes
 exécution 101
 rapports de sortie 100
 fin de l'entrée interactive 96

incidents
 à distance 131
 local 101

longueur maximale des lignes 99

prises en charge par MQSeries pour
 Compaq NSK 335

redirection des données en entrée et
 en sortie 98

sécurité des canaux 151

MQSC 31 (*suite*)
 utilisation des commandes 98
 vérification des commandes 101

MQSC interactive
 fin 96
 informations renvoyées 96
 utilisation 95

MQSeries pour Compaq NSK
 architecture 21
 commandes MQSC prises en charge
 par 335
 composants 25
 construction d'applications 347
 évolutivité 219
 exécution d'applications 347
 exits utilisateur 365
 fichiers 27
 Modèle de fichier d'événement EMS
 utilisé par 405
 performances 219
 prise en charge des clients 395
 résumé 317
 sous-volumes 27
 structure du volume 62
 super utilisateur, MQM 139
 support de TM/MP (TMF) 181

MQSeries pour Compaq NSK
 (généralités) 317

MQSET 344

MQSINI (fichier de configuration)
 exemple 190
 généralités 189
 modification 198

MQSNOAUT 323

MQTRANSACTIONLIFE
 régénération d'une transaction
 TM/MP 347

MQZAO (constantes et droits
 d'accès) 152

MsgId
 effet sur les performances 208

N

nettoyage d'une file d'attente locale 108

NOHARDENBO
 attribut de processus 338

nom de serveur de file d'attente par
 défaut 59

nom de serveur de statut par défaut 57

noms
 autorisés pour les objets 251
 objets 10

NonStop TM/MP (gestionnaire de
 transactions) 181

NonStop Tuxedo
 environnement de transaction avec
 TM/MP 238

notification des événements 174

NSKSegidRange 54

O

OAM (gestionnaire des droits d'accès aux
 objets)
 remplacement 404

- objets
 - accès aux 139
 - administration éloignée 123
 - conventions de dénomination 251
 - définition de processus 15
 - file d'attente 12
 - gestionnaire de files d'attente
 - appels MQI 12
 - noms 95
 - pour le déclenchement 114
 - système
 - restauration 67
 - valeur par défaut 16, 321
 - types 10
 - valeur par défaut
 - attributs 104
 - restauration 67
 - objets système
 - restauration 67
 - Objets système et objets par défaut 321
 - objets système par défaut 16
 - OpenTMF 237
 - optimisation de MQSeries pour Compaq
 - NSK 219
 - option de signal 341
 - option MQGMO_SET_SIGNAL 341
- P**
- paire de processus NonStop 24
 - paramètre CCSID 51
 - paramètre DefaultPrefix (commande crtmqm) 48
 - paramètre HomeTerminalName 49
 - paramètre NumECs (commande crtmqm) 49
 - paramètres
 - commande altmqfls 255
 - commande altmqusr 259
 - commande cleanrdf 261
 - commande cnvclchl 263
 - commande crtmqcvx 265
 - commande crtmqm 267
 - commande dltnqm 271
 - commande dspmqaut 273
 - commande dspmqcsv 277
 - commande dspmqfls 278
 - commande dspmqtrc 281
 - commande dspmqusr 282
 - commande endmqcsv 284
 - commande endmqm 286
 - commande endmqtrc 288
 - commande instmqm 289
 - commande runmqchi 290
 - commande runmqchl 291
 - commande runmqdlq 292
 - commande runmqslr 294
 - commande runmqsc 297
 - commande runmqtrm 299
 - commande setmqaut 302
 - commande strmqcsv 306
 - commande strmqm 307
 - commande strmqtrc 309
 - commande upgmqm 313
 - PARAMs (variables d'environnement) 323
 - partitionnement
 - fichiers de file d'attente 224
 - PathmonProcName 50
 - PATHWAY
 - classe de serveur 42
 - classe de serveur pour les serveurs de file d'attente 222
 - configuration pour la disponibilité 246
 - erreurs 203
 - exemple de configuration 36
 - PCF (commandes programmables) 400
 - descripteur de message 400
 - en-tête (MQCFH) 400
 - entier (MQCFIN) 401
 - liste de chaînes (MQCFSL) 402
 - MQCFH (en-tête) 400
 - MQCFIN (entier) 401
 - MQCFSL (liste de chaînes) 402
 - MQCFST (paramètre chaîne) 401
 - MQCFST (paramètre liste d'entiers) 401
 - paramètre chaîne (MQCFST) 401
 - paramètre PCF liste d'entiers (MQCFIL) 401
 - pris en charge par MQSeries pour Compaq NSK 402
 - PDF (Portable Document Format) 414
 - performances 219
 - Considérations sur le développement d'applications 207
 - lors du traçage 215
 - PMSEARCH 323
 - point de contrôle
 - des messages non persistants par un serveur de file d'attente 226
 - point de défaillance unique 232
 - point de synchronisation 236
 - limites 182
 - point de synchronisation (considérations de performance) 208
 - points de défaillance multiples 232
 - PRIDB, base de données de principaux 157
 - principal OAM 238
 - principaux
 - appellation 141
 - base de données, PRIDB 157
 - but de 19
 - création 66, 145
 - priorité des processus de files d'attente 53
 - prise en charge transactionnelle 181
 - problèmes de commandes 204
 - processeur
 - charge importante 227
 - configuration pour la disponibilité 246
 - équilibre de la charge de travail 34
 - optimisation 219
 - serveur de file d'attente 222
 - processus (géré par messages) 7
 - processus agents 26, 52
 - processus agents en veille 52
 - processus de canal répondeur LU6.2 376
 - processus EC (nombre de) 49
 - processus géré par messages 7
 - processus gérés par messages 7
 - processus TCP/IP 55
 - processus TCP/IP personnalisé 56
 - programme d'écoute LU6.2 25
 - programmes (exemples fournis) 357
 - programmes d'écoute TCP/IP
 - arrêt 32
 - configuration 32
 - démarrage 32
 - reprise sur incident 240
 - programmes exemples
 - construction de versions C 358
 - construction de versions C++ 360
 - construction de versions COBOL 361
 - construction de versions OSS 363
 - construction de versions TAL 363
 - fournis avec MQSeries pour Compaq NSK 357
 - protocole SNA 375
 - protocole TCP/IP 375
 - publications connexes 415
 - publications MQSeries 413
- Q**
- QMDefaultVolume 48
 - QMINI (fichier de configuration)
 - configuration permettant de prendre en charge les programmes d'écoute TCP/IP 393
 - exemple 193
 - généralités 191
 - modification 198
- R**
- RDF (Fonction de copie de la base de données éloignée)
 - commande cleanrdf 261
 - RDF (Remote Database Duplication Facility)
 - commande cleanrdf 67
 - reprise après incident grave 188
 - redémarrage d'un gestionnaire de files d'attente 69
 - redirection des données en entrées et en sortie (commandes MQSC) 98
 - réinitialisation d'un numéro de séquence de message (MSN) 90
 - répertoires, gestionnaire de files d'attente 148
 - Réponses des commandes PCF 404
 - reprise sur incident 240
 - ressources, protection 139
 - ressources protégées 142
 - restauration
 - objets par défaut 67
 - objets système 67
 - restrictions
 - accès aux objets MQM 139
 - noms d'objet 251
 - return codes
 - commande cleanrdf 261
 - commande crtmqcvx 265
 - commande crtmqm 269
 - commande dltnqm 271

- return codes (*suite*)
 - commande dspmqaut 275
 - commande dspmqfls 279
 - commande endmqm 287
 - commande endmqtrc 288
 - commande runmqchi 290
 - commande runmqchl 291
 - commande runmqsc 297
 - commande runmqtrm 299
 - commande setmqaut 304
 - commande strmqcsv 306
 - commande strmqm 307
 - commande strmqtrc 310
- runmqsc
 - commande 296
 - émission de commandes MQSC 94
 - fin 96
 - incidents 101
 - mode file d'attente 129
 - redirection des données en entrée et en sortie 98
 - résultat 96
 - spécification d'un gestionnaire de files d'attente 97
 - utilisation 98
 - utilisation interactive 95
 - vérification 101

S

- SAFEGUARD 19, 142, 318
- SAVE-ENVIRONMENT ON 323
- sécurité 139
 - activation 143
 - éloignée 150
 - Gestionnaire des droits d'accès aux objets (OAM) 19
 - principaux (création) 66
 - utilisation des commandes 144, 147
- segments de mémoire partagée 25
- serveur de commandes
 - administration éloignée 121
 - affichage 277
 - affichage de l'état 121
 - arrêt d'un serveur de commande 122
 - commande d'arrêt 284
 - commande de lancement 306
 - configuration à l'aide des commandes PATHCOM 32
 - lancement d'un serveur de commandes 121
 - reprise sur incident 240
- serveur de file d'attente
 - ajout et suppression de serveurs de statut personnalisés 60
 - compteur Measure pour la file d'attente 226
 - construction de la structure de données au démarrage du serveur de file d'attente 225
 - description 24
 - gestion des données non persistantes 235
 - messages non persistants avec point de contrôle 226
 - nom par défaut 59
 - optimisation 222

- serveur de file d'attente (*suite*)
 - PARAMs 325
 - processus
 - ajout 35
 - répartition des files d'attente 223
 - répartition entre processeurs 222
 - reprise et redémarrage 187
 - reprise sur incident 240
 - stockage des messages en mémoire 225
 - suppression 61
 - verrouillage en mémoire des chaînes et des structures de données 226
- serveur de file d'attente (ajout et suppression) 60
- serveur de file d'attente NonStop 221
- serveur de gestionnaire de files d'attente
 - reprise sur incident 240
- serveur de statut
 - ajout et suppression de serveurs de statut personnalisés 58
 - nom par défaut 57
 - reprise sur incident 240
 - suppression 59
- serveur de statut personnalisé (ajout et suppression) 58
- serveur MQSS 25
- service annuaire 19
- service d'autorisation 19
 - interface 404
- services optionnels
 - gestionnaire des droits d'accès aux objets (OAM) 140
 - désactivation 143
 - service annuaire 404
 - service d'autorisation 404
- SNA
 - charge processeur 228
 - programme d'écoute SNA 212
- sortie incorrecte 209
- sous-volume 27
- sous-volume de fichiers de données (gestionnaire de files d'attente) 62
- sous-volume de files d'attente de messages 63
- sous-volume de synchronisation des canaux 64
- START SERVER (commande PATHCOM) 32
- status server
 - classe de serveur MQS-STATUS00 34
 - processus
 - ajout 35
 - reprise et redémarrage 187
- STATUS SERVER (commande PATHCOM) 32
- stdin, dans runmqsc 98
- stdout, dans runmqsc 98
- STOP SERVER (commande PATHCOM) 32
- strophe AllQueueManagers (fichier MQSINI) 190
- strophe Authority (fichier QMINI) 192
- strophe ChannelInitiator (fichier QMINI) 192
- strophe Channels (fichier QMINI) 193

- strophe de configuration (fichier QMINI) 191
- strophe DefaultProcess (fichier QMINI) 191
- strophe DefaultQueueManager (fichier MQSINI) 190
- strophe EC (fichier QMINI) 191
- strophe ECBoss (fichier QMINI) 191
- strophe LQMAgent (fichier QMINI) 192
- strophe MCACaller (fichier QMINI) 192
- strophe MCALU62Responder (fichier QMINI) 192
- strophe MCATCPResponder (fichier QMINI) 192
- strophe MQIServer (fichier QMINI) 192
- strophe QueueManager (fichier MQSINI) 190
- strophe Service (fichier QMINI) 192
- strophe ServiceComponent (fichier QMINI) 192
- strophe TCPConfig
 - configuration de TCP/IP 391
- strophe TCPConfig (fichier QMINI) 193
- strophe TCPListener (fichier QMINI) 192
- strophe TuningParameters (fichier QMINI) 193
- strophes
 - MQSINI 190
 - QMINI 191
- super utilisateur, MQM
 - MQM 139
- Support de l'euro 319
- support de TM/MP (TMF) 181, 348
- SupportPac 415
- suppression
 - canal 88
 - file d'attente locale 79, 108
 - gestionnaire de files d'attente 70, 271
 - suppression d'un serveur de file d'attente 61
 - suppression d'un serveur de file d'attente personnalisé 60
 - suppression d'un serveur de statut 59
 - suppression d'un serveur de statut personnalisé 58
 - suppression manuelle de gestionnaires de file d'attente 331
- SYSTEM.CLUSTER.TRANSMIT.QUEUE 223

T

- table de règles (gestionnaire DLQ) 162
 - exemple 170
 - modèles et actions (règles)
 - mot clé ACTION 165
 - mot clé APPLIDAT 164
 - mot clé APPLNAME 164
 - mot clé APPLTYPE 164
 - mot clé DESTQ 164
 - mot clé DESTQM 164
 - mot clé FEEDBACK 164
 - mot clé FORMAT 165
 - mot clé FWDQ 166
 - mot clé FWDQM 166
 - mot clé HEADER 166
 - mot clé MSGTYPE 165

- table de règles (gestionnaire DLQ) 162
 - (suite)
 - mot clé PERSIST 165
 - mot clé PUTAUT 166
 - mot clé REASON 165
 - mot clé REPLYQ 165
 - mot clé REPLYQM 165
 - mot clé RETRY 166
 - mot clé USERID 165
 - saisie de données de contrôle
 - mot clé INPUTQ 162
 - mot clé INPUTQM 163
 - mot clé RETRYINT 163
 - mot clé WAIT 163
 - syntaxe 167
 - traitement 168
- taille de message (diminution) 106
- taille du fichier d'audit, TM/MP 183
- TAL non-natif
 - langage pris en charge 351
- tampon
 - consultation des messages 225
- tampons mémoire 222
- TCP/IP
 - administration éloignée 17
 - canaux 381
- TCPListenerPort 55
- TCPNumListenerPorts 55
- TCPPort 54
- Terminologie utilisée dans le manuel 417
- THAW SERVER (commande PATHCOM) 32
- TM/MP 20
 - contrôle des fichiers de base de données 236
 - coordination avec les bases de données internes 236
 - fichier d'audit 220
- TP ICE 377
- traçage des objets MQSeries 73
- trace
 - considérations sur les performances 215
 - exemple de données 216
- transaction
 - gestion des transactions simultanées 238
- transformation de nom (objet) 64
- transformation des noms d'objets 64
- Tuxedo (NonStop)
 - environnement de transaction avec TM/MP 238
- types d'événements 173
- types d'objets 10
- types de données
 - types de données de structure 339
- types de données de structure 339

U

- unité de travail
 - description 19
- unité de travail globale
 - description 19
- unité de travail locale
 - description 19

- unités d'exécution
 - dans les applications 351
- USERDATA
 - attribut de processus 338
- utilisateurs
 - appartenant à plus d'un groupe 141
 - groupes 141

V

- valeur par défaut
 - file d'attente de transmission 46, 133
 - gestionnaire de files d'attente 46
 - commandes traitées 95
 - modification 69, 97
 - modification par erreur 69
 - suppression accidentelle 268
 - objets 16, 67
 - objets système 321
 - port TCP/IP 54
 - serveur de file d'attente (nom) 59
 - serveur de statut (nom) 57
- validation (commit) et annulation (backout) 181
- validation à deux phases 181
- validation monophasée 181
- validation ou annulation d'un canal 91
- variable d'environnement 323
- variable d'environnement MQEMSEVENTS 179
- variable d'environnement MQSNOAUT 143
- Variables d'environnement TAQL 323
- vérification des commandes MQSC 101
- verrouillage en cache 226
- visualisation des files d'attente 108
- volume
 - structure 62
- volume, modification 106
- volume disque
 - partitionnement de fichiers de file d'attente 224
- volume origine d'un gestionnaire de files d'attente 48



Référence : CT8Z1FR

SC11-1796-00



(1P) P/N: CT8Z1FR

