

MQSeries pour Compaq OpenVMS Alpha



# Administration du système

*Version 5.1*



MQSeries pour Compaq OpenVMS Alpha



# Administration du système

*Version 5.1*

**Remarque**

Avant d'utiliser ce manuel et le produit associé, prenez connaissance des informations générales figurant à l'«Annexe L. Remarques» à la page 383.

**Première édition - mai 2001**

Réf. US : GC34-5884-00

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 1994, 2001. All rights reserved.**

# Table des matières

<b>Figures</b> . . . . .	<b>ix</b>
<b>Tableaux</b> . . . . .	<b>xi</b>
<b>Avis aux lecteurs canadiens</b> . . . . .	<b>xiii</b>
<b>Préface</b> . . . . .	<b>xv</b>
A qui s'adresse ce manuel . . . . .	xv
Connaissances préalables nécessaires à la lecture du manuel . . . . .	xv
Utilisation du manuel . . . . .	xv
Utilisation des annexes . . . . .	xv
Informations sur MQSeries disponibles sur Internet . . . . .	xvi
<b>Nouvelles fonctions de MQSeries pour Compaq OpenVMS Alpha, V5.1</b> . . . . .	<b>xvii</b>
<b>Partie 1. Exploitation</b> . . . . .	<b>1</b>
<b>Chapitre 1. Introduction à MQSeries</b> . . . . .	<b>7</b>
MQSeries et la mise en file d'attente de messages . . . . .	7
Applications asynchrones . . . . .	7
Processus gérés par messages . . . . .	7
Messages et files d'attente . . . . .	7
Définition d'un message . . . . .	8
Définition d'une file d'attente . . . . .	8
Objets . . . . .	9
Noms des objets . . . . .	10
Gestion des objets . . . . .	10
Gestionnaires de files d'attente MQSeries . . . . .	11
Files d'attente MQSeries . . . . .	11
Définitions de processus . . . . .	14
Canaux . . . . .	15
Clusters . . . . .	15
Listes de noms . . . . .	15
Objets système par défaut . . . . .	16
Administration locale et éloignée . . . . .	16
Clients et serveurs . . . . .	16
Applications MQSeries en environnement client-serveur . . . . .	17
Extension des fonctions du gestionnaire de files d'attente . . . . .	17
Exits utilisateur . . . . .	18
Services optionnels . . . . .	18
Sécurité . . . . .	19
Gestionnaire des droits d'accès aux objets (OAM) . . . . .	19
Sécurité DCE . . . . .	19
Prise en charge transactionnelle . . . . .	19
<b>Chapitre 2. Présentation des commandes d'administration MQSeries</b> . . . . .	<b>21</b>
Administration locale et éloignée . . . . .	21
Utilisation des commandes de contrôle . . . . .	21

Administration éloignée à l'aide de commandes MQSC . . . . .	22
Exécution de tâches administratives à l'aide de commandes PCF . . . . .	23
Attributs dans les commandes MQSC et PCF . . . . .	23
Commandes PCF d'échappement . . . . .	23
Signification des noms de fichiers MQSeries . . . . .	23
Transformation de nom du gestionnaire de file d'attente . . . . .	24
Transformation des noms d'objets . . . . .	24
Utilisation des caractères en majuscules et en minuscules . . . . .	25
Utilisation des majuscules et minuscules dans les commandes de contrôle . . . . .	25
Utilisation des majuscules et minuscules dans les commandes MQSC . . . . .	26

<b>Chapitre 3. Gestion des gestionnaires de files d'attente à l'aide de commandes de contrôle</b> . . . . .	<b>27</b>
Commandes de contrôle . . . . .	27
Utilisation des commandes de contrôle . . . . .	27
Création d'un gestionnaire de files d'attente . . . . .	28
Instructions pour la création de gestionnaires de files d'attente . . . . .	28
Création d'un gestionnaire de files d'attente par défaut . . . . .	32
Lancement d'un gestionnaire de files d'attente . . . . .	32
Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant . . . . .	33
Arrêt d'un gestionnaire de files d'attente . . . . .	33
Arrêt progressif . . . . .	33
Arrêt immédiat . . . . .	34
Arrêt forcé . . . . .	34
En cas d'incident . . . . .	34
Redémarrage d'un gestionnaire de files d'attente . . . . .	35
Suppression d'un gestionnaire de files d'attente . . . . .	35

<b>Chapitre 4. Administration des objets MQSeries locaux</b> . . . . .	<b>37</b>
Prise en charge des applications utilisant l'interface MQI . . . . .	37
Administration locale à l'aide de commandes MQSC . . . . .	38
Préambule . . . . .	39
Utilisation de la fonction MQSC de façon interactive . . . . .	39
Informations renvoyées par les commandes MQSC . . . . .	40
Fin de l'entrée interactive sur MQSC . . . . .	41
Affichage des attributs d'un gestionnaire de files d'attente . . . . .	41
Utilisation d'un gestionnaire différent du gestionnaire par défaut . . . . .	42
Modification des attributs d'un gestionnaire de files d'attente . . . . .	42

Exécution de commandes MQSC à partir de fichiers texte . . . . .	42
Fichiers de commandes MQSC . . . . .	43
Rapports MQSC . . . . .	44
Exécution des fichiers de commandes MQSC fournis . . . . .	45
Utilisation de runmqsc pour vérifier des commandes . . . . .	45
Résolution des incidents à l'aide des commandes MQSC . . . . .	46
Manipulation des files d'attente locales . . . . .	47
Définition d'une file d'attente locale . . . . .	48
Définition d'une file d'attente de rebut . . . . .	48
Affichage des attributs d'objet par défaut . . . . .	49
Copie d'une définition de file d'attente locale . . . . .	50
Modification des attributs d'une file d'attente locale . . . . .	50
Nettoyage d'une file d'attente locale . . . . .	51
Suppression d'une file d'attente locale . . . . .	51
Visualisation des files d'attente . . . . .	51
Manipulation des files d'attente alias . . . . .	55
Définition d'une file d'attente d'alias . . . . .	55
Utilisation d'autres commandes avec les files d'attente alias. . . . .	56
Manipulation des files d'attente modèles . . . . .	56
Définition d'une file d'attente modèle . . . . .	57
Utilisation d'autres commandes avec les modèles . . . . .	57
Gestion des objets pour le déclenchement . . . . .	58
Définition d'une file d'attente d'application pour le déclenchement . . . . .	58
Définition d'une file d'attente d'initialisation . . . . .	59
Création d'une définition de processus . . . . .	59
Affichage de la définition de processus . . . . .	60

## Chapitre 5. Automatisation des tâches 61

Commandes programmables PCF . . . . .	61
Attributs dans les commandes MQSC et PCF . . . . .	62
Commandes PCF d'échappement . . . . .	62
Utilisation de l'interface MQAI pour simplifier l'utilisation des commandes programmables PCF. . . . .	62
Gestion du serveur de commandes pour l'administration éloignée . . . . .	63
Lancement du serveur de commandes . . . . .	63
Affichage de l'état du serveur de commandes . . . . .	63
Arrêt d'un serveur de commandes. . . . .	64

## Chapitre 6. Administration des objets

### MQSeries éloignés . . . . . 65

Canaux, clusters et mise en file d'attente éloignée . . . . .	65
Administration éloignée à l'aide de clusters . . . . .	66
Administration éloignée à partir d'un gestionnaire de files d'attente local à l'aide de commandes MQSC . . . . .	67
Préparation des gestionnaires pour l'administration éloignée . . . . .	67
Préparation des canaux et des files d'attente de transmission pour l'administration éloignée . . . . .	68
Définition des canaux et des files d'attente de transmission . . . . .	69
Lancement des canaux. . . . .	70

Définition automatique de canaux . . . . .	71
Emission de commandes MQSC à distance . . . . .	71
Opérations sur des gestionnaires de files d'attente sous MVS/ESA . . . . .	72
En cas d'incident . . . . .	73
Création d'une définition locale de file d'attente éloignée . . . . .	73
Fonctionnement de la définition locale d'une file d'attente éloignée . . . . .	73
Autre mode d'insertion de messages dans une file d'attente éloignée . . . . .	75
Utilisation d'autres commandes avec les files d'attente éloignées . . . . .	75
Création d'une file d'attente de transmission . . . . .	76
Utilisation des définitions de files d'attente éloignées pour les alias . . . . .	77
Gestionnaire de files d'attente alias . . . . .	77
Files d'attente de réponse alias . . . . .	77
Conversion de données . . . . .	77
Modification du CCSID d'un gestionnaire de files d'attente . . . . .	79

## Chapitre 7. Protection des objets

### MQSeries . . . . . 81

Nécessité de protéger les ressources MQSeries. . . . .	81
Avant de commencer . . . . .	81
ID utilisateur dans MQSeries pour Compaq OpenVMS avec l'identificateur de ressource MQM . . . . .	82
Informations complémentaires . . . . .	82
Présentation du gestionnaire des droits d'accès aux objets (OAM). . . . .	83
Mode de fonctionnement de l'OAM . . . . .	83
Gestion de l'accès par le biais d'identificateurs de droits . . . . .	83
Identificateurs de droits par défaut . . . . .	84
Ressources pouvant être protégées par l'OAM. . . . .	84
Utilisation des identificateurs de droits pour les droits d'accès. . . . .	85
Désactivation de l'OAM . . . . .	85
Utilisation des commandes de l'OAM . . . . .	85
Précision des informations indiquées dans les commandes . . . . .	86
Utilisation de la commande setmqaut. . . . .	86
Droits d'accès . . . . .	87
Commande d'affichage des droits d'accès . . . . .	87
Caractéristiques de l'OAM . . . . .	88
ID utilisateur . . . . .	88
Répertoires du gestionnaire de files d'attente . . . . .	88
Files d'attente . . . . .	88
Droits d'accès d'utilisateur secondaire . . . . .	89
Droits de contexte . . . . .	89
Gestion éloignée de la sécurité . . . . .	90
Sécurité des commandes de canal . . . . .	90
Tables de spécification des droits d'accès . . . . .	91
Droits d'accès MQI. . . . .	92
Droits d'administration . . . . .	95
Droits des commandes MQSC dans les PCF d'échappement . . . . .	95
Fichiers de droits d'accès . . . . .	98
Chemins des fichiers de droits d'accès . . . . .	98

Contenu des fichiers de droits d'accès . . . . .	99	Utilisation du journal pour la reprise . . . . .	145
Gestion des fichiers de droits d'accès . . . . .	101	Reprise sur incident . . . . .	145
<b>Chapitre 8. Gestionnaire de files d'attente de rebut (DLQ) MQSeries . . . . .</b>	<b>103</b>	Reconstitution de support . . . . .	145
Lancement du gestionnaire DLQ . . . . .	103	Reconstitution des objets altérés lors du démarrage . . . . .	146
Gestionnaire DLQ exemple (amqsdlq) . . . . .	104	Reconstitution des objets altérés à d'autres moments . . . . .	147
règle du gestionnaire DLQ . . . . .	104	Protection des fichiers journaux MQSeries . . . . .	147
Données de contrôle . . . . .	105	Sauvegarde et restauration . . . . .	148
Règles (modèles et actions) . . . . .	106	Sauvegarde de MQSeries . . . . .	148
Conventions de la règles . . . . .	109	Restauration de MQSeries . . . . .	148
Traitement de la règles . . . . .	110	Scénarios de reprise . . . . .	149
Assurance du traitement de tous les messages . . . . .	111	Incidents d'unité de disque . . . . .	149
Exemple de règle de la règles d'un gestionnaire DLQ . . . . .	113	Objet gestionnaire de files d'attente altéré . . . . .	150
		Simple objet altéré . . . . .	150
		Echec de reconstitution de support automatique . . . . .	151
		Cliché du contenu du fichier journal par la commande dmpmqlog . . . . .	151
<b>Chapitre 9. Événements outils . . . . .</b>	<b>115</b>	<b>Chapitre 12. Utilisation du service annuel . . . . .</b>	<b>171</b>
Description des événements outils . . . . .	115	Utilisation de l'environnement DCE pour partager des files d'attente entre plusieurs gestionnaires de files d'attente . . . . .	171
Utilisation des événements outils . . . . .	116	Tâches de configuration des files d'attente partagées . . . . .	171
Types d'événements . . . . .	117	Configuration DCE . . . . .	172
Notification d'événements via les files d'attente d'événements . . . . .	118	<b>Chapitre 13. Configuration de MQSeries . . . . .</b>	<b>173</b>
Activation et désactivation des événements . . . . .	119	Fichiers de configuration MQSeries . . . . .	173
Messages d'événements . . . . .	119	Modification des fichiers de configuration . . . . .	173
		Fichier de configuration de MQSeries, mqs.ini . . . . .	174
		Fichier de configuration de gestionnaire de files d'attente, qm.ini . . . . .	175
<b>Chapitre 10. Prise en charge transactionnelle . . . . .</b>	<b>121</b>	Attributs permettant de modifier les données de configuration de MQSeries . . . . .	175
Coordination des bases de données . . . . .	122	Strophe AllQueueManagers . . . . .	175
Restrictions . . . . .	123	Strophe ClientExitPath . . . . .	177
Connexions aux bases de données . . . . .	123	Strophe DefaultQueueManager . . . . .	177
Configuration des gestionnaires de bases de données . . . . .	123	Strophe ExitProperties . . . . .	177
Configuration pour Oracle . . . . .	125	Strophe LogDefaults . . . . .	178
Versions minimales requises pour Oracle . . . . .	125	Strophe QueueManager . . . . .	180
Vérification du paramétrage des variables d'environnement . . . . .	125	Modification des données de configuration du gestionnaire de files d'attente . . . . .	181
Activation de la prise en charge de XA Oracle . . . . .	125	Strophe Service . . . . .	181
Création du fichier de commutation Oracle . . . . .	126	Strophe ServiceComponent . . . . .	181
Ajout des données de configuration XAResourceManager pour Oracle . . . . .	128	Strophe Log . . . . .	182
Modification des paramètres de configuration d'Oracle . . . . .	129	Strophe XAResourceManager . . . . .	184
Tâches d'administration . . . . .	130	Strophe Channels . . . . .	186
Unités de travail en attente de validation . . . . .	130	Strophes LU62 et TCP . . . . .	187
Utilisation de la commande dspmqtrn . . . . .	131	Strophe ExitPath . . . . .	189
Utilisation de la commande rsvmqtrn . . . . .	132	Exemples de fichiers mqs.ini et qm.ini . . . . .	189
Résultats incohérents et erreurs . . . . .	132	<b>Chapitre 14. Identification des incidents . . . . .</b>	<b>193</b>
Modification des fichiers de configuration . . . . .	133	Vérifications préalables . . . . .	193
		MQSeries a-t-il fonctionné correctement auparavant ? . . . . .	193
		Existe-t-il des messages d'erreur ? . . . . .	193
<b>Chapitre 11. Reprise et redémarrage . . . . .</b>	<b>135</b>		
Assurance de récupération des messages (journalisation) . . . . .	135		
Aspect des journaux . . . . .	136		
Types de journalisation . . . . .	136		
Définition de points de contrôle – Garantie d'une reprise complète . . . . .	138		
Calcul de la taille des journaux . . . . .	141		
Gestion des journaux . . . . .	142		
Que se passe-t-il lorsqu'un disque est saturé ? . . . . .	143		
Gestion des fichiers journaux . . . . .	143		

Existe-t-il des codes retour clarifiant l'incident ?	194
L'incident peut-il être reproduit ?	194
Des modifications ont-elles été apportées depuis la dernière exécution réussie ?	194
L'exécution de l'application a-t-elle déjà abouti ?	194
L'incident affecte-t-il des parties spécifiques du réseau ?	195
L'incident se produit-il à certaines heures de la journée ?	196
L'incident se produit-il par intermittence ?	196
Avez-vous effectué des mises à jour de fonctions ?	196
Une mise à jour est-elle nécessaire ?	196
Erreurs courantes de programmation	196
Vérifications approfondies	197
Avez-vous obtenu une sortie incorrecte ?	197
Une réponse de la commande PCF vous manque-t-elle ?	197
L'incident est-il lié à certaines de vos files d'attente ?	198
L'incident n'affecte-t-il que les files d'attente éloignées ?	199
Considérations sur le développement d'applications	199
Effet sur la longueur des messages	200
Effet de la persistance d'un message	200
Recherche d'un message particulier	200
Files d'attente contenant des messages de différentes longueurs	200
Fréquence des points de synchronisation	200
Utilisation de l'appel MQPUT1	201
Sortie incorrecte	201
Messages qui n'apparaissent pas dans la file d'attente	201
Messages contenant des informations inattendues ou altérés	202
Sortie incorrecte lors de l'utilisation des files d'attente réparties	203
Journaux d'erreurs	203
Fichiers journaux	204
Erreurs précoces	204
Messages opérateur	205
Exemple de journal des erreurs	205
Files d'attente de rebut	208
Fichiers de configuration et identification des incidents	208
Traçage de MQSeries	208
Fichiers de trace	209
Exemple de données de trace	209
First failure support technology (FFST)	209
Consultation des fichiers FFST	209
Résolution des incidents liés aux clients	214
Arrêt des clients	215
Messages d'erreur sur les postes clients	215

## Chapitre 15. Optimisation des performances . . . . . 217

Définition de la valeur des paramètres spécifiques aux processus	218
--	-----

## Chapitre 16. MQSeries pour OpenVMS et le clustering . . . . . 221

Installation de MQSeries sur un cluster OpenVMS	221
Groupe de reprise de cluster OpenVMS	222
Présentation des groupes de reprise de cluster OpenVMS	222
Concepts des groupes de reprise de cluster OpenVMS	223
Préparation à la configuration d'un groupe de reprise dans un cluster OpenVMS	224
Configuration d'un groupe de reprise dans un cluster OpenVMS	224
Tâches à effectuer après la configuration d'un groupe de reprise dans un cluster OpenVMS	225
Edition du fichier de configuration FAILOVER.INI	225
Procédures de commande utilisées par les groupes de reprise	226
Administration des groupes de reprise	228
Démarrage des moniteurs de reprise	228
Démarrage d'un GFA dans un groupe de reprise	228
Arrêt d'un GFA dans un groupe de reprise	229
Déplacement d'un GFA dans un groupe de reprise	229
Affichage de l'état du groupe de reprise	229
Paramétrage des symboles DCL pour indiquer l'état d'un groupe de reprise	231
Arrêt d'un moniteur de reprise	232
Exécution de commandes pendant une mise à jour	232
Modification de l'état d'un groupe de reprise	232
Paramétrage de la sécurité pour les associations ICC	233
Résolution des incidents liés aux groupes de reprise	234
Utilisation de MultiNet for OpenVMS avec des groupes de reprise	235
Exemple d'utilisation des groupes de reprise	235

## Partie 2. Section de référence . . . 241

### Chapitre 17. Commandes de contrôle MQSeries . . . . . 243

Noms d'objets MQSeries	243
Consultation des fichiers objets	244
Lecture des diagrammes de syntaxe	244
Aide sur la syntaxe	245
Exemples	246
Codes retour de MQSeries	247
crtmqcvx (conversion de données)	248
crtmqm (création d'un gestionnaire de files d'attente)	250
dltmqm (suppression d'un gestionnaire de files d'attente)	254
dmpmqlog (cliché du journal)	257
dspmqaout (affichage des droits d'accès)	259
dspmqcsv (affichage du serveur de commandes)	263
dspmqlfls (affichage des fichiers MQSeries)	264
dspmqrtrc (affichage de la sortie de trace formatée MQSeries)	266



dspmqrn (affichage des transactions MQSeries)	268
endmqcqv (arrêt d'un serveur de commandes)	270
endmqslr (arrêt du programme d'écoute)	273
endmqm (arrêt d'un gestionnaire de files d'attente)	274
endmqtrc (arrêt de la trace MQSeries)	277
failover (gestion d'un groupe de reprise)	278
rcdmqimg (enregistrement d'une image de support)	282
rcrmqobj (recréation d'un objet)	284
rsvmqtrn (résolution des transactions MQSeries)	287
runmqchi (exécution de l'initialisateur de canal)	289
runmqchl (démarrage d'un canal)	290
runmqdlq (lancement du gestionnaire de files d'attente de rebut)	291
runmqfm (lancement d'un moniteur de reprise)	293
runmqslr (exécution du programme d'écoute)	294
runmqsc (exécution des commandes MQSeries)	296
runmqtmc (lancement du moniteur de déclenchement client)	299
runmqtrm (lancement du moniteur de déclenchement)	300
setmqaut (définition ou réinitialisation des droits d'accès)	301
strmqcqv (lancement d'un serveur de commandes)	308
strmqm (lancement d'un gestionnaire de files d'attente)	309
strmqtrc (lancement de la trace MQSeries)	311

## Partie 3. Annexes . . . . . 315

### Annexe A. MQSeries pour Compaq OpenVMS, généralités . . . . . 317

Programme et référence	317
Caractéristiques matérielles requises	317
Caractéristiques logicielles requises	317
Connectivité	317
Sécurité	317
Fonctions de maintenance	318
Compatibilité	318
Compilateurs pris en charge	318
Choix de la langue	318
Compatibilité internationale	318

### Annexe B. Objets système et objets par défaut . . . . . 319

### Annexe C. Structure de répertoires 321

Répertoire et fichiers dans MQS_ROOT:[MQM]	322
Répertoires et fichiers du sous-répertoire MQS_ROOT:[MQM.QMGRS.QMNAME]	322

### Annexe D. Comparaison des jeux de commandes . . . . . 327

Commandes d'administration des gestionnaires de files d'attente	327
Commandes d'administration des serveurs de commandes	327
Commandes d'administration des files d'attente	328
Commandes d'administration des processus	328

Commandes d'administration des canaux	329
Autres commandes de contrôle	329

## Annexe E. Modèles de programmes

### MQI et de fichiers MQSC . . . . . 331

Modèles de fichiers de commandes MQSC	331
Modèles de programmes C et COBOL	331
Outils divers	332

## Annexe F. Modèles de groupe de reprise de cluster OpenVMS . . . . . 333

Modèle de fichier de configuration	
FAILOVER.TEMPLATE	333
Modèle de procédure StartCommand	
START_QM.TEMPLATE	335
Modèle de procédure EndCommand	
END_QM.TEMPLATE	336
Modèle de procédure TidyCommand	
TIDY_QM.TEMPLATE	339

## Annexe G. Jeux de codes acceptés par MQSeries pour Compaq OpenVMS 341

## Annexe H. Utilitaire de diagnostic

### MONMQ. . . . . 343

Généralités	343
Variables utilisées dans MONMQ	344
Attribution des valeurs par défaut	346
Ouverture et création d'une section de trace et de sa boîte aux lettres	346
Affichage de la définition de l'unité logique	346
Fermeture et suppression d'une LU	347
Affichage des informations sur les canaux	347
Affichage du masque de trace actuel d'un canal	348
Affichage du contenu des piles d'unités d'exécution cibles	348
Affichage des processus MQSeries actifs et de l'utilisation de la mémoire	349
Affichage de tous les messages d'un canal	349
Affichage de toutes les sections globales MQSeries d'un noeud	350
Ordre d'envoi de la table mutex au processus de traçage par l'unité d'exécution cible	350
Ordre d'envoi de la table interne des événements au processus de traçage par l'unité d'exécution cible	351
Ordre d'envoi de la table interne de mémoire partagée mappée au processus de traçage par l'unité d'exécution cible	352
Affichage des composants MQSeries actifs, par nom et identificateurs hexadécimaux	353
Affichage des fonctions du composant spécifié	353
Activation du traçage dès le lancement d'un processus	354
Désactivation du traçage dès le lancement d'un processus MQSeries	354
Connexion de l'unité d'exécution cible au canal indiqué	354

Déconnexion de l'unité d'exécution cible du canal indiqué . . . . .	355
Affichage des messages de trace en temps réel destinés à la boîte aux lettres de trace de la LU . . . . .	355
Arrêt d'un processus client en cours. . . . .	355
Indication des données de trace . . . . .	355
Suppression d'une entrée donnée de la table de filtre de trace . . . . .	356
Ecriture de messages de trace dans un fichier binaire par un processus client . . . . .	356
Fermeture d'un fichier binaire de messages de trace . . . . .	357
Ecriture des messages de trace dans un fichier texte . . . . .	357
Fermeture du fichier texte de messages de trace	357
Horodatage des messages . . . . .	357
Désactivation de l'horodatage des messages . . . . .	357
Activation du traçage. . . . .	357
Désactivation du traçage . . . . .	358
Enregistrement de l'historique des messages . . . . .	358
Désactivation de l'historique de message . . . . .	358
Suppression de l'historique des messages . . . . .	358
Définition de la profondeur de l'historique . . . . .	358
Réinitialisation des données de la pile et d'historique pour un canal . . . . .	358
Activation ou désactivation d'un bit de masque	359
Attribution d'une couleur à un canal . . . . .	360
Réacheminement de la sortie vers un fichier . . . . .	360
Analyse du fichier de trace binaire . . . . .	360
Affichage de l'état actuel des unités d'exécution MQSeries. . . . .	363
Fermeture d'une trace et sortie de MONMQ . . . . .	363
Sortie de MONMQ sans fermeture de la trace . . . . .	363
Gestion de la mémoire partagée avec MONMQ	364
Scripts et macros dans MONMQ . . . . .	365
Exemple de session de trace . . . . .	366

**Annexe I. Exits utilisateur . . . . . 377**

Exits de canal et de pondération de charge . . . . .	377
Exits de pondération de charge de cluster MQSeries. . . . .	377

**Annexe J. Applications sécurisées 379**

Applications utilisateur . . . . .	379
Configuration des applications sécurisées . . . . .	379
Exécution de canaux et de programmes d'écoute en tant qu'applications sécurisées. . . . .	380
Messages non persistants rapides. . . . .	380

**Annexe K. Informations utiles pour la configuration . . . . . 381**

Guide de programmation d'applications . . . . .	381
Déclenchement d'application . . . . .	381

**Annexe L. Remarques. . . . . 383**

Marques . . . . .	385
-------------------	-----

**Bibliographie . . . . . 387**

Publications multi-plateformes MQSeries . . . . .	387
Publications MQSeries spécifiques d'une plateforme . . . . .	387
Manuel sous forme électronique . . . . .	388
format HTML . . . . .	388
Format PDF (Portable Document Format) . . . . .	388
Format BookManager . . . . .	389
Format PostScript . . . . .	389
Format d'aide Windows. . . . .	389
Informations sur MQSeries disponibles sur Internet	389
Publications connexes . . . . .	389

**Glossaire . . . . . 391**

**Index . . . . . 405**

**Envoi de vos remarques à IBM. . . . . 415**

---

## Figures

1. Files d'attente, messages et applications	37	18. Exemple de sortie de la commande dmpmqlog	156
2. Sortie type générée par la commande DISPLAY QMGR	41	19. Exemple de fichier de configuration MQSeries pour les systèmes MQSeries pour Compaq OpenVMS	189
3. Extrait du fichier de commandes MQSC monprog.in	44	20. Exemple de fichier de configuration de gestionnaire de files d'attente pour MQSeries pour Compaq OpenVMS.	190
4. Extrait du fichier de rapport MQSC monprog.out	45	21. Exemple de trace MQSeries pour Compaq OpenVMS	209
5. Résultats renvoyés par un afficheur de file d'attente	52	22. Exemple des entrées requises pour ICC\$SYSTARTUP.COM	234
6. Administration éloignée	68	23. Failover.template pour la création d'un fichier de configuration FAILOVER.INI	236
7. Configuration des canaux et des files d'attente pour l'administration éloignée	69	24. Procédure de commande start_failover_set	238
8. Exemple de règle de la règles d'un gestionnaire DLQ	106	25. Procédure de commande end_failover_set	240
9. Présentation des événements outils	116	26. Arborecence de fichiers par défaut après le démarrage d'un gestionnaire de files d'attente	321
10. Contrôle des gestionnaires de files d'attente sur différentes plateformes sur un seul noeud.	117	27. Fichier de configuration modèle : failover.template	334
11. Code source du fichier de commutation Oracle, oraswit.c	126	28. Modèle de procédure StartCommand : Start_QM.template	335
12. Exemple d'entrée XAResourceManager pour Oracle	129	29. Modèle de procédure EndCommand : END_QM.template	338
13. Exemple de sortie de dspmqtrn	131	30. Modèle de procédure TidyCommand : TIDY_QM.template	339
14. Exemple de sortie de dspmqtrn pour une transaction générant une erreur	133		
15. Strophe XAResourceManager mise en commentaire	134		
16. Définition des points de contrôle	139		
17. Définition de points de contrôle pour une transaction de longue durée.	140		



---

## Tableaux

1. Catégories des commandes de contrôle	27	17. Indication des droits d'accès pour différents types d'objets	304
2. Droits d'accès nécessaires pour les appels MQI	92	18. Objets système et objets par défaut - Files d'attente	319
3. Commandes MQSC et droits d'accès nécessaires	95	19. Objets système et objets par défaut - Canaux	320
4. Commandes PCF et droits d'accès nécessaires	96	20. Objets système et objets par défaut - Listes de noms	320
5. Bases de données relationnelles conformes à la norme XA	122	21. Objets système et objets par défaut - Processus	320
6. Tailles des journaux (toutes les valeurs sont approximatives)	141	22. Commandes d'administration des gestionnaires de files d'attente	327
7. Liste des CCSID ISO possibles	176	23. Commandes d'administration des serveurs de commandes	327
8. Valeurs par défaut pour le nombre maximal de demandes de connexion en attente (TCP)	188	24. Commandes d'administration des files d'attente	328
9. Description des champs du fichier FAILOVER.INI	225	25. Commandes d'administration des processus	328
10. Paramètres transmis aux procédures de commande	226	26. Commandes d'administration des canaux	329
11. Etat du GFA dans un groupe de reprise	230	27. Autres commandes de contrôle	329
12. Etats des GFA des noeuds du groupe de reprise	230	28. Fichiers de commandes MQSC	331
13. Etats des moniteurs de noeud du groupe de reprise	230	29. Modèles de programmes - fichiers source	331
14. Symboles DCL et description	231	30. Fichiers divers	332
15. Syntaxe des commandes	244	31. Paramètres nationaux et CCSID	341
16. Droits d'accès de sécurité avec la commande dspmqaut	260		



---

## 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 OpenVMS Alpha, V5.1 (également appelé MQSeries pour Compaq OpenVMS 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 OpenVMS Alpha, V5.1 et des services qu'il fournit pour prendre en charge la messagerie interapplicative en environnement OpenVMS. 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 OpenVMS 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 OpenVMS, 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.

## Utilisation des annexes

Les annexes fournissent des informations de référence. Certaines d'entre elles seront intégrés aux autres manuels MQSeries lors de leur prochaine mise à jour.

## 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/ts/mqseries/>

---

# Nouvelles fonctions de MQSeries pour Compaq OpenVMS Alpha, V5.1

Ces nouvelles fonctions sont décrites dans cette nouvelle édition du manuel *MQSeries pour Compaq OpenVMS Alpha, V5.1 - Administration du système*.

## 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

Pour plus d'informations, consultez la section «Clusters» à la page 15.

**Remarque :** Les clusters MQSeries et les clusters OpenVMS sont deux choses différentes. Le terme *cluster* désigne un cluster de gestionnaires de files d'attente MQSeries. Un cluster OpenVMS est toujours désigné par le terme *cluster OpenVMS*. Pour plus d'informations sur les clusters, reportez-vous au «Chapitre 16. MQSeries pour OpenVMS et le clustering» à la page 221.

## Interface d'administration MQSeries (MQAI)

MQSeries pour Compaq OpenVMS 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*.

## Taille des files d'attente de messages

La taille d'une file d'attente de messages peut aller jusqu'à 2 Go.

## Arrêt 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 OpenVMS fonctionne désormais avec les compilateurs Java.

## Administration Web

Vous pouvez à présent effectuer à distance les tâches d'administration suivantes pour MQSeries pour Compaq OpenVMS à l'aide d'un navigateur HTML (par exemple Netscape Navigator ou Microsoft Internet Explorer) sur un système Windows NT :

- 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



---

# Partie 1. Exploitation

<b>Chapitre 1. Introduction à MQSeries</b> . . . . .	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 . . . . .	7
Définition d'un message . . . . .	8
Tailles de message . . . . .	8
Définition d'une file d'attente. . . . .	8
Envoi et réception de messages par les applications. . . . .	8
Files d'attente prédéfinies et files d'attente dynamiques . . . . .	9
Extraction de messages des files d'attente. . . . .	9
Objets . . . . .	9
Noms des objets . . . . .	10
Gestion des objets . . . . .	10
Attributs d'objet . . . . .	10
Gestionnaires de files d'attente MQSeries . . . . .	11
Appels MQI . . . . .	11
Files d'attente MQSeries . . . . .	11
Utilisation des objets de file d'attente. . . . .	12
Files d'attente locales spécifiques utilisées par MQSeries . . . . .	12
Définitions de processus . . . . .	14
Canaux . . . . .	15
Clusters . . . . .	15
Listes de noms . . . . .	15
Objets système par défaut . . . . .	16
Administration locale et éloignée . . . . .	16
Clients et serveurs . . . . .	16
Applications MQSeries en environnement client-serveur. . . . .	17
Extension des fonctions du gestionnaire de files d'attente . . . . .	17
Exits utilisateur . . . . .	18
Services optionnels . . . . .	18
Sécurité. . . . .	19
Gestionnaire des droits d'accès aux objets (OAM) . . . . .	19
Sécurité DCE . . . . .	19
Prise en charge transactionnelle. . . . .	19
<b>Chapitre 2. Présentation des commandes d'administration MQSeries</b> . . . . .	21
Administration locale et éloignée . . . . .	21
Utilisation des commandes de contrôle . . . . .	21
Administration éloignée à l'aide de commandes MQSC . . . . .	22
Exécution de tâches administratives à l'aide de commandes PCF. . . . .	23
Attributs dans les commandes MQSC et PCF . . . . .	23
Commandes PCF d'échappement . . . . .	23
Signification des noms de fichiers MQSeries . . . . .	23
Transformation de nom du gestionnaire de file d'attente . . . . .	24
Transformation des noms d'objets . . . . .	24

Utilisation des caractères en majuscules et en minuscules . . . . .	25
Utilisation des majuscules et minuscules dans les commandes de contrôle . . . . .	25
Utilisation des majuscules et minuscules dans les commandes MQSC . . . . .	26

<b>Chapitre 3. Gestion des gestionnaires de files d'attente à l'aide de commandes de contrôle</b> . . . . .	27
Commandes de contrôle . . . . .	27
Utilisation des commandes de contrôle . . . . .	27
Création d'un gestionnaire de files d'attente . . . . .	28
Instructions pour la création de gestionnaires de files d'attente . . . . .	28
Définition d'un nom de gestionnaire de files d'attente unique. . . . .	29
Limitation du nombre de gestionnaires de files d'attente . . . . .	29
Définition du gestionnaire de files d'attente par défaut . . . . .	29
Définition de la file d'attente de rebut . . . . .	30
Définition de la file d'attente de transmission . . . . .	31
Définition des paramètres de journalisation nécessaires. . . . .	31
Sauvegarde des fichiers de configuration après la création d'un gestionnaire de files d'attente . . . . .	31
Création d'un gestionnaire de files d'attente par défaut . . . . .	32
Lancement d'un gestionnaire de files d'attente. . . . .	32
Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant . . . . .	33
Arrêt d'un gestionnaire de files d'attente . . . . .	33
Arrêt progressif . . . . .	33
Arrêt immédiat . . . . .	34
Arrêt forcé. . . . .	34
En cas d'incident . . . . .	34
Redémarrage d'un gestionnaire de files d'attente . . . . .	35
Suppression d'un gestionnaire de files d'attente . . . . .	35

<b>Chapitre 4. Administration des objets MQSeries locaux</b> . . . . .	37
Prise en charge des applications utilisant l'interface MQI. . . . .	37
Administration locale à l'aide de commandes MQSC . . . . .	38
Préambule. . . . .	39
Noms des objets MQSeries . . . . .	39
Redirection des données en entrée et en sortie . . . . .	39
Utilisation de la fonction MQSC de façon interactive . . . . .	39
Informations renvoyées par les commandes MQSC . . . . .	40
Fin de l'entrée interactive sur MQSC . . . . .	41
Affichage des attributs d'un gestionnaire de files d'attente . . . . .	41
Utilisation d'un gestionnaire différent du gestionnaire par défaut . . . . .	42

Modification des attributs d'un gestionnaire de files d'attente . . . . .	42	Lancement des canaux. . . . .	70
Exécution de commandes MQSC à partir de fichiers texte. . . . .	42	Définition automatique de canaux . . . . .	71
Fichiers de commandes MQSC . . . . .	43	Emission de commandes MQSC à distance . . . . .	71
Rapports MQSC . . . . .	44	Opérations sur des gestionnaires de files d'attente sous MVS/ESA . . . . .	72
Exécution des fichiers de commandes MQSC fournis . . . . .	45	Recommandations pour la mise en file d'attente éloignée . . . . .	73
Utilisation de runmqsc pour vérifier des commandes . . . . .	45	En cas d'incident . . . . .	73
Résolution des incidents à l'aide des commandes MQSC . . . . .	46	Création d'une définition locale de file d'attente éloignée . . . . .	73
Manipulation des files d'attente locales . . . . .	47	Fonctionnement de la définition locale d'une file d'attente éloignée . . . . .	73
Définition d'une file d'attente locale . . . . .	48	Exemple . . . . .	74
Définition d'une file d'attente de rebut . . . . .	48	Autre mode d'insertion de messages dans une file d'attente éloignée . . . . .	75
Affichage des attributs d'objet par défaut . . . . .	49	Utilisation d'autres commandes avec les files d'attente éloignées . . . . .	75
Copie d'une définition de file d'attente locale . . . . .	50	Création d'une file d'attente de transmission . . . . .	76
Modification des attributs d'une file d'attente locale . . . . .	50	Files d'attente de transmission par défaut . . . . .	76
Nettoyage d'une file d'attente locale . . . . .	51	Utilisation des définitions de files d'attente éloignées pour les alias . . . . .	77
Suppression d'une file d'attente locale . . . . .	51	Gestionnaire de files d'attente alias . . . . .	77
Visualisation des files d'attente . . . . .	51	Files d'attente de réponse alias . . . . .	77
Manipulation des files d'attente alias . . . . .	55	Conversion de données . . . . .	77
Définition d'une file d'attente d'alias . . . . .	55	Quand le gestionnaire de files d'attente ne peut pas convertir les messages dans des formats livrés d'origine . . . . .	78
Utilisation d'autres commandes avec les files d'attente alias. . . . .	56	Fichier ccsid.tbl . . . . .	78
Manipulation des files d'attente modèles . . . . .	56	Conversion de messages en formats définis par l'utilisateur . . . . .	79
Définition d'une file d'attente modèle . . . . .	57	Modification du CCSID d'un gestionnaire de files d'attente . . . . .	79
Utilisation d'autres commandes avec les modèles . . . . .	57		
Gestion des objets pour le déclenchement . . . . .	58		
Définition d'une file d'attente d'application pour le déclenchement . . . . .	58		
Définition d'une file d'attente d'initialisation . . . . .	59		
Création d'une définition de processus . . . . .	59		
Affichage de la définition de processus . . . . .	60		
<b>Chapitre 5. Automatisation des tâches . . . . .</b>	<b>61</b>	<b>Chapitre 7. Protection des objets MQSeries. . . . .</b>	<b>81</b>
Commandes programmables PCF . . . . .	61	Nécessité de protéger les ressources MQSeries. . . . .	81
Attributs dans les commandes MQSC et PCF . . . . .	62	Avant de commencer . . . . .	81
Commandes PCF d'échappement . . . . .	62	ID utilisateur dans MQSeries pour Compaq OpenVMS avec l'identificateur de ressource MQM . . . . .	82
Utilisation de l'interface MQAI pour simplifier l'utilisation des commandes programmables PCF. . . . .	62	Informations complémentaires . . . . .	82
Gestion du serveur de commandes pour l'administration éloignée . . . . .	63	Présentation du gestionnaire des droits d'accès aux objets (OAM). . . . .	83
Lancement du serveur de commandes . . . . .	63	Mode de fonctionnement de l'OAM . . . . .	83
Affichage de l'état du serveur de commandes . . . . .	63	Gestion de l'accès par le biais d'identificateurs de droits . . . . .	83
Arrêt d'un serveur de commandes. . . . .	64	Identificateurs de droits et identificateur de droits primaires . . . . .	83
		Principal doté de plusieurs identificateurs de droits d'accès. . . . .	84
		Identificateurs de droits par défaut . . . . .	84
		Ressources pouvant être protégées par l'OAM. . . . .	84
		Utilisation des identificateurs de droits pour les droits d'accès. . . . .	85
		Désactivation de l'OAM . . . . .	85
		Utilisation des commandes de l'OAM . . . . .	85
		Précision des informations indiquées dans les commandes . . . . .	86
		Listes de droits d'accès . . . . .	86
		Utilisation de la commande setmqaut. . . . .	86
		Commandes de droits d'accès et services optionnels . . . . .	87

Droits d'accès . . . . .	87	Définition des gestionnaires de bases de données . . . . .	124
Commande d'affichage des droits d'accès . . . . .	87	Configuration pour Oracle . . . . .	125
Caractéristiques de l'OAM . . . . .	88	Versions minimales requises pour Oracle . . . . .	125
ID utilisateur . . . . .	88	Vérification du paramétrage des variables d'environnement . . . . .	125
Répertoires du gestionnaire de files d'attente . . . . .	88	Activation de la prise en charge de XA Oracle . . . . .	125
Files d'attente . . . . .	88	Création du fichier de commutation Oracle . . . . .	126
Droits d'accès d'utilisateur secondaire . . . . .	89	Création du fichier de commutation Oracle sur les systèmes OpenVMS . . . . .	126
Droits de contexte . . . . .	89	Ajout des données de configuration . . . . .	128
Gestion éloignée de la sécurité . . . . .	90	XAResourceManager pour Oracle . . . . .	128
Sécurité des commandes de canal . . . . .	90	Modification des paramètres de configuration d'Oracle . . . . .	129
Commandes PCF . . . . .	90	Tâches d'administration . . . . .	130
Commandes de canal MQSC . . . . .	91	Unités de travail en attente de validation . . . . .	130
Commandes de contrôle des canaux . . . . .	91	Utilisation de la commande dspmqtrn . . . . .	131
Tables de spécification des droits d'accès . . . . .	91	Utilisation de la commande rsvmqtrn . . . . .	132
Droits d'accès MQI . . . . .	92	Résultats incohérents et erreurs . . . . .	132
Droits d'administration . . . . .	95	Modification des fichiers de configuration . . . . .	133
Droits des commandes MQSC dans les PCF d'échappement . . . . .	95	Suppression d'instances de gestionnaire de bases de données . . . . .	134
Droits pour les commandes PCF . . . . .	96	<b>Chapitre 11. Reprise et redémarrage . . . . .</b>	<b>135</b>
Fichiers de droits d'accès . . . . .	98	Assurance de récupération des messages (journalisation) . . . . .	135
Chemins des fichiers de droits d'accès . . . . .	98	Aspect des journaux . . . . .	136
Répertoires de droits d'accès . . . . .	98	Fichier de contrôle de journal . . . . .	136
Contenu des fichiers de droits d'accès . . . . .	99	Types de journalisation . . . . .	136
Fichiers de droits d'accès aux classes . . . . .	100	Journalisation circulaire . . . . .	136
Fichiers de droits d'accès à toutes les classes . . . . .	101	Journalisation linéaire . . . . .	137
Gestion des fichiers de droits d'accès . . . . .	101	Définition de points de contrôle – Garantie d'une reprise complète . . . . .	138
Accès aux fichiers de droits d'accès . . . . .	101	Calcul de la taille des journaux . . . . .	141
<b>Chapitre 8. Gestionnaire de files d'attente de rebut (DLQ) MQSeries . . . . .</b>	<b>103</b>	Gestion des journaux . . . . .	142
Lancement du gestionnaire DLQ . . . . .	103	Que se passe-t-il lorsqu'un disque est saturé ? . . . . .	143
Gestionnaire DLQ exemple (amqsdlq) . . . . .	104	Gestion des fichiers journaux . . . . .	143
règle du gestionnaire DLQ . . . . .	104	Emplacement des fichiers journaux . . . . .	144
Données de contrôle . . . . .	105	Utilisation du journal pour la reprise . . . . .	145
Règles (modèles et actions) . . . . .	106	Reprise sur incident . . . . .	145
Mots clés de modèle . . . . .	106	Reconstitution de support . . . . .	145
Mots clés d'action . . . . .	107	Reconstitution d'images de support . . . . .	146
Conventions de la règles . . . . .	109	Reconstitution des objets altérés lors du démarrage . . . . .	146
Traitement de la règles . . . . .	110	Reconstitution des objets altérés à d'autres moments . . . . .	147
Assurance du traitement de tous les messages . . . . .	111	Protection des fichiers journaux MQSeries . . . . .	147
Exemple de règle de la règles d'un gestionnaire DLQ . . . . .	113	Sauvegarde et restauration . . . . .	148
<b>Chapitre 9. Événements outils . . . . .</b>	<b>115</b>	Sauvegarde de MQSeries . . . . .	148
Description des événements outils . . . . .	115	Restauration de MQSeries . . . . .	148
Utilisation des événements outils . . . . .	116	Scénarios de reprise . . . . .	149
Types d'événements . . . . .	117	Incidents d'unité de disque . . . . .	149
Notification d'événements via les files d'attente d'événements . . . . .	118	Objet gestionnaire de files d'attente altéré . . . . .	150
Utilisation de files d'attente d'événements déclenchées . . . . .	118	Simple objet altéré . . . . .	150
Activation et désactivation des événements . . . . .	119	Echec de reconstitution de support automatique . . . . .	151
Messages d'événements . . . . .	119	Cliché du contenu du fichier journal par la commande dmpmqlog . . . . .	151
<b>Chapitre 10. Prise en charge transactionnelle . . . . .</b>	<b>121</b>	<b>Chapitre 12. Utilisation du service annuaire . . . . .</b>	<b>171</b>
Coordination des bases de données . . . . .	122		
Restrictions . . . . .	123		
Connexions aux bases de données . . . . .	123		
Configuration des gestionnaires de bases de données . . . . .	123		
Création de fichiers de commutation . . . . .	124		



Utilisation de l'environnement DCE pour partager des files d'attente entre plusieurs gestionnaires de files d'attente . . . . .	171	Une réponse de la commande PCF vous manque-t-elle ? . . . . .	197
Tâches de configuration des files d'attente partagées . . . . .	171	L'incident est-il lié à certaines de vos files d'attente ? . . . . .	198
Configuration DCE . . . . .	172	L'incident n'affecte-t-il que les files d'attente éloignées ? . . . . .	199
<b>Chapitre 13. Configuration de MQSeries</b> . . . . .	173	Considérations sur le développement d'applications . . . . .	199
Fichiers de configuration MQSeries . . . . .	173	Effet sur la longueur des messages . . . . .	200
Modification des fichiers de configuration . . . . .	173	Effet de la persistance d'un message . . . . .	200
Quand modifier un fichier de configuration	174	Recherche d'un message particulier . . . . .	200
Priorités valables dans les fichiers de configuration . . . . .	174	Files d'attente contenant des messages de différentes longueurs . . . . .	200
Implémentation des modifications dans les fichiers de configuration . . . . .	174	Fréquence des points de synchronisation . . . . .	200
Fichier de configuration de MQSeries, mqs.ini	174	Utilisation de l'appel MQPUT1 . . . . .	201
Fichier de configuration de gestionnaire de files d'attente, qm.ini . . . . .	175	Sortie incorrecte . . . . .	201
Attributs permettant de modifier les données de configuration de MQSeries . . . . .	175	Messages qui n'apparaissent pas dans la file d'attente . . . . .	201
Strophe AllQueueManagers . . . . .	175	Messages contenant des informations inattendues ou altérées . . . . .	202
Strophe ClientExitPath . . . . .	177	Sortie incorrecte lors de l'utilisation des files d'attente réparties . . . . .	203
Strophe DefaultQueueManager . . . . .	177	Journaux d'erreurs . . . . .	203
Strophe ExitProperties . . . . .	177	Fichiers journaux . . . . .	204
Strophe LogDefaults . . . . .	178	Erreurs précoces . . . . .	204
Strophe QueueManager . . . . .	180	Messages opérateur . . . . .	205
Modification des données de configuration du gestionnaire de files d'attente . . . . .	181	Exemple de journal des erreurs . . . . .	205
Strophe Service . . . . .	181	Files d'attente de rebut . . . . .	208
Strophe ServiceComponent . . . . .	181	Fichiers de configuration et identification des incidents . . . . .	208
Strophe Log . . . . .	182	Traçage de MQSeries . . . . .	208
Strophe XAResourceManager . . . . .	184	Fichiers de trace . . . . .	209
Strophe Channels . . . . .	186	Exemple de données de trace . . . . .	209
Strophes LU62 et TCP . . . . .	187	First failure support technology (FFST) . . . . .	209
Strophe ExitPath . . . . .	189	Consultation des fichiers FFST . . . . .	209
Exemples de fichiers mqs.ini et qm.ini . . . . .	189	Résolution des incidents liés aux clients . . . . .	214
<b>Chapitre 14. Identification des incidents</b> . . . . .	193	Arrêt des clients . . . . .	215
Vérifications préalables . . . . .	193	Messages d'erreur sur les postes clients . . . . .	215
MQSeries a-t-il fonctionné correctement auparavant ? . . . . .	193	Systèmes clients OS/2, UNIX, et OpenVMS	215
Existe-t-il des messages d'erreur ? . . . . .	193	Clients DOS et Windows* . . . . .	215
Existe-t-il des codes retour clarifiant l'incident ?	194	<b>Chapitre 15. Optimisation des performances</b> . . . . .	217
L'incident peut-il être reproduit ? . . . . .	194	Définition de la valeur des paramètres spécifiques aux processus . . . . .	218
Des modifications ont-elles été apportées depuis la dernière exécution réussie ? . . . . .	194	<b>Chapitre 16. MQSeries pour OpenVMS et le clustering</b> . . . . .	221
L'exécution de l'application a-t-elle déjà abouti ?	194	Installation de MQSeries sur un cluster OpenVMS	221
Si l'application ne s'est jamais exécutée correctement auparavant . . . . .	195	Groupe de reprise de cluster OpenVMS . . . . .	222
L'incident affecte-t-il des parties spécifiques du réseau ? . . . . .	195	Présentation des groupes de reprise de cluster OpenVMS . . . . .	222
L'incident se produit-il à certaines heures de la journée ? . . . . .	196	Concepts des groupes de reprise de cluster OpenVMS . . . . .	223
L'incident se produit-il par intermittence ? . . . . .	196	Préparation à la configuration d'un groupe de reprise dans un cluster OpenVMS . . . . .	224
Avez-vous effectué des mises à jour de fonctions ? . . . . .	196	Configuration d'un groupe de reprise dans un cluster OpenVMS . . . . .	224
Une mise à jour est-elle nécessaire ? . . . . .	196	Tâches à effectuer après la configuration d'un groupe de reprise dans un cluster OpenVMS . . . . .	225
Erreurs courantes de programmation . . . . .	196	Edition du fichier de configuration FAILOVER.INI . . . . .	225
Vérifications approfondies . . . . .	197		
Avez-vous obtenu une sortie incorrecte ? . . . . .	197		



Procédures de commande utilisées par les groupes de reprise. . . . .	226
Administration des groupes de reprise . . . . .	228
Démarrage des moniteurs de reprise . . . . .	228
Démarrage d'un GFA dans un groupe de reprise	228
Arrêt d'un GFA dans un groupe de reprise . . . . .	229
Déplacement d'un GFA dans un groupe de reprise. . . . .	229
Affichage de l'état du groupe de reprise . . . . .	229
Paramétrage des symboles DCL pour indiquer l'état d'un groupe de reprise . . . . .	231
Arrêt d'un moniteur de reprise . . . . .	232
Exécution de commandes pendant une mise à jour . . . . .	232
Modification de l'état d'un groupe de reprise	232
Paramétrage de la sécurité pour les associations ICC . . . . .	233
Résolution des incidents liés aux groupe de reprise. . . . .	234
Utilisation de MultiNet for OpenVMS avec des groupes de reprise. . . . .	235
Exemple d'utilisation des groupes de reprise	235
Personnalisation de failover.template . . . . .	235
Modification des procédures de commandes d'un groupe de reprise . . . . .	237
Exemple de procédure de commande de démarrage de groupe de reprise :	
start_failover_set.com. . . . .	237
Exemple de procédure de commande d'arrêt de groupe de reprise, end_failover_set.com . . . . .	239



---

## Chapitre 1. Introduction à MQSeries

Ce chapitre présente MQSeries pour Compaq OpenVMS du point de vue d'un administrateur, et décrit les concepts de base de MQSeries et de la messagerie. Il contient les sections suivantes :

- «MQSeries et la mise en file d'attente de messages»
- «Messages et files d'attente»
- «Objets» à la page 9
- «Objets système par défaut» à la page 16
- «Administration locale et éloignée» à la page 16
- «Clients et serveurs» à la page 16
- «Extension des fonctions du gestionnaire de files d'attente» à la page 17
- «Sécurité» à la page 19

---

### MQSeries et la mise en file d'attente de messages

MQSeries permet aux programmes d'application d'utiliser la mise en file d'attente pour participer à des processus gérés par messages. Les programmes d'application installés sur différentes plateformes peuvent communiquer par le biais de logiciels de mise en file d'attente de messages. Par exemple, les applications OpenVMS et MVS/ESA\* peuvent respectivement communiquer à l'aide de MQSeries pour Compaq OpenVMS 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 d'application 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.

## Messages et files d'attente

### 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 d'application 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*.

### 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.

### 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 2 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 visitez le site Web suivant :

<http://www.ibm.com/software/ts/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 ces appels, 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.

### 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 du manuel chapitre :

- «Chapitre 2. Présentation des commandes d'administration MQSeries» à la page 21
- «Chapitre 3. Gestion des gestionnaires de files d'attente à l'aide de commandes de contrôle» à la page 27

## Objets

- «Chapitre 4. Administration des objets MQSeries locaux» à la page 37
- «Chapitre 6. Administration des objets MQSeries éloignés» à la page 65

### Noms des objets

Les conventions de dénomination adoptées pour MQSeries dépendent de l'objet concerné.

Chaque instance d'un objet gestionnaire de files d'attente est connue par son nom. Ce nom doit être unique sur le réseau de gestionnaires interconnectés, de façon qu'un gestionnaire de files d'attente puisse identifier sans ambiguïté le gestionnaire cible auquel un message donné doit être envoyé.

Pour les autres types d'objet, chacun possède un nom et peut être référencé par ce nom dans MQSeries. 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 de *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 243.

### Gestion des objets

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

- Commandes de contrôle saisies à partir du clavier
- Commandes MQSeries (MQSC), saisies à partir d'un clavier et lues à partir d'un fichier.
- Messages PCF (format de commande programmable) utilisés dans un programme d'automatisation.
- Appels MQSeries Administration Interface (MQAI) dans un programme.

Pour plus d'informations, reportez-vous au «Chapitre 2. Présentation des commandes d'administration MQSeries» à la page 21.

#### Attributs d'objet

Les propriétés d'un objet sont définies par ses attributs. Vous pouvez en définir 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* configurable par l'utilisateur lors de la création d'une file. L'attribut *DefinitionType* indique comment la file d'attente a été créée. Vous pouvez seulement le visualiser.

Il existe deux façons de référencer un attribut dans MQSeries :

- à l'aide de son nom PCF ; par exemple *MaxMsgLength*.
- 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 files d'attente MQSeries

Un gestionnaire de files d'attente fournit aux applications des fonctions de mise en file d'attente, et gère les files qui lui appartiennent. 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 *local queue*. 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 les files d'attente sont définies à l'aide des commandes suivantes :

- Commande appropriée MQSC DEFINE
- Commande PCF Create Queue (création d'une file d'attente)

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 ;
- 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 de détails sur la définition des objets file d'attente, reportez vous au manuel *MQSeries - Guide de référence des commandes* ou 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 cluster, reportez-vous à la section «Administration éloignée à l'aide de clusters» à la page 66.

- **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 spécifiques 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.

Reportez-vous aux sections «Gestion des objets pour le déclenchement» à la page 58, et «runmqtrm (lancement du moniteur de déclenchement)» à la page 300. Pour plus d'informations sur le déclenchement, reportez-vous au manuel *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. Les files de ce type sont également utilisées pour l'administration éloignée ; reportez-vous à la section «Administration éloignée à partir d'un gestionnaire de files d'attente local à l'aide de commandes MQSC» à la page 67. 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. Dans les versions 5.1 de MQSeries, la définition de cette file 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 en 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.

## Objets

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.

Une file de commandes est automatiquement définie pour chaque gestionnaire de files d'attente créé.

**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 Version 5.1 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 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.

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 68.

## 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.

**Remarque :** Les clusters MQSeries et les clusters OpenVMS sont deux choses différentes. Le terme *cluster* désigne un cluster de gestionnaires de files d'attente MQSeries. Un cluster OpenVMS est toujours désigné par le terme *cluster OpenVMS*. Pour plus d'informations sur les clusters, reportez-vous au «Chapitre 16. MQSeries pour OpenVMS et le clustering» à la page 221.

Pour plus d'informations sur les clusters, reportez-vous au «Chapitre 6. Administration des objets MQSeries éloignés» à la page 65 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.

## Objets

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. Vous pouvez copier et modifier chacune de ces définitions d'objet pour les utiliser dans les applications de votre installation. La racine du nom des objets par défaut est SYSTEM.DEF ; par exemple, la file d'attente locale par défaut s'appelle SYSTEM.DEFAULT.LOCAL.QUEUE, et le canal récepteur par défaut est SYSTEM.DEF.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.

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 319.

---

### 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 prend en charge les configurations client-serveur pour les applications MQSeries.

Un *client MQSeries* 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 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 MQSeries locales normales.

La différence entre un gestionnaire de files d'attente ordinaire et un serveur MQI est que ce dernier dispose d'une liaison dédiée avec chaque client. Pour plus d'informations sur la création de canaux client et serveur, reportez-vous au manuel *MQSeries Intercommunication*.

Pour plus d'informations sur le support client en général, reportez-vous au manuel *MQSeries - Clients*.

### Applications MQSeries en environnement client-serveur

Une fois associées à un serveur, les applications MQSeries 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 lier vos applications aux bibliothèques clientes appropriées. Consultez le manuel *MQSeries - Clients* pour plus d'informations.

---

## 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.

## Extension des fonctions

### 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 *Guide de programmation d'applications*.

- **Exit de pondération de charge du cluster**

La fonction de cet exit est définie par son fournisseur. Pour plus d'informations, consultez le manuel *MQSeries Queue Manager Clusters*.

Tous les types d'exits sont liés à la mise en file d'attente répartie. Pour plus d'informations sur ces exits et leur utilisation, consultez le manuel *MQSeries Intercommunication*.

### Services optionnels

Les services optionnels sont plus puissants que les exits, dans le sens où ils offrent des interfaces formalisées (API) 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. L'OAM est actif par défaut, et vous n'avez rien à faire pour le configurer. 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 7. Protection des objets MQSeries» à la page 81.

- **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. Ce service est mis en oeuvre par un composant par défaut livré avec MQSeries Version 5.1. Il utilise l'environnement DCE (Distributed Computing Environment) de l'OSF (Open Software Foundation). Vous pouvez également écrire votre propre composant de service annuaire. Par défaut, le service annuaire est inactif.

Consultez le «Chapitre 12. Utilisation du service annuaire» à la page 171 et le manuel *MQSeries Programmable System Management*.

---

## Sécurité

Il existe deux moyens d'assurer la sécurité dans les produits MQSeries Version 5 :

- Le gestionnaire des droits d'accès aux objets (OAM)
- La sécurité DCE

### 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 le biais des groupes d'utilisateurs MQSeries et de l'OAM. Une interface de ligne de commande est fournie pour permettre aux administrateurs d'accorder ou de révoquer des droits selon les besoins.

Pour plus d'informations sur la création de composants d'autorisation, reportez-vous au manuel *MQSeries Programmable System Management*.

### Sécurité DCE

Les exits de canal qui utilisent la fonction DCE de sécurité générique (GSS) sont fournis par MQSeries. Pour plus d'informations, reportez-vous au manuel *MQSeries Intercommunication*.

---

## Prise en charge transactionnelle

Un programme d'application 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.

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 (GFA) 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.

Pour plus d'informations, reportez-vous au «Chapitre 10. Prise en charge transactionnelle» à la page 121.





---

## Chapitre 2. Présentation des commandes d'administration MQSeries

Ce chapitre présente les commandes d'administration de MQSeries.

Les tâches d'administration comprennent la création, le démarrage, la modification, la visualisation, l'arrêt et la suppression d'objets MQSeries (à savoir gestionnaires de files d'attente, files d'attente, processus, noms de listes, clusters et canaux).

Ce chapitre contient les sections suivantes :

- «Administration locale et éloignée»
- «Utilisation des commandes de contrôle»
- «Administration éloignée à l'aide de commandes MQSC» à la page 22
- «Exécution de tâches administratives à l'aide de commandes PCF» à la page 23
- «Signification des noms de fichiers MQSeries» à la page 23.
- «Utilisation des caractères en majuscules et en minuscules» à la page 25

---

### Administration locale et éloignée

Les objets MQSeries peuvent être administrés localement ou à distance.

*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.

Pour plus d'informations sur l'administration à distance, reportez-vous au «Chapitre 6. Administration des objets MQSeries éloignés» à la page 65.

---

### Utilisation des commandes de contrôle

Les *commandes de contrôle* vous permettent d'effectuer des tâches administratives sur les gestionnaires de files d'attente.

## Commandes de contrôle

Pour plus de détails sur les commandes de contrôle, reportez-vous au «Chapitre 3. Gestion des gestionnaires de files d'attente à l'aide de commandes de contrôle» à la page 27.

---

## Administration éloignée à l'aide de commandes MQSC

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

Vous pouvez émettre des commandes MQSC en utilisant la commande **runmqsc**. Vous pouvez travailler :

- de façon interactive en entrant les commandes au clavier. Consultez la section «Utilisation de la fonction MQSC de façon interactive» à la page 39.
- en exécutant une séquence de commandes à partir d'un fichier texte ASCII. Consultez la section «Exécution de commandes MQSC à partir de fichiers texte» à la page 42.

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.

Pour plus de détails sur l'utilisation des commandes MQSC, consultez la section «Exécution de commandes MQSC à partir de fichiers texte» à la page 42. Vous trouverez davantage d'informations sur la commande **runmqsc** dans la section «runmqsc (exécution des commandes MQSeries)» à la page 296.

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 commandes MQSC sont disponibles sur d'autres plateformes, notamment AS/400 et OS/390.

Les commandes MQSC sont répertoriées dans l'«Annexe D. Comparaison des jeux de commandes» à la page 327.

Le manuel *MQSeries - Guide de référence des commandes* contient une description de chaque commande MQSC et de sa syntaxe.

Pour plus de détails sur l'utilisation des commandes MQSC pour l'administration locale, reportez-vous à la section «Administration locale à l'aide de commandes MQSC» à la page 38.

---

## Exécution de tâches administratives à l'aide de commandes PCF

L'objet des commandes programmables PCF est de permettre l'intégration de tâches d'administration dans un programme d'administration. Ainsi, vous pouvez créer des files d'attente et des définitions de processus, ou 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.

Pour plus d'informations, reportez-vous à la section «Commandes programmables PCF» à la page 61.

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

Vous pouvez utiliser l'interface MQAI (interface d'administration MQSeries) pour accéder plus facilement aux messages PCF. Pour plus de détails, reportez-vous à la section «Utilisation de l'interface MQAI pour simplifier l'utilisation des commandes programmables PCF» à la page 62.

### 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 attributs étant limités à huit caractères, il peut parfois s'avérer difficile de leur donner un sens (par exemple, QDPHIEV). Les attributs d'objets dans PCF apparaissent en italique, ne sont pas limités à huit caractères et sont donc plus simples à interpréter. 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*.

---

## Signification des noms de fichiers MQSeries

Chaque objet file d'attente, gestionnaire de files d'attente, liste de noms et 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.

Le chemin vers un répertoire de gestionnaire de files est composé des éléments suivants :

- Un préfixe - la première partie du nom :

MQS\_ROOT:[MQM]

il est défini dans le fichier de configuration du gestionnaire de files d'attente.

- Un littéral :

QMGRS

## Signification des noms de fichiers MQSeries

- Un nom de gestionnaire de files d'attente codé, qui est le nom du gestionnaire transformé en un nom de répertoire valide. Par exemple, le gestionnaire de files d'attente :

GEST.FILES

serait représenté par :

GEST\$FILES

Ce processus est appelé *transformation de nom*.

## Transformation de nom du gestionnaire de file d'attente

Dans MQSeries, un nom de gestionnaire de files d'attente peut comporter jusqu'à 48 caractères.

Par exemple, vous pouvez appeler un gestionnaire de files :

GESTION.FILES.SERVICES.COMPTA

Néanmoins, chaque gestionnaire de files est représenté par un fichier et il existe des restrictions quant à la longueur maximale d'un nom de fichier et aux caractères utilisables dans ce nom. En conséquence, les noms des fichiers représentant des objets sont automatiquement transformés pour correspondre aux besoins du système de fichiers.

Les règles gouvernant la transformation des noms de gestionnaires de files d'attente, en prenant l'exemple d'un gestionnaire de files appelé GEST.FILES, sont les suivantes :

1. Transformation des caractères individuels :
  - devient \$
  - / devient \_
  - % devient \_
2. Si le nom est toujours incorrect :
  - a. Coupure à huit caractères
  - b. Ajout d'un suffixe numérique de trois caractères

Par exemple, en prenant le préfixe par défaut, le nom du gestionnaire de files d'attente devient :

MQS\_ROOT:[MQM.QMGRS.GEST\$FILES]

L'algorithme de transformation permet également la distinction entre des noms qui ne diffèrent que par les majuscules ou minuscules sur des systèmes de fichiers qui ne tiennent pas compte de cette distinction.

## 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 une machine, il peut y avoir un grand nombre d'objets pour chaque gestionnaire de files. Seules les définitions de processus, les files d'attente et les listes de noms sont représentées dans le système de fichiers ; les canaux ne sont pas concernés par ces considérations.

## Signification des noms de fichiers MQSeries

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

Les noms de files d'attentes commencent par la lettre "Q".

Pour plus d'informations sur la dénomination des objets, consultez la section «Noms d'objets MQSeries» à la page 243.

---

## Utilisation des caractères en majuscules et en minuscules

### Utilisation des majuscules et minuscules dans les commandes de contrôle

OpenVMS est habituellement considéré comme un système d'exploitation ne faisant pas la distinction entre majuscules et minuscules. En d'autres termes, cela signifie qu'en règle générale, les trois commandes suivantes créent toutes un gestionnaire de files d'attente appelé "GESTFILES".

```
$ crtmqm GestFiles
$ crtmqm gestfiles
$ crtmqm GESTFILES
```

Avec MQSeries pour Compaq OpenVMS, vous pouvez aussi mettre des guillemets autour du nom du gestionnaire de files d'attente (ou d'un autre paramètre) pour protéger sa casse. Si vous utilisez les guillemets, les trois commandes suivantes créent trois gestionnaires de files d'attente différents.

```
$ crtmqm "GestFiles"   crée un gestionnaire de files d'attente appelé GestFiles
$ crtmqm "gestfiles"  crée un gestionnaire de files d'attente appelé gestfiles
$ crtmqm "GESTFILES" crée un gestionnaire de files d'attente appelé GESTFILES
```

OpenVMS Version 7.2 introduit la nouvelle commande suivante :

```
$ set process /parse_style = ( traditional | extended )
```

Cette commande modifie la façon dont OpenVMS gère les caractères en majuscules et en minuscules.

Si la commande **set process /parse\_style** n'est pas utilisée, ou si elle est utilisée avec l'option **traditional**, OpenVMS se comporte comme à l'ordinaire.

Si cette commande est utilisée avec l'option **extended**, la routine de la librairie d'exécution LIB\$GET\_FOREIGN permet de conserver la casse du texte récupéré. MQSeries utilisant cette routine pour obtenir les paramètres de ligne de commande, la casse des paramètres est conservée même lorsqu'ils ne sont pas mis entre guillemets.

Par exemple, la séquence de commandes suivante crée trois gestionnaires de files d'attente différents. Notez que les paramètres ne sont pas entre guillemets.

## Signification des noms de fichiers MQSeries

```
$ set process /parse_style = extended
$ crtmqm GestFiles   crée un gestionnaire de files d'attente appelé GestFiles
$ crtmqm gestfiles  crée un gestionnaire de files d'attente appelé gestfiles
$ crtmqm GESTFILES  crée un gestionnaire de files d'attente appelé GESTFILES
```

Outre la distinction entre majuscules et minuscules, la commande OpenVMS `set process /parse_style` apporte diverses modifications. Consultez les informations fournies dans le dictionnaire DCL OpenVMS avant d'appliquer la commande à votre système.

## Utilisation des majuscules et minuscules dans les commandes MQSC

Les commandes de contrôle MQSeries (par exemple, **runmqsc** qui appelle la fonction MQSC) ne font pas la distinction entre majuscules et minuscules.

Les commandes MQSC, ainsi que leurs attributs peuvent s'écrire indifféremment en minuscules ou en majuscules. Les noms d'objets figurant dans les commandes MQSC sont automatiquement convertis en majuscules sauf s'ils sont placés entre *apostrophes*. Si vous n'utilisez pas d'apostrophes, l'objet est traité avec un nom en majuscules. Consultez le manuel *MQSeries - Guide de référence des commandes* pour plus d'informations.

---

## Chapitre 3. Gestion des gestionnaires de files d'attente à l'aide de commandes de contrôle

Le présent chapitre décrit comment effectuer des opérations sur les gestionnaires de files d'attente et les serveurs de commandes. Il contient les sections suivantes :

- «Commandes de contrôle»
- «Instructions pour la création de gestionnaires de files d'attente» à la page 28
- «Création d'un gestionnaire de files d'attente par défaut» à la page 32
- «Lancement d'un gestionnaire de files d'attente» à la page 32
- «Définition d'un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant» à la page 33
- «Arrêt d'un gestionnaire de files d'attente» à la page 33
- «Redémarrage d'un gestionnaire de files d'attente» à la page 35
- «Suppression d'un gestionnaire de files d'attente» à la page 35
- «Consultation des fichiers objets» à la page 244

---

### Commandes de contrôle

Les commandes de contrôle permettent d'effectuer des opérations sur les gestionnaires de files d'attente, les serveurs de commandes et les canaux. Elles sont classées en trois catégories, comme indiqué dans le tableau 1 à la page 27.

Tableau 1. Catégories des commandes de contrôle

Catégorie	Description
Gestionnaires de files d'attente	Commandes de création, de démarrage, d'arrêt et de suppression des gestionnaires de files d'attente et des serveurs de commandes.
Canaux	Commandes de démarrage et d'arrêt des canaux et initialisateurs de canal.
Utilitaires	Commandes liées aux opérations suivantes : <ul style="list-style-type: none"><li>• Exécution de commandes MQSC</li><li>• Exits de conversion</li><li>• Gestion des droits d'accès</li><li>• Enregistrement et reconstitution d'images de support des ressources de gestionnaire de files d'attente</li><li>• Affichage et résolution des transactions</li><li>• Moniteurs de déclenchement</li><li>• Affichage des noms de fichiers des objets MQSeries</li></ul>

Pour plus de détails sur les tâches d'administration relatives aux canaux, reportez-vous au manuel *MQSeries Intercommunication*.

### Utilisation des commandes de contrôle

Sous MQSeries pour Compaq OpenVMS, les commandes de contrôle sont entrées à une invite DCL. Les noms de commande et les indicateurs ne font pas de différence entre les majuscules et les minuscules. Par contre, les arguments peuvent devoir être convertis en majuscules selon l'état d'une option de processus OpenVMS, et si les arguments ont ou non été inclus entre guillemets doubles. Vous

## Commandes de contrôle

trouvez plus de détails sur l'influence de l'option OpenVMS et des guillemets sur la casse, à la section «Utilisation des caractères en majuscules et en minuscules» à la page 25.

Par exemple, dans la commande :

```
crtmqm -u system.dead.letter.queue "jupiter.gest.files"
```

- La file d'attente de rebut est SYSTEM.DEAD.LETTER.QUEUE., bien qu'elle ait été tapée en minuscules. La conversion automatique de la casse dépend de l'état de la commande OpenVMS **set process/parse\_style**. Consultez la section «Utilisation des caractères en majuscules et en minuscules» à la page 25.
- Le nom du gestionnaire de files d'attente est bien "jupiter.gest.files", car il a été tapé entre guillemets doubles ("JUPITER.gest.files" serait un nom différent).

Par conséquent, veillez à entrer des commandes rigoureusement identiques à celles montrées dans les exemples.

---

## Création d'un gestionnaire de files d'attente

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

Pour pouvoir utiliser des messages et des files d'attente, vous devez au préalable créer au moins un gestionnaire de files d'attente ainsi que les objets qui lui sont associés. Pour créer un gestionnaire de files d'attente, utilisez la commande de contrôle MQSeries **crtmqm**. La commande **crtmqm** crée automatiquement les objets par défaut et les objets système requis. 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. Une fois le gestionnaire de files d'attente et ses objets associés créés, utilisez la commande **strmqm** pour lancer le gestionnaire.

## Instructions pour la création de gestionnaires de files d'attente

Avant de créer un gestionnaire de files d'attente, en particulier dans un environnement de production, vous devez respecter les points suivants, dans l'ordre indiqué :

- Définition d'un nom de gestionnaire de files d'attente unique
- Limitation du nombre de gestionnaires de files d'attente.
- Définition d'un gestionnaire de files d'attente par défaut.
- Définition de la file d'attente de rebut.
- Définition de la file d'attente de transmission.
- Définition des paramètres de journalisation nécessaires.
- Sauvegarde des fichiers de configuration après la création d'un gestionnaire de files d'attente

Ces tâches sont détaillées dans les sections suivantes.



### 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 créer de canaux pour la mise en file d'attente répartie.

Une des méthodes pour garantir l'unicité des noms consiste à utiliser le nom de noeud du gestionnaire, qui est unique, comme préfixe du nom du gestionnaire. 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, mais faites attention, `comptes.saturne` et `comptes.saturne.gest.files` sont des noms de gestionnaires *différents*.

Si vous utilisez MQSeries pour communiquer avec d'autres entreprises, vous pouvez également ajouter le nom de votre entreprise en préfixe. Pour des raisons de commodité, nous ne le faisons pas dans nos exemples.

**Remarque :** Les noms de commande et les indicateurs ne différencient pas les majuscules et les minuscules. Par contre, les arguments peuvent être convertis en majuscules selon l'état d'une option de processus OpenVMS, et si les arguments ont ou non été inclus entre guillemets doubles. Vous avez donc la possibilité de créer deux gestionnaires différents nommés `jupiter.gest.files` et `JUPITER.gest.files`. Vous trouverez plus de détails sur l'influence de l'option OpenVMS et des guillemets doubles sur la casse, à la section «Utilisation des caractères en majuscules et en minuscules» à la page 25.

### Limitation du nombre de gestionnaires de files d'attente

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 en général préférable de créer un gestionnaire avec 100 files d'attente sur un noeud, plutôt que dix gestionnaires avec dix files d'attente chacun.

Dans les systèmes de production, plusieurs noeuds sont exécutés avec un seul gestionnaire de files d'attente, mais des serveurs plus importants peuvent fonctionner avec des gestionnaires multiples.

### Définition du 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.

Pour créer un gestionnaire de files d'attente par défaut, utilisez 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 250.

### Qu'est-ce qu'un gestionnaire de files d'attente 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.

## Création des gestionnaires de files d'attente

### Comment définir le 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**. Si vous ne voulez pas créer de gestionnaire de files d'attente par défaut, n'ajoutez pas cet indicateur.

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

### Que se passe-t-il si je change le gestionnaire de files d'attente par défaut ?

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.

C'est peut-être l'effet escompté, mais vous devez tenir compte de ces éléments avant de changer l'option par défaut.

### Définition de la 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. Dans le cas contraire, des erreurs dans des programmes d'application entraîneront la fermeture des canaux ou les réponses aux commandes d'administration ne seront pas reçues.

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 tout simplement 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 42 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 8. Gestionnaire de files d'attente de rebut (DLQ) MQSeries» à la page 103, pour plus d'informations sur le gestionnaire 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 de la file d'attente de transmission

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. Cette opération ne crée pas effectivement la file d'attente. Vous devrez le faire plus tard. Pour plus d'informations, consultez la section «Manipulation des files d'attente locales» à la page 47.

### Définition des paramètres de journalisation nécessaires

Vous pouvez indiquer des paramètres de journalisation dans la commande `crtmqm`, y compris le type de journalisation, le chemin d'accès et la taille des fichiers journaux. Dans un environnement de développement, les paramètres de journalisation par défaut devraient convenir. Cependant, vous pouvez modifier les options par défaut, par exemple, dans les cas suivants :

- Vous avez une configuration système d'entrée de gamme qui ne peut pas prendre en charge des journaux importants.
- Vous prévoyez de recevoir simultanément dans vos files d'attente un nombre important de longs messages.

Pour plus d'informations sur la définition des paramètres de journalisation, reportez-vous à :

- La section «`crtmqm` (création d'un gestionnaire de files d'attente)» à la page 250, pour l'utilisation de la commande `crtmqm`.
- La section «Strophe Log» à la page 182, pour l'utilisation des fichiers de configuration.

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

Il existe deux types de fichiers de configuration :

1. Lorsque vous installez le produit, le fichier de configuration MQSeries (`mqs.ini`) est créé. 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 seul fichier `mqs.ini` par noeud.
2. Lorsque vous créez un nouveau gestionnaire de files d'attente, un nouveau fichier de configuration du gestionnaire de files d'attente (`qm.ini`) est créé automatiquement. 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, plus tard, 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. La règle générale consiste à faire 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. Configuration de MQSeries» à la page 173.

### 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** ci-dessous 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.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE "saturne.gest.files"
```

Où :

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

**-d MY.DEFAULT.XMIT.QUEUE**  
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.

**"saturn.queue.manager"**  
est le nom de ce gestionnaire de files d'attente. Ce nom doit être le dernier paramètre de la commande **crtmqm**.

La création d'un gestionnaire de files d'attente par défaut vous permet de lui envoyer certaines commandes (comme **strmqm** ou **runmqsc**), sans avoir à préciser un nom de gestionnaire de files d'attente. D'autres commandes (comme **endmqm** et **dltmqm**) exigent un nom de gestionnaire de files d'attente.

Vous remarquerez que dans cet exemple, le nom du gestionnaire de files d'attente est en minuscules, et qu'il est protégé par les guillemets doubles. Vous trouverez davantage d'informations concernant la gestion de la casse pour les paramètres dans la section «Utilisation des caractères en majuscules et en minuscules» à la page 25.

---

### 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 lancé 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.

## 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 (mqs.ini). 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 :

- **Pour définir un gestionnaire de files d'attente par défaut à partir d'un gestionnaire existant.** Pour cela, vous devez modifier dans la strophe le nom du gestionnaire de files d'attente dans l'attribut Name en indiquant 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.

Pour plus de détails sur les fichiers de configuration, reportez-vous au «Chapitre 13. Configuration de MQSeries» à la page 173.

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

## 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`, tapez :

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

## Arrêt progressif

Par défaut, la commande **endmqm** 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.

Utilisez ce type d'arrêt pour notifier aux applications qu'elles doivent s'arrêter. Lorsque vous tapez la commande :

```
endmqm -c "saturne.gest.files"
```

le système ne vous informe pas de l'arrêt des applications. (La commande `-c "saturne.gest.files"` est équivalente à la commande `endmqm "saturne.gest.files"`).

## Création des gestionnaires de files d'attente

### Arrêt immédiat

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

Utilisez ce type d'arrêt comme procédure d'arrêt normale du gestionnaire de files d'attente, éventuellement après une période de mise au repos. Pour exécuter un arrêt immédiat, tapez :

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

### Arrêt forcé

**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.

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"
```

Cette commande arrête immédiatement le code de gestionnaire de files d'attente.

**Remarque** : Après un arrêt forcé, ou si le gestionnaire de files d'attente se plante, il se peut que la mémoire partagée qu'il utilisait ne soit pas nettoyée. Ceci peut provoquer des problèmes au redémarrage. Pour des informations sur l'utilisation de MONMQ pour nettoyer la mémoire après un arrêt brutal de ce type, consultez la section «Gestion de la mémoire partagée avec MONMQ» à la page 364.

### 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,
- s'arrêtent sans se déconnecter du gestionnaire (en émettant un appel MQDISC).

Si l'arrêt d'un gestionnaire de files d'attente est très lent, et si vous pensez que le gestionnaire de files d'attente ne va pas s'arrêter, vous pouvez annuler la commande **endmqm** par Ctrl-Y. Vous pouvez alors émettre une autre commande **endmqm**, en spécifiant cette fois-ci un indicateur correspondant au au type d'arrêt souhaité (immédiat ou forcé).

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 274.

## Redémarrage d'un gestionnaire de files d'attente

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

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

---

## 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** : La suppression d'un gestionnaire de files d'attente est une opération dangereuse car vous supprimez également toutes les ressources qui lui sont associées. Les ressources comprennent toutes les files d'attente avec leurs messages, ainsi que les définitions d'objets.

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 254. Vous devez vous assurer que seuls les administrateurs autorisés ont les droits d'accès pour utiliser cette commande.





---

## Chapitre 4. 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). Dans ce contexte, l'administration locale comprend les opérations de création, d'affichage, de modification, de copie et de suppression des 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 38
- «Exécution de commandes MQSC à partir de fichiers texte» à la page 42
- «Résolution des incidents à l'aide des commandes MQSC» à la page 46
- «Manipulation des files d'attente locales» à la page 47
- «Manipulation des files d'attente alias» à la page 55
- «Manipulation des files d'attente modèles» à la page 56
- «Gestion des objets pour le déclenchement» à la page 58

---

### Prise en charge des applications utilisant l'interface MQI

Les programmes d'application MQSeries nécessitent certains objets pour pouvoir s'exécuter. Par exemple, la figure 1 montre une application qui extrait des messages d'une file d'attente, les traite, puis envoie les résultats à une autre file d'attente située sur le même gestionnaire.

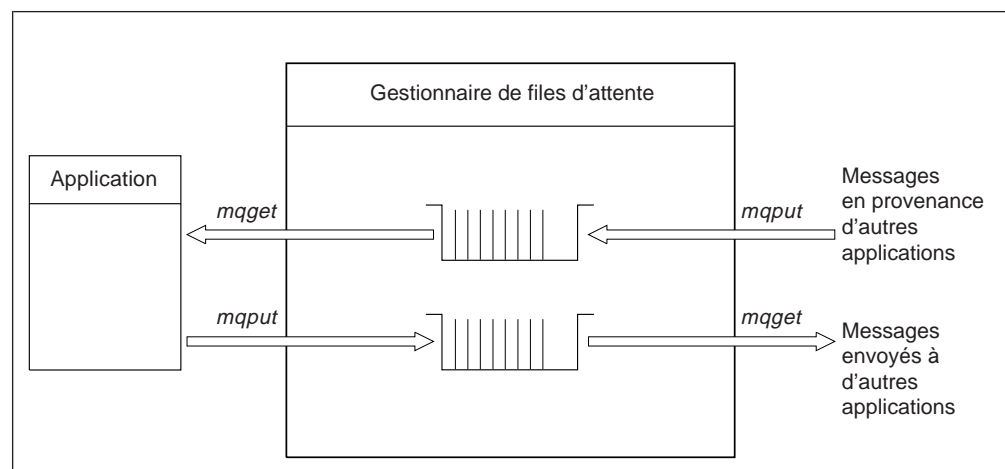


Figure 1. Files d'attente, messages et applications

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.
- La première file d'attente d'application, dans laquelle les messages doivent être récupérés, doit être définie.

## Programmes d'application

- La seconde file d'attente, dans laquelle l'application insère les messages, doit également être définie.
- L'application doit pouvoir se connecter au gestionnaire de files d'attente. Pour cela, elle doit être en liaison avec le code du produit. Consultez le *Guide de programmation d'applications* pour plus d'informations.
- 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. Cette partie du système n'est pas montrée par la figure 1 à la page 37.

---

## Administration locale à l'aide de commandes MQSC

Dans cette section, nous supposons que vous effectuez vos opérations d'administration en utilisant la commande **runmqsc**. Vous pouvez travailler de façon interactive en entrant les commandes au clavier, ou bien rediriger l'unité d'entrée standard SYS\$INPUT pour exécuter une séquence de commandes à partir d'un fichier texte ASCII. Dans les deux cas, le format des commandes est le même.

Le manuel *MQSeries - Guide de référence des commandes* contient une description de chaque commande MQSC et de sa syntaxe.

Les commandes de script MQSeries (MQSC) permettent de gérer des objets gestionnaire de files d'attente, y compris le gestionnaire lui-même, les clusters, les canaux, les files d'attente, les listes de noms et les 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 (pour plus d'informations sur cette commande, consultez la section «Utilisation des majuscules et minuscules dans les commandes MQSC» à la page 26). 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 32.

### 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.

### Redirection des données en entrée et en sortie

Pour simplifier la migration depuis les autres systèmes d'exploitation vers OpenVMS, MQSeries prend en charge les types d'indicateurs de redirection UNIX\* pour sys\$input, sys\$output, et sys\$error, de la façon suivante :

```
< indique la source pour SYS$INPUT
>indique la source pour SYS$OUTPUT
2> indique la source pour SYS$error
```

Cette fonction est aussi disponible sur les versions exécutables des programmes exemples. Par contre, elle ne figure pas dans la source des versions d'exemple et ne sera donc pas disponible si vous reconstruisez les exemples à partir du code source.

## Utilisation de la fonction MQSC de façon interactive

Pour exécuter des commandes de façon interactive, entrez le texte suivant à une invite DCL :

```
runmqsc
```

## Emission de commandes MQSC

Aucun nom de gestionnaire de files d'attente n'a été indiqué dans cette commande. Par conséquent, les commandes MQSC seront traitées par le gestionnaire par défaut. Maintenant, vous pouvez entrer n'importe quelle commande MQSC, en fonction des opérations à effectuer. Par exemple :

```
DEFINE QLOCAL (FILE.LOCALE.ORANGE)
```

Pour indiquer qu'une commande doit se poursuivre sur la ligne suivante, utilisez un caractère de continuation.

- 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. Mais vous pouvez aussi terminer la saisie de façon explicite en entrant un point-virgule (;). (Ce système est particulièrement utile si vous entrez par erreur un caractère de suite à la fin de la dernière ligne de la commande).

## 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: File d'attente MQSeries créée.  
.  
.  
AMQ8405: Erreur de syntaxe détectée vers la fin du segment de  
commande situé sous : -  
z  
  
AMQ8426: Commandes MQSC valides :  
  
ALTER  
CLEAR  
DEFINE  
DELETE  
DISPLAY  
END  
PING  
REFRESH  
RESET  
RESOLVE  
RESUME  
START  
STOP  
SUSPEND
```

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. Ces messages sont envoyés à l'unité de sortie standard. Si vous n'avez pas entré la commande correctement, reportez-vous au manuel *MQSeries - Guide de référence des commandes* pour connaître la syntaxe correcte.

## Fin de l'entrée interactive sur MQSC

Pour mettre fin à la saisie interactive de commandes MQSC, entrez la commande MQSC END :

```
END
```

Vous pouvez également utiliser le caractère de fin de fichier (EOF) <CTRL Z>.

Si vos données en entrée proviennent d'une autre source que l'unité standard, telle qu'un fichier texte, cette opération est inutile.

## 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 MQSC suivante :

```
DISPLAY QMGR ALL
```

Vous obtenez une sortie identique à celle présentée à la figure 2.

```
1 : display qmgr all
AMQ8408: Détails sur l'affichage du gestionnaire de files d'attente.
DESCR( )                DEADQ( )
DEFXMITQ( )             CHADEXIT( )
CLWLEXIT( )             CLWLDATA( )
REPOS( )                REPOSNL( )
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)  QMNAME(saturne.gest.files)
CRDATE(2001-01-16)      CRTIME(11.13.56)
ALTDATA(2001-01-16)     ALTTIME(11.13.56)
QMID(saturne.gest.files_2001-01-16_11.13.56)
TRIGINT(999999999)      MAXHANDS(256)
MAXUMSGS(10000)         AUTHOREV(DISABLED)
INHIBTEV(DISABLED)     LOCALEV(DISABLED)
REMOTEEV(DISABLED)     PERFMV(DISABLED)
STRSTPEV(ENABLED)      CHAD(DISABLED)
CHADEV(DISABLED)       CLWLEN(100)
MAXMSGL(4194304)       CCSID(819)
MAXPRTY(9)             CMDLEVEL(510)
PLATFORM(OpenVMS)      SYNCPT
DISTL(YES)
```

Figure 2. Sortie type générée par la commande `DISPLAY QMGR`

Le paramètre `ALL` de la commande `DISPLAY QMGR` permet d'afficher tous les attributs du gestionnaire de files d'attente. Aucun nom de gestionnaire de files d'attente n'ayant été indiqué lors de l'exécution de la commande, cette sortie nous donne notamment le nom du gestionnaire par défaut (`saturne.gest.files`), et les noms de file de rebut (`SYSTEM.DEAD.LETTER.QUEUE`) et de la file de commandes (`SYSTEM.ADMIN.COMMAND.QUEUE`).

## Emission de commandes MQSC

Vous pouvez vérifier que ces files d'attente ont été créées en entrant la commande :

```
DISPLAY QUEUE (SYSTEM.*)
```

La liste des files dont le nom contient le préfixe 'SYSTEM.\*'s'affiche. Notez que les parenthèses sont obligatoires.

## Utilisation d'un gestionnaire différent du gestionnaire par défaut

Pour exécuter les commandes MQSC sur un gestionnaire de files d'attente 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 MQSC que vous entrerez à partir de maintenant 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 71.

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

Pour modifier les attributs du gestionnaire de files d'attente indiqué 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 d'inhibition.

---

## Exécution de commandes MQSC à partir de fichiers texte

L'exécution de commandes MQSC 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 pour lire les données dans un fichier texte (consultez la section «Redirection des données en entrée et en sortie» à la page 39 pour plus d'informations sur les indicateurs de redirection.) Vous devez commencer par créer un fichier texte contenant les commandes MQSC, à l'aide de votre éditeur de texte habituel. Par exemple,

## Exécution de commandes MQSC

L'instruction suivante exécute une séquence de commandes qui se trouvent dans le fichier `monprog.in` :

```
runmqsc <monprog.in
```

Vous pouvez également rediriger les données en sortie vers un fichier. Un fichier qui contient les commandes MQSC 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 `SYS$INPUT` et `SYS$OUTPUT` dans la commande **runmqsc**, employez la syntaxe suivante :

```
runmqsc < monprog.in > monprog.out
```

Cette instruction appelle les commandes MQSC contenues dans le fichier de commandes MQSC `monprog.in`. 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 `monprog.out`. La figure 3 à la page 44 montre un extrait du fichier de commandes MQSC `monprog.in` et la figure 4 à la page 45 montre la sortie correspondante extraite du fichier `monprog.out`.

Pour rediriger `SYS$INPUT` et `SYS$OUTPUT` dans la commande **runmqsc** pour un gestionnaire de files d'attente (`saturne.gest.files`) qui n'est pas celui par défaut, employez la syntaxe ci-dessous :

```
runmqsc "saturne.gest.files" < monprog.in > monprog.out
```

## Fichiers de commandes MQSC

Les commandes MQSC sont écrites sous une forme directement lisible par l'utilisateur, c'est-à-dire en texte au format ASCII. La figure 3 à la page 44 présente un extrait d'un fichier de commandes MQSC montrant la commande `DEFINE QLOCAL` avec ses attributs. Le manuel *MQSeries - Guide de référence des commandes* contient une description de chaque commande MQSC et de sa syntaxe.

## Exécution de commandes MQSC

```
.  
. .  
. .  
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 3. Extrait du fichier de commandes MQSC *monprog.in*

Pour des raisons de portabilité entre les différents environnements MQSeries, il est recommandé de limiter à 72 caractères la longueur des lignes dans les fichiers de commandes MQSC. Le signe plus indique que la commande se poursuit sur la ligne suivante.

Pour MQSeries pour Compaq OpenVMS, vous devez limiter à 80 caractères la longueur des lignes, caractère de continuation compris. Le signe plus indique que la commande se poursuit sur la ligne suivante.

## Rapports MQSC

La commande **runmqsc** génère un *rapport*, qui est envoyé à l'unité SYS\$OUTPUT. Ce rapport contient les éléments suivants :

- Un en-tête indiquant que MQSC est à l'origine du rapport :  
Commandes MQSeries en cours de lancement.
- Une liste numérotée des commandes MQSC émises (en option). Par défaut, le texte en entrée est répercuté dans celui en sortie. Dans le rapport, chaque commande est précédée d'un numéro de séquence, comme le montre la figure 4 à la page 45. 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: File d'attente MQSeries créée.
- 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.



```

Commandes MQSeries en cours de lancement.
.
.
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: File d'attente MQSeries créée.
:
.
.
15 commandes MQSC lues
0 commande avec erreur de syntaxe.
0 commande non traitée.

```

Figure 4. Extrait du fichier de rapport MQSC *monprog.out*

## Exécution des fichiers de commandes MQSC fournis

À l'installation de MQSeries pour Compaq OpenVMS, fichiers de commandes MQSC suivants sont fournis :

### **amqscos0.tst**

Définitions des objets utilisés par les programmes exemples.

Ces fichiers se trouvent dans le répertoire `MQS_EXAMPLES`:

## Utilisation de `runmqsc` pour vérifier des commandes

Vous pouvez utiliser la commande **runmqsc** pour vérifier des commandes MQSC sur un gestionnaire de files d'attente local sans avoir à les exécuter réellement. Pour cela, ajoutez l'indicateur `-v` dans la commande **runmqsc**, comme ci-dessous :

```
runmqsc -v < monprog.in > monprog.out
```

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 opération est d'une grande importance si vous exécutez un grand nombre de commandes MQSC à partir d'un fichier de commandes.

Le rapport envoyé est du même type que celui montré dans la figure 4.

Vous ne pouvez pas employer cette méthode pour vérifier des commandes MQSC à distance. Par exemple, si vous lancez cette commande :

```
runmqsc -w 30 -v "jupiter.gest.files" < monprog.in > monprog.out
```

## Exécution de commandes MQSC

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 pour 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 ordinaires répertoriés. Il n'est pas toujours évident de trouver de quel type d'incident il s'agit lorsque vous lisez l'erreur générée.

Lorsque vous utilisez la commande `runmqsc`, n'oubliez pas la règle suivante :

- Utilisez l'opérateur `<` 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: Gestionnaire de files de messages MQSeries inexistant.

- Si vous redirigez les données de sortie vers un fichier, utilisez l'opérateur `>`. Par défaut, le fichier est créé dans le répertoire à partir duquel la commande `runmqsc` a été appelée. Indiquez un nom de fichier complet pour envoyer vos données en sortie vers un fichier et un répertoire spécifiques.
- 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 répertoire `MQS_ROOT:[MQM]`. Il contient le nom des gestionnaires de files d'attente et celui du gestionnaire par défaut, si vous en avez un.

- Le gestionnaire doit être actif. S'il ne l'est pas, lancez-le. Reportez vous à la section «Lancement d'un gestionnaire de files d'attente» à la page 32. 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: Gestionnaire de files d'attente MQSeries non disponible.

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 33.

- 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 DCL avant d'avoir lancé **runmqsc**. Par exemple :

```
DEFINE QLOCAL(Queue1)

%DCL-W-PARMDL, invalid parameter delimiter - check use of special
characters
```

- 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
0790997, 5724-A38 (C) Copyright IBM Corp. 1996, 2001 TOUS DROITS RESERVES
Commandes MQSeries en cours de lancement.

strmqm saturne.gest.files
  1 : strmqm saturne.gest.files
AMQ8405: Erreur de syntaxe détectée vers la fin du segment de
commande situé sous : -
s

AMQ8426: Commandes MQSC valides :

ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND

*CANCEL*

Une commande MQSC lue.
Une commande avec erreur de syntaxe.
Toutes les commandes MQSC valides ont été traitées.
$
```

Consultez aussi la section «En cas d'incident» à la page 73.

---

## 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* pour une description complète de ces commandes.

## Manipulation des files d'attente locales

### 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 d'attente 'ordinaire', c'est-à-dire qu'elle ne sert ni à l'initialisation ni à la transmission, et qu'elle ne peut pas générer des messages de déclenchement.
- La capacité maximale de la file est 1000 messages et la taille maximale de chaque message est 2000 octets.

Cette définition correspond à la commande suivante

```
DEFINE QLOCAL (FILE.LOCALE.ORANGE) +
  DESCR('File pour les messages provenant d'autres systèmes') +
  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. Reportez-vous également à la section «Affichage des attributs d'objet par défaut» à la page 49.
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 50.

### 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 au rebut (c'est-à-dire, qui n'ont pas pu être transmis vers leur cible) en attendant d'en être extraits ultérieurement. Vous devez spécifier explicitement une file d'attente de rebut au gestionnaire. Pour ce faire, indiquez son nom dans la

## Manipulation des files d'attente locales

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 automatiquement lorsque vous exécutez l'exemple. 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 8. Gestionnaire de files d'attente de rebut (DLQ) `MQSeries`» à la page 103.

## 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) ALL
```

**Remarque :** La syntaxe de cette commande diffère de celle de la commande `DEFINE` correspondante.

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: Affichage des détails de la file.  
QUEUE(FILE.LOCALE.ORANGE)  
MAXDEPTH (1000)  
MAXMSGL(2000)  
CURDEPTH(0)
```

## Manipulation des files d'attente locales

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 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 48, 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.

- Avec 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.

## Manipulation des files d'attente locales

- 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.

## 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 OpenVMS 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

```
MQS_EXAMPLES:AMQSBG0.C
```

### Exécutable

```
[.BIN]AMQSBG.EXE, sous
```

```
MQS_EXAMPLES:
```









## 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 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 qui indiquent 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.

## Manipulation des files d'attente alias

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 :

```
* Affichage des attributs de la file d'attente alias
* ALL = Affichage de tous les attributs

DISPLAY QUEUE (ALPHAS.ALIAS.QUEUE) ALL

* Modification (ALTER) du nom de la file de base, sur lequel l'alias pointe.
* FORCE = forcer le changement, même si la file est ouverte.

ALTER QALIAS (FILE.ALIAS.ALPHA) TARGQ(FILE.LOCALE.ORANGE) FORCE

* Suppression de cet alias, si possible.

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*.

---

## 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.

## Manipulation des files d'attente modèles

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

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('File pour les messages provenant de l'application X') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL) +
  DEFTYPE (PERDYN)
```

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 :

```
* Affichage des attributs de la file d'attente modèle
* ALL = Affichage de tous les attributs

DISPLAY QUEUE (FILE.MODELE.VERTE) ALL

* Modification (ALTER) du modèle pour autoriser les insertions
* dans les files dynamiques qu'il sert à créer.

ALTER QMODEL (FILE.MODELE.BLEUE) PUT(ENABLED)

* Suppression de cette file modèle

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 *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 OpenVMS

#### Définition d'une file d'attente d'application pour le déclenchement

Une file d'attente d'application est une file locale utilisée par des applications, via l'interface MQI, pour la messagerie. 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 ou supérieure :

```
DEFINE QLOCAL (FILE.ASS.AUTO) +
PROCESS (MOTOR.INS.PROC) +
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 que les messages doivent avoir pour être pris en compte par le gestionnaire de files d'attente en vue de générer l'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, donné dans cet exemple :

```
DEFINE PROCESS (EVALUATION.ASS.AUTO) +
  DESCR ('Traitement des messages de demande d'assurance') +
  APPLTYPE (OPENVMS) +
  APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE') +
  USERDATA ('open, close, 235')
```

Où :

### EVALUATION.ASS.AUTO

indique le nom de la définition de processus.

### DESCR ('Traitement des messages de demande d'assurance')

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 (OPENVMS)

indique le type de l'application à démarrer sous OpenVMS.

### APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE')

est le nom du fichier exécutable de l'application.

## Gestion des objets pour le déclenchement

### 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 (EVALUATION.ASS.AUTO) ALL
AMQ8407: Détails sur le processus de commande.
DESCR ('Traitement des messages de demande d'assurance') +
APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE') +
USERDATA (open, close, 235) +
PROCESS (MOTOR.INS.PROC) +
APPLTYPE (OPENVMS)
```

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 5. Automatisation des tâches

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 «Commandes programmables PCF»
- l'utilisation du serveur de commandes, dans la section «Gestion du serveur de commandes pour l'administration éloignée» à la page 63

---

### 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 nomset 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 italiques et ne sont pas limités à huit caractères. Par exemple, l'équivalent PCF de RQMNAME est *RemoteQMgrName*.

## 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 OpenVMS.

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.

- **mettre en oeuvre des applications d'administration automatique et des outils d'administration**

Par exemple, l'interface ADSI fournie par MQSeries pour Windows NT et Windows 2000 Version 5.2 utilise MQAI. (Il n'existe actuellement pas d'exemple de cette utilisation sur la plateforme OpenVMS.)

- **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 le 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 émises. 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.

## 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

## Administration éloignée du serveur de commandes

fonctionnement, les commandes éloignées ne pourront être traitées. Tous les messages contenant des commandes seront insérés dans la file d'attente du gestionnaire cible.

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 sur la machine cible. Si le serveur de commandes est en fonctionnement, le message suivant est envoyé :

```
AMQ8027    Serveur de commandes MQSeries lancé.
```

## 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 6. Administration des objets MQSeries éloignés

Ce chapitre décrit comment gérer des 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 :

- «Canaux, clusters et mise en file d'attente éloignée»
- «Administration éloignée à partir d'un gestionnaire de files d'attente local à l'aide de commandes MQSC» à la page 67
- «Création d'une définition locale de file d'attente éloignée» à la page 73
- «Utilisation des définitions de files d'attente éloignées pour les alias» à la page 77

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 d'application écrits par l'utilisateur qui transfèrent des données d'un noeud sur un autre.
- Programmes d'application d'administration écrits par l'utilisateur et qui ont recours aux commandes PCF, à l'interface MQAI ou aux interfaces ADSI.
- 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 qui lui permet de détecter les messages entrants et de les transporter. 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.

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 68 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 (CLUSSDR) 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 éloignée à partir d'un gestionnaire de files d'attente local à l'aide de commandes MQSC

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, 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 AIX\*, AS/400, MVS/ESA, et OS/2\*.

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 6 à la page 68 montre la configuration des gestionnaires de files d'attente et des canaux requis pour l'administration éloignée à l'aide de la commande **runmqsc**. Le gestionnaire `source.gest.files` est le gestionnaire *source* à partir duquel vous émettrez des commandes MQSC et qui recevra, éventuellement, le résultat de ces commandes (par des messages opérateur). Le gestionnaire `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 250.



## Administration éloignée

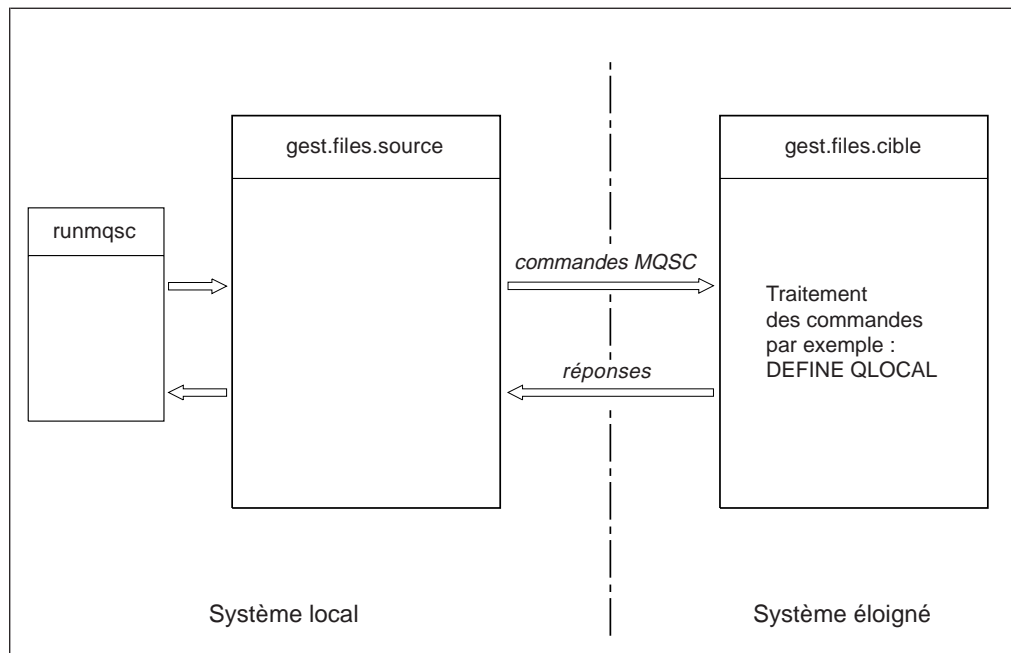


Figure 6. Administration éloignée

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**.

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 7 à la page 69.



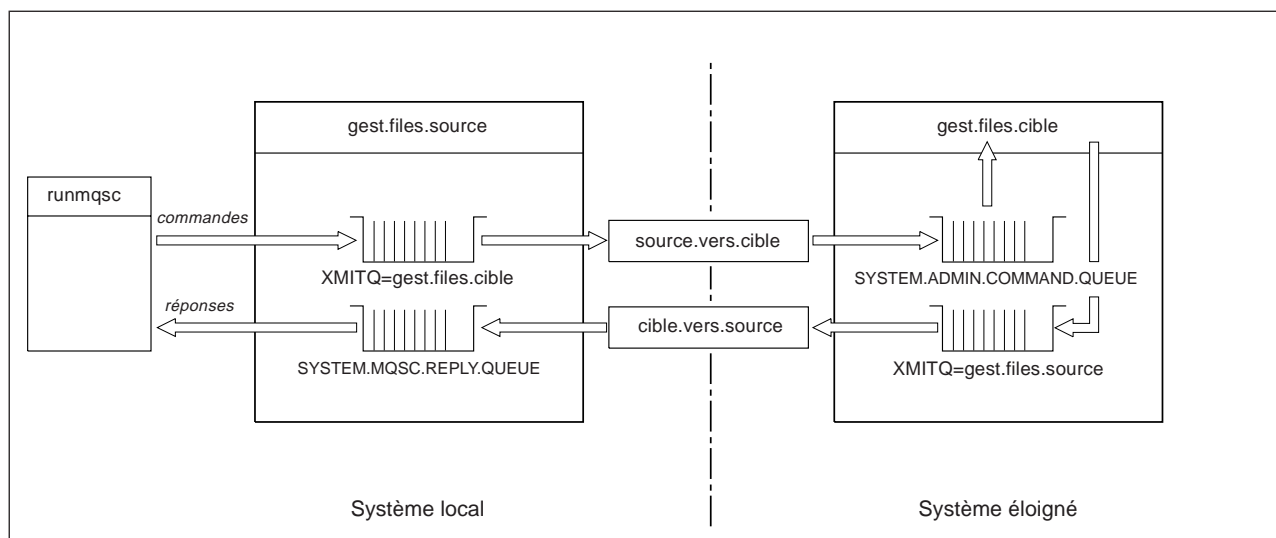


Figure 7. Configuration des canaux et des files d'attente pour l'administration éloignée

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 :

```
* Définition du canal émetteur sur le gestionnaire source
DEFINE CHANNEL ('source.vers.cible') +
  CHLTYPE(SDR) +
  CONNAME (RHX5498) +
  XMITQ ('cible.gest.files') +
  TRPTYPE(TCP)

* Définition du canal récepteur sur le gestionnaire source
DEFINE CHANNEL ('cible.vers.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Définition de la file de transmission sur le gestionnaire source
DEFINE QLOCAL ('cible.gest.files') +
  USAGE (XMITQ)
```

## Administration éloignée

Sur le gestionnaire cible (cible.gest.files), émettez ces commandes pour créer les canaux et file d'attente de transmission :

```
* Définition du canal émetteur sur le gestionnaire de files d'attente cible

DEFINE CHANNEL ('cible.vers.source') +
  CHLTYPE(SDR) +
  CONNAME (RHX7721) +
  XMITQ ('source.gest.files') +
  TRPTYPE(TCP)

* Définition du canal récepteur sur le gestionnaire cible

DEFINE CHANNEL ('source.vers.cible') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Définition de la file d'attente de transmission sur le gestionnaire cible

DEFINE QLOCAL ('source.gest.files') +
  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. Ce nom correspond au nom 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 OpenVMS. Dans le cas contraire, consultez la documentation appropriée pour l'extrémité non-OpenVMS du canal.

Pour lancez les deux canaux, assurez-vous d'abord que les services TCP/IP ont été configurés pour MQSeries sur les deux noeuds et fonctionnent à chaque extrémité des connexions.

Démarrez un programme d'écoute sur l'extrémité cible de chaque canal.

- Sur le gestionnaire source, entrez :

```
runmqtsr -m "source.gest.files" -t tcp
```

- Sur le gestionnaire destination, entrez :

```
runmqtsr -m "cible.gest.files" -t tcp
```

Il n'est pas nécessaire d'indiquer le nom du gestionnaire de files d'attente dans la commande **runmqchl** car le gestionnaire de files d'attente source doit être le gestionnaire par défaut. Indiquez le nom du gestionnaire de files d'attente dans la commande **runmchl** pour le gestionnaire de destination, s'il n'est pas le gestionnaire par défaut de son noeud.

Lancez ensuite les canaux :

- Sur le gestionnaire source, entrez :

```
runmqchl -m "source.gest.files" -c "source.vers.cible"
```

- Sur le gestionnaire destination, entrez :

```
runmqchl -m "cible.gest.files" -c "cible.vers.source"
```

Les commandes **runmqclr** et **runmqchl** sont des commandes de contrôle MQSeries. Elles ne peuvent pas être émises en utilisant **runmqsc**. Vous pouvez lancer les programmes d'écoute et les canaux à l'aide de commandes **runmqsc** ou de scripts (lancement du canal et lancement du programme d'écoute).

## Définition automatique de canaux

La définition automatique de canaux ne s'applique qu'aux cas dans lesquels 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 and 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"
```

## Administration éloignée

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: Le délai imparti à MQSC pour l'attente d'un
réponse venant du serveur de commandes est dépassé.
```

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.

En mode indirect, vous pouvez également exécuter un fichier de commandes MQSC sur un gestionnaire de files d'attente éloigné. Par exemple :

```
runmqsc -w 60 "cible.gest.files" < mescomds.in > état.out
```

où `mescomds.in` est un fichier contenant les commandes MQSC et `état.out` 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 OpenVMS. 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 OpenVMS :

```
runmqsc -w 30 -x "cible.gest.files"
```

Sur le canal émetteur, définissez l'attribut **CONVERT** sur **YES**. Il indique que la conversion des données nécessaire entre les systèmes est réalisée du côté OpenVMS. La commande de définition de canal devient :

```
* Définition du canal émetteur sur le gestionnaire source sous OpenVMS

DEFINE CHANNEL ('source.vers.cible') +
  CHLTYPE(SDR) +
  CONNAME (RHX5498) +
  XMITQ ('cible.gest.files') +
  TRPTYPE(TCP) +
  CONVERT (YES)
```

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 46.

---

## Création d'une définition locale de file d'attente éloignée

Vous pouvez utiliser une définition de file d'attente éloignée comme une définition locale d'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

## Définition locale d'une file d'attente éloignée

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

**Objectif :** Une application doit insérer un message dans une file d'attente qui appartient à un gestionnaire de files d'attente éloigné.

**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 (FILE.TRANSMIS.COTEE)
```

Où :

#### **QREMOTE (FILE.ELOIGNEE.CYAN)**

Indique le nom local de l'objet file d'attente éloignée. Il s'agit du nom que

## Définition locale d'une file d'attente éloignée

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. Ce nom doit être mis entre guillemets **simples**.

### XMITQ (FILE.TRANSMIS.COTEE)

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 utilisé.

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(XMIT)).

## 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 :

## Définition locale d'une file d'attente éloignée

```
* Affichage des attributs de la file d'attente éloignée.  
* ALL = Affichage de tous les attributs  
  
DISPLAY QUEUE (FILE.ELOIGNEE.CYAN) ALL  
  
* Modification de la file d'attente éloignée pour permettre les  
* insertions. Ceci n'affecte pas la file d'attente cible, mais  
* seulement les applications qui spécifient cette file éloignée.  
  
ALTER QREMOTE (FILE.ELOIGNEE.CYAN) PUT(ENABLED)  
  
* Suppression de cette file d'attente éloignée  
* Ceci n'affecte pas la file d'attente cible, mais  
* seulement sa définition locale.  
  
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. 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 ('cible.gest.files') +  
  DESCR ('File de transmission par défaut du gest. files cible') +  
  USAGE (XMITQ)
```

Les applications peuvent insérer des messages directement dans une file d'attente de transmission, avec l'en-tête approprié, ou indirectement, par exemple par le



## Définition locale d'une file d'attente éloignée

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 73.

---

## 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 si le message doit arriver 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

## Alias

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 «Strophe AllQueueManagers» à la page 175.

les conversions prises en charge sont définies dans les 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».

### **Fichier ccsid.tbl**

Le fichier ccsid.tbl est utilisé pour les opérations suivantes :

- Il spécifie les jeux de codes supplémentaires. Pour ce faire, vous devez modifier le fichier ccsid.tbl (le fichier vous fournit des indications sur la procédure à suivre).
- Il spécifie la conversion des données par défaut.

Vous pouvez mettre à jour les informations contenues dans le fichier ccsid.tbl. 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.

Dans l'environnement MQSeries pour Compaq OpenVMS, un fichier ccsid.tbl exemple est fourni sous le nom de :

```
MQS_EXAMPLES:CCSID.TBL
```

et le fichier ccsid.tbl actif se trouve dans le répertoire :

```
MQS_ROOT:[MQM.CONV.TABLE]
```

**Conversion des données par défaut :** Pour mettre en oeuvre la conversion de données par défaut, vous devez modifier le fichier ccsid.tbl 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.tbl 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.tbl.

**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 qu'elle 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 *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 7. Protection des objets MQSeries

Ce chapitre décrit les fonctions de contrôle de la sécurité de MQSeries pour Compaq OpenVMS et leur mise en oeuvre.

Il comprend les sections suivantes :

- «Nécessité de protéger les ressources MQSeries»
- «Avant de commencer»
- «Présentation du gestionnaire des droits d'accès aux objets (OAM)» à la page 83
- «Utilisation des commandes de l'OAM» à la page 85
- «Caractéristiques de l'OAM» à la page 88
- «Tables de spécification des droits d'accès» à la page 91
- «Fichiers de droits d'accès» à la page 98

---

### 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 de protéger contre les accès non autorisés aux ressources dont le gestionnaire de files d'attente est propriétaire. 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.

---

### Avant de commencer

Toutes les ressources de gestionnaire de files d'attente fonctionnent avec l'identificateur de droits VMS :

MQM

Cet identificateur de droits est créé lors de l'installation de MQSeries. Vous **devez** lui attribuer cet attribut de ressource, pour tous les utilisateurs qui ont besoin de contrôler des ressources MQSeries.

## ID utilisateur dans MQSeries pour Compaq OpenVMS avec l'identificateur de ressource MQM

Si votre ID utilisateur dispose de l'identificateur de droits OpenVMS MQM, vous bénéficiez de tous les droits à toutes les ressources MQSeries. Votre ID utilisateur *doit* disposer de l'identificateur de droits OpenVMS MQM pour pouvoir utiliser toutes les commandes de contrôle MQSeries pour Compaq OpenVMS (à l'exception de **crtmqcvx**). Vous devez notamment disposer de ces droits d'accès pour :

- utiliser la commande **runmqsc** pour exécuter les commandes MQSC.
- administrer les droits d'accès sous MQSeries pour Compaq OpenVMS 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 éloigné, vérifiez que votre ID utilisateur dispose de l'identificateur de droits MQM OpenVMS 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 90.

De plus, l'installation de MQSeries crée un identificateur MQS\_SERVER. Il dispose des droits de propriété du domaine de ressources dans lequel VMS conserve les informations de verrouillage pour MQSeries. Par défaut, les droits d'accès à cet identifiant sont accordés aux utilisateurs suivants :

- ceux qui figurent dans le même groupe que l'utilisateur MQM, ou
- les utilisateurs système, ou
- les utilisateurs qui ont des droits SYSPRV, SYSLCK, ou BYPASS

Pour permettre aux autres utilisateurs d'accéder aux ressources MQ, vous devez veiller à ce que l'identificateur MQS\_SERVER ait les droits WORLD appropriés en exécutant la commande suivante :

```
SET SECURITY/CLASS=RESOURCE [MQS_SERVER] /PROTECTION=(W:RWL)
```

**Remarque :** 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

## Informations complémentaires

Pour plus d'informations sur :

- les jeux de commandes MQSeries pour Compaq OpenVMS, consultez le «Chapitre 2. Présentation des commandes d'administration MQSeries» à la page 21.
- les commandes de contrôle MQSeries pour Compaq OpenVMS, consultez le «Chapitre 17. Commandes de contrôle MQSeries» à la page 243.
- les commandes PCF et commandes d'échappement PCF, consultez le manuel *MQSeries Programmable System Management*.
- les appels MQI, consultez les manuels *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 d'autorisation. Ce composant est appelé Gestionnaire des droits d'accès aux objets (OAM) pour MQSeries pour Compaq OpenVMS. Fourni avec MQSeries pour Compaq OpenVMS, il 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. Vous pouvez ainsi :

- remplacer l'OAM livré avec le produit par votre propre système d'autorisation à l'aide de l'interface fournie ;
- 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 fonctionne en exploitant les fonctions de sécurité du système d'exploitation OpenVMS. En particulier, l'OAM utilise les ID utilisateur, de groupe et de droits d'accès de OpenVMS. 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 d'identificateurs de droits

Dans l'interface, nous utilisons le terme *principal* plutôt qu'ID utilisateur car 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. Cependant, dans ce chapitre, les principaux et les ID utilisateurs sont toujours des identificateurs utilisateur OpenVMS.

#### Identificateurs de droits et identificateur de droits primaires

La gestion des droits d'accès aux ressources MQSeries est basée sur les *identificateurs de droits* OpenVMS c'est-à-dire les identificateurs de principaux. Un principal peut contenir un ou plusieurs identificateurs de droits OpenVMS. Un groupe est l'ensemble de tous les principaux qui ont été accordés à un identificateur de droits spécifique.

L'OAM gère les droits d'accès au niveau des identificateurs de droits plutôt qu'au niveau des principaux individuels. Le mappage des principaux avec les noms

## Gestionnaire des droits d'accès aux objets.

d'identificateurs est effectué dans OAM et les opérations sont réalisées au niveau de l'identificateur des droits d'accès. Vous pouvez toutefois afficher les droits d'accès d'un principal individuel.

### Principal doté de plusieurs identificateurs de droits d'accès

Les droits d'accès dont dispose un principal sont l'union de ses identificateurs de droits d'accès, c'est-à-dire ses droits de traitement. Lorsqu'un principal demande l'accès à une ressource, l'OAM calcule cette union, et vérifie qu'elle est bien conforme aux droits d'accès. Vous pouvez utiliser la commande **setmqaut** pour définir les droits d'accès d'un principal 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. Toutefois, les modifications apportées aux appartenances d'un principal ne seront prises en compte que lorsque le gestionnaire de files d'attente sera réinitialisé, c'est-à-dire arrêté puis redémarré.

Les droits d'accès associés à un principal sont mis en cache lorsqu'ils sont calculés par l'OAM. Une fois mises en cache les modifications apportées aux droits d'accès d'un identificateur, elles ne sont effectives qu'une fois le gestionnaire de files d'attente redémarré. Evitez de modifier les droits d'accès lorsque le gestionnaire de files d'attente est en cours d'exécution.

## Identificateurs de droits par défaut

L'OAM reconnaît un groupe d'utilisateurs par défaut auquel tous les utilisateurs sont affectés de manière nominale. Ce groupe est défini par le pseudo-identificateur de droits 'NOBODY'. 'NOBODY' peut être utilisé comme un identificateur de droits valide pour attribuer des droits d'accès à l'aide de commandes MQSeries. Par défaut, aucun droit d'accès n'est accordé à cet identificateur. Les utilisateurs n'ayant pas de droits d'accès spécifiques peuvent se voir accorder l'accès aux ressources MQSeries via cet identificateur de droits.

## Ressources pouvant être protégées par l'OAM

L'OAM peut contrôler :

- L'accès aux objets MQSeries via l'interface MQI. Lorsqu'un programme d'application tente d'accéder à un objet, l'OAM vérifie si l'ID utilisateur auteur de la demande possède les droits d'accès (via l'identificateur) 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 principaux dotés de l'identificateur de droits MQM 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 principaux dotés de l'identificateur de droits MQM peuvent exécuter des commandes de contrôle, comme la création d'un gestionnaire de files d'attente, le démarrage d'un serveur de commandes ou l'utilisation de **runmqsc**.
- L'autorisation d'utiliser les commandes PCF.

Différents utilisateurs peuvent se voir accorder différents types de droits d'accès à un même objet. Par exemple, pour une file d'attente particulière, un utilisateur doté d'un identificateur peut être autorisé à exécuter des opérations d'insertion et



## Gestionnaire des droits d'accès aux objets.

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, les utilisateurs dotés d'identificateurs 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 des identificateurs de droits pour les droits d'accès

L'utilisation d'identificateurs (au lieu de principaux) 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 attribuant l'identificateur approprié à leur ID utilisateur OpenVMS.

Essayez de réduire le nombre d'identificateurs au minimum, en répartissant par exemple les principaux entre un groupe d'utilisateurs d'applications et un groupe d'administrateurs.

## Désactivation de l'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.

```
$ DEFINE/SYSTEM MQSNOAUT TRUE
```

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 (qm.ini).

---

## 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 contenir l'identificateur de droits d'accès OpenVMS MQM). Cet identificateur a dû être défini lorsque vous avez installé MQSeries.

Si votre ID utilisateur contient l'identificateur MQM, vous disposez d'un droit de gestion 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 deux commandes que vous pouvez appeler depuis votre DCL OpenVMS pour gérer les droits d'accès des utilisateurs. Il s'agit des commandes suivantes :

- **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.

## Utilisation des commandes de l'OAM

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.

## Précision des informations indiquées dans les 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 le principal ou l'identificateur 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 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 identificateur de droits d'accès à accéder à un objet particulier. L'exemple suivant indique comment utiliser la commande **setmqaut** :

```
setmqaut -m "saturne.gest.files" -t queue -n FILE.LOCALE.ROUGE -g GROUPA +browse -get +put
```

Dans cet exemple :

Le terme...	désigne...
<b>saturne.gest.files</b>	Le gestionnaire de files d'attente.
<b>queue</b>	Le type d'objet.
<b>FILE.LOCALE.ROUGE</b>	L'objet.
<b>GROUPA</b>	L' ID du groupe auquel les droits d'accès sont accordés.
<b>+browse -get +put</b>	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...	a l'effet suivant...
<b>+browse</b>	Ajoute les droits d'accès permettant de consulter (MQGET avec option de consultation) les messages de la file.

## Utilisation des commandes de l'OAM

<b>-get</b>	Supprime les droits d'accès permettant d'extraire (MQGET) les messages de la file.
<b>+put</b>	Ajoute les droits d'accès permettant d'insérer (MQPUT) des messages dans la file.

Cela signifie que les applications lancées avec des ID utilisateur contenant l'identificateur OpenVMS GROUPA possèdent ces autorisations.

Vous pouvez indiquer un ou plusieurs principaux et, en même temps, un ou plusieurs identificateurs. Par exemple, la commande suivante révoque le droit d'accès d'insertion dans la file d'attente MaFile du principal FVUSER et des identificateurs GROUPA et GROUPB.

```
setmqaut -m "saturne.gest.files" -t queue -n "MaFile" -p FVUSER -g GROUPA -g GROUPB -put
```

**Remarque :** Cette commande révoque également le droit d'accès d'insertion pour tous les identificateurs de droits de FVUSER, c'est-à-dire tous les groupes auxquels appartient FVUSER.

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 301.

### 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 service 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 301.

## 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 identificateur 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 identificateur 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 259.

## Utilisation des commandes de l'OAM

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.

```
dspmqaout -m "GestFiles1" -t process -n "Annuités" -g "GpAdmin"
```

Les mots clés affichés en retour de cette commande indiquent les droits d'accès activés.

---

## Caractéristiques de l'OAM

Certaines opérations sont particulièrement sensibles et doivent être limitées à des utilisateurs privilégiés. Par exemple :

- 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.
- la création et la copie de files d'attente d'applications.

### ID utilisateur

L'ID utilisateur spécial MQM que vous avez créé lors de l'installation ne doit être utilisé que par le produit. Il ne doit jamais être mis à la disposition des utilisateurs non privilégiés.

L'ID utilisateur utilisé pour la vérification des droits d'accès, associé à un processus MQ, est l'ID utilisateur OpenVMS.

### Répertoires du gestionnaire de files d'attente

Le répertoire 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 OpenVMS, liés à leurs droits d'accès OAM. N'utilisez pas les commandes standard de OpenVMS pour accorder ou supprimer les 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 «Signification des noms de fichiers MQSeries» à la page 23.
- Tous les objets sont la propriété de l'ID utilisateur MQM.

### 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 plus d'informations, consultez la note 1 à la page 95.

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 ID utilisateur à une file d'attente alias qui correspond à une file d'attente locale à laquelle l'ID utilisateur 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.

**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 et 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*

#### 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

## Caractéristiques OAM

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.

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 *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.

### 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 2 à la page 92.

### 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 OpenVMSéloigné. L'ID utilisateur, tel qu'il est spécifié dans le descripteur de message PCF, doit contenir l'identificateur de droits d'accès MQM sur le système cible. Ces commandes sont les suivantes :

- *ChangeChannel*
- *CopyChannel*
- *CreateChannel*
- *DeleteChannel*
- *PingChannel*
- *ResetChannel*
- *StartChannel*

- *StartChannelInitiator*
- *StartChannelListener*
- *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 OpenVMS 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 contenir l'identificateur de droits d'accès 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
- START LISTENER
- 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.

### Commandes de contrôle des canaux

En ce qui concerne les commandes de contrôle de canaux, l'ID utilisateur qui les émet doit contenir l'identificateur de droits MQM. Ces commandes sont les suivantes :

- **runmqchi** (exécution de l'initialisateur de canal)
- **runmqchl** (lancement du canal)

---

## Tables de spécification des droits d'accès

Les tables de spécification des droits d'accès présentées dans le 92. Ces tables concernent les situations ci-après :

- 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\_'.



## Tables de spécification des droits d'accès

Dans les tables, 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 cmqzc.h, qui est fourni avec le produit. Pour plus d'informations, reportez-vous à la section «Contenu des fichiers de droits d'accès» à la page 99.

## 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.

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 2 résume les droits nécessaires pour chaque appel.

Tableau 2. Droits d'accès nécessaires pour les appels MQI

Droits requis pour :	File d'attente (1)	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	MQZAO_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)	MQZAO_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 (7)	Non applicable



## Tables de spécification des droits d'accès

Tableau 2. Droits d'accès nécessaires pour les appels MQI (suite)

Droits requis pour :	File d'attente (1)	Objet processus	Gestionnaire de file d'attente	Listes de noms
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)	(10)
Option MQPUT1				
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 :

1. 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.

## Tables de spécification des droits d'accès

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. MQOO\_INPUT\_\* doit également être indiqué. Cela est valable pour une file d'attente locale, modèle, ou alias.
4. Cette vérification est effectuée pour tous les cas de sortie, sauf celui indiqué à la remarque 9.
5. MQOO\_OUTPUT doit également être indiqué.
6. MQOO\_PASS\_IDENTITY\_CONTEXT est également concerné par cette option.
7. Ce droit est nécessaire à la fois pour l'objet gestionnaire de files d'attente et la file d'attente particulière.
8. MQOO\_PASS\_IDENTITY\_CONTEXT, MQOO\_PASS\_ALL\_CONTEXT et MQOO\_SET\_IDENTITY\_CONTEXT sont également concernés par cette option.
9. 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 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.

## Tables de spécification des droits d'accès

- 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 94) 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 3 résume les droits nécessaires à chaque commande MQSC contenue dans la fonction Escape PCF.

Tableau 3. 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
Commande MQSC				
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

## Tables de spécification des droits d'accès

### 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 est indiqué, 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. Cela 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 4 résume les droits nécessaires pour chaque commande PCF.

Tableau 4. Commandes PCF et droits d'accès nécessaires

(2) Droits requis pour :	Objet de file d'attente	Objet processus	Gestionnaire de file d'attente	Listes de noms
<b>commande PCF</b>				
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
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)

## Tables de spécification des droits d'accès

Tableau 4. Commandes PCF et droits d'accès nécessaires (suite)

(2) Droits requis pour :	Objet de file d'attente	Objet processus	Gestionnaire de file d'attente	Listes de noms
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, liste de noms 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
  - MQZAO\_CLEAR
  - MQZAO\_DELETE

## Tables de spécification des droits d'accès

- 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 une procédure.

---

## Fichiers de droits d'accès

**Remarque :** Les informations de cette section concernent l'identification des incidents. Dans des conditions normales, utilisez les commandes permettant d'afficher et de modifier les informations sur les droits d'accès.

MQSeries pour Compaq OpenVMS utilise une structure de fichiers particulière pour assurer la sécurité. Aucune opération ne doit être effectuée sur ces fichiers, mais vous devez tout de même assurer l'intégrité et la sécurité de leurs données.

La sécurité est assurée par le biais de fichiers de droits d'accès. Il existe trois types de droits d'accès :

- les droits qui ne s'appliquent qu'à un seul objet (par exemple, le droit d'insérer un message dans une file d'attente) ;
- les droits qui s'appliquent à une classe d'objets (par exemple, le droit de créer une file d'attente) ;
- les droits portant sur toutes les classes d'objets, par exemple, le droit d'exécuter des opérations pour différents utilisateurs.

## Chemins des fichiers de droits d'accès

Le chemin d'un fichier de droits d'accès dépend de son type. Lorsque vous indiquez un droit sur un objet, par exemple, le gestionnaire de files d'attente crée les fichiers de droits appropriés. Il les place ensuite dans un sous-répertoire dont le chemin d'accès est défini par le nom du gestionnaire de files d'attente, le type de droit d'accès, et le cas échéant, le nom de l'objet.

Tous les droits ne s'appliquent pas directement aux instances d'objets. Par exemple, le droit de créer un objet s'applique à la classe d'objets à laquelle il appartient, et non à une instance particulière de cet objet. De même, certains droits concernent la totalité d'un gestionnaire de files d'attente (par exemple, droit d'utilisateur secondaire signifie qu'un utilisateur peut disposer des droits associés à un autre utilisateur).

### Répertoires de droits d'accès

Par défaut les répertoires de droits d'accès d'un gestionnaire de files d'attente appelé saturne sont les suivants :

**MQS\_ROOT: [MQM.QMGRS.SATURNE.AUTH.QUEUES]**

Fichiers de droits pour files d'attente.

**MQS\_ROOT: [MQM.QMGRS.SATURNE.AUTH.PROCDEF]**

Fichiers de droits pour définitions de procédures.

**MQS\_ROOT: [MQM.QMGRS.SATURNE.AUTH.QMANAGER]**

Fichiers de droits pour le gestionnaire de files d'attente.

**MQS\_ROOT: [MQM.QMGRS.SATURNE.AUTH]\$ACCLASS**

Droits d'accès s'appliquant à toutes les classes.

**MQS\_ROOT: [MQM.QMGRS.SATURNE.AUTH.NAMELIST]**

Droits d'accès s'appliquant à toutes les listes de noms.

Dans les répertoires objets les fichiers \$CLASS contiennent les droits relatifs à la classe toute entière.

**Remarque :** Il existe une différence entre \$CLASS (fichier spécifiant les droits d'une classe particulière) et \$ACCLASS (le fichier de droits définissant les droits d'accès à toutes les classes).

Les chemins des fichiers de droits d'accès des objets sont basés sur ceux de l'objet lui-même, où AUTH est inséré avant le répertoire de type d'objet. Vous pouvez utiliser la commande **dspmqls** pour afficher le chemin d'accès à un objet spécifique.

Par exemple, si le nom et le chemin d'accès de SYSTEM.DEFAULT.LOCAL.QUEUE est:

**MQS\_ROOT : [MQM.QMGRS.SATURNE.QUEUES.SYSTEM\$DEFAULT\$LOCAL\$QUEUE]**

Le nom et le chemin du fichier de droits correspondant est :

**MQS\_ROOT: [MQM.QMGRS.SATURNE.AUTH.QUEUES.SYSTEM\$DEFAULT\$LOCAL\$QUEUE]**

**Remarque :** Dans ce cas, les noms réels des fichiers associés à la file d'attente ne sont pas identiques à ceux de la file d'attente elle-même. Consultez la section «Signification des noms de fichiers MQSeries» à la page 23 pour plus d'informations.

## Contenu des fichiers de droits d'accès

Les droits d'accès d'un identificateur particulier sont définis par une série de strophes dans le fichier de droits d'accès. Pour plus d'informations, reportez-vous à la section «Fichiers de droits d'accès» à la page 98. Les droits s'appliquent à l'objet associé au fichier. Par exemple :

```
groupb:  
  Authority=0x0040007
```

## Fichiers de droits d'accès

Cette strophe définit le droit d'accès pour l'identificateur GROUPB. Les droits d'accès obtenus correspondent à la réunion des schémas de bits individuels basés sur les affectations suivantes :

Mot clé de droits d'accès	Nom formel	Valeur hexadécimale
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
-----	
Droit :	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.

### Fichiers de droits d'accès aux classes

Les fichiers de droits d'accès de classe contiennent des droits qui correspondent à la classe entière. Ces fichiers sont appelés «\$CLASS» et existent dans le même répertoire que les fichiers des objets particuliers. L'entrée MQZAO\_CRT dans le fichier \$CLASS donne le droit de créer un objet dans cette classe. Il s'agit du seul droit de classe.



### Fichiers de droits d'accès à toutes les classes

Le fichier de droits d'accès à toutes les classes contient les droits qui s'appliquent à un gestionnaire de files d'attente entier. Ce fichier est appelé \$ACCLASS et existe dans le sous-répertoire auth du gestionnaire de files d'attente.

Les droits suivants s'appliquent au gestionnaire de files d'attente dans son entier et sont contenus dans le fichier de droits à toutes les classes :

**L'entrée...      Autorise à...**

**MQZAO\_ALTERNATE\_USER\_AUTHORITY**

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.

### Gestion des fichiers de droits d'accès

Voici quelques éléments à prendre en compte pour la gestion des fichiers de droits d'accès :

1. Vous devez vérifier que les fichiers de droits d'accès sont sécurisés et que des utilisateurs non autorisés n'y ont pas accès en écriture. Consultez la section «Accès aux fichiers de droits d'accès».
2. Pour pouvoir reproduire votre fichier de droits, effectuez au moins l'une des tâches suivantes :
  - Sauvegarde du sous-répertoire AUTH après toute mise à jour importante.
  - Conservation des fichiers de commande DCL contenant les commandes utilisées.
3. Vous pouvez copier et éditer les fichiers de droits d'accès. Cependant, vous ne devez normalement pas avoir besoin de les créer ou de les modifier manuellement. En cas d'urgence, les informations données ici peuvent être utilisées pour récupérer les fichiers perdus ou altérés.

### Accès aux fichiers de droits d'accès

Les fichiers d'autorisation doivent pouvoir être lus par n'importe quel principal. Cependant, seul le gestionnaire système et l'utilisateur dotés de l'identificateur MQM doivent être autorisés à les mettre à jour.

Les droits sur les fichiers de droits d'accès créés par l'OAM sont les suivants :  
S:RWD, O:RWD, G:RWD, W:R (ID=MQM, ACCESS=R+W+E+D+C)

Ne les modifiez pas sans avoir vérifié la sécurité des données.

Pour pouvoir modifier les droits à l'aide de la commande fournie avec MQSeries pour Compaq OpenVMS, votre processus doit être doté de l'identificateur de droits d'accès MQM.



---

## Chapitre 8. Gestionnaire de files d'attente de rebut (DLQ) MQSeries

Une *file d'attente de rebut* (DLQ), (parfois appelée *file d'attente des messages non livrés*), est une file d'attente qui contient les messages ne pouvant pas être livrés dans leurs files d'attente cibles. 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 des DLQ doivent être préfixés à l'aide d'une structure *d'en-tête de rebut* (MQDLH). Les messages insérés en DLQ par un gestionnaire de files d'attente ou par un agent MCA possèdent toujours un MQDLH. En ce qui concerne les applications qui insèrent des messages en DLQ, il est fortement recommandé de fournir un 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.

Dans tous les environnements MQSeries, il doit exister une routine traitant régulièrement les messages en 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 règles, le gestionnaire DLQ exécute l'action associée à cette entrée.

Ce chapitre contient les sections suivantes :

- «Lancement du gestionnaire DLQ»
- «règle du gestionnaire DLQ» à la page 104
- «Traitement de la règles» à la page 110
- «Exemple de règle de la règles d'un gestionnaire DLQ» à la page 113

---

### Lancement du gestionnaire DLQ

Pour lancer le gestionnaire DLQ, vous devez utiliser la commande **runmqdlq**. Vous pouvez désigner la file DLQ à traiter et le gestionnaire de files d'attente à utiliser de deux manières :

- sous forme de paramètre de **runmqdlq** à partir de la ligne de commande.  
Exemple :

```
runmqdlq File.ATT.REBUT.ABC1 ABC1.GEST.FILES < qrule.rul
```

- dans la règles. Exemple :

```
INPUTQ(FILE.ATT.REBUT.ABC1) INPUTQM(ABC1.GEST.FILES)
```

## Gestionnaire de files d'attente de rebut

Les exemples ci-dessus s'appliquent à la DLQ appelée FILE.ATT.REBUT.ABC1, dont le gestionnaire est ABC1.GEST.FILES.

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** prend ses données d'entrée dans SYS\$INPUT : vous associez la règles à **runmqdlq** en réacheminant SYS\$INPUT à partir de la règles.

**Attention:** le fait d'exécuter le gestionnaire DLQ sans rediriger SYS\$INPUT vers un fichier de règles provoque le fonctionnement en boucle du gestionnaire.

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** dans la section «runmqdlq (lancement du gestionnaire de files d'attente de rebut)» à la page 291.

## Gestionnaire DLQ exemple (amqsdldq)

Outre le gestionnaire DLQ lancé à partir de la commande **runmqdlq**, MQSeries fournit le code source d'un gestionnaire DLQ exemple, **amqsdldq**, dont les fonctions sont similaires à celles fournies via la commande **runmqdlq**. Les sources sont fournies comme modèle uniquement, et doivent être personnalisées pour créer un gestionnaire DLQ répondant à vos besoins spécifiques. Par exemple, vous pouvez souhaiter avoir un gestionnaire qui traite les messages sans en-tête de rebut. (En effet, le gestionnaire DLQ par défaut et le gestionnaire exemple **amqsdldq**, ne traitent que les messages en file DLQ qui commencent par un en-tête de rebut (MQDLH). Les messages qui ne commencent pas par un MQDLH sont identifiés comme étant en erreur et restent définitivement en DLQ.)

Le code source de **amqsdldq** se trouve dans le répertoire :

[.DLQ], sous MQS\_EXAMPLES

La version compilée est dans le répertoire suivant :

[.BIN], sous MQS\_EXAMPLES

---

## règle du gestionnaire DLQ

La 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, qui est 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 règles doit contenir au moins une règle.

Chaque entrée de la règles comporte un ou plusieurs mots clés.

## Données de contrôle

Cette section décrit les mots clés que vous pouvez inclure dans une entrée de données de contrôle d'une 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*|' ')

Permet d'indiquer la file DLQ à traiter :

1. Si vous indiquez une valeur INPUTQ comme paramètre de la commande **runmqdlq**, cette valeur remplace toute valeur INPUTQ de la règles.
2. Si vous ne précisez aucune valeur INPUTQ comme paramètre de la commande **runmqdlq** mais que vous *indiquez* une valeur dans la règles, c'est cette dernière qui est utilisée.
3. Si aucune file DLQ n'est indiquée ou si vous spécifiez INPUTQ(' ') dans la 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** qui est utilisé.
4. Si vous ne précisez aucune valeur INPUTQ comme paramètre de la commande **runmqdlq** ou comme valeur dans la règles, c'est la file DLQ appartenant au gestionnaire de files d'attente indiqué dans le mot clé INPUTQM de la règles qui est utilisé.

### **INPUTQM** (*NomGestFiles*|' ')

Permet d'indiquer le gestionnaire de files d'attente propriétaire de la file DLQ indiquée dans le mot clé INPUTQ :

1. Si vous indiquez une valeur INPUTQM comme paramètre de la commande **runmqdlq**, cette valeur remplace toute valeur INPUTQM de la règles.
2. Si vous ne précisez aucune valeur INPUTQM comme paramètre de la commande **runmqdlq** c'est la valeur INPUTQM indiquée dans la règles qui est utilisée.
3. Si aucun gestionnaire de files d'attente n'est spécifié ou si vous indiquez INPUTQM(' ') dans la règles, c'est le gestionnaire de files d'attente par défaut de l'installation qui est utilisé.

### **RETRYINT** (*Intervalle*|60)

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 ont été demandées. L'intervalle par défaut est de 60 secondes.

### **WAIT** (YES|NO|*nnn*)

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.

## Gestionnaire de files d'attente de rebut

Il est recommandé de spécifier WAIT (YES) pour les DLQ très actives et WAIT (NO) ou WAIT (*nmn*) pour les DLQ peu actives. Si le gestionnaire DLQ s'arrête, il est recommandé de le relancer par déclenchement.

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 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 de la table.

## Règles (modèles et actions)

La figure 8 présente un exemple de règle de la règles d'un gestionnaire DLQ.

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Figure 8. Exemple de règle de la règles d'un gestionnaire DLQ. Cette règle demande au gestionnaire DLQ d'effectuer 3 tentatives d'insertion dans la file d'attente cible pour les messages persistants insérés en DLQ en raison d'une désactivation de MQPUT et MQPUT1.

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

Vous trouverez ci-après la description des mots clés de modèle, que vous utilisez pour indiquer les valeurs auxquelles les messages en DLQ doivent être comparés. Tous les mots clés de modèle sont facultatifs.

#### APPLIDAT (*IdAppl*|\*)

Valeur *IdAppl* spécifiée dans le descripteur, MQMD, du message.

#### 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*|\*)

Valeur *TypeApplPut* spécifiée 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 *MsgType* est MQFB\_REPORT, *Retour* décrit la nature de l'état.

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 des messages de réponse indiqué 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

Vous trouverez ci-après les mots clés d'action utilisés pour 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.

## Gestionnaire de files d'attente de rebut

### 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.

### 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 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* à blanc, 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 dans la zone *NomGestFilesCible* de la structure MQDLH.

### &REPLYQM

Le nom est récupéré dans la zone *FileAttRép* du descripteur de message, MQMD.

' ' FWDQM(' '), qui est la valeur par défaut, 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.

**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)

Nombre de tentatives, compris entre 1 et 999 999 999 devant être effectuées pour une action (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 règles

La règles doit être conforme aux règles suivantes en ce qui concerne sa syntaxe, sa structure et son contenu :

- Une 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 au moins par 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 du début de cette dernière. 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 règles.
- Les lignes blanches ne sont pas prises en compte.
- Chaque entrée de la règles du gestionnaire DLQ comprend un ou plusieurs mots clés et les paramètres associés. Les paramètres doivent être conformes aux règles de syntaxe ci-après :

- 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 ( )**

## Gestionnaire de files d'attente de rebut

### FORMAT ()

#### FORMAT

- Les caractères génériques sont pris en charge : vous pouvez utiliser le point d'interrogation (?) à la place d'un caractère (exception faite du blanc de fin). Vous pouvez utiliser l'astérisque (\*) à la place de zéro ou de plusieurs caractères adjacents. L'astérisque (\*) et le point d'interrogation (?) sont *toujours* interprétés comme des caractères génériques dans les valeurs de paramètres.
- Les caractères génériques ne peuvent pas figurer dans les paramètres des mots clés suivants : ACTION, HEADER, RETRY, FWDQ, FWDQM, et PUTAUT.
- Les blancs de fin dans les valeurs de paramètres et dans les zones correspondantes du message en DLQ ne sont pas significatifs lors des mises en correspondance de caractères génériques. Cependant, les blancs d'en-tête ou insérés dans des chaînes entre des points d'interrogation sont significatifs pour les mises en correspondance des caractères génériques.
- Les paramètres numériques ne peuvent pas inclure 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 ci-après sont corrects :
  - 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 999 999 999. Si la valeur du paramètre se situe dans cette plage, il est accepté, même s'il n'est pas valide dans la zone à laquelle le mot clé fait référence. Les noms symboliques peuvent être utilisés comme paramètres numériques.
- Si la valeur d'une chaîne est plus courte que la zone du MQDLH ou MQMD à laquelle le mot clé fait référence, la valeur est complétée par des blancs pour remplir la zone. Si la valeur (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. Lorsque la longueur de la chaîne est calculée, chaque occurrence de deux apostrophes consécutives est comptée comme un caractère simple.

---

## Traitement de la règle

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. Le gestionnaire DLQ augmente de 1 le nombre de tentatives chaque fois qu'il tente d'appliquer une 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. Les formats de règles ayant une correspondance ne sont recherchés que pour les messages en DLQ qui commencent par un MQDLH. Les messages qui ne commencent pas par un MQDLH sont identifiés comme étant en erreur et restent définitivement en DLQ.
2. Tous les mots clés de format peuvent être désactivés, 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. Vous pouvez modifier la 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

Le gestionnaire DLQ conserve un enregistrement de tous les messages en DLQ qui ont été détectés, 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 (first-in-first-out). Ainsi, si la file d'attente n'est pas vide, une consultation périodique de la DLQ est effectuée pour vérifier tous les messages. Vous devez donc vous assurer que la DLQ contient 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.

## Gestionnaire de files d'attente de rebut

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 pris 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. Exemple :

```
ACTION (FWD) FWDQ (FILE.ATTENTE.IGNOREE) HEADER (YES)
```

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 adressés par aucune autre règle de la table. Par exemple, la dernière règle de la table peut se présenter de la façon suivantes :

```
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 règle de la règles d'un gestionnaire DLQ

La table de règles 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)

* 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.

```

## Gestionnaire de files d'attente de rebut

```
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)
```

---

## Chapitre 9. Événements outils

Les événement outils MQSeries permettent de contrôler le fonctionnement des gestionnaires de files d'attente. Ils sont brièvement présentés dans ce chapitre. Pour une description plus complète, reportez-vous au manuel *MQSeries Programmable System Management*.

---

### Description des événements outils

Les événements outils provoquent la création de messages spéciaux appelés *messages d'événements*, lorsque le gestionnaire de files d'attente détecte une série de conditions prédéfinies. Par exemple, les conditions suivantes provoquent l'événement *Queue Full (File d'attente saturée)* :

- Les événements Queue Full (File d'attente saturée) 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 maximum 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.
- Sous MQSeries pour Compaq OpenVMS, 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).

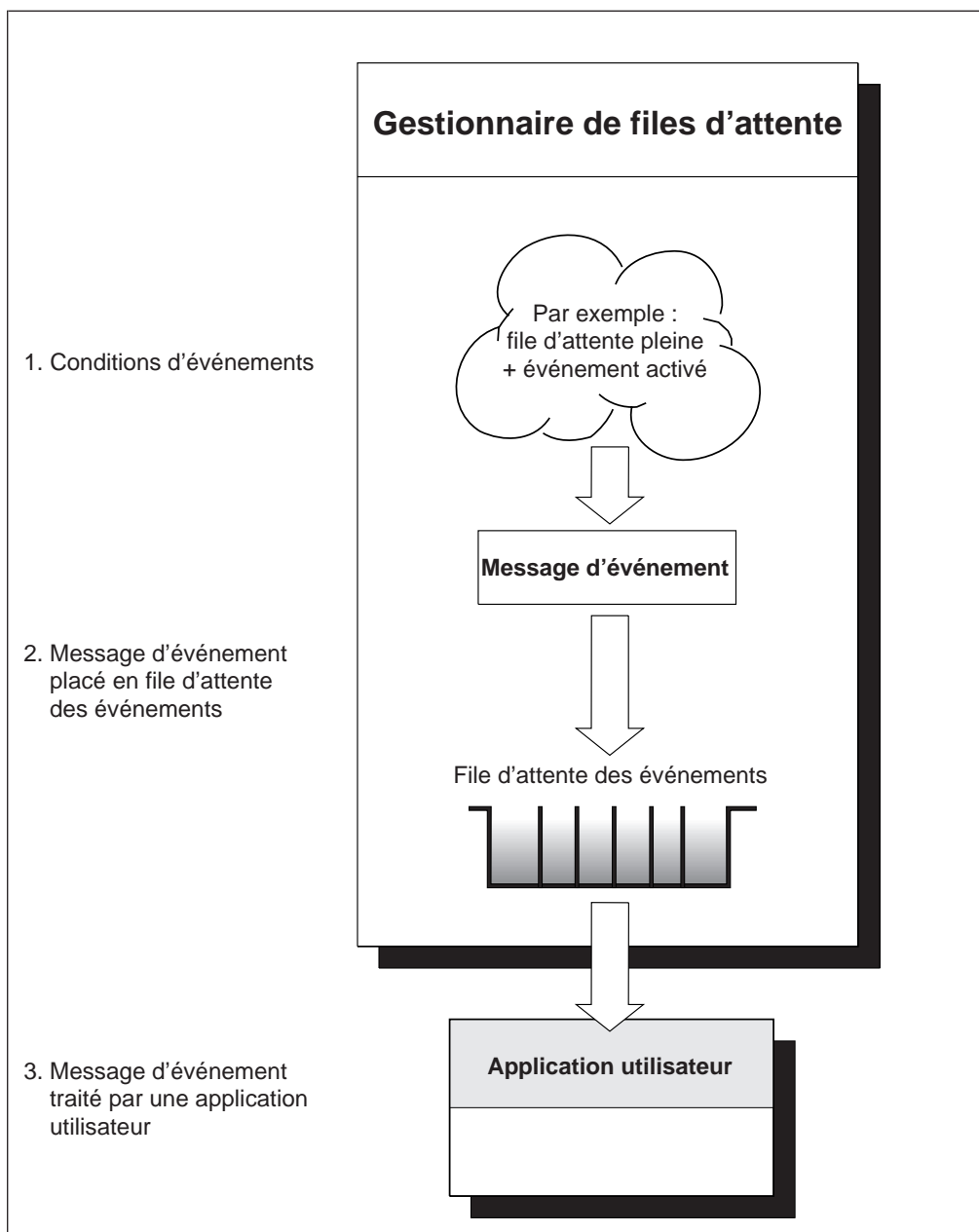


Figure 9. Présentation des événements outils. Lorsqu'un gestionnaire de files d'attente détecte la présence de conditions donnant lieu à un événement particulier, il insère un message d'événement dans la file d'événements appropriée.

Le message d'événement contenant des informations sur les conditions provoquant l'événement est placé dans une *file d'attente des événements*. Une application peut le retirer de la file d'attente d'événements pour l'analyser.

## Utilisation des événements outils

Si vous définissez les files d'attente d'événements en tant que files d'attente éloignées, vous pouvez placer toutes ces files d'attente 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. La figure 10 à la page 117 illustre ces opérations.



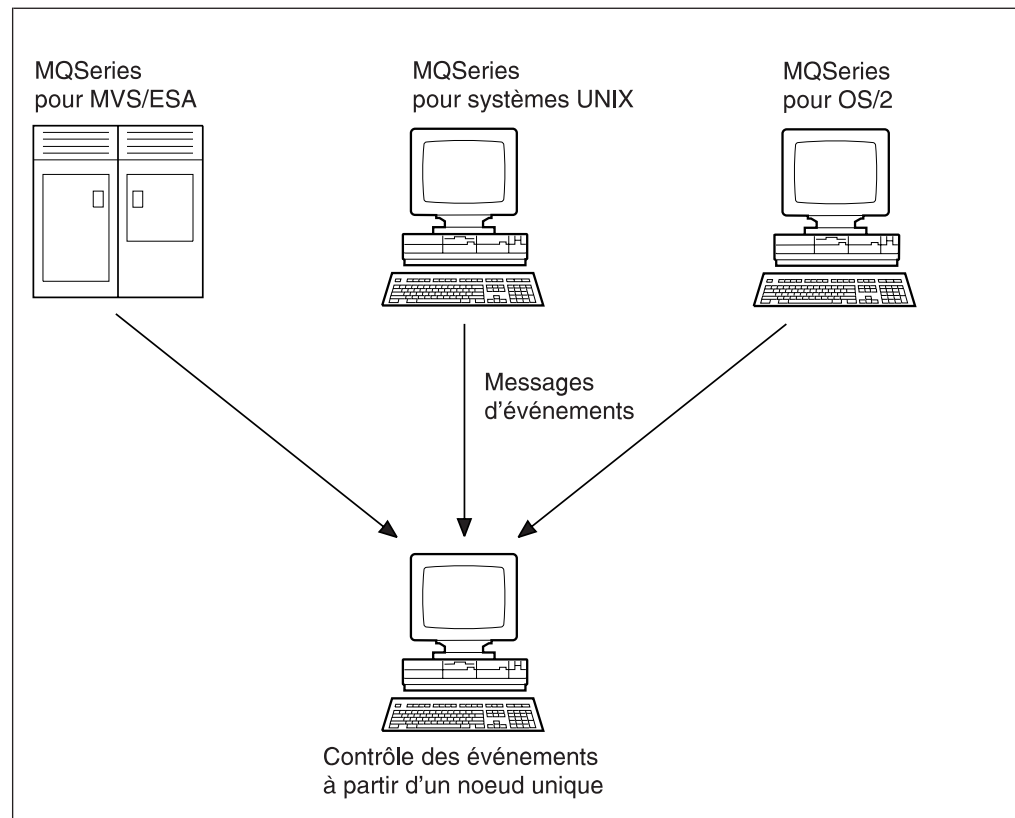


Figure 10. Contrôle des gestionnaires de files d'attente sur différentes plateformes sur un seul nœud

## Types d'événements

Les événements MQSeries se classent dans différentes catégories :

### 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.

### Événements 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.

### Événement 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).

### Événements de déclenchement

Lorsque ce manuel ou d'autres manuels MQSeries évoquent la fonction de déclenchement, il est parfois fait référence aux *événements de déclenchement*. Ce type d'événement intervient lorsqu'un gestionnaire de files d'attente détecte une condition favorable à l'apparition d'un événement de déclenchement. 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'émission d'un message de déclenchement qui est inséré dans une file d'attente d'initialisation et un programme d'application est parfois lancé.

## 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 d'application MQI adapté qui effectue les opérations ci-après.

- Extraction du message dans 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 la même file d'attente.

### Cette file d'attente d'événement...

Contient les messages provenant des...

#### SYSTEM.ADMIN.QMGR.EVENT

Événements Gestionnaire de files d'attente

#### SYSTEM.ADMIN.PERFM.EVENT

Événements Performances

#### SYSTEM.ADMIN.CHANNEL.EVENT

Événements Canal

Vous pouvez définir les files d'attente d'événements comme des files d'attente locales ou éloignées. Le fait de définir toutes vos files d'attente d'événements comme étant éloignées sur le même gestionnaire de files d'attente vous permet de centraliser vos activités de gestion.

## 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, le message d'événement inséré en file d'attente démarre 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 certaines tâches d'administration.

### Activation et désactivation des événements

Pour activer et désactiver les événements, vous devez spécifier des valeurs appropriées pour les attributs du gestionnaire de files d'attente ou de la file d'attente, ou des deux (selon le type d'événement). Pour cela, vous utilisez l'une des procédures suivantes :

- Commandes MQSC. Pour plus d'informations, reportez-vous au manuel *MQSeries - Guide de référence des commandes*.
- Commandes PCF pour les gestionnaires de files d'attente sous UNIX, OpenVMS et OS/2. Pour plus d'informations, consultez le manuel *MQSeries Programmable System Management*.
- Commandes MQAI. Pour plus d'informations, reportez-vous au manuel *MQSeries Administration Interface Programming Guide and Reference*.

L'activation d'un événement varie selon la catégorie de l'événement :

- Les événements Gestionnaire de files d'attente sont activés par la définition d'attributs sur le gestionnaire de files d'attente.
- Les événements Performances doivent être activés globalement sur le gestionnaire de files d'attente. Si cela n'est pas fait, aucun événement Performances ne peut intervenir. Vous activez ensuite les événements Performances spécifiques en définissant les attributs de file d'attente appropriés. Vous devez également identifier les conditions qui provoquent l'événement, telles que l'atteinte du seuil de longueur maximale pour une file d'attente.
- Les événements Canal interviennent automatiquement. Il n'est pas nécessaire de les activer. Si vous ne souhaitez pas les gérer, vous pouvez désactiver les demandes MQPUT vers la file d'attente d'événements Canal.

### 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 de structures de données décrites dans le manuel *MQSeries Programmable System Management*.



---

## Chapitre 10. Prise en charge transactionnelle

Le *Guide de programmation d'applications* propose une présentation complète de ce sujet qui est brièvement traité dans ce chapitre.

Un programme d'application 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.

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 (GFA) 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 monophasé.

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.

En résumé, les ressources d'un GFA peuvent être mises à jour dans le cadre d'unités de travail locales ou globales :

### Unité de travail locale

Utilisez des unités de travail locales lorsque les seules ressources à mettre à jour sont celles du gestionnaire de files d'attente MQSeries. Les opérations sont validées à l'aide de l'instruction MQCMIT ou annulées à l'aide de l'instruction MQBACK.

### Unité de travail globale

Utilisez des unités de travail globales lorsque vous avez également besoin de mettre à jour les ressources des gestionnaires de bases de données (GBD) conformes à la norme XA. Dans ce cas, la coordination peut être interne ou externe au GFA.

### Coordination des gestionnaires de files d'attente

Les unités de travail globales sont démarrées à l'aide de l'instruction MQBEGIN, puis validées avec MQCMIT ou annulées avec MQBACK. Un processus de validation à deux phases est utilisé pour demander, dans un premier temps, à tous les gestionnaires de ressources conformes à la norme XA tels que Oracle de se préparer à la validation. Ensuite, et seulement si tous les gestionnaires sont prêts, il leur est demandé de valider les transactions. Si l'un des gestionnaires signale qu'il n'est pas en mesure de valider, tous devront annuler les opérations.

## Prise en charge transactionnelle

### Coordination externe

Avec ce système, la coordination est réalisée par un gestionnaire de transactions conforme à la norme XA tel que IBM CICS, Transarc Encina, ou BEA Tuxedo. Les unités de travail sont démarrées et validées sous le contrôle de ce gestionnaire. Les instructions MQBEGIN, MQCMIT et MQBACK ne sont pas disponibles.

Le présent chapitre décrit comment activer la prise en charge des unités de travail globales (aucune activation explicite n'est nécessaire pour les unités de travail locales).

Il contient les sections suivantes :

- «Coordination des bases de données»
- «Configuration pour Oracle» à la page 125
- «Tâches d'administration» à la page 130

---

## Coordination des bases de données

Lorsque le gestionnaire de files d'attente (GFA) coordonne lui-même des unités de travail globales, il devient possible d'intégrer des mises à jour de bases de données au sein d'unités de travail MQ. En d'autres termes, vous pouvez écrire des applications combinant MQI et SQL, et les instructions MQCMIT et MQBACK permettent de valider ou d'annuler les modifications apportées à la fois aux files d'attente et aux bases de données.

Pour ce faire, le GFA utilise un protocole de validation à deux phases. Lorsqu'une unité de travail est sur le point d'être validée, le GFA demande tout d'abord à chaque gestionnaire de bases de données (GBD) concerné s'il est prêt à valider ses mises à jour. Ensuite, la validation des mises à jour des files d'attente et des bases de données n'est effectuée que si la totalité des intervenants, y compris le GFA lui-même, sont prêts. Si l'un des intervenants signale qu'il ne parvient pas à préparer ses opérations, l'unité de travail est annulée.

Une fonction de reprise intégrale est fournie si le GFA perd le contact avec l'un des GBD au cours du processus de validation. Si un GBD devient indisponible tandis qu'il est en attente de validation (c'est-à-dire qu'il a reçu la demande de préparation, mais pas encore l'ordre de valider ou d'annuler), le GFA se souvient du résultat de l'unité de travail jusqu'à ce que son traitement ait abouti. De même, si le GFA s'arrête alors qu'il reste des opérations de validation incomplètes, il s'en souviendra après son redémarrage.

Il faut employer l'instruction MQI MQBEGIN pour indiquer les unités de travail qui doivent aussi impliquer des mises à jour de base de données. Le *Guide de programmation d'applications* décrit les programmes exemples qui effectuent des mises à jour de MQSeries et de bases de données au sein de la même unité de travail.

Le GFA communique avec les GBD à l'aide de l'interface XA, comme décrit dans l'ouvrage *X/Open Distributed Transaction Processing: The XA Specification* (ISBN 1 872630 24 3). Cela signifie que le gestionnaire de files d'attente peut dialoguer avec tous les gestionnaires de bases de données qui sont conformes à la norme XA.

Le tableau 5 à la page 123 présente les GBD conformes à la norme XA pris en charge par les produits MQSeries Version 5.1.

Tableau 5. Bases de données relationnelles conformes à la norme XA

Produit MQSeries	DB2	Oracle	Sybase
MQSeries pour AIX	Oui	Oui	Oui
MQSeries pour HP-UX	Oui	Oui	Non
MQSeries pour Sun Solaris	Oui	Oui	Oui
MQSeries pour Compaq OpenVMS Alpha	Non	Oui	Non
MQSeries pour Windows NT et Windows 2000	Oui	Non	Oui

## Restrictions

Les restrictions suivantes s'appliquent à la coordination des bases de données :

- La possibilité de coordonner des mises à jour de bases de données au sein d'unités de travail MQSeries **n'est pas** prise en charge dans une application cliente MQI.
- Les mises à jour MQI et celles des bases doivent être effectuées sur la même machine serveur (celle du GFA).
- Le serveur de bases de données peut être installé sur une machine différente de celle du serveur de files d'attente. Le cas échéant, vous devez accéder à la base de données via une fonction conforme à la norme XA cliente fournie par le GBD lui-même.
- Bien que le GFA soit conforme à la norme XA, il n'est pas possible de configurer plusieurs gestionnaires de ce type en tant qu'intervenants dans des unités de travail globales. En effet, le système ne prend en charge qu'une seule connexion à la fois.

## Connexions aux bases de données

Une application qui établit une connexion standard au GFA est associée à une unité d'exécution dans un agent GFA local distinct. Lorsque l'application émet un appel MQBEGIN, elle et le processus agent doivent se connecter aux bases de données qui doivent participer à l'unité de travail. Les connexions aux bases sont maintenues tant que l'application reste connectée au GFA. Cette considération est importante si la base ne prend en charge qu'un nombre limité d'utilisateurs ou de connexions.

Pour réduire le nombre des connexions, l'application a la possibilité de lancer un appel MQCONNX afin de demander une édition de liens raccourcie. Dans ce cas, l'application et l'agent GFA local deviennent un unique processus et peuvent, par conséquent, utiliser la même connexion avec les bases de données. Avant d'employer cette méthode d'édition de liens, consultez le manuel *Guide de programmation d'applications* pour connaître les restrictions qui s'y appliquent.

## Configuration des gestionnaires de bases de données

Il est nécessaire d'effectuer plusieurs opérations avant qu'un GBD soit en mesure d'intervenir dans des unités de travail globales coordonnées par le GFA :

1. Création d'un *fichier de commutation XA*<sup>1</sup> pour le GBD.
2. Définition du GBD dans le fichier de configuration du gestionnaire de files d'attente `qm.ini`.

---

1. Un fichier de commutation XA est un objet chargé de façon dynamique qui permet au GFA et au GBD de communiquer.

## Configuration des gestionnaires de bases de données

Divers paramètres (y compris le nom du fichier de commutation) doivent être définis dans le fichier `qm.ini`.

### Création de fichiers de commutation

Les instructions nécessaires à la création de fichiers de commutation figurent dans la section «Création du fichier de commutation Oracle» à la page 126.

Pour plus d'informations sur la procédure d'installation, consultez la documentation MQSeries appropriée.

Tous les modules source exemples servant à générer les fichiers de commutation contiennent une seule fonction, appelée MQStart. Lorsque le fichier de commutation est chargé, le GFA appelle cette fonction pour obtenir l'adresse d'une structure nommée commutateur XA. Le fichier de commutation est lié à une bibliothèque fournie par le GBD, ce qui permet à MQSeries d'appeler ce dernier.

Les modules source exemples servant à générer les fichiers de commutation pour Oracle sont `oraswit.c`.

### Définition des gestionnaires de bases de données

Une fois que vous avez créé un fichier de commutation pour votre GBD, vous devez le déclarer à votre GFA. Pour cela, vous utilisez le fichier `qm.ini` dans la strophe `XAResourceManager` du gestionnaire de file d'attente.

Il convient d'ajouter une strophe `XAResourceManager` pour chaque gestionnaire de bases de données que votre gestionnaire de files d'attente doit coordonner.

Les attributs de la strophe `XAResourceManager` se présentent comme suit.

#### **Name=nom**

Chaîne choisie par l'utilisateur, qui définit l'instance du gestionnaire de bases de données.

Ce nom est obligatoire et peut comporter jusqu'à 31 caractères. Il doit être unique. Vous pouvez simplement indiquer le nom du gestionnaire ; cependant, afin de maintenir l'unicité du nom dans des configurations plus complexes, il est conseillé d'y inclure également le nom de la base de données à mettre à jour (par exemple).

Le nom choisi doit être significatif, car le GFA s'en sert pour faire référence à cette instance de GBD, à la fois dans les messages et dans le résultat de la commande `dspmqrn`.

**Ne modifiez pas cet attribut par la suite.** Vous trouverez des informations sur la modification de la configuration à la section «Modification des fichiers de configuration» à la page 133.

#### **SwitchFile=nom**

Nom complet du fichier de commutation XA du GBD. Cet attribut est obligatoire.

#### **XAOpenString=chaîne**

Chaîne de données transmise lors d'appels au point d'entrée `xa_open` du gestionnaire de base de données. Le format de cette chaîne dépend du gestionnaire employé, mais il doit généralement indiquer le nom de la base de données à mettre à jour.

Cet attribut est facultatif ; s'il est omis, le système le remplace par une chaîne vide.



## Configuration des gestionnaires de bases de données

### **XACloseString=chaîne**

Chaîne de données transmise lors d'appels au point d'entrée `xa_close` du gestionnaire de bases de données. Le format de cette chaîne dépend du gestionnaire employé.

Cet attribut est facultatif ; s'il est omis, le système le remplace par une chaîne vide.

### **ThreadOfControl=THREAD | PROCESS**

La valeur *ThreadOfControl* peut être `THREAD` ou `PROCESS`. Le GFA s'en sert à des fins de sérialisation.

Si le GBD accepte les "connexions multiples", la valeur de *ThreadOfControl* peut être `THREAD`, et le GFA peut appeler le GBD à partir de plusieurs unités d'exécution simultanées.

Dans le cas contraire, la valeur de *ThreadOfControl* doit être `PROCESS`. Le GFA met en série tous ses appels vers le GBD, de sorte qu'un processus spécifique ne puisse émettre qu'un seul appel à la fois.

Pour plus de détails sur ces attributs, reportez vous à la section «Strophe `XAResourceManager`» à la page 184

La section «Configuration pour Oracle» décrit plus en détail les tâches de configuration de `MQSeries` avec chacun des GBD pris en charge.

---

## Configuration pour Oracle

Vous devez effectuer les opérations suivantes :

- Vérification de la version d'Oracle et application des correctifs (si ce n'est déjà fait).
- Vérification du paramétrage des variables d'environnement ;
- Activation de la prise en charge de XA Oracle ;
- Création du fichier de commutation Oracle ;
- Ajout des données de configuration `XAResourceManager` dans le fichier `qm.ini` ;
- Modification éventuelle des paramètres de configuration d'Oracle.

## Versions minimales requises pour Oracle

La versions minimale d'Oracle prise en charge sur `OpenVMS` est 8.1.6.

## Vérification du paramétrage des variables d'environnement

Vous devez vérifier que vos variables d'environnement Oracle sont définies pour les processus du GFA et pour ceux de vos applications. Les variables suivantes, notamment, doivent toujours être définies **avant** de démarrer le GFA :

### **ORACLE\_HOME**

Répertoire de travail d'Oracle

### **ORACLE\_SID**

Numéro d'identification SID d'Oracle

## Activation de la prise en charge de XA Oracle

Vous devez vérifier que la prise en charge de XA Oracle est activée. Il faut notamment qu'une bibliothèque partagée ait été installée, ce qui se produit lors de l'installation de la bibliothèque XA Oracle.

## Configuration pour Oracle

Cette bibliothèque est créée automatiquement lors de l'installation de Oracle8. S'il s'avère nécessaire de reconstituer la bibliothèque, consultez le *Oracle8 Administrator's Reference* approprié à votre plate-forme.

### Création du fichier de commutation Oracle

La méthode la plus simple pour créer le fichier de commutation Oracle est d'utiliser le fichier exemple fourni. Le code source ayant servi à générer le fichier de commutation Oracle est représenté à la figure 11.

```
#include <cmqc.h>
#include "xa.h"

extern struct xa_switch_t xaosw;

struct xa_switch_t * MQENTRY MQStart(void)
{
    return(&xaosw);
}
```

Figure 11. Code source du fichier de commutation Oracle, oraswit.c

Le fichier d'en-tête xa.h livré avec MQSeries est installé dans le même répertoire que oraswit.c.

### Création du fichier de commutation Oracle sur les systèmes OpenVMS

Pour créer le fichier de commutation Oracle sur un système OpenVMS, oraswit.c doit être compilé puis lié à la librairie client Oracle "oraclient\_v816.exe".

1. Créez le répertoire dans lequel le fichier de commutation Oracle, oraswit, sera compilé.
2. Copiez les fichiers suivants depuis mqs\_examples:[xatm] dans ce nouveau répertoire :
  - xa.h
  - oraswit.c
3. Compilez le fichier source copié (oraswit.c). Par exemple :

```
$ cc oraswit0.c
```

4. Générez le fichier de commutation :

```
$ link/share oraswit0.obj, sys$input/options
ora_root:[util]oraclient_v816.exe/share
SYMBOL_VECTOR=(MQStart=PROCEDURE)
```

## Configuration pour Oracle

5. La procédure MQStart est chargée dynamiquement lors de l'exécution à partir de l'image générée. Il est donc nécessaire de définir un nom logique dans la table de noms logiques système pour pointer vers le fichier généré sans extension ".exe", ou de copier le fichier sur sys\$share. Par exemple, si l'emplacement du fichier créé est disk\$a\_device:[a\_directory]oraswit0.exe, utilisez l'une des commandes suivantes :

```
$ define/sys/exec oraswit0 disk$a_device:[a_directory]oraswit0
```

ou

```
$ copy disk$a_device:[a_directory]oraswit0.exe sys$share:oraswit0.exe
```

## Configuration pour Oracle

### Ajout des données de configuration XAResourceManager pour Oracle

Vous devez maintenant modifier le fichier de configuration du gestionnaire de file d'attente `qm.ini` pour définir Oracle comme intervenant dans des unités de travail globales. Pour cela, ajoutez une strophe XAResourceManager dotée des attributs suivants :

#### **Name=nom**

Cet attribut est obligatoire. Choisissez un nom approprié à cet intervenant. Il est conseillé d'y inclure le nom de la base de données à mettre à jour.

#### **SwitchFile=nom**

Cet attribut est obligatoire. Nom complet du fichier de commutation Oracle.

#### **XAOpenString=chaîne**

Le format de la chaîne d'ouverture XA pour Oracle est le suivant :

```
Oracle_XA+Acc=P//|P/nomUtilisateur/motPasse
      +SesTm=duréeInactivitéSession
      [+DB=nomBaseDonnées]
      [+GPwd=P/motPasseGroupe]
      [+LogDir=répJournal]
      [+MaxCur=maxCurseurOuverts]
      [+SqlNet=chaîneConnexion]
```

Où :

**Acc=** Option obligatoire servant à spécifier des informations d'accès. **P//** indique qu'aucune information utilisateur ou de mot de passe n'est explicitement fournie, et qu'il faut utiliser la forme **ops\$login**. **P/nomUtilisateur/motPasse** indique un ID utilisateur reconnu par ORACLE, avec le mot de passe correspondant.

#### **SesTm=**

Option obligatoire servant à définir la durée maximale, en secondes, pendant laquelle une transaction peut rester inactive avant d'être automatiquement supprimée par le système. L'unité de temps est la seconde.

**DB=** Indique le nom de la base de données ; ce paramètre est utilisé par les précompilateurs Oracle. Cette option n'est requise que si des applications spécifient explicitement le nom de la base (c'est-à-dire qu'elles utilisent une clause AT dans leurs instructions SQL).

#### **GPwd=**

Sert à spécifier le mot de passe correspondant au groupe de sécurité du serveur. Les groupes de sécurité offrent un niveau de protection supplémentaire pour différentes applications sollicitant la même instance d'ORACLE. La valeur par défaut est un groupe défini par ORACLE.

#### **LogDir=**

Indique le nom du répertoire, sur une machine locale, dans lequel les erreurs de la bibliothèque XA Oracle et les informations de suivi doivent être consignées. Si aucune valeur n'est spécifiée, le système utilise le répertoire en cours. Assurez-vous que l'utilisateur `mqm` possède l'accès en écriture à ce répertoire.

#### **MaxCur=**

Sert à définir le nombre de curseurs à allouer lors de l'ouverture de la

base de données. Ce paramètre est identique à l'option maxopencursors du précompilateur.

### SqlNet=

Définit la chaîne de connexion SQL\*Net qui sert à se connecter au système. Il peut s'agir d'une chaîne SQL\*Net version 1 ou 2, ou d'un alias SQL\*Net version 2. Cette option est requise lorsque vous configurez Oracle sur une machine distincte de celle du GFA.

Pour plus de détails, reportez-vous aux ouvrages *Oracle8 Server Application Developer's Guide* (référence A54642-01).

### XACloseString=chaîne

Oracle n'a pas besoin de chaîne de fermeture XA.

### ThreadOfControl=THREAD | PROCESS

Il est inutile de préciser ce paramètre sur les plateformes MQSeries pour Compaq OpenVMS.

Pour plus de détails sur ces attributs, reportez-vous à la section «Strophe XAResourceManager» à la page 184.

Dans la figure 12, la base à mettre à jour s'appelle MQBankDB. Notez qu'il est recommandé d'ajouter l'option LogDir dans la chaîne d'ouverture XA, afin que toutes les informations d'erreur et de suivi soient consignées au même endroit. Nous supposons que le fichier de commutation Oracle a été copié dans le répertoire sys\$share après sa création.

```
XAResourceManager:
  Name=Oracle MQBankDB
  SwitchFile=sys$share:oraswit0
  XAOpenString=Oracle_XA+Acc=P/jim/tiger+SesTm=35+LogDir=/tmp/ora.log+DB=MQBankDB
```

Figure 12. Exemple d'entrée XAResourceManager pour Oracle

## Modification des paramètres de configuration d'Oracle

Lorsqu'ils se connectent à Oracle, le GFA et les applications utilisateur se servent de l'ID utilisateur spécifié dans la chaîne d'ouverture XA.

- **Définition des droits d'accès à la base de données** L'ID utilisateur Oracle spécifié dans la chaîne d'ouverture doit détenir les droits d'accès à la vue DBA\_PENDING\_TRANSACTIONS.

Les droits requis peuvent être octroyés à l'aide de la commande suivante, où IDutilisateur représente l'ID de l'utilisateur auquel l'autorisation d'accès est donnée.

```
grant select on DBA_PENDING_TRANSACTIONS to IDutilisateur;
```

Pour plus d'informations sur la sécurité, reportez-vous au «Chapitre 7. Protection des objets MQSeries» à la page 81.

### Tâches d'administration

Au cours du fonctionnement normal, seule une administration minimale est nécessaire une fois que vous avez terminé la configuration. Les tâches d'administration sont facilitées du fait que le GFA tolère l'éventuelle indisponibilité des GBD dont il a besoin. Cela signifie notamment que :

- Le GFA peut être démarré à tout moment sans qu'il soit utile de lancer chaque GBD au préalable.
- Il n'est pas nécessaire d'arrêter et de redémarrer le GFA si l'un des GBD devient indisponible.

Ainsi, vous pouvez démarrer et arrêter le GFA indépendamment des GBD, et inversement si ces derniers acceptent ce type d'opération.

Chaque fois que le contact entre le GFA et un GBD est perdu, les deux gestionnaires doivent se resynchroniser dès qu'ils redeviennent disponibles.

La resynchronisation est le processus grâce auquel sont résolues les unités de travail en attente de validation auxquelles participe cette base de données. En général, elle se produit de façon automatique, sans qu'une intervention de l'utilisateur soit nécessaire. Le GFA demande au GBD la liste des unités de travail dans lesquelles il est en attente de validation. Ensuite, il lui demande de valider ou d'annuler chacune de ces unités.

Lorsque le GFA s'arrête, il doit se resynchroniser avec toutes les instances de GBD lors de son redémarrage. Lorsqu'un GBD devient indisponible, lui seul doit être resynchronisé dès que le GFA s'aperçoit qu'il est de nouveau accessible.

Le GFA tente automatiquement de reprendre contact avec un GBD indisponible quand de nouvelles unités de travail globales sont lancées. Il est également possible d'utiliser la commande **rsvmqtrn** pour forcer la résolution de toutes les unités en attente.

### Unités de travail en attente de validation

Si un GBD perd le contact avec le GFA après avoir reçu l'ordre de se préparer (instruction PREPARE), il peut se retrouver avec des unités de travail en attente de validation. Tant que le GBD n'a pas reçu le résultat de l'instruction COMMIT ou ROLLBACK de la part du GFA, il doit maintenir les verrous de bases de données associés aux mises à jour.

Ces verrous empêchant les autres applications de mettre à jour, voire de lire, certains enregistrements de la base de données, la resynchronisation doit s'effectuer dès que possible.

Si, pour une quelconque raison, vous ne pouvez pas attendre que le GFA se resynchronise automatiquement avec la base, vous pouvez utiliser les fonctions offertes par le GBD pour valider ou annuler manuellement les mises à jour. Cette méthode, appelée une prise de décision *heuristique* ne doit être employée qu'en dernier recours, car elle peut compromettre l'intégrité des données ; vous pouvez vous retrouver à valider les mises à jour des bases de données alors que tous les autres intervenants annulent, ou inversement.

Il est largement préférable de redémarrer le GFA, ou d'utiliser la commande **rsvmqtrn** lorsque la base de données a été relancée, afin de déclencher la resynchronisation automatique.

## Utilisation de la commande `dspmqrn`

Lorsqu'un GBD est indisponible, il est possible d'employer la commande `dspmqrn` pour vérifier l'état des unités de travail en attente auxquelles participe cette base de données.

Dès qu'un GBD devient indisponible, avant le processus de validation à deux phases, toutes les unités de travail en cours auxquelles il participait sont annulées. Lors de son redémarrage, le GBD lui-même annule les mises à jour en cours.

La commande `dspmqrn` n'affiche que les unités de travail dans lesquelles un ou plusieurs intervenants sont en attente de validation, nécessitant une instruction COMMIT ou ROLLBACK de la part du GFA.

Pour chacune de ces unités, l'état des intervenants est indiqué. Si une unité n'a pas mis à jour les ressources d'un gestionnaire de ressources particulier, elle n'est pas affichée.

Par rapport à une unité de travail en attente de validation, un gestionnaire de ressources peut présenter l'état :

<b>Préparé</b>	Le gestionnaire est prêt à valider ses mises à jour.
<b>Validé</b>	Le gestionnaire a validé ses mises à jour.
<b>Annulé</b>	Le gestionnaire a annulé ses mises à jour.
<b>Intervenant</b>	Le gestionnaire est un intervenant, mais il n'a pas préparé, validé ou annulé ses mises à jour.

Notez que, lorsque le GFA est redémarré, il ne se souvient plus de l'état de chacun des intervenants. S'il est redémarré mais qu'il ne parvient pas à se connecter à un GBD, les unités de travail en attente de validation dans lesquelles ce GBD intervenait ne sont pas résolues au cours du redémarrage. Dans ce cas, le GBD est considéré comme étant dans l'état *préparé* jusqu'à la prochaine resynchronisation.

Chaque fois que la commande `dspmqrn` affiche une unité de travail en attente de validation, elle indique en premier tous les gestionnaires de ressources qui pourraient y participer. Elle attribue à chacun un identificateur unique, *RMID*, qui est utilisé à la place du *nom* du gestionnaire lors de l'indication de son état par rapport à une unité de travail.

La figure 13 montre le résultat de la commande suivante :

```
dspmqrn -m MY_QMGR
```

```
AMQ7107: Le gestionnaire de ressources 0 est MQSeries.
AMQ7107: Le gestionnaire de ressource 1 est Oracle MQBankDB
AMQ7107: Le gestionnaire de ressources 2 est Oracle MQFeeDB

AMQ7056: Transaction numéro 0,1.
      XID: formatID 5067085, gtrid_length 12, bqual_length 4
          gtrid [3291A5060000201374657374]
          bqual [00000001]
AMQ7105: Le gestionnaire de ressources 0 a validé.
AMQ7104: Le gestionnaire de ressources 1 s'est préparé.
AMQ7104: Le gestionnaire de ressources 2 s'est préparé.
```

Figure 13. Exemple de sortie de `dspmqrn`

## Tâches d'administration

La sortie de la figure 13 à la page 131 montre qu'il existe trois gestionnaires de ressources associés au gestionnaire de files d'attente. Le premier est le gestionnaire de ressources 0, c'est à dire le GFA lui-même. Les deux autres instances sont les bases de données Oracle MQBankDB et MQFeeDB Oracle.

L'exemple montre une seule unité de travail en attente de validation. Un message est émis pour les trois gestionnaires de ressources, ce qui signifie que les mises à jour ont été effectuées (dans l'unité de travail) sur le GFA et les deux bases Oracle.

Les mises à jour du GFA (gestionnaire de ressources) 0, ont été *validées*. Les mises à jour des bases de données Oracle sont dans l'état *préparé*, ce qui signifie que Oracle a dû devenir indisponible avant d'être appelé pour la validation relative à *MQBankDB* et *MQFeeDB*.

L'unité de travail en attente de validation possède un identificateur externe, appelé un *XID*. C'est cet identificateur que Oracle associe aux mises à jour.

## Utilisation de la commande `rsvmqtrn`

Le résultat indiqué dans la figure 13 à la page 131 montre une seule unité de travail pour laquelle la décision de validation n'a pas encore été transmise aux deux bases de données Oracle.

Pour pouvoir résoudre cette unité, le GFA et Oracle doivent se resynchroniser dès que Oracle redevient disponible. Chaque fois que le GFA détecte le lancement de nouvelles unités de travail, il en profite pour tenter de reprendre contact avec Oracle. Vous pouvez également forcer le GFA à se resynchroniser, à l'aide de la commande `rsvmqtrn`. Il est recommandé d'effectuer cette opération juste après le redémarrage de Oracle, de sorte que tous les verrous de base de données associés à l'unité de travail en attente de validation soient libérés dès que possible.

Pour ce faire, utilisez l'option `-a` qui demande au GFA de résoudre toutes les unités en attente. Dans l'exemple ci-après, Oracle a été redémarré pour que le GFA soit en mesure d'effectuer cette résolution :

```
rsvmqtrn -m MY_QMGR -a
```

Toutes les unités de travail en attente de validation ont été résolues.

## Résultats incohérents et erreurs

Bien que le GFA utilise un protocole de validation à deux phases, cela n'élimine pas complètement la possibilité de générer un résultat incohérent lors du traitement de certaines unités de travail. Ce phénomène survient lorsque certains intervenants valident leurs mises à jour, tandis que d'autres annulent les leurs.

Les implications des unités de travail présentant de tels résultats sont particulièrement importantes, car les ressources partagées ne sont plus dans un état cohérent.

Les résultats incohérents surviennent principalement lorsque des décisions heuristiques sont prises concernant des unités de travail, au lieu de laisser le GFA les résoudre lui-même.



## Tâches d'administration

Chaque fois que le GFA détecte des dommages heuristiques, il génère des informations FFST\* et documente l'incident dans ses journaux d'erreurs avec l'un des deux messages suivants :

- Si un GBD a annulé au lieu de valider ;

```
AMQ7606 Une transaction a été validée, mais un ou plusieurs
gestionnaires de ressources ont annulé.
```

- Si un GBD a validé au lieu d'annuler ;

```
AMQ7607 Une transaction a été annulée, mais un ou plusieurs
gestionnaires de ressources ont validé.
```

D'autres messages sont émis pour identifier les bases de données ayant subi des dommages heuristiques. Vous avez alors la responsabilité d'effectuer des opérations de récupération locales dans les bases affectées, afin de restaurer l'intégrité des données. Il s'agit d'une procédure complexe dans laquelle vous devez d'abord isoler la mise à jour qui a été validée ou annulée par erreur, puis défaire ou refaire manuellement la modification dans la base de données.

Les dommages dus à des erreurs logicielles sont beaucoup plus rares. Le cas échéant, le numéro de transaction des unités de travail affectées est signalé par le message AMQ7112. Il est possible que les intervenants soient dans un état incohérent.

```
rsvmqtrn -m MON_GFA

AMQ7107: Le gestionnaire de ressources 0 est MQSeries.
AMQ7107: Le gestionnaire de ressource 1 est Oracle MQBankDB
AMQ7107: Le gestionnaire de ressources 2 est Oracle MQFeedB

AMQ7112: La transaction numéro 0,1 a rencontré une erreur.
        XID: formatID 5067085, gtrid_length 12, bqual_length 4
            gtrid [3291A5060000201374657374]
            bqual [00000001]
AMQ7105: Le gestionnaire de ressources 0 a validé.
AMQ7104: Le gestionnaire de ressources 1 s'est préparé.
AMQ7104: Le gestionnaire de ressources 2 a annulé.
```

Figure 14. Exemple de sortie de `dspmqtrn` pour une transaction générant une erreur

Suite à un incident de ce type, le GFA ne tente jamais d'opération de reprise avant d'avoir redémarré. Par rapport à la figure 14, cela signifierait que les mises à jour du gestionnaire de ressources 1 (la base de données MQBankDB) seraient laissées dans l'état *préparé* même si la commande `rsvmqtrn` était émise afin de résoudre l'unité de travail.

## Modification des fichiers de configuration

Dès que le GFA a commencé à coordonner avec succès des unités de travail globales, vous devez être très prudent quant aux éventuelles modifications apportées aux strophes XAResourceManager du fichier `qm.ini`.

## Tâches d'administration

Si vous avez besoin de modifier le fichier `qm.ini` vous pouvez le faire à tout moment, mais les changements ne prendront effet qu'après le redémarrage du GFA. Par exemple, si vous voulez modifier la chaîne d'ouverture XA passée au GBD, vous devrez redémarrer le GFA pour que le changement soit appliqué.

Notez que, si vous supprimez une strophe `XAResourceManager`, vous ôtez toute possibilité au GFA de pouvoir se connecter au GBD concerné.

Vous ne devez *jamais* modifier l'attribut `Name` dans une quelconque strophe `XAResourceManager`. Pour le GFA, cet attribut identifie de façon unique cette instance de GBD. S'il est modifié, le GFA considère que l'instance de GBD a été supprimée et qu'une instance entièrement nouvelle a été ajoutée. Le GFA associe toujours à l'ancien *nom*, les unités de travail en attente, ce qui peut laisser la base de données dans un état d'attente de validation.

### Suppression d'instances de gestionnaire de bases de données

Si vous devez retirer de façon permanente une base de données ou un GBD de votre configuration, commencez par vous assurer que la base n'est pas en attente de validation. Effectuez cette vérification avant de redémarrer le GFA. La plupart des GBD offrent des commandes qui permettent de répertorier les transactions en attente de validation. S'il existe des transactions dans cet état, attendez d'abord que le GFA se soit resynchronisé avec le GBD avant de tenter de supprimer sa strophe `XAResourceManager`.

Si vous n'observez pas cette procédure (ou si elle échoue), le GFA se souviendra toujours de toutes les unités de travail en attente de validation auxquelles participent cette base de données. Le message d'avertissement AMQ7623 est émis à chaque redémarrage du GFA. Si vous n'avez pas l'intention de reconfigurer cette base de données avec le GFA, vous pouvez lui ordonner d'oublier ces transactions en attente à l'aide de l'option `-r` de la commande `rsvmqtrn`.

**Remarque :** Le GFA n'oubliera les transactions qu'une fois la synchronisation effectuée avec tous les intervenants.

Parfois, vous aurez besoin de supprimer une strophe `XAResourceManager` de façon temporaire uniquement. La meilleure méthode est de mettre la strophe en commentaire, afin de pouvoir la réinstaurer facilement par la suite. Vous pouvez envisager cette opération si des erreurs se produisent chaque fois que le GFA se connecte à une base de données ou à un GBD spécifique. Ce retrait temporaire de l'entrée `XAResourceManager` concernée permet au GFA de lancer des unités de travail globales impliquant tous les autres intervenants. Voici un exemple de strophe `XAResourceManager` mise en commentaire :

```
# Cette base de données a été temporairement supprimée
#XAResourceManager:
# Name=Oracle MQBankDB
# SwitchFile=sys$share:oraswit0
# XAOpenString=MQBankDB
```

Figure 15. Strophe `XAResourceManager` mise en commentaire

---

## Chapitre 11. Reprise et redémarrage

Un système de messagerie garantit que les messages entrés dans le système sont livrés à destination. Cela implique qu'il fournisse une méthode de suivi des messages sur le système, ainsi qu'une méthode de récupération des messages si le système échoue pour une raison quelconque.

MQSeries garantit que les messages ne sont pas perdus en gardant des enregistrements (journaux) des activités des gestionnaires de files d'attente qui traitent la réception, la transmission et la communication des messages. Il utilise ces journaux pour trois types de reprises :

1. *Reprise au redémarrage*, lorsque vous arrêtez MQSeries de façon planifiée.
2. *Reprise sur incident*, lorsque MQSeries est arrêté à la suite d'une erreur inattendue.
3. *Reconstitution du support*, pour restaurer les objets.

Dans tous les cas, 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 :

- «Assurance de récupération des messages (journalisation)»
- «Définition de points de contrôle – Garantie d'une reprise complète» à la page 138
- «Calcul de la taille des journaux» à la page 141
- «Gestion des journaux» à la page 142
- «Utilisation du journal pour la reprise» à la page 145
- «Protection des fichiers journaux MQSeries» à la page 147
- «Sauvegarde et restauration» à la page 148
- «Scénarios de reprise» à la page 149
- «Cliché du contenu du fichier journal par la commande `dmpmqlog`» à la page 151.

---

### Assurance de récupération des messages (journalisation)

MQSeries enregistre dans un journal toutes les modifications importantes apportées aux données contrôlées par le gestionnaire de files d'attente, y compris la création et la suppression d'objets, les mises à jour de tous les messages persistants, l'état des transactions, les modifications apportées aux attributs d'objets et les activités des canaux. Le journal contient par conséquent les informations dont vous avez besoin pour récupérer toutes les mises à jour de files d'attente de messages. Il vous permet de :

- garder des enregistrements des modifications du gestionnaire de files d'attente,
- garder des enregistrements des mises à jour des files d'attente, qui seront utilisés par la procédure de redémarrage,
- restaurer les données après un incident matériel ou logiciel.

### Aspect des journaux

Un journal MQSeries se compose de deux éléments :

1. un ou plusieurs fichiers de données,
2. un fichier de contrôle.

Les données sont enregistrées dans plusieurs fichiers journaux. Vous pouvez définir le nombre et la taille des fichiers journaux (comme indiqué dans le «Calcul de la taille des journaux» à la page 141), ou accepter les valeurs système par défaut de 3 fichiers.

Lorsque vous créez un gestionnaire de files d'attente, le nombre de fichiers journaux que vous définissez est le nombre de fichiers journaux *primaire* alloués. Si vous n'indiquez pas de nombre, la valeur par défaut est utilisée. Si vous n'avez pas modifié le chemin du journal, les fichiers journaux sont créés dans le répertoire :

```
MQS_ROOT:[MQM.LOG.QmName.ACTIVE]
```

MQSeries commence par utiliser ces fichiers journaux primaires, mais s'ils arrivent à saturation, il alloue des fichiers journaux *secondaires* de façon dynamique et les supprime lorsque la demande d'espace de journalisation diminue. Par défaut, il est possible d'allouer jusqu'à deux fichiers journaux secondaires pour une capacité disque supplémentaire de 8 Mo. Il est également possible de changer ce nombre par défaut. Consultez le «Chapitre 13. Configuration de MQSeries» à la page 173.

#### Fichier de contrôle de journal

Le fichier de contrôle du journal contient les informations nécessaires pour contrôler l'utilisation des fichiers journaux, c'est-à-dire leur taille et leur emplacement, le nom du fichier disponible suivant, etc.

**Remarque :** Vous devez vous assurer que les journaux créés lors du démarrage d'un gestionnaire de files d'attente sont suffisamment grands par rapport à la taille et au volume des messages que vos applications vont traiter. Le nombre de fichiers journaux et leur taille par défaut devront être modifiés pour répondre à vos besoins. Pour plus d'informations sur la modification des valeurs par défaut, consultez la section 141.

### Types de journalisation

Dans MQSeries, le nombre de fichiers requis pour la journalisation varie en fonction de la taille des fichiers, du nombre de messages reçus et de la longueur des messages. Il existe deux façons de gérer les enregistrements des activités du gestionnaire de files d'attente : la journalisation circulaire et la journalisation linéaire.

#### Journalisation circulaire

Utilisez la journalisation circulaire si vous voulez uniquement une reprise au redémarrage en utilisant le journal pour abandonner les transactions qui étaient en cours à l'arrêt du système.

La journalisation circulaire garde toutes les données de redémarrage dans une chaîne de fichiers journaux. La journalisation est effectuée dans le premier fichier de la chaîne, puis, une fois ce dernier plein, dans le suivant. La procédure se poursuit jusqu'à ce que tous les fichiers soient remplis. Elle retourne ensuite dans

le premier fichier de la chaîne et recommence. La procédure continue tant que le produit est en cours d'utilisation et de cette façon, vous n'êtes jamais à court de fichiers journaux.

Ceci est une explication simple de la journalisation circulaire, cependant, cette notion revêt des aspects plus complexes. Les entrées de journal nécessaires au redémarrage du gestionnaire de files d'attente sans perte de données sont conservées jusqu'à ce qu'elles ne soient plus nécessaires pour garantir la récupération des données du gestionnaire de files d'attente. Le mécanisme de libération des fichiers journaux pour réutilisation est décrit dans la section «Définition de points de contrôle – Garantie d'une reprise complète» à la page 138. Pour le moment, vous devez juste savoir que MQSeries utilise des fichiers journaux secondaires pour augmenter la capacité de journalisation si nécessaire.

### Journalisation linéaire

Utilisez la journalisation linéaire si vous voulez assurer la reprise au redémarrage et la reconstitution de support ou la reprise intégrale (nouvelle création des données perdues ou altérées par relecture du contenu du journal).

La journalisation linéaire conserve les données de journalisation dans une série continue de fichiers. L'espace disque n'est pas réutilisé, par conséquent, vous pouvez toujours extraire tout enregistrement consigné depuis la création du gestionnaire de files d'attente.

L'espace disque étant limité, vous devez envisager une forme d'archivage. Il incombe à l'administrateur de gérer l'espace disque de journalisation, en réutilisant ou en augmentant l'espace existant si nécessaire.

Le nombre de fichiers journaux utilisés pour la journalisation linéaire peut être très grand selon le flux de messages et l'âge de votre gestionnaire de files d'attente. Cependant, un certain nombre de fichiers sont dits actifs. Les fichiers actifs contiennent les entrées de journal requises pour redémarrer le gestionnaire de files d'attente. Le nombre de fichiers journaux actifs est généralement égal au nombre de fichiers journaux primaires défini dans les fichiers de configuration (pour plus de détails sur le nombre défini, reportez-vous à la section «Calcul de la taille des journaux» à la page 141).

L'événement clé qui contrôle si un fichier journal est considéré actif ou non est un *point de contrôle*. Un point de contrôle MQSeries est un groupe d'enregistrements de journal contenant les informations permettant le redémarrage du gestionnaire de files d'attente. Toute information précédemment enregistrée n'est pas nécessaire pour redémarrer le gestionnaire de files d'attente et peut donc être définie comme inactive (consultez la section «Définition de points de contrôle – Garantie d'une reprise complète» à la page 138 pour plus d'informations sur la définition de points de contrôle.)

Vous devez décider du moment où les fichiers journaux inactifs ne sont plus requis. Vous pouvez choisir de les archiver ou de les supprimer si vous n'en avez plus besoin. Pour plus de détails sur la disposition des fichiers journaux, reportez-vous à la section «Gestion des journaux» à la page 142.

Si un nouveau point de contrôle est enregistré dans le second fichier journal primaire (ou dans un fichier ultérieur), le premier fichier devient inactif et un nouveau fichier primaire est formaté et ajouté à la fin du pool primaire, ce qui restaure le nombre de fichiers primaires disponibles pour la journalisation. De cette façon, le pool de fichiers journaux primaires peut être considéré comme un

## Journalisation

ensemble actuel de fichiers dans une liste de fichiers journaux toujours croissante. Il incombe également à l'administrateur de gérer les fichiers inactifs selon les besoins du système.

Bien que des fichiers journaux secondaires soient définis pour la journalisation linéaire, il ne sont pas utilisés en mode de fonctionnement normal. Lorsque, probablement suite à des transactions de longue durée, il s'avère impossible de libérer un fichier du pool actif car il peut encore être nécessaire pour un redémarrage, des fichiers secondaires sont formatés et ajoutés au pool de fichiers journaux actifs.

Lorsque tous les fichiers secondaires disponibles sont utilisés, les demandes d'opérations supplémentaires nécessitant une activité de journalisation sont refusées et le message MQRC\_RESOURCE\_PROBLEM est renvoyé à l'application.

Les deux types de journalisation peuvent faire face à une coupure d'alimentation inattendue en supposant qu'il n'y ait pas d'incident matériel.

---

## Définition de points de contrôle – Garantie d'une reprise complète

Les mises à jour persistantes de files d'attente de messages se produisent en deux étapes. Tout d'abord, les enregistrements représentant la mise à jour sont écrits dans le journal, puis la file d'attente est mise à jour. Les fichiers journaux peuvent ainsi être plus à jour que les fichiers de la file d'attente. Pour s'assurer que la procédure de redémarrage commence à un point de cohérence, MQSeries utilise des points de contrôle. Un point de contrôle correspond à un moment où l'enregistrement décrit dans le journal est identique à celui de la file d'attente. Le point de contrôle lui-même consiste en une série d'enregistrements de journal nécessaires pour redémarrer le gestionnaire de files d'attente par exemple, l'état de toutes les transactions (unités de travail) actives au moment du point de contrôle.

Les points de contrôle sont générés automatiquement par MQSeries. à des moments définis : lorsque le gestionnaire de files d'attente démarre, à l'arrêt système, lorsque l'espace de journalisation diminue et après toutes les 1000 journalisations d'opération. Lorsque les files d'attente traitent d'autres messages, l'enregistrement de points de contrôle n'est plus conforme à l'état en cours des files d'attente.

Lorsque MQSeries est relancé, il localise le dernier enregistrement de point de contrôle du journal. Ces informations sont conservées dans le fichier de points de contrôle qui est mis à jour à la fin de chaque point de contrôle. L'enregistrement de point de contrôle représente le point de cohérence le plus récent entre le journal et les données. Les données de ce point de contrôle sont utilisées pour recréer les files d'attente telles qu'elles étaient au moment du point de contrôle. Lorsque les files d'attente sont recréées, la lecture du journal se poursuit afin qu'elles retrouvent l'état qui était le leur avant la panne système ou l'arrêt.

MQSeries insère des pointeurs internes à la tête et à la queue du journal. Il déplace le pointeur de tête au niveau du point de contrôle le plus récent nécessaire à la récupération des données des messages.

Les points de contrôle rendent la procédure de reprise plus efficace et permettent de contrôler la réutilisation des fichiers journaux primaires et secondaires.

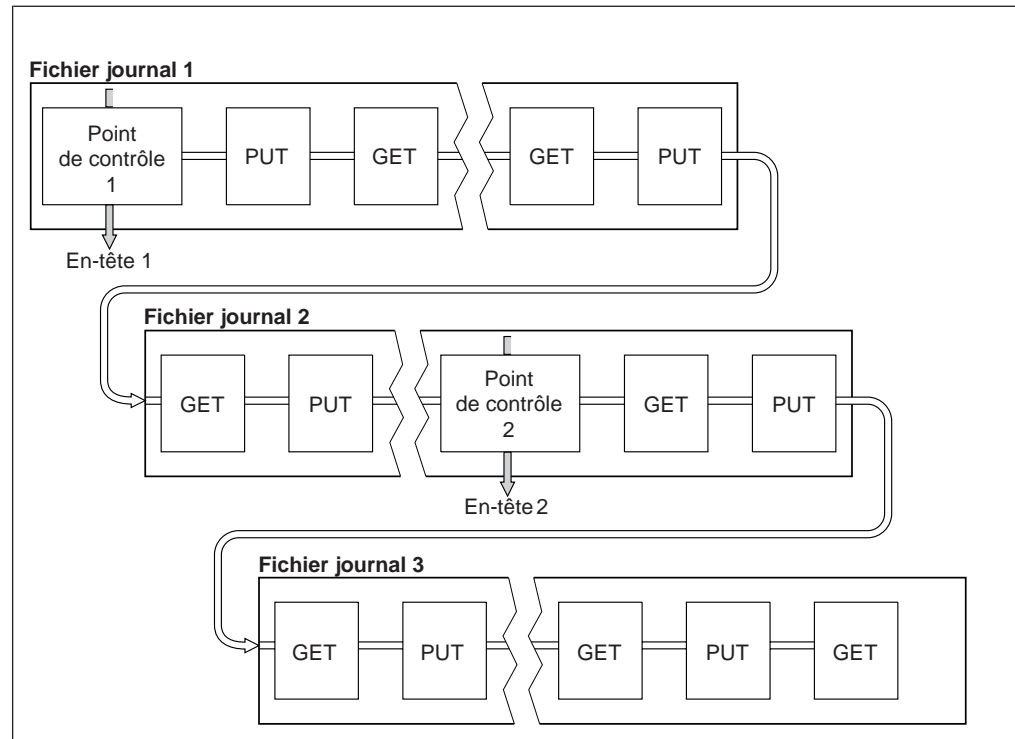


Figure 16. Définition des points de contrôle. Pour plus de simplicité, seules les extrémités des fichiers journaux sont représentées.

Dans la figure 16, les enregistrements précédant le dernier point de contrôle, Point de contrôle 2, ne sont plus requis par MQSeries. Les files d'attente peuvent être récupérées à partir des informations de point de contrôle et des entrées de journal suivantes. Dans le cas d'une journalisation circulaire, tous les fichiers libérés avant le point de contrôle peuvent être réutilisés. Dans le cas d'une journalisation linéaire, les fichiers journaux libérés n'ont plus à être accessibles en mode de fonctionnement normal et ils deviennent inactifs. Dans l'exemple, le pointeur de tête de la file d'attente est déplacé au niveau du dernier point de contrôle, Pt de ctrl 2, qui devient ensuite le nouveau pointeur de tête de la file d'attente, Tête 2. Le fichier journal 1 peut maintenant être réutilisé.



## Définition des points de contrôle

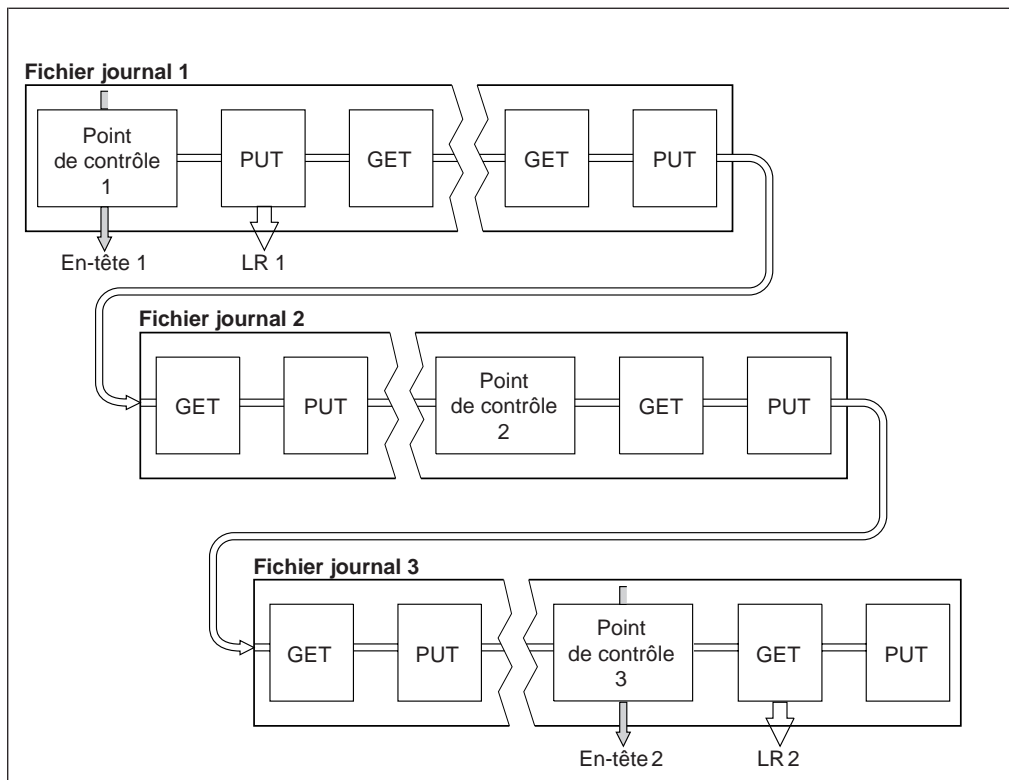


Figure 17. Définition de points de contrôle pour une transaction de longue durée. Pour plus de simplicité, seules les extrémités des fichiers journaux sont représentées.

La figure 17 montre comment une transaction de longue durée affecte la réutilisation des fichiers journaux. Dans l'exemple, une transaction de longue durée a entraîné l'enregistrement d'une nouvelle entrée (LR 1) dans le journal après le premier point de contrôle indiqué. La fin de la transaction (LR 2) ne se produit pas avant le troisième point de contrôle. A partir de LR 1, toutes les informations du journal sont retenues pour permettre la reprise éventuelle de cette transaction, jusqu'à ce qu'elle ait abouti.

Une fois que la transaction de longue durée a abouti (au niveau de LR 2), le pointeur de tête du journal est déplacé au niveau du point de contrôle 3, dernier point de contrôle consigné. Les fichiers contenant les enregistrements de journal précédant le point de contrôle 3 (Tête 2) ne sont plus nécessaires. Si vous utilisez la journalisation circulaire, l'espace disque peut être réutilisé.

Si les fichiers journaux primaires sont saturés avant la fin de la transaction de longue durée, des fichiers journaux secondaires sont utilisés pour éviter, si possible, une situation de complète saturation.

Lorsque le pointeur de tête du journal est déplacé et que vous utilisez la journalisation circulaire, les fichiers journaux primaires sont susceptibles d'être réutilisés, et l'enregistreur automatique réutilise le premier fichier primaire disponible une fois qu'il a rempli le fichier en cours. En revanche, si vous utilisez la journalisation linéaire, le pointeur de tête du journal est toujours déplacé vers le bas dans le pool actif et le premier fichier devient inactif. Un nouveau fichier primaire est formaté et ajouté en bas du pool en prévision de futures activités de journalisation.



## Calcul de la taille des journaux

Après avoir déterminé si le gestionnaire de files d'attente doit utiliser la journalisation circulaire ou linéaire, vous devez estimer la taille du journal. Cette dernière est déterminée par les paramètres de configuration suivants :

### LogFilePages

Taille de chaque fichier journal primaire et secondaire en unités de pages de 4 Ko

### LogPrimaryFiles

Nombre de fichiers journaux primaires préalloués.

### LogSecondaryFiles

Nombre de fichiers journaux secondaires pouvant être créés lorsque les journaux primaires sont saturés.

Le tableau 6 indique les quantités de données consignées par le gestionnaire pour diverses opérations. La plupart des opérations réalisées par le gestionnaire de files d'attente occupent un espace réduit du journal. Cependant, lorsqu'un message persistant est mis en file d'attente, l'intégralité du contenu du message doit être écrite dans le journal pour rendre la récupération du message possible. C'est pourquoi la taille du journal dépend en général du nombre et de la taille des messages persistants gérés par le gestionnaire de files d'attente.

Tableau 6. Tailles des journaux (toutes les valeurs sont approximatives)

Opération	Taille
Insertion d'un message persistant	750 octets + longueur du message. Si le message est long, il est divisé en segments de 15700 octets, avec un surcroît d'occupation de 300 octets pour chacun.
Extraction d'un message	260 octets
Point de synchronisation, validation	750 octets
Point de synchronisation, annulation	1000 octets + 12 octets par extraction ou insertion annulée
Création d'objet	1500 octets
Suppression d'un objet	300 octets
Modification d'attributs	1024 octets
Enregistrement d'images de support	800 octets + image. L'image est divisée en segments de 15700 octets avec un surcroît d'occupation de 300 octets pour chacun.
Point de contrôle	750 octets + 200 octets pour chaque unité de travail active. Il est possible que des données supplémentaires soient consignées pour toute insertion ou extraction non validée ayant été mise en mémoire tampon pour des raisons de performances.

### Remarques :

1. Le nombre de fichiers journaux primaires et secondaires peut être modifié à chaque démarrage du gestionnaire de files d'attente.
2. La taille des fichiers journaux ne peut pas être modifiée et doit être fixée **avant** que le gestionnaire de files d'attente soit créé.

## Définition des points de contrôle

3. Le nombre de fichiers journaux primaires et leur taille déterminent l'espace de journalisation qui est préalloué lors de la création du gestionnaire de files d'attente. Il est conseillé d'organiser cet espace en créant un petit nombre de fichiers journaux de grande taille plutôt qu'un grand nombre de petits fichiers journaux.
4. Le nombre total de fichiers journaux primaires et secondaires ne peut pas dépasser 63, ce qui, en présence de transactions longues, limite l'espace de journalisation maximal pouvant être mis à la disposition du gestionnaire de files d'attente pour la reprise sur redémarrage. L'espace de journalisation dont le gestionnaire peut avoir besoin pour la reconstitution de supports n'entre pas dans cette limite.
5. Dans le cas de la journalisation *circulaire*, le gestionnaire réutilise l'espace du journal primaire. C'est-à-dire que la taille réelle du journal du gestionnaire peut être inférieure à votre estimation. Le gestionnaire de files d'attente allouera, dans une certaine limite, un fichier journal secondaire en cas de saturation du journal et d'indisponibilité du journal primaire suivant.
6. Les fichiers journaux primaires deviennent réutilisables lors d'un point de contrôle. Le gestionnaire de files d'attente tient compte à la fois de l'espace des journaux primaires et secondaires avant d'effectuer un point de contrôle en raison d'un manque d'espace.

Si vous ne définissez pas un nombre de journaux primaires supérieur au nombre de journaux secondaires, le gestionnaire de files d'attente peut allouer des fichiers journaux secondaires avant d'effectuer un point de contrôle. Ceci permet de réutiliser les fichiers journaux primaires.

---

## Gestion des journaux

Au fil du temps, certains enregistrements de journal écrits deviennent inutiles au redémarrage du gestionnaire de files d'attente, et ce dernier nécessite que de l'espace soit libéré dans les fichiers journaux. Cette activité est transparente pour l'utilisateur et vous ne voyez généralement pas la quantité d'espace disque utilisé diminuer car l'espace alloué est rapidement réutilisé.

Parmi les enregistrements de journal, seuls ceux écrits depuis le début du dernier point de contrôle effectué et ceux écrits par des transactions actives sont nécessaires pour redémarrer le gestionnaire de files d'attente. Le journal peut ainsi devenir saturé s'il n'y a pas eu de point de contrôle depuis longtemps, ou s'il y a longtemps qu'une transaction de longue durée a écrit un enregistrement de journal. Le gestionnaire de files d'attente essaie de prendre des points de contrôle suffisamment souvent pour éviter le premier cas.

Lorsqu'une transaction de longue durée sature le journal, les tentatives d'écriture d'enregistrements de journal échouent et certains appels MQI renvoient le message MQRC\_RESOURCE\_PROBLEM. (Un certain espace est réservé pour valider ou annuler toutes les transactions en cours ; par conséquent, MQCMIT et MQBACK ne peuvent pas échouer.)

Le gestionnaire de files d'attente annule les transactions qui consomment trop d'espace de journalisation. Une application dont la transaction est annulée de cette manière est incapable d'exécuter des opérations MQPUT ou MQGET ultérieures indiquant un point de synchronisation dans la même transaction. Une telle tentative d'insertion ou d'extraction de message au niveau du point de synchronisation renvoie le message MQRC\_BACKED\_OUT. L'application peut ensuite émettre MQCMIT, qui renvoie MQRC\_BACKED\_OUT, ou émettre MQBACK pour démarrer une nouvelle transaction. Lorsque la transaction

consommant trop d'espace de journalisation est annulée, son espace est libéré et le gestionnaire de files d'attente reprend son fonctionnement normal.

Si le journal arrive à saturation, un message d'avertissement est envoyé (AMQ7463). En outre, si le journal arrive à saturation parce qu'une transaction de longue durée a empêché la libération d'espace, le message AMQ7465 est émis.

Enfin, si les enregistrements sont écrits dans le journal plus rapidement que ne peuvent les traiter les processus de maintenance asynchrone, le message AMQ7466 est émis. Si vous recevez ce message, vous devez augmenter le nombre de fichiers journaux ou réduire la quantité de données traitées par le gestionnaire de files d'attente.

### Que se passe-t-il lorsqu'un disque est saturé ?

Le module de journalisation du gestionnaire de files d'attente peut faire face à un disque saturé et à des fichiers journaux saturés. Si le disque contenant le journal arrive à saturation, le gestionnaire de files d'attente émet le message AMQ6708 et un enregistrement d'erreur est consigné.

Les fichiers journaux sont créés à la taille maximale, plutôt que d'être agrandis au fur et à mesure que des enregistrements y sont écrits. Cela signifie que MQSeries ne peut être à court d'espace disque que lorsqu'il crée un nouveau fichier et non lorsqu'il écrit un enregistrement dans le journal. MQSeries connaît toujours la quantité d'espace disponible dans les fichiers journaux existants et il gère l'espace des fichiers en en tenant compte.

Si l'unité contenant les fichiers journaux arrive à saturation, vous pouvez peut-être libérer de l'espace disque. Si vous utilisez un journal linéaire, vous pouvez peut-être copier certains fichiers journaux inactifs du répertoire de journalisation sur une autre unité. Si cela ne suffit pas, vérifiez la configuration du journal dans le fichier de configuration du gestionnaire de files d'attente. Vous pouvez peut-être réduire le nombre de fichiers journaux primaires ou secondaires, de sorte que le journal ne dépasse pas l'espace disponible. Notez qu'il est impossible de modifier la taille des fichiers journaux d'un gestionnaire de files d'attente existant. Le gestionnaire de files d'attente suppose que tous les fichiers journaux ont la même taille.

### Gestion des fichiers journaux

Si vous utilisez la journalisation circulaire, assurez-vous qu'il y a suffisamment d'espace pour contenir les fichiers journaux. Faites-le lorsque vous configurez votre système (reportez-vous aux sections «Strophe LogDefaults» à la page 178 et «Strophe Log» à la page 182). La quantité d'espace disque utilisée par le journal, y compris l'espace nécessaire pour les fichiers secondaires créés si nécessaire, est limitée à la taille configurée du disque.

Si vous utilisez un journal linéaire, les fichiers journaux sont ajoutés de façon continue au fur et à mesure de la journalisation des données, et la quantité d'espace disque utilisée augmente au fil du temps. Si la quantité de données consignées est importante, l'espace disque est rapidement consommé par les nouveaux fichiers journaux.

## Gestion des journaux

Au fil du temps, les anciens fichiers journaux de journalisation linéaire ne sont plus requis pour redémarrer le gestionnaire de files d'attente ou pour restaurer des objets endommagés. Périodiquement, le gestionnaire de files d'attente émet deux messages pour indiquer quels sont les fichiers journaux requis :

- Le message AMQ7467 donne le nom du plus ancien fichier journal nécessaire pour redémarrer le gestionnaire de files d'attente. Ce fichier journal et tous les fichiers ultérieurs doivent être disponibles lors du redémarrage du gestionnaire de files d'attente.
- Le message AMQ7468 donne le nom du plus ancien fichier journal nécessaire pour effectuer une reconstitution de support.

Tous les fichiers journaux antérieurs à ceux-ci n'ont plus à être en ligne. Vous pouvez les copier sur un support d'archivage tel qu'une bande, en vue d'une reprise sur incident, et les supprimer du répertoire de journalisation actif. Tous les fichiers journaux nécessaires pour une reprise sur incident mais pas pour le redémarrage peuvent également être transférés sur un support d'archivage.

Si l'un des fichiers journaux nécessaires est introuvable, le message opérateur AMQ6767 est émis. Faites en sorte que le fichier journal indiqué dans le message et tous les fichiers journaux suivants soient disponibles pour le gestionnaire de files d'attente, puis faites une nouvelle tentative.

**Remarque :** Lorsque vous effectuez une reconstitution de support, tous les fichiers journaux requis doivent être disponibles en même temps dans le répertoire des fichiers journaux. Assurez-vous d'enregistrer régulièrement des images de support des objets à restaurer, afin d'éviter de manquer d'espace disque pour contenir tous les fichiers journaux requis.

### Emplacement des fichiers journaux

Lorsque vous choisissez un emplacement pour vos fichiers journaux, rappelez-vous que le fonctionnement de vos applications sera gravement compromis si MQSeries ne peut pas formater un nouveau journal à cause d'un manque d'espace disque.

Si vous utilisez un journal circulaire, vérifiez qu'il y a au moins suffisamment d'espace sur l'unité pour les fichiers journaux primaires configurés. Vous devez également laisser de l'espace pour au moins un fichier journal secondaire, qui sera nécessaire si le journal doit s'agrandir.

Si vous utilisez un journal linéaire, vous devez prévoir considérablement plus d'espace. En effet, l'espace consommé par le journal augmente de façon continue au fur et à mesure de la journalisation des données.

Dans l'idéal, les fichiers journaux doivent être placés sur une unité de disque distincte de celle des données du gestionnaire de files d'attente, ce qui s'avère positif en terme de performances. Il est également possible de placer les fichiers journaux sur plusieurs unités de disque dans un environnement miroir, ce qui protège contre une éventuelle panne de l'unité contenant le journal. Sans écriture miroir, vous pouvez être obligé de revenir à la dernière copie de sauvegarde de votre système MQSeries.

## Utilisation du journal pour la reprise

Vos données peuvent être altérées de plusieurs manières. MQSeries pour Compaq OpenVMS vous donne des solutions de reprise en cas :

- d'objet de données altéré,
- de coupure d'alimentation du système,
- d'incident de communication,
- de volume de journalisation altéré.

La présente section indique comment les journaux sont utilisés pour permettre la reprise.

### Reprise sur incident

MQSeries permet la reprise sur incident de communication et après une coupure d'alimentation. En outre, la reprise est parfois possible après d'autres types d'incidents, tels que la suppression par inadvertance d'un fichier.

En cas d'incident de communication, les messages restent simplement en file d'attente jusqu'à ce qu'ils soient récupérés par une application de réception. Si le message est en cours de transmission, il reste dans la file d'attente de transmission jusqu'à ce que l'opération aboutisse. Pour effectuer une reprise sur incident de communication, il suffit normalement de redémarrer les canaux à l'aide de la liaison qui a échoué.

En cas de coupure d'alimentation, lorsque le gestionnaire de files d'attente est redémarré, MQSeries restaure les files d'attente dans l'état qui était le leur au moment de l'incident, ce qui garantit qu'aucun message persistant n'est perdu. Les messages non persistants sont supprimés.

Il peut arriver qu'un objet MQSeries devienne inutilisable, par exemple, suite à des dégâts provoqués par inadvertance. Vous devez alors restaurer tout ou partie de votre système. L'action requise dépend du moment où l'incident est détecté, des objets altérés et de la méthode de journalisation sélectionnée (prise en charge de la reconstitution de support ou pas).

### Reconstitution de support

La reconstitution de support consiste à recréer des objets à partir d'informations enregistrées dans un journal linéaire. Cette fonction n'est pas prise en charge par la consignation circulaire. Par exemple, si un fichier objet est supprimé par inadvertance ou s'il devient inutilisable pour toute autre raison, il est possible d'utiliser la reconstitution de support pour le recréer. Les informations du journal nécessaires à la reconstitution de support d'un objet constituent une *image de support*. Les images de support peuvent être enregistrées manuellement à l'aide de la commande **rcdmqimg** ou automatiquement dans certains cas.

Une image de support est une série d'enregistrements de journal contenant une image d'un objet à partir de laquelle l'objet lui-même peut être recréé.

Le premier enregistrement de journal requis pour recréer un objet est appelé *enregistrement de reconstitution de support*; il s'agit du début de la dernière image de support de l'objet. L'enregistrement de reconstitution de support de chaque objet est l'une des informations enregistrées au cours d'un point de contrôle.

## Utilisation du journal

Lors de la recréation d'un objet à partir de son image support, il est également nécessaire de relire tous les enregistrements de journal décrivant les mises à jour apportées à l'objet depuis la dernière image.

Imaginez par exemple une file d'attente locale dont une image a été enregistrée avant l'insertion d'un message persistant. Pour recréer la dernière image de l'objet, il est nécessaire de relire les entrées de journal ayant enregistré l'insertion du message en file d'attente, tout comme de relire l'image elle-même.

Lorsqu'un objet est créé, les enregistrements de journal écrits contiennent suffisamment d'informations pour le recréer entièrement. Ces enregistrements constituent la première image de support de l'objet. Par la suite, les images de support sont enregistrées automatiquement par le gestionnaire de files d'attente :

- lorsque des images de tous les objets du processus et des files d'attente non locales sont enregistrées, à chaque arrêt système.
- lorsque des images de files d'attente locales vides sont enregistrées.

Les images de support peuvent être enregistrées manuellement à l'aide de la commande **rcdmqimg** décrite à la section «rcdmqimg (enregistrement d'une image de support)» à la page 282.

### Reconstitution d'images de support

MQSeries recrée automatiquement certains objets à partir de leur image de support s'il détecte une altération. Cela s'applique en particulier aux objets dont une altération est détectée lors du démarrage normal du gestionnaire de files d'attente. Si une transaction n'avait pas abouti au moment du dernier arrêt système du gestionnaire de files d'attente, toute file d'attente affectée est également recrée automatiquement afin de terminer l'opération de démarrage.

Vous devez reconstituer les autres objets manuellement, à l'aide de la commande **rcrmqobj**. Cette commande relit les enregistrements du journal pour recréer l'objet MQSeries. L'objet est recréé à partir de la dernière image trouvée dans le journal, ainsi que de tous les événements applicables consignés entre le moment où l'image a été sauvegardée et celui où la commande de reconstitution a été émise. Si un objet MQSeries est altéré, les seules actions correctes possibles sont la suppression ou la reconstitution à l'aide de cette méthode. Notez cependant que les messages non persistants ne peuvent pas être récupérés de cette façon.

Reportez-vous à la section «rcrmqobj (recréation d'un objet)» à la page 284 pour plus d'informations sur la commande **rcrmqobj**.

Rappelez-vous que le fichier journal contenant l'enregistrement de reconstitution de support et tous les fichiers journaux ultérieurs doivent être disponibles dans le répertoire de fichiers journaux lors de la tentative de reconstitution de support d'un objet. Si l'un des fichiers requis est introuvable, le message opérateur AMQ6767 est émis et l'opération de reconstitution de support échoue. Si vous n'enregistrez pas régulièrement des images de support des objets que vous pouvez vouloir recréer, vous risquez de ne pas avoir suffisamment d'espace disque pour contenir tous les fichiers journaux requis pour recréer un objet.

## Reconstitution des objets altérés lors du démarrage

Si le gestionnaire de files d'attente détecte un objet altéré lors du démarrage, l'action qu'il entreprend dépend du type d'objet et de la prise en charge ou non de la reconstitution de support par le gestionnaire de files d'attente.



Si l'objet gestionnaire de files d'attente est altéré, le gestionnaire de files d'attente ne peut pas démarrer sauf s'il peut reconstituer l'objet. Si le gestionnaire de files d'attente est configuré avec un journal linéaire et prend par conséquent en charge la reconstitution de support, MQSeries essaie automatiquement de recréer l'objet MQSeries à partir de ses images de support. Si la méthode de journalisation sélectionnée ne prend pas en charge la reconstitution de support, vous pouvez soit restaurer une copie de sauvegarde du gestionnaire de files d'attente, soit supprimer ce dernier.

Si des transactions étaient actives lorsque le gestionnaire de files d'attente s'est arrêté, les files d'attente locales contenant les messages persistants non validés insérés dans ou extraits de ces transactions sont également nécessaires pour que le démarrage du gestionnaire de files d'attente aboutisse. Si certaines de ces files d'attente locales sont altérées et que le gestionnaire de files d'attente prend en charge la reconstitution de support, il tente automatiquement de les recréer à partir de leurs images de support. Si ces files d'attente ne peuvent pas toutes être reconstituées, MQSeries ne peut pas démarrer.

Si des files d'attente locales altérées contenant des messages non validés sont détectées lors de la procédure de démarrage d'un gestionnaire de files d'attente qui ne prend pas en charge la reconstitution de support, elles sont marquées comme des objets altérés et les messages non validés qu'elles contiennent ne sont pas pris en compte. Il est en effet impossible d'effectuer une reconstitution de support d'objets altérés sur ce type de gestionnaire de files d'attente et la seule action admise est la suppression. Le message AMQ7472 est émis pour signaler tout incident.

### Reconstitution des objets altérés à d'autres moments

La reconstitution de support d'objets n'est automatique qu'au moment du démarrage. En dehors de cette période, lorsqu'un objet altéré est détecté, le message opérateur AMQ7472 est émis et la plupart des opérations utilisant cet objet échouent. Si l'objet gestionnaire de files d'attente est altéré après le démarrage du gestionnaire, le gestionnaire de files d'attente exécute un arrêt système forcé. Lorsqu'un objet a été altéré, vous pouvez le supprimer ou, si le gestionnaire de files d'attente utilise un journal linéaire, tenter de le reconstituer à partir de son image de support à l'aide de la commande `rcrmqobj` (pour plus de détails, reportez-vous à la section «rcrmqobj (recréation d'un objet)» à la page 284).

---

## Protection des fichiers journaux MQSeries

Il est important de ne pas supprimer de fichiers journaux manuellement lorsqu'un gestionnaire de files d'attente MQSeries est en cours d'exécution. Si un utilisateur supprime par inadvertance (ou malveillance) les fichiers journaux requis par un gestionnaire de files d'attente pour redémarrer, MQSeries n'émet pas de messages d'erreur et continue de traiter les données, y compris les messages persistants. Le gestionnaire de files d'attente s'arrêtera normalement, mais ne redémarrera pas. La reconstitution de support pour les messages devient alors impossible.

Tout utilisateur possédant le droit de supprimer des journaux en cours d'utilisation a également le droit de supprimer d'autres ressources de gestionnaire importantes (fichiers de droits d'accès, de files d'attente, catalogue des objets, et fichiers exécutables MQSeries). Il peut en conséquence endommager (par inexpérience ou intentionnellement) un gestionnaire de files d'attente en cours d'exécution ou en sommeil sans que MQSeries ne puisse s'en protéger.

## Utilisation du journal

Il convient d'être extrêmement prudent lorsque vous accordez les droits d'accès de super utilisateur ou MQM.

---

## Sauvegarde et restauration

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. Cependant, la durée de vie des données des messages étant souvent courte, vous pouvez choisir de ne pas faire de copies de sauvegarde.

### 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 répertoires dans lesquels le gestionnaire de files d'attente place ses données et ses fichiers journaux.

Vous pouvez utiliser les informations des fichiers de configuration pour déterminer quels sont ces répertoires. Pour plus d'informations, reportez-vous au «Chapitre 13. Configuration de MQSeries» à la page 173.

**Remarque :** Vous pouvez éprouver certaines difficultés à comprendre les noms qui apparaissent dans le répertoire. En effet, les noms sont 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 «Signification des noms de fichiers MQSeries» à la page 23.

3. Faites des copies de toutes les données du gestionnaire de files d'attente et des répertoires et sous-répertoires de journalisation.  
Vérifiez que vous n'oubliez aucun fichier, en particulier le fichier de contrôle du journal et les fichiers de configuration. Certains répertoires peuvent être vides, mais ils seront tous requis si vous restaurez ultérieurement la copie de sauvegarde ; par conséquent, il est recommandé de les sauvegarder également.
4. Vous devez également vous assurer que vous préservez la propriété des fichiers. Pour ce faire, utilisez la commande **BACKUP** et le paramètre **/BY\_OWNER**.

### 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 répertoires dans lesquels le gestionnaire de files d'attente place ses données et ses fichiers journaux. Ces informations se trouvent dans le fichier de configuration.
3. Videz les répertoires dans lesquels vous allez placer les données sauvegardées.



4. Copiez les données du gestionnaire de files d'attente et les fichiers journaux sauvegardés aux emplacements appropriés.

Contrôlez la structure de répertoires obtenue pour vérifier que tous les répertoires requis sont présents.

Pour plus d'informations sur les répertoires et sous-répertoires de MQSeries, reportez-vous à l'«Annexe C. Structure de répertoires» à la page 321.

Vérifiez que vous disposez du fichier de contrôle du journal et des fichiers journaux. Vérifiez également 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.

**Remarque :** Même si les données du gestionnaire de files d'attente et les fichiers journaux se trouvent dans des répertoires différents, vous devez sauvegarder et restaurer les répertoires en même temps. Si les données du gestionnaire de files d'attente et les fichiers journaux n'ont pas le même âge, l'état du gestionnaire de files d'attente est incorrect et ce dernier ne démarrera probablement pas. S'il démarre, vos données seront certainement altérées.

---

## Scénarios de reprise

Cette section présente un certain nombre d'incidents possibles et les méthodes de reprise correspondantes.

### Incidents d'unité de disque

Des incidents peuvent se produire sur une unité de disque contenant les données du gestionnaire de files d'attente, le journal ou les deux. Ces incidents comprennent la perte ou l'altération de données. Les trois cas diffèrent uniquement quant aux données qui survivent, le cas échéant.

Dans *tous* les cas, vous devez commencer par contrôler la structure de répertoires et réparer les dégâts éventuellement détectés. Si vous perdez des données du gestionnaire de files d'attente, l'arborescence de ce dernier risque d'avoir été altérée. Dans ce cas, vous devez recréer l'arborescence manuellement avant d'essayer de redémarrer le gestionnaire de files d'attente. Après avoir recherché les éventuels dégâts structurels, vous pouvez entreprendre un certain nombre de mesures, en fonction du type de journalisation que vous utilisez.

- **En cas de dégâts majeurs au niveau de la structure de répertoires ou de dégâts au niveau du journal**, supprimez tous les anciens fichiers jusqu'au niveau `NomGestFiles` y compris les fichiers de configuration, le journal et le répertoire du gestionnaire de files d'attente, restaurez la dernière copie de sauvegarde et tentez de redémarrer le gestionnaire de files d'attente.
- **Dans le cas d'une journalisation linéaire avec reconstitution de support**, assurez-vous que la structure de répertoires est intacte et essayez de redémarrer le gestionnaire de files d'attente. Si le gestionnaire de files d'attente ne redémarre pas, restaurez une copie de sauvegarde. Si le gestionnaire redémarre, vérifiez à l'aide des commandes MQSC qu'aucun objet n'a été endommagé. Le cas échéant, reconstituez-les à l'aide de la commande `rcrmqobj`. Exemple :

## Scénarios de reprise

```
rcrmqobj -m NomGestFiles -t * *
```

où `NomGestFiles` est le gestionnaire de files d'attente reconstitué. `-t * *` indique que tous les objets de tout type seront reconstitués. Si seulement un ou deux objets altérés ont été détectés, vous pouvez définir ces objets par nom et par type.

**Remarque :** Ces commandes ne s'appliquent pas aux canaux.

- **Dans le cas d'une journalisation linéaire avec reconstitution de support et un journal non altéré**, vous pouvez restaurer une copie de sauvegarde des données du gestionnaire de files d'attente en gardant les fichiers journaux et le fichier de contrôle du journal existants. Le démarrage du gestionnaire de files d'attente entraîne l'application des modifications à partir du journal pour que le gestionnaire de files d'attente revienne à l'état qui était le sien au moment où l'incident s'est produit.

Cette méthode repose sur deux conditions. Tout d'abord, il est vital que le fichier de points de contrôle soit restauré comme faisant partie des données du gestionnaire de files d'attente. Ce fichier contient les informations déterminant la quantité de données du journal nécessaire pour obtenir un gestionnaire de files d'attente cohérent.

Ensuite, vous devez disposer dans le répertoire de journalisation du plus ancien fichier journal requis pour démarrer le gestionnaire de files d'attente au moment du lancement, ainsi que tous les fichiers journaux ultérieurs.

Si c'est impossible, vous devez restaurer une copie de sauvegarde des données du gestionnaire de files d'attente du journal, les deux copies de sauvegarde ayant été effectuées au même moment.

- **Dans le cas d'une journalisation circulaire ou linéaire sans reconstitution de support**, vous devez restaurer le gestionnaire de files d'attente à partir de la dernière copie de sauvegarde dont vous disposez. Après avoir restauré cette copie de sauvegarde, redémarrez le gestionnaire de files d'attente et vérifiez s'il n'y a pas d'objets altérés. Cependant, étant donné que vous ne disposez pas de la reconstitution de support, vous devez trouver d'autres moyens pour recréer les éventuels objets altérés.

## Objet gestionnaire de files d'attente altéré

Si l'objet gestionnaire de files d'attente a été altéré en cours de fonctionnement normal, le gestionnaire de files d'attente exécute un arrêt système forcé. Dans ces circonstances, il y a deux modes de reprise selon le type de journalisation utilisé :

- **Dans le cas d'une journalisation linéaire uniquement**, supprimez manuellement le fichier contenant l'objet altéré et redémarrez le gestionnaire de files d'attente. La reconstitution de support de l'objet altéré est automatique.
- **Dans le cas d'une journalisation circulaire ou linéaire**, restaurez la dernière copie de sauvegarde des données du gestionnaire de files d'attente et du journal, puis redémarrez le gestionnaire de files d'attente.

## Simple objet altéré

Si un seul objet a été altéré en cours de fonctionnement normal, il y a deux modes de reprise selon le type de journalisation utilisé :

- **Dans le cas d'une journalisation linéaire**, recréez l'objet à partir de son image de support.

- Dans le cas d'une journalisation circulaire, restaurez la dernière copie de sauvegarde des données du gestionnaire de files d'attente et du journal, puis redémarrez le gestionnaire de files d'attente.

### Echec de reconstitution de support automatique

Si une file d'attente locale nécessaire au démarrage du gestionnaire de files d'attente en mode de journalisation linéaire est altérée et que la reconstitution de support automatique échoue, restaurez la dernière copie de sauvegarde des données du gestionnaire de files d'attente et du journal, puis redémarrez le gestionnaire de files d'attente.

---

### Cliché du contenu du fichier journal par la commande `dmpmqlog`

La commande `dmpmqlog` peut être utilisée pour cliquer le contenu du journal du gestionnaire de files d'attente. Par défaut, tous les enregistrements de journal actifs sont cliqués, c'est-à-dire que la commande commence à cliquer à partir de la tête du journal. En temps normal, la commande commence à cliquer à partir du dernier point de contrôle achevé.

Le journal peut être cliqué uniquement lorsque le gestionnaire de files d'attente est inactif. Puisque le gestionnaire de files d'attente prend un point de contrôle au cours de l'arrêt, la partie active du journal contient généralement un petit nombre d'enregistrements de journal. Cependant, la commande `dmpmqlog` peut être définie de manière à cliquer plus d'enregistrements de journal, en utilisant l'une des options suivantes pour modifier le point de départ du cliché :

- L'option la plus simple consiste à commencer le cliché à partir de la *base* du journal. Il s'agit du premier enregistrement de journal se trouvant dans le fichier journal qui contient la tête du journal. La quantité de données supplémentaires cliquées dans ce cas varie en fonction du positionnement de la tête du journal dans le fichier journal. S'il se situe près du début du fichier journal, seul un petit nombre de données sera cliqué. S'il est proche de la fin du fichier journal, une quantité beaucoup plus importante de données sera cliquée.
- Une autre option permet de spécifier le point de départ du cliché comme étant un enregistrement de journal déterminé. Chaque enregistrement de journal est identifié par un *numéro d'ordre de journal (LSN)* unique. Dans le cas de la journalisation circulaire, cet enregistrement de journal de départ ne peut pas être antérieur à la base du journal (cette restriction ne s'applique pas aux journaux linéaires). Il se peut que vous ayez à rétablir les fichiers journaux inactifs avant d'exécuter cette commande. Pour cette option, un numéro d'ordre de journal correct doit être indiqué en tant que point de départ. Il doit être extrait de la sortie de la commande `dmpmqlog`.

Par exemple, avec la journalisation linéaire, vous pouvez indiquer le `nextlsn` à partir de la dernière sortie de la commande `dmpmqlog`. "Next LSN" apparaît dans Log File Header et indique le numéro d'ordre de journal du prochain enregistrement de journal à écrire. Il peut donc être utilisé comme point de départ pour formater tous les enregistrements de journal écrits depuis le dernier cliché du journal.

- La troisième option concerne uniquement les journaux linéaires. Le programme de cliché peut être défini de sorte qu'il démarre le formatage des enregistrements de journal à partir d'une plage de fichier journal donné. Dans ce cas, le programme de cliché du journal s'attend à trouver ce fichier journal, et chacun des suivants, dans le même répertoire que les fichiers journaux actifs.

## Utilisation de la commande **dmpmqlog**

Cette option ne s'applique pas aux journaux circulaires, car dans ces cas, le programme de cliché du journal ne peut pas accéder aux enregistrements de journal antérieurs à la base du journal.

La sortie de la commande **dmpmqlog** consiste en un Log File Header et une série d'enregistrements de journal formatés. Le gestionnaire de files d'attente utilise plusieurs enregistrements de journal pour enregistrer les modifications apportées à ses données.

Certaines informations formatées n'ont qu'un usage interne. La liste suivante inclut les enregistrements de journal les plus utiles :

### Log File Header

Chaque journal dispose d'un en-tête de fichier journal, qui est toujours formaté en premier par la commande **dmpmqlog**. Il comprend les zones suivantes :

<i>logactive</i>	Nombre de domaines de journal primaire.
<i>loginactive</i>	Nombre de domaines de journal secondaire.
<i>logsize</i>	Nombre de pages de 4 Ko par domaine.
<i>baselsn</i>	Premier numéro d'ordre de journal du domaine de journal contenant la tête du journal.
<i>nextlsn</i>	Numéro d'ordre de journal du prochain enregistrement de journal à écrire.
<i>headlsn</i>	Numéro d'ordre de journal de l'enregistrement de journal se trouvant à la tête du journal.
<i>tailsn</i>	Numéro d'ordre de journal identifiant la position de queue du journal.
<i>hflag1</i>	Indique si le journal est de type CIRCULAR ou LOG RETAIN (linéaire).
<i>HeadExtentID</i>	Domaine de journal contenant la tête du journal.

### Log Record Header

Chaque enregistrement du journal comporte un en-tête fixe contenant les informations suivantes :

<i>LSN</i>	Numéro d'ordre de journal.
<i>LogRecdType</i>	Type de l'enregistrement de journal.
<i>XTranid</i>	Identificateur de transaction associé à cet enregistrement de journal (le cas échéant).  Un <i>TranType</i> de type MQI indique une transaction MQ uniquement. Un <i>TranType</i> de XA indique une transaction impliquant d'autres gestionnaires de ressources. Les mises à jour faisant partie de la même unité de travail ont le même <i>XTranid</i> .
<i>QueueName</i>	File d'attente associée à cet enregistrement de journal (le cas échéant).
<i>Qid</i>	Identificateur interne unique pour la file d'attente.
<i>PrevLSN</i>	Numéro d'ordre de journal de l'enregistrement de journal précédent au sein de la même transaction (le cas échéant).

### Start Queue Manager

Consigne le démarrage du gestionnaire de files d'attente.

*StartDate* Date du démarrage du gestionnaire de files d'attente.

*StartTime* Heure du démarrage du gestionnaire de files d'attente.

### Stop Queue Manager

Consigne l'arrêt du gestionnaire de files d'attente.

*StopDate* Date de l'arrêt du gestionnaire de files d'attente.

*StopTime* Heure de l'arrêt du gestionnaire de files d'attente.

*ForceFlag* Type d'arrêt utilisé.

### Start Checkpoint

Indique le début d'un point de contrôle de gestionnaire de files d'attente.

### End Checkpoint

Indique la fin d'un point de contrôle de gestionnaire de files d'attente.

*ChkPtLSN* Numéro d'ordre de journal de l'enregistrement de journal ayant démarré le point de contrôle.

### Put Message

Consigne la mise en file d'attente d'un message persistant. Si le message a été mis en file d'attente lors d'un point de synchronisation, l'en-tête de l'enregistrement de journal contient un *XTranid* autre que zéro. Le reste de l'enregistrement contient :

*SpcIndex* Un identificateur pour le message mis en file d'attente. Il peut être utilisé pour être mis en correspondance avec le MQGET utilisé pour extraire ce message de la file d'attente. Dans ce cas, un enregistrement de journal *Get Message* ultérieur contenant le même *QueueName* et *SpcIndex* peut être trouvé. A ce stade, l'identificateur *SpcIndex* peut être réutilisé pour un autre message d'insertion dans cette file d'attente.

*Data* Dans le cliché en hexadécimal pour cet enregistrement de journal se trouvent diverses données suivies par le descripteur du message (eyecatcher) et les données propres du message.

### Put Part

Les messages persistants trop importants pour un seul enregistrement de journal sont consignés sous la forme d'un enregistrement *Put Message* suivi de plusieurs enregistrements de journal *Put Part*.

*Data* Continue les données du message à l'endroit où le précédent enregistrement de journal s'est arrêté.

### Get Message

Seuls les extractions de messages persistants sont consignées. Si le message a été mis en file d'attente lors d'un point de synchronisation, l'en-tête de l'enregistrement de journal contient un *XTranid* autre que zéro. Le reste de l'enregistrement contient :

*SpcIndex* Identifie le message extrait de la file d'attente. L'enregistrement de journal *Put Message* le plus récent contenant le même *QueueName* et le même *SpcIndex* identifie le message extrait.

*QPriority* Priorité du message extrait de la file d'attente.

### Start Transaction

Indique le début d'une nouvelle transaction. Un *TranType* de type MQI

## Utilisation de la commande `dmpmqlog`

indique une transaction MQ uniquement. Un `TranType` de type XA indique une transaction impliquant d'autres gestionnaires de ressources. Toutes les mises à jour effectuées par cette transaction auront le même *XTranid*.

### Prepare Transaction

Indique que le gestionnaire de files d'attente est préparé à valider les mises à jour associées au *XTranid*. Cet enregistrement de journal est écrit dans le cadre d'une validation à deux phases impliquant d'autres gestionnaires de ressources.

### Commit Transaction

Indique le gestionnaire de files d'attente a validé toutes les mises à jour effectuées par une transaction.

### Rollback Transaction

Cet enregistrement de journal indique l'intention du gestionnaire de files d'attente d'annuler une transaction.

### End Transaction

Cet enregistrement de journal indique la fin d'une transaction annulée.

### Transaction Table

Cet enregistrement est écrit lors d'un point de synchronisation. Il enregistre l'état de chaque transaction ayant effectué des mises à jour permanentes. Pour chaque transaction, les informations suivantes sont enregistrées :

<i>XTranid</i>	Identificateur de transaction.
<i>FirstLSN</i>	Numéro d'ordre de journal du premier enregistrement de journal associé à la transaction.
<i>LastLSN</i>	Numéro d'ordre de journal du dernier enregistrement de journal associé à la transaction.

### Transaction Participants

Cet enregistrement de journal est écrit par le composant XA Transaction Manager du gestionnaire de files d'attente. Il enregistre les gestionnaires de ressources externes participant aux transactions. Pour chaque participant, les éléments suivants sont enregistrés :

<i>RMName</i>	Nom du gestionnaire de ressources.
<i>RMIId</i>	Identificateur du gestionnaire de ressources. Il est également consigné dans les enregistrements de journal <i>Transaction Prepared</i> ultérieurs qui enregistrent les transactions globales auxquelles le gestionnaire de ressources participe.
<i>SwitchFile</i>	Fichier de commutation pour ce gestionnaire de ressources.
<i>XAOpenString</i>	Chaîne d'ouverture XA pour ce gestionnaire de ressources.
<i>XACloseString</i>	Chaîne de fermeture XA pour ce gestionnaire de ressources.

### Transaction Prepared

Cet enregistrement de journal est écrit par le composant XA Transaction Manager du gestionnaire de files d'attente. Il indique que la préparation de la transaction globale spécifiée a abouti. Chacun des gestionnaires de ressources participant recevra l'ordre de valider. Le *RMIId* de chaque gestionnaire de ressources préparé est enregistré dans l'enregistrement de journal. Si le gestionnaire de files d'attente participe à la transaction, un *Participant Entry* avec un *RMIId* de zéro sera présent.

### Transaction Forget

Cet enregistrement de journal est écrit par le composant XA Transaction

## Utilisation de la commande **dmpmqlog**

Manager du gestionnaire de files d'attente. Il suit l'enregistrement de journal *Transaction Prepared* lorsque la décision de validation a été transmise à chaque participant.

### **Purge Queue**

Consigne le fait que tous les messages sur une file d'attente ont été purgés (par exemple, à l'aide de la commande `RUNMQSC CLEAR`).

### **Queue Attributes**

Consigne l'initialisation ou la modification des attributs d'une file d'attente.

### **Create Object**

Consigne la création d'un objet MQSeries

*ObjName*            Nom de l'objet créé.

*UserId*            ID utilisateur effectuant la création.

### **Delete Object**

Consigne la création d'un objet MQSeries

*ObjName*            Nom de l'objet supprimé.

La figure 18 à la page 156 montre un exemple de sortie d'une commande **dmpmqlog**. Le cliché, qui a démarré à partir du numéro d'ordre de journal d'un enregistrement de journal spécifique, a été produit à l'aide des commandes suivantes :

```
dmpmqlog -m "testqm" -s 0:0:0:44162
```

## Utilisation de la commande dmpmqlog

```
AMQ7701: DMPMQLLOG command is starting.
LOG FILE HEADER
*****

counter1 . . . . : 23                counter2 . . . . : 23
FormatVersion . . : 2                logtype . . . . : 10
logactive . . . . : 3                loginactive . . . : 2
logsize . . . . . : 1024             pages
base1sn . . . . . : <0:0:0:0>
next1sn . . . . . : <0:0:0:60864>
lowtran1sn . . . . : <0:0:0:0>
minbuff1sn . . . . : <0:0:0:58120>
head1sn . . . . . : <0:0:0:58120>
tail1sn . . . . . : <0:0:0:60863>
logfilepath . . . : ""
hflag1 . . . . . : 1
                    -> CONSISTENT
                    -> CIRCULAR
HeadExtentID . . . : 1                LastEID . . . . . : 846249092
LogId . . . . . : 846249061           LastCommit . . . : 0
FirstArchNum . . . : 4294967295       LastArchNum . . . : 4294967295
nextArcFile . . . : 4294967295        firstRecFile . . . : 4294967295
firstDl1teFile . . : 4294967295       lastDeleteFile . . : 4294967295
RecHeadFile . . . : 4294967295        FileCount . . . . : 3
frec_trunc1sn . . . : <0:0:0:0>
frec_read1sn . . . : <0:0:0:0>
frec_extnum . . . . : 0                LastCid . . . . . : 0
onlineBkupEnd . . . : 0                softmax . . . . . : 4194304

LOG RECORD - LSN <0:0:0:44162>
*****

HLG Header: 1recsize 212, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM Start Checkpoint (1025)
Eyecatcher . . . . : ALRH              Version . . . . . : 1
LogRecdLen . . . . : 192              LogRecdOwnr . . . : 1024 (ALM)
XTranid . . . . . : TranType: NULL
QueueName . . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

No data for Start Checkpoint Record
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 1 de 13)



## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:44374>
*****

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Transaction Table (773)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 200                      LogRecdOwnr . . : 768   (ATM)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . : 1
TranCount . . . : 0

LOG RECORD - LSN <0:0:0:44594>
*****

HLG Header: lreclsize 1836, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : Transaction Participants (1537)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 1816                    LogRecdOwnr . . : 1536   (T)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Id . . . . . : TLPH
Version . . . . : 1                        Flags . . . . . : 3
Count . . . . . : 2

Participant Entry 0
RMName . . . . : DB2 MQBankDB
RMId . . . . . : 1
SwitchFile . . : /Development/sbolam/build/devlib/tstxasw
XAOpenString . :
XACloseString . :

Participant Entry 1
RMName . . . . : DB2 MQBankDB
RMId . . . . . : 2
SwitchFile . . : /Development/sbolam/build/devlib/tstxasw
XAOpenString . :
XACloseString . :
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 2 de 13)

## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:46448>
*****

HLG Header: lrecsize 236, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM End Checkpoint (1026)
Eyecatcher   . . . : ALRH                      Version . . . . : 1
LogRecdLen   . . . : 216                      LogRecdOwnr . . : 1024 (ALM)
XTranid      . . . : TranType: NULL
QueueName    . . . : NULL
Qid          . . . : {NULL_QID}
ThisLSN      . . . : <0:0:0:0>
PrevLSN      . . . : <0:0:0:0>

ChkPtLSN     . . . : <0:0:0:44162>
OldestLSN    . . . : <0:0:0:0>
MedialLSN    . . . : <0:0:0:0>

LOG RECORD - LSN <0:0:0:52262>
*****

HLG Header: lrecsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Start Transaction (769)
Eyecatcher   . . . : ALRH                      Version . . . . : 1
LogRecdLen   . . . : 200                      LogRecdOwnr . . : 768 (ATM)
XTranid      . . . : TranType: MQI           TranNum{High 0, Low 1}
QueueName    . . . : NULL
Qid          . . . : {NULL_QID}
ThisLSN      . . . : <0:0:0:0>
PrevLSN      . . . : <0:0:0:0>

Version . . . . : 1
SoftLogLimit . . : 10000
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 3 de 13)

## Utilisation de la commande dmpmqlog

```

LOG RECORD - LSN <0:0:0:52482>
*****

HLG Header: lreclsize 730, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Put Message (257)
Eyecatcher   . . . : ALRH                      Version . . . . . : 1
LogRecdLen   . . . : 710                      LogRecdOwnr . . . : 256   (AQM)
XTranid      . . . : TranType: MQI      TranNum{High 0, Low 1}
QueueName    . . . : Queue1
Qid          . . . : {Hash 196836031, Counter: 0}
ThisLSN      . . . : <0:0:0:0>
PrevLSN      . . . : <0:0:0:52262>

Version . . . . . : 3
SpIndex  . . . . . : 1
PrevLink.Locn . . : 36                      PrevLink.Length : 8
PrevDataLink . . . : {High 0, Low 2048}
Data.Locn . . . . : 2048                    Data.Length . . . : 486
Data . . . . . :
00000: 41 51 52 48 00 00 00 04 FF FF FF FF FF FF FF FF  AQRH.....
00016: 00 00 00 00 00 00 00 00 00 00 00 01 00 01 01 C0  .....&#192;
00032: 00 00 00 00 00 00 00 01 00 00 00 22 00 00 00 00  .....".....
00048: 00 00 00 00 41 4D 51 20 74 65 73 74 71 6D 20 20  ....AMQ testqm
00064: 20 20 20 20 33 80 2D D2 00 00 10 13 00 00 00 00  3C-&#30;.....
00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00112: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01  .....
00128: 00 00 00 00 00 00 00 00 22 00 00 00 00 00 00 00  .....".....
00144: 00 00 00 00 00 00 00 00 C9 2C B5 C0 25 FF FF FF FF  .....&#26;,&#181;&#192;%....
00160: 4D 44 20 20 00 00 00 01 00 00 00 00 00 00 00 08  MD .....
00176: 00 00 00 00 00 00 01 11 00 00 03 33 20 20 20 20  .....3
00192: 20 20 20 20 00 00 00 00 00 00 00 01 20 20 20 20  .....
00208: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00224: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00240: 20 20 20 20 20 20 20 20 20 20 20 20 20 74 65 73 74  .....          test
00256: 71 6D 20 20 20 20 20 20 20 20 20 20 20 20 20 20  qm
00272: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00288: 20 20 20 20 20 20 20 20 20 20 20 20 73 62 6F 6C  .....          sbo1
00304: 61 6D 20 20 20 20 20 20 04 37 34 38 30 00 00 00  am          .7480...
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00336: 00 00 00 00 00 00 00 00 20 20 20 20 20 20 20 20  .....
00352: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00368: 20 20 20 20 20 20 20 20 00 00 00 06 75 74 7A 61  .....          ....utza
00384: 70 69 20 20 20 20 20 20 20 20 20 20 20 20 20 20  pi
00400: 20 20 20 20 20 20 20 20 31 39 39 37 30 35 31 39  .....          19970519
00416: 31 30 34 32 31 35 32 30 20 20 20 20 00 00 00 00  10421520  ....
00432: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00448: 50 65 72 73 69 73 74 65 6E 74 20 6D 65 73 73 61  Persistent messa
00464: 67 65 20 70 75 74 20 75 6E 64 65 72 20 73 79 6E  ge put under syn
00480: 63 70 6F 69 6E 74  cpoint

```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 4 de 13)

## Utilisation de la commande dmpmqlog

```

LOG RECORD - LSN <0:0:0:53458>
*****

HLG Header: lreclsize 734, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Put Message (257)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 714                      LogRecdOwnr . . . : 256   (AQM)
XTranid . . . . : TranType: NULL
QueueName . . . : Queue2
Qid . . . . . : {Hash 184842943, Counter: 2}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . . : 3
SpIndex . . . . . : 1
PrevLink.Locn . . : 36                      PrevLink.Length : 8
PrevDataLink . . : {High 0, Low 2048}
Data.Locn . . . . : 2048                    Data.Length . . . : 490
Data . . . . . :
00000: 41 51 52 48 00 00 00 04 FF FF FF FF FF FF FF FF  AQRH.....
00016: 00 00 00 00 00 00 00 00 00 00 00 01 00 01 01 C0  .....&#192;
00032: 00 00 00 00 00 00 00 01 00 00 00 26 00 00 00 00  .....&;...
00048: 00 00 00 00 41 4D 51 20 74 65 73 74 71 6D 20 20  ....AMQ testqm
00064: 20 20 20 20 33 80 2D D2 00 00 10 13 00 00 00 00  3C-&#30;.....
00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00112: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01  .....
00128: 00 00 00 00 00 00 00 26 00 00 00 00 00 00 00 00  .....&;.....
00144: 00 00 00 00 00 00 00 C9 2C B6 D8 DD FF FF FF FF  .....&#26;,.&#216;.....
00160: 4D 44 20 20 00 00 00 01 00 00 00 00 00 00 00 08  MD .....
00176: 00 00 00 00 00 00 01 11 00 00 03 33 20 20 20 20  .....3
00192: 20 20 20 20 00 00 00 00 00 00 00 01 20 20 20 20  .....
00208: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00224: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00240: 20 20 20 20 20 20 20 20 20 20 20 20 20 74 65 73 74  ..... test
00256: 71 6D 20 20 20 20 20 20 20 20 20 20 20 20 20 20  qm .....
00272: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00288: 20 20 20 20 20 20 20 20 20 20 20 20 73 62 6F 6C  ..... sbol
00304: 61 6D 20 20 20 20 20 04 37 34 38 30 00 00 00 00  am .7480...
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00336: 00 00 00 00 00 00 00 20 20 20 20 20 20 20 20 20  .....
00352: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00368: 20 20 20 20 20 20 20 00 00 00 06 75 74 7A 61  .....utza
00384: 70 69 20 20 20 20 20 20 20 20 20 20 20 20 20 20  pi .....
00400: 20 20 20 20 20 20 20 31 39 39 37 30 35 31 39  ..... 19970519
00416: 31 30 34 33 32 37 30 36 20 20 20 20 00 00 00 00  10432706 ....
00432: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00448: 50 65 72 73 69 73 74 65 6E 74 20 6D 65 73 73 61  Persistent messa
00464: 67 65 20 6E 6F 74 20 70 75 74 20 75 6E 64 65 72  ge not put under
00480: 20 73 79 6E 63 70 6F 69 6E 74  ..... syncpoint

```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 5 de 13)

## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:54192>
*****

HLG Header: lreclsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Commit Transaction (774)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 196                      LogRecdOwnr . . . : 768   (ATM)
XTranid . . . . : TranType: MQI      TranNum{High 0, Low 1}
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:52482>

Version . . . . : 1
LOG RECORD - LSN <0:0:0:54408>
*****

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Start Transaction (769)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 200                      LogRecdOwnr . . . : 768   (ATM)
XTranid . . . . : TranType: MQI      TranNum{High 0, Low 3}
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . : 1
SoftLogLimit . . : 10000

LOG RECORD - LSN <0:0:0:54628>
*****

HLG Header: lreclsize 240, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Get Message (259)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 220                      LogRecdOwnr . . . : 256   (AQM)
XTranid . . . . : TranType: MQI      TranNum{High 0, Low 3}
QueueName . . . : Queue1
Qid . . . . . : {Hash 196836031, Counter: 0}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:54408>

Version . . . . : 2
SpIndex . . . . : 1                          QPriority . . . . : 0
PrevLink.Locn . . : 36                      PrevLink.Length : 8
PrevDataLink . . . : {High 4294967295, Low 4294967295}
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 6 de 13)

## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:54868>
*****

HLG Header: lreclsize 240, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Get Message (259)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 220                      LogRecdOwnr . . . : 256   (AQM)
XTranid . . . . : TranType: NULL
QueueName . . . : Queue2
Qid . . . . . : {Hash 184842943, Counter: 2}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . . : 2
SpIndex . . . . . : 1                      QPriority . . . . . : 0
PrevLink.Locn . . : 36                    PrevLink.Length : 8
PrevDataLink . . : {High 4294967295, Low 4294967295}
LOG RECORD - LSN <0:0:0:55108>
*****

HLG Header: lreclsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Commit Transaction (774)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 196                      LogRecdOwnr . . . : 768   (ATM)
XTranid . . . . : TranType: MQI    TranNum{High 0, Low 3}
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:54628>

Version . . . . . : 1

LOG RECORD - LSN <0:0:0:55324>
*****

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Start Transaction (769)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 200                      LogRecdOwnr . . . : 768   (ATM)
XTranid . . . . : TranType: XA
      XID: formatID 5067085, gtrid_length 14, bqual_length 4
            gtrid [3270BDB40000102374657374716D]
            bqual [00000001]
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . . : 1
SoftLogLimit . . : 10000
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 7 de 13)

## Utilisation de la commande dmpmqlog

LOG RECORD - LSN <0:0:0:55544>  
 \*\*\*\*\*

HLG Header: lreccsize 738, version 1, rmid 0, eyecatcher HLRH

```
LogRecdType . . . : AQM Put Message (257)
Eyecatcher   . . . : ALRH                      Version . . . . . : 1
LogRecdLen   . . . : 718                      LogRecdOwncr . . . : 256   (AQM)
XTranid      . . . : TranType: XA
  XID: formatID 5067085, gtrid_length 14, bqual_length 4
      gtrid [3270BDB40000102374657374716D]
      bqual [00000001]
QueueName    . . . : Queue2
Qid          . . . : {Hash 184842943, Counter: 2}
ThisLSN      . . . : <0:0:0:0>
PrevLSN      . . . : <0:0:0:55324>
```

```
Version . . . . . : 3
SpIndex  . . . . . : 1
PrevLink.Locn . . . : 36                      PrevLink.Length : 8
PrevDataLink . . . : {High 0, Low 2048}
Data.Locn . . . . . : 2048                   Data.Length . . . : 494
```

```
Data . . . . . :
00000: 41 51 52 48 00 00 00 04 FF FF FF FF FF FF FF FF  AQRH.....
00016: 00 00 00 00 00 00 00 00 00 00 01 00 01 01 C0    .....&#192;
00032: 00 00 00 00 00 00 00 01 00 00 00 2A 00 00 00 00  .....*....
00048: 00 00 00 01 41 4D 51 20 74 65 73 74 71 6D 20 20  ....AMQ testqm
00064: 20 20 20 20 33 80 2D D2 00 00 10 13 00 00 00 00  3E-&#30;.....
00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00112: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01  .....
00128: 00 00 00 00 00 00 00 2A 00 00 00 00 00 00 00 00  .....*.....
00144: 00 00 00 00 00 00 00 C9 2C B8 3E E8 FF FF FF FF  .....&#26;,&#184;>.....
00160: 4D 44 20 20 00 00 00 01 00 00 00 00 00 00 00 08  MD .....
00176: 00 00 00 00 00 00 01 11 00 00 03 33 20 20 20 20  .....3
00192: 20 20 20 20 00 00 00 00 00 00 00 01 20 20 20 20  .....
00208: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00224: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00240: 20 20 20 20 20 20 20 20 20 20 20 20 74 65 73 74  test
00256: 71 6D 20 20 20 20 20 20 20 20 20 20 20 20 20 20  qm
00272: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00288: 20 20 20 20 20 20 20 20 20 20 20 20 20 73 62 6F 6C  sbol
00304: 61 6D 20 20 20 20 20 04 37 34 38 30 00 00 00 00  am .7480...
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00336: 00 00 00 00 00 00 00 20 20 20 20 20 20 20 20 20  .....
00352: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00368: 20 20 20 20 20 20 20 00 00 00 06 75 74 7A 61  ....utza
00384: 70 69 20 20 20 20 20 20 20 20 20 20 20 20 20 20  pi
00400: 20 20 20 20 20 20 20 31 39 39 37 30 35 31 39  19970519
00416: 31 30 34 34 35 38 37 32 20 20 20 20 00 00 00 00  10445872 ....
00432: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00448: 41 6E 6F 74 68 65 72 20 70 65 72 73 69 73 74 65  Another persiste
00464: 6E 74 20 6D 65 73 73 61 67 65 20 70 75 74 20 75  nt message put u
00480: 6E 64 65 72 20 73 79 6E 63 70 6F 69 6E 74  nder syncpoint
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 8 de 13)

## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:56282>
*****

HLG Header: lreclsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Prepare Transaction (770)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 196                      LogRecdOwnr . . . : 768   (ATM)
XTranid . . . . : TranType: XA
  XID: formatID 5067085, gtrid_length 14, bqual_length 4
      gtrid [3270BDB40000102374657374716D]
      bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:55544>

Version . . . . . : 1

LOG RECORD - LSN <0:0:0:56498>
*****

HLG Header: lreclsize 708, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : Transaction Prepared (1538)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 688                      LogRecdOwnr . . . : 1536 (T)
XTranid . . . . : TranType: XA
  XID: formatID 5067085, gtrid_length 14, bqual_length 4
      gtrid [3270BDB40000102374657374716D]
      bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

Id. . . . . : TLPR
Version . . . . : 1                      Flags . . . . . : 1
Count . . . . . : 3

Participant Entry 0
RMID . . . . . : 0                      State . . . . . : 2

Participant Entry 1
RMID . . . . . : 1                      State . . . . . : 2

Participant Entry 2
RMID . . . . . : 2                      State . . . . . : 2
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 9 de 13)



## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:57206>
*****

HLG Header: lrecsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Commit Transaction (774)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 196                      LogRecdOwnr . . : 768   (ATM)
XTranid . . . . : TranType: XA
    XID: formatID 5067085, gtrid_length 14, bqual_length 4
        gtrid [3270BDB40000102374657374716D]
        bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:56282>

Version . . . . : 1
LOG RECORD - LSN <0:0:0:57440>
*****

HLG Header: lrecsize 224, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : Transaction Forget (1539)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 204                      LogRecdOwnr . . : 1536 (T)
XTranid . . . . : TranType: XA
    XID: formatID 5067085, gtrid_length 14, bqual_length 4
        gtrid [3270BDB40000102374657374716D]
        bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Id . . . . . : TLFG
Version . . . . : 1                          Flags . . . . . : 0
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 10 de 13)

## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:58120>
*****

HLG Header: lreclsize 212, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM Start Checkpoint (1025)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 192                      LogRecdOwnr . . . : 1024 (ALM)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

No data for Start Checkpoint Record

LOG RECORD - LSN <0:0:0:58332>
*****

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Transaction Table (773)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 200                      LogRecdOwnr . . . : 768 (ATM)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . . : 1
TranCount . . . . : 0
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 11 de 13)

## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:58552>
*****

HLG Header: lreclsize 1836, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : Transaction Participants (1537)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 1816                      LogRecdOwnr . . . : 1536 (T)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Id. . . . . : TLPH
Version . . . : 1                            Flags . . . . . : 3
Count . . . . : 2

Participant Entry 0
RMName . . . . : DB2 MQBankDB
RMId . . . . . : 1
SwitchFile . . : /Development/sbolam/build/devlib/tstxasw
XAOpenString . . :
XACloseString . . :

Participant Entry 1
RMName . . . . : DB2 MQFeeDB
RMId . . . . . : 2
SwitchFile . . : /Development/sbolam/build/devlib/tstxasw
XAOpenString . . :
XACloseString . . :

LOG RECORD - LSN <0:0:0:60388>
*****

HLG Header: lreclsize 236, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM End Checkpoint (1026)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 216                      LogRecdOwnr . . . : 1024 (ALM)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

ChkPtLSN . . . . : <0:0:0:58120>
OldestLSN . . . . : <0:0:0:0>
MediaLSN . . . . : <0:0:0:0>
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 12 de 13)

## Utilisation de la commande dmpmqlog

```
LOG RECORD - LSN <0:0:0:60624>
*****

HLG Header: lreclsize 240, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM Stop Queue Manager (1028)
Eyecatcher . . . : ALRH                               Version . . . . . : 1
LogRecdLen . . . : 220                               LogRecdOwnc . . . : 1024 (ALM)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . : 1
StopDate . . . : 19970519                            StopTime . . . : 10490868
SessionNumber . : 0                                  ForceFlag . . . : Quiesce

AMQ7702: DMPMQLOG command has finished successfully.
```

Figure 18. Exemple de sortie de la commande dmpmqlog (Numéro 13 de 13)

### Remarque concernant la figure 18 à la page 156:

1. La valeur *headlsn* dans *Log File Header* est <0:0:0:58120>. Il s'agit de l'emplacement auquel le cliché aurait démarré si nous n'avions pas demandé un numéro d'ordre de journal de départ différent.
2. Le numéro d'ordre de journal *nextlsn* est <0:0:0:60864>. Il s'agit de celui du premier enregistrement de journal que le gestionnaire de files d'attente écrira lors de son prochain redémarrage.
3. La valeur de *HeadExtentID* est 1, ce qui indique que la tête du journal se trouve actuellement dans le fichier journal S0000001.LOG.
4. Le premier enregistrement de journal formaté est *Start Checkpoint*. Ce point de contrôle porte sur les enregistrements de journal suivants jusqu'à l'enregistrement *End CheckPoint* à l'emplacement <0:0:0:46448>.
5. L'un des enregistrements consignés au cours du point de contrôle est *Transaction Participants* à l'emplacement <0:0:0:44594>. Cet enregistrement indique les gestionnaires de ressources participant aux transactions globales coordonnées par le gestionnaire de files d'attente.
6. L'enregistrement de journal *Start Transaction* à l'emplacement <0:0:0:52262>, indique le début d'une transaction. Le *XTranid* affiche un *TranType* de type MQI, ce qui indique qu'il s'agit d'une transaction locale comprenant uniquement des mises à jour MQ.
7. L'enregistrement de journal suivant est *Put Message*. Il enregistre la commande MQPUT persistante lors du point de synchronisation qui a démarré la transaction. Cette commande a été effectuée sur la file d'attente *Queue1* et les données de message sont consignées en tant que *Persistent message put under syncpoint*. Un *SpcIndex* de valeur 1, a été alloué à ce message. Il sera mis en correspondance avec la commande MQGET ultérieure concernant ce message.
8. L'enregistrement de journal suivant au niveau du numéro d'ordre <0:0:0:53458> est aussi un enregistrement *Put Message*. Ce message persistant a été placé sur une autre file d'attente, *Queue2*, mais pas lors d'un point de synchronisation puisque la valeur de *XTranid* est *NULL*. Il s'est également vu allouer un *SpcIndex* de valeur 1, qui est un identificateur unique pour cette file d'attente particulière.

## Utilisation de la commande dmpmqlog

9. L'enregistrement de journal suivant au niveau du numéro d'ordre <0:0:0:54192> valide le message inséré lors d'un point de synchronisation.
10. Dans les enregistrements de journal <0:0:0:54408> et <0:0:0:54628>, une nouvelle transaction est démarrée par une commande MQGET lors d'un point de synchronisation pour la file d'attente *Queue1*. Le *SpcIndex* de l'enregistrement de journal *Get Message* a une valeur de 1, indiquant qu'il s'agit du même message que celui qui a été placé dans la file d'attente *Queue1* du numéro d'ordre <0:0:0:52262>.
11. L'enregistrement de journal suivant indique le message qui a été placé dans la file d'attente *Queue2* par l'autre enregistrement de journal *Put Message*.
12. La commande MQGET lancée lors d'un point de synchronisation a été validée comme l'indique l'enregistrement de journal *Commit Transaction* au numéro d'ordre <0:0:0:55108>.
13. Enfin, une commande MQBEGIN est utilisée pour démarrer une transaction globale dans l'enregistrement de journal *Start Transaction* au numéro d'ordre <0:0:0:55324>. Le *XTranid* dans cet enregistrement de journal a un *TranType* de type XA.
14. Le *Put Message* suivant enregistre un message persistant inséré dans la file *Queue2*. Il partage le même *XTranid* que l'enregistrement de journal précédent.
15. Si un enregistrement de journal *Transaction Prepared* est consigné pour ce *XTranid*, la transaction dans son ensemble doit être validée. L'absence d'un tel enregistrement de journal peut être considérée comme une indication que la transaction a été annulée. Dans ce cas, un enregistrement de journal *Transaction Prepared* figure au niveau du numéro d'ordre <0:0:0:56498>. Il indique que le gestionnaire de files d'attente participe lui-même avec un *RMId* de zéro. Il existe deux autres participants, dont les *RMId* (1 et 2) peuvent être mis en correspondance avec l'enregistrement de journal *Transaction Participants*.
16. Au cours de la phase de validation, le composant XA Transaction Manager du gestionnaire de files d'attente ne consigne pas les réponses individuelles des participants. Le journal indique seulement si les mises à jour du gestionnaire de files d'attente ont été validées. L'enregistrement de journal *Commit Transaction* à l'emplacement <0:0:0:57206> indique que le message a bien été validé dans la file d'attente *Queue2*.
17. L'enregistrement de journal *Transaction Forget* au niveau du numéro d'ordre <0:0:0:57440> indique que la décision de validation a également été transmise aux autres gestionnaires de ressources. Si ces derniers ne sont pas parvenus à valider leurs mises à jour, cela est diagnostiqué dans le journal des erreurs du gestionnaire de files d'attente.

## Utilisation de la commande dmpmqlog

---

## Chapitre 12. Utilisation du service annuaire

Le service annuaire est un service optionnel qui permet à une application connectée à un gestionnaire de files d'attente, d'ouvrir ce qu'elle pense être des files d'attente locales. Ces files sont en réalité définies sur un autre gestionnaire de files d'attente (une autre machine) avec l'attribut SCOPE défini sur CELL.

De cette façon, l'application peut procéder à toutes les opérations autorisées sur les files d'attente distantes. La mise en oeuvre proposée utilise un environnement DCE (Distributed Computing Environment). Vous êtes toutefois libre d'écrire votre propre composant n'utilisant pas cet environnement.

Pour utiliser le composant de service d'annuaire fourni, vous devez définir le service d'annuaire et son composant installé dans le gestionnaire de files d'attente. Pour ce faire, insérez la strophe appropriée dans le fichier de configuration (qm.ini) du gestionnaire de files d'attente. Consultez le manuel *MQSeries Programmable System Management* pour plus d'informations. Vous devrez aussi configurer l'environnement DCE.

---

### Utilisation de l'environnement DCE pour partager des files d'attente entre plusieurs gestionnaires de files d'attente

Si vos gestionnaires de files d'attente se trouvent sur des noeuds faisant partie d'une cellule DCE (Distributed Computing Environment (DCE), vous pouvez les configurer de façon à partager les files d'attente. Les applications peuvent alors se connecter à un gestionnaire de files d'attente et ouvrir une file sur un *autre* gestionnaire de files d'attentes d'un autre noeud. Cette opération est totalement transparente pour l'application : elle ne se rend pas compte que la file d'attente se trouve en réalité sur un autre gestionnaire de files d'attente. (Normalement, le gestionnaire de files d'attente rejette les requêtes émanant d'une application locale si la file d'attente n'existe pas sur ce gestionnaire de files d'attente.)

### Tâches de configuration des files d'attente partagées

Cette section décrit la configuration de files d'attente partagées sur des gestionnaires de files d'attente résidant sur des noeuds d'une cellule DCE.

Pour chaque gestionnaire de files d'attente :

1. Configurez le service d'annuaire en ajoutant la strophe de service d'annuaire souhaitée au fichier de configuration du gestionnaire de files d'attente. Le contenu de cette strophe est décrit dans le manuel *MQSeries Programmable System Management*. Pour appeler le service d'annuaire, vous devez redémarrer le gestionnaire de files d'attente.
2. Utilisez la commande **endmqm** pour arrêter le gestionnaire de files d'attente s'il fonctionne.
3. Utilisez la commande **strmqm** pour redémarrer le gestionnaire de files d'attente.
4. Définissez des canaux de transmission de messages entre les gestionnaires de files d'attente ; reportez-vous à la section «Préparation des canaux et des files d'attente de transmission pour l'administration éloignée» à la page 68.

## Partage de files d'attente

Pour chaque file d'attente que vous souhaitez partager, indiquez l'attribut SCOPE en tant que CELL. Par exemple, utilisez les commandes MQSC suivantes :

```
DEFINE QLOCAL(FILE.PUBLIQUE.GRISE) SCOPE(CELL)
```

ou

```
ALTER QLOCAL(FILE.LOCALE.ROSE) SCOPE(CELL)
```

La file d'attente créée ou modifiée doit faire partie d'un gestionnaire de files d'attente situé sur un noeud de la cellule DCE.

---

## Configuration DCE

Pour utiliser le composant de service d'annuaire fourni, l'environnement DCE OSF doit être installé. Ce service permet à des applications qui se connectent à un gestionnaire de files d'attente d'ouvrir des files d'attente situées sur un autre gestionnaire de la même cellule DCE.

Un exemple de script d'interpréteur de commande DCL permettant l'exécution du service d'annuaire figure dans `mqs_examples:dcesetu.com`.



---

## Chapitre 13. Configuration de MQSeries

Ce chapitre explique comment modifier le comportement d'un gestionnaire de files d'attente individuel ou d'un noeud, en fonction des besoins de votre installation.

Vous changez les informations de configuration MQSeries en modifiant les valeurs de l'ensemble d'attributs (ou paramètres) de configuration qui régit MQSeries.

La façon de modifier ces données de configuration et l'endroit où MQSeries enregistre ces changements dépendent de la plateforme. Les utilisateurs de MQSeries pour Compaq OpenVMS peuvent modifier les informations de configuration en éditant les **fichiers de configuration MQSeries**.

Le présent chapitre :

- décrit les attributs permettant de modifier les données de configuration de MQSeries dans la section «Attributs permettant de modifier les données de configuration de MQSeries» à la page 175.
- décrit les attributs permettant de modifier les données de configuration des gestionnaires de file d'attente, dans la section «Modification des données de configuration du gestionnaire de files d'attente» à la page 181.
- fournit des exemples de fichiers `mqs.ini` et `qm.ini` pour MQSeries pour Compaq OpenVMS dans la section «Exemples de fichiers `mqs.ini` et `qm.ini`» à la page 189.

---

### Fichiers de configuration MQSeries

Les utilisateurs de MQSeries pour Compaq OpenVMS modifient les attributs de configuration de MQSeries dans :

- le fichier de configuration MQSeries (`mqs.ini`) pour que les changements affectent MQSeries sur l'ensemble du noeud. Il existe un fichier `mqs.ini` par noeud.
- Le fichier de configuration du gestionnaire de files d'attentes (`qm.ini`) pour modifier les caractéristiques d'un gestionnaire de file d'attente. Il existe un fichier `qm.ini` pour chaque gestionnaire de files d'attente sur le noeud.

Un fichier de configuration (également appelé fichier de *strophes*, ou fichier *.ini*) peut contenir une ou plusieurs strophes. Une strophe étant 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).

Toute modification apportée à un fichier de configuration ne prendra effet qu'au démarrage suivant du gestionnaire de files d'attente.

### Modification des fichiers de configuration

Avant toute tentative de modification d'un fichier de configuration, sauvegardez ce dernier de manière à disposer d'une copie permettant de restaurer le fichier, si nécessaire !

Vous pouvez modifier les fichiers de configuration soit :

- automatiquement, par des commandes qui modifient la configuration des gestionnaires de files d'attente sur le noeud ;

## Fichiers de configuration

- manuellement, à l'aide d'un éditeur de texte standard.

Vous pouvez modifier les valeurs par défaut des fichiers de configuration MQSeries après l'installation.

Si vous attribuez une valeur incorrecte à un attribut du fichier de configuration, la valeur est ignorée et un message opérateur est émis pour signaler l'incident. (Le résultat est le même que si vous aviez omis la valeur de l'attribut.)

Lorsque vous créez un nouveau gestionnaire de files d'attente, il convient de :

- faire une copie de sauvegarde du fichier de configuration MQSeries,
- faire une copie de sauvegarde du fichier de configuration du nouveau gestionnaire de files d'attente.

### Quand modifier un fichier de configuration

Vous pouvez être amené à modifier un fichier de configuration, par exemple :

- si vous perdez un fichier de configuration (récupérez-le à partir d'une copie de sauvegarde, si possible) ;
- si vous devez déplacer un ou plusieurs gestionnaires de files d'attente dans un nouveau répertoire ;
- 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.

### Priorités valables dans les fichiers de configuration

Les valeurs des attributs d'un fichier de configuration sont définies conformément aux priorités suivantes :

- Les paramètres entrés en ligne de commande sont prioritaires par rapport à ceux définis dans les fichiers de configuration.
- Les paramètres définis dans le fichier `qm.ini` sont prioritaires par rapport à ceux qui sont définis dans le fichier `mqs.ini`.

### 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 des fichiers de configuration de MQSeries sont pris en compte au démarrage de MQSeries. Les modifications apportées à un fichier de configuration de gestionnaire de files d'attente 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.

## Fichier de configuration de MQSeries, `mqs.ini`

Le fichier de configuration de `mqs.ini` contient les informations relatives à tous les gestionnaires de files d'attente se trouvant sur le noeud. 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.

Le fichier `mqs.ini` est enregistré dans le répertoire de données par défaut, `MQS_ROOT:[MQM]`.

Le fichier `mqs.ini` contient :

- les noms des gestionnaires de files d'attente,

## Fichier de configuration MQSeries

- le nom du gestionnaire de files d'attente par défaut,
- l'emplacement des fichiers associés à chacun d'eux.

Vous trouverez davantage d'informations concernant le contenu du fichier `mqs.ini`, dans la section «Attributs permettant de modifier les données de configuration de MQSeries».

## Fichier de configuration de gestionnaire de files d'attente, `qm.ini`

Chaque fichier de configuration d'un gestionnaire de files d'attente, `qm.ini`, contient les informations relatives à ce gestionnaire. Il existe un fichier de configuration par gestionnaire de files d'attente. Le fichier `qm.ini` est créé automatiquement lors de la création du gestionnaire de files d'attente auquel il est associé.

Ce fichier se trouve dans le répertoire principal de l'arborescence occupée par le gestionnaire de files d'attente.

Sous MQSeries pour Compaq OpenVMS, le chemin et le nom du fichier de configuration du gestionnaire de files d'attente QMNAME serait :

```
MQS_ROOT:[MQM.QMGRS.QMNAME]QM.INI
```

**Remarque :** Le nom du gestionnaire de files d'attente peut comporter jusqu'à 48 caractères. Cependant, cela ne garantit pas que le nom est correct ou unique. C'est pourquoi un nom de répertoire est généré en fonction du nom du gestionnaire de files d'attente. Ce processus s'appelle la **transformation de nom**. Pour une description plus détaillée, reportez-vous à la section «Signification des noms de fichiers MQSeries» à la page 23.

Vous trouverez davantage d'informations concernant le fichier `qm.ini`, à la section «Modification des données de configuration du gestionnaire de files d'attente» à la page 181.

---

## Attributs permettant de modifier les données de configuration de MQSeries

Les groupes d'attributs suivants figurent dans le fichier `mqs.ini`:

- Strophe `AllQueueManagers`
- «Strophe `ClientExitPath`» à la page 177
- «Strophe `DefaultQueueManager`» à la page 177
- «Strophe `ExitProperties`» à la page 177
- «Strophe `LogDefaults`» à la page 178
- «Strophe `QueueManager`» à la page 180

Vous trouverez un exemple de fichier `mqs.ini` à la section «Exemples de fichiers `mqs.ini` et `qm.ini`» à la page 189.

### Strophe `AllQueueManagers`

La strophe `AllQueueManagers` peut définir :

- le chemin du répertoire `qmgrs` contenant les fichiers associés à un gestionnaire de files d'attente,
- la méthode de conversion d'EBCDIC en ASCII.

## Modification du fichier de configuration de MQSeries

### **DefaultPrefix=nom\_répertoire**

Définit le chemin du répertoire qmgrs qui contient les informations relatives au gestionnaire de files d'attente.

Si vous modifiez le préfixe par défaut du gestionnaire de files d'attente, vous devez reconstituer l'arborescence créée au moment de l'installation (voir l'«Annexe C. Structure de répertoires» à la page 321).

Il faut en particulier créer la structure qmgrs. Vous devez arrêter MQSeries avant de modifier le préfixe par défaut. Ne redémarrez MQSeries qu'une fois que les structures ont été déplacées dans le nouvel emplacement et que le préfixe par défaut a été modifié.

Au lieu de modifier le préfixe par défaut DefaultPrefix, vous pouvez le remplacer par la variable d'environnement MQSPREFIX dans la commande **crtmqm**.

### **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**

L'affectation de la valeur NL\_TO\_LF à l'attribut ConvEBCDICNewline imposera la conversion du caractère EBCDIC NL (X'15') vers le caractère ASCII LF (line feed, 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 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\_LF.

La liste des CCSID ISO possibles se trouve dans le tableau 7.

Tableau 7. 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

## Modification du fichier de configuration de MQSeries

Tableau 7. Liste des CCSID ISO possibles (suite)

CCSID	Jeu de codes
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 *Guide de programmation d'applications* ou la section «Conversion de données» à la page 77.

### Strophe ClientExitPath

La strophe ClientExitPath définit le chemin par défaut de l'exit de canal sur le client.

**ExitsDefaultPath=préfixepardéfaut**

Définit le préfixe par défaut de la plateforme.

### Strophe DefaultQueueManager

La strophe DefaultQueueManager définit le gestionnaire de files d'attente par défaut pour le noeud.

**Name=gest\_files\_défaut**

Le gestionnaire de files d'attente par défaut traite toutes les commandes pour lesquelles vous n'avez pas indiqué de nom de gestionnaire. L'attribut DefaultQueueManager est automatiquement mis à jour lorsque vous créez un nouveau gestionnaire de files d'attente par défaut. Si vous créez par erreur un nouveau gestionnaire de files d'attente par défaut et voulez ensuite revenir à l'original, vous devez modifier l'attribut DefaultQueueManager manuellement.

### Strophe ExitProperties

La strophe ExitProperties définit les options de configuration utilisées par les programmes d'exit du gestionnaire de files d'attente.

**CLWLMode=SAFE | FAST**

L'exit de charge de travail de cluster, CLWL, permet de définir la file d'attente à ouvrir dans le cluster en réponse à un appel MQAPI (MQOPEN ou MQPUT, etc.). L'exit CLWL s'exécute en mode FAST ou SAFE selon la valeur que vous définissez pour l'attribut CLWLMode. Si vous ne précisez rien, l'exit de charge de travail de cluster s'exécute en mode SAFE.

#### SAFE

Indique que l'exit CLWL s'exécute sous forme d'un processus distinct de celui du gestionnaire de files d'attente. Il s'agit du mode par défaut.

Au cas où il se produirait un incident avec l'exit utilisateur CLWL en mode SAFE :

- le processus serveur CLWL (amqzlw0) échoue,
- le gestionnaire de files d'attente redémarre le processus serveur CLWL,
- l'erreur est consignée dans le journal des erreurs. Lorsqu'un appel MQAPI est en cours de traitement, vous êtes averti par un code retour d'incident.

L'intégrité du gestionnaire de files d'attente est préservée.

## Modification du fichier de configuration de MQSeries

**Remarque :** L'exécution de l'exit CLWL sous forme de processus distinct peut avoir une incidence sur les performances.

### FAST

Indique que l'exit de cluster s'exécute en ligne avec le processus du gestionnaire de files d'attente.

Cette option permet d'améliorer les performances en évitant les temps de traitement associés au mode SAFE, mais il existe un risque pour l'intégrité du gestionnaire de files d'attente. C'est pourquoi il convient de ne choisir l'exécution de l'exit CLWL en mode FAST que si vous êtes certain qu'il n'y aura **pas** de problèmes avec votre exit CLWL, et que les performances sont essentielles.

En cas de problème avec un exit CLWL qui s'exécute en mode FAST, le gestionnaire de files d'attente échoue et l'intégrité du gestionnaire de files d'attente risque d'être compromise.

## Strophe LogDefaults

La strophe `LogDefaults` définit les attributs de journalisation par défaut du noeud. Ces attributs sont utilisés comme valeurs par défaut lors de la création d'un gestionnaire de files d'attente, mais peuvent être remplacés si vous les redéfinissez avec la commande `crtmqm`. Pour plus d'informations, reportez-vous à la section «`crtmqm` (création d'un gestionnaire de files d'attente)» à la page 250.

Une fois le gestionnaire de files d'attente créé, ses attributs de journalisation sont lus à partir de la strophe de journalisation du fichier `qm.ini` ;

L'attribut `DefaultPrefix` (strophe `AllQueueManagers`) et l'attribut `LogPath` de la strophe `LogDefaults` vous permettent de placer le gestionnaire et son journal sur des unités physiques différentes. Il s'agit de la méthode conseillée, bien que par défaut ils se trouvent sur la même unité.

Pour plus d'informations sur la façon de calculer la taille des journaux, reportez-vous à la section «Calcul de la taille des journaux» à la page 141.

**Remarque :** Les limites indiquées dans la liste de paramètres qui suit sont fixées par MQSeries. Les limites imposées par le système d'exploitation peuvent être inférieures à la valeur maximale indiquée.

### `LogPrimaryFiles=3|2-62`

Les fichiers journaux primaires sont les fichiers journaux alloués au moment de la création en vue d'une utilisation ultérieure.

Le nombre minimal de fichiers journaux primaires est de 2, et le nombre maximal, de 62. Valeur par défaut : 3.

Le nombre total de fichiers journaux primaires et secondaires ne doit pas dépasser 63 et ne peut être inférieur à 3.

Cette valeur est écrasée par le paramètre `-lp` de la commande `crtmqm`, lors de la création du gestionnaire de files d'attente.

### `LogSecondaryFiles=2|1-61`

Les fichiers journaux secondaires sont les fichiers journaux alloués lorsque les fichiers primaires sont saturés.

Le nombre minimal de fichiers journaux secondaires est de 1, et le nombre maximal de 61. Leur nombre par défaut est de 2.

## Modification du fichier de configuration de MQSeries

Le nombre total de fichiers journaux primaires et secondaires ne doit pas dépasser 63 et ne peut être inférieur à 3.

Cette valeur est écrasée par le paramètre `-ls` de la commande `crtmqm`, lors de la création du gestionnaire de files d'attente.

### **LogFilePages=nombre**

Les données de journalisation sont contenues dans une série de fichiers appelés fichiers journaux. La taille du fichier journal est indiquée en nombre de pages de 4 ko.

Sous MQSeries pour Compaq OpenVMS, le nombre de pages par défaut d'un fichier journal est de 1024, ce qui équivaut à une taille de fichier journal de 4 Mo. Le nombre minimal de pages d'un fichier journal est de 64, et le nombre maximal, de 16 384

Cette valeur est écrasée par le paramètre `-lf` de la commande `crtmqm`, lors de la création du gestionnaire de files d'attente.

### **LogType=CIRCULAR | LINEAR**

L'attribut `LogType` définit le type de de fichier journal utilisé. Valeur par défaut : CIRCULAR.

#### **CIRCULAR**

Permet une reprise au redémarrage en utilisant le journal pour abandonner les transactions qui étaient en cours à l'arrêt du système.

Pour plus d'informations, reportez-vous à la section «Journalisation circulaire» à la page 136.

#### **LINEAR**

Permet la reprise au redémarrage et la reconstitution de support ou la reprise intégrale (recréation des données perdues ou altérées par relecture du contenu du journal).

Pour plus d'informations, reportez-vous à la section «Journalisation linéaire» à la page 137.

Si vous voulez modifier la valeur par défaut, vous devez éditer l'attribut `LogType` du fichier `mqs.ini`. Vous pouvez également activer la journalisation linéaire avec le paramètre `-ll` de la commande `crtmqm` command. Vous ne pouvez pas changer de méthode de journalisation après la création du gestionnaire de files d'attente.

### **LogBufferPages=17 | 4-32**

La quantité de mémoire allouée aux enregistrements tampon pour l'écriture est configurable. La taille des tampons est indiquée en nombre de pages de 4 ko.

Le nombre minimal de pages de mémoire tampon est de 4, et le nombre maximal, de 32. Plus les tampons sont grands et plus le débit de transmission est élevé, en particulier pour les longs messages.

Le nombre de pages de mémoire tampon par défaut est de 17, ce qui équivaut à 68 ko.

La valeur est examinée lorsque le gestionnaire de files d'attente est créé ou démarré et elle peut être augmentée ou réduite aux mêmes moments. Cependant, la modification de la valeur ne sera pas effective avant que le gestionnaire de files d'attente ne soit redémarré.

### **LogDefaultPath=nom\_répertoire**

Vous pouvez préciser le répertoire dans lequel se trouvent les fichiers journaux d'un gestionnaire de files d'attente. Le répertoire doit exister sur une unité



## Modification du fichier de configuration de MQSeries

locale à laquelle le gestionnaire de files d'attente a accès en écriture et de préférence différente de celle sur laquelle se trouvent les files d'attente de messages. En effet, l'utilisation d'une unité différente accroît la protection en cas de panne système.

Sous MQSeries pour Compaq OpenVMS, la valeur par défaut est :  
MQS\_ROOT: [MQM.LOG].

Vous pouvez indiquer un autre nom de répertoire dans la commande **crtmqm** à l'aide du paramètre **-ld**. Lorsqu'un gestionnaire de files d'attente est créé, un sous-répertoire est également créé dans le répertoire du gestionnaire de files d'attente, afin de contenir les fichiers journaux. Le nom de ce sous-répertoire est basé sur celui du gestionnaire de files d'attente. Cela garantit que le chemin des fichiers journaux est unique et qu'il est conforme aux limitations de longueur des noms de répertoires.

Si vous n'utilisez pas le paramètre **-ld** dans la commande **crtmqm**, c'est la valeur de l'attribut `LogDefaultPath` du fichier `mqs.ini` qui est utilisée, soit :  
MQS\_ROOT: [MQM.LOG].

Le nom du gestionnaire de files d'attente est ajouté au nom du répertoire pour garantir que les différents gestionnaires de files d'attente utilisent différents répertoires de journalisation.

Une fois que le gestionnaire de files d'attente a été créé, une valeur `LogPath` est créée dans la strophe `Log` du fichier `qm.ini`, décrivant le nom complet du répertoire contenant les fichiers de journalisation du gestionnaire de files d'attente. Cette valeur est utilisée pour localiser le journal lorsque le gestionnaire de files d'attente est démarré ou arrêté.

## Strophe QueueManager

Il existe une strophe `QueueManager` par gestionnaire de files d'attente. Ses attributs définissent le nom du gestionnaire de files d'attente et le nom du répertoire contenant les fichiers associés au gestionnaire. Ce nom est basé sur celui du gestionnaire, mais il peut être transformé s'il existe déjà ou si le nom du gestionnaire n'est pas un nom de fichier correct.

Pour plus d'informations sur la transformation des noms, reportez-vous à la section «Signification des noms de fichiers MQSeries» à la page 23.

**Name**=*nom\_gest\_files*

Définit le nom du gestionnaire de files d'attente.

**Prefix**=*préfixe*

Définit l'emplacement des fichiers du gestionnaire de files d'attente. Par défaut, il s'agit de la même valeur que pour l'attribut `DefaultPrefix` de la strophe `AllQueueManager` du fichier `mqs.ini`.

**Directory**=*nom*

Définit le nom du sous-répertoire où sont enregistrés les fichiers du gestionnaire de files d'attente. Il s'agit normalement de `MQS_ROOT: [MQM.QMGRS]`, sauf si une autre valeur du préfixe a été définie. Ce nom est basé sur celui du gestionnaire, mais il peut être transformé s'il existe déjà ou si le nom du gestionnaire n'est pas un nom de fichier correct.



## Modification des données de configuration du gestionnaire de files d'attente

Les groupes d'attributs suivants peuvent figurer dans le fichier `qm.ini` correspondant à un gestionnaire de files d'attente donné, ou être utilisés pour remplacer les valeurs définies dans `mq5.ini`.

- «Strophe Service»
- «Strophe ServiceComponent»
- «Strophe Log» à la page 182
- «Strophe XAResourceManager» à la page 184
- «Strophe Channels» à la page 186
- «Strophes LU62 et TCP» à la page 187
- «Strophe ExitPath» à la page 189

### Strophe Service

La strophe `Service` indique le nom d'un service optionnel et le nombre de ses points d'entrée. Il doit exister une strophe `Service` par service utilisé.

Pour chaque composant d'un service, il doit exister une strophe `ServiceComponent` ; elle indique le nom et le chemin du module contenant le code du composant. Pour plus d'informations, reportez-vous à la section `Strophe ServiceComponent`.

**Name=AuthorizationService | NameService**

Définit le nom du service requis.

#### AuthorizationService

Pour MQSeries, le composant `Authorization Service` (service d'autorisation) est le Gestionnaire des droits d'accès aux objets ou OAM.

Sous MQSeries pour Compaq OpenVMS, la strophe `AuthorizationService` et sa strophe associée `ServiceComponent` sont ajoutées automatiquement lors de la création du gestionnaire de files d'attente. Cette action automatique peut être désactivée en fixant la valeur de la logique `mq5noaut` avant de créer le gestionnaire de files d'attente. (Pour plus d'informations, consultez la section «Désactivation de l'OAM» à la page 85.) Toute autre strophe `ServiceComponent` doit être ajoutée manuellement.

#### NameService

La strophe `NameService` doit être ajoutée manuellement au fichier `qm.ini` pour activer le service d'annuaire fourni.

**EntryPoints=nombre-d'entrées**

Définit le nombre de points d'entrée du service y compris les points d'entrée d'initialisation et de fin.

Pour plus d'informations sur les services et composants optionnels, reportez-vous au manuel *MQSeries Programmable System Management*.

Pour plus de détails sur les services de sécurité, reportez-vous au «Chapitre 7. Protection des objets MQSeries» à la page 81.

### Strophe ServiceComponent

La strophe `ServiceComponent` indique le nom et le chemin du module contenant le code du composant.

## Modification du fichier de configuration de MQSeries

Il peut exister plusieurs strophes ServiceComponent par service, mais chacune doit concorder avec la strophe Service correspondante.

Sous MQSeries pour Compaq OpenVMS, la strophe de service d'autorisation est présente par défaut et le composant associé (l'OAM) est actif.

**Service=service\_annuaire**

Définit le nom du service requis. Ce nom doit coïncider avec la valeur de l'attribut Name de la strophe Service.

**Name=nom\_composant**

Définit le nom descriptif du composant. Ce nom doit être unique et se composer exclusivement de caractères autorisés dans les noms des objets MQSeries (par exemple, les noms de files d'attente). Ce nom figure dans les messages opérateur générés par le service. Il est par conséquent vivement conseillé que ce nom commence par une marque de la société ou toute autre chaîne distinctive.

**Module=nom\_module**

Définit le nom du module qui contient le code du composant.

**Remarque :** Indiquez un chemin complet.

**ComponentDataSize=taille**

Indique la taille en octets de la zone de données composant transmise au composant à chaque appel. Indiquez zéro si aucune donnée composant n'est requise.

Pour plus d'informations sur les services et composants optionnels, reportez-vous au manuel *MQSeries Programmable System Management*.

## Strophe Log

La strophe Log définit les attributs de journalisation d'un gestionnaire de files d'attente donné. Par défaut, ils sont hérités des paramètres de la strophe LogDefaults du fichier `mqs.ini`, à la création du gestionnaire de files d'attente, sauf s'ils sont écrasés par des paramètres précisés par la commande `crtmqm`. Pour plus d'informations, reportez vous aux sections «Strophe LogDefaults» à la page 178 et «crtmqm (création d'un gestionnaire de files d'attente)» à la page 250.

Ne modifiez les attributs de cette strophe que si le gestionnaire en question nécessite une configuration différente des autres.

Les valeurs indiquées pour les attributs dans le fichier `qm.ini` sont lues lors du démarrage du gestionnaire de files d'attente. Le fichier est créé lors de la création du gestionnaire de files d'attente.

Pour plus d'informations sur la façon de calculer la taille des journaux, reportez-vous à la section «Calcul de la taille des journaux» à la page 141.

**Remarque :** Les limites indiquées dans la liste de paramètres qui suit sont fixées par MQSeries. Les limites imposées par le système d'exploitation peuvent être inférieures à la valeur maximale indiquée.

**LogPrimaryFiles=3|2-62**

Les fichiers journaux primaires sont les fichiers journaux alloués au moment de la création en vue d'une utilisation ultérieure.

## Modification du fichier de configuration de MQSeries

Le nombre minimal de fichiers journaux primaires est de 2, et le nombre maximal, de 62. Valeur par défaut : 3.

Le nombre total de fichiers journaux primaires et secondaires ne doit pas dépasser 63 et ne peut être inférieur à 3.

La valeur est examinée lorsque le gestionnaire de files d'attente est créé ou démarré. Vous pouvez la modifier une fois que le gestionnaire de files d'attente a été créé. Cependant, la modification de la valeur ne sera effective que lorsque le gestionnaire de files d'attente aura été redémarré, et il se peut qu'elle ne soit pas prise en compte immédiatement.

### **LogSecondaryFiles=2 | 1-61**

Les fichiers journaux secondaires sont les fichiers journaux alloués lorsque les fichiers primaires sont saturés.

Le nombre minimal de fichiers journaux secondaires est de 1, et le nombre maximal, de 61. Leur nombre par défaut est de 2.

Le nombre total de fichiers journaux primaires et secondaires ne doit pas dépasser 63 et ne peut être inférieur à 3.

La valeur est examinée lorsque le gestionnaire de files d'attente est démarré. Vous pouvez modifier cette valeur, mais les modifications ne seront effectives que lorsque le gestionnaire de files d'attente aura été redémarré, et leurs effets ne seront pas forcément immédiats.

### **LogFilePages=*nombre***

Les données de journalisation sont contenues dans une série de fichiers appelés fichiers journaux. La taille du fichier journal est indiquée en nombre de pages de 4 ko.

Sous MQSeries pour Compaq OpenVMS, le nombre de pages par défaut d'un fichier journal est de 1024, ce qui équivaut à une taille de fichier journal de 4 Mo. Le nombre minimal de pages d'un fichier journal est de 64, et le nombre maximal, de 16 384

**Remarque :** La taille des fichiers journaux est précisée lors de la création du gestionnaire de files d'attente et elle ne peut être modifiée pour un gestionnaire de files d'attente existant.

### **LogType=CIRCULAR | LINEAR**

L'attribut LogType définit le type de journalisation utilisé par le gestionnaire de files d'attente. Cependant, vous ne pouvez pas modifier le type de journalisation une fois le gestionnaire créé. Pour plus de détails sur la création d'un gestionnaire de files d'attente avec le type de journalisation approprié, reportez-vous à la description de l'attribut LogType dans la section «Strophe LogDefaults» à la page 178.

#### **CIRCULAR**

Permet une reprise au redémarrage en utilisant le journal pour abandonner les transactions qui étaient en cours à l'arrêt du système.

Pour plus d'informations, reportez-vous à la section «Journalisation circulaire» à la page 136.

#### **LINEAR**

Permet la reprise au redémarrage et la reconstitution de support ou la reprise intégrale (recréation des données perdues ou altérées par relecture du contenu du journal).

## Modification du fichier de configuration de MQSeries

Pour plus d'informations, reportez-vous à la section «Journalisation linéaire» à la page 137.

### **LogBufferPages=17|4-32**

La quantité de mémoire allouée aux enregistrements tampon pour l'écriture est configurable. La taille des tampons est indiquée en nombre de pages de 4 ko.

Le nombre minimal de pages de mémoire tampon est de 4, et le nombre maximal, de 32. Plus les tampons sont grands et plus le débit de transmission est élevé, en particulier pour les longs messages.

Le nombre de pages de mémoire tampon par défaut est de 17, ce qui équivaut à 68 ko.

La valeur est examinée lorsque le gestionnaire de files d'attente est démarré et elle peut être augmentée ou réduite aux mêmes moments. Cependant, la modification de la valeur ne sera pas effective avant que le gestionnaire de files d'attente ne soit redémarré.

### **LogPath=nom\_répertoire**

Vous pouvez préciser le répertoire dans lequel se trouvent les fichiers journaux d'un gestionnaire de files d'attente. Le répertoire doit exister sur une unité locale à laquelle le gestionnaire de files d'attente a accès en écriture et de préférence différente de celle sur laquelle se trouvent les files d'attente de messages. En effet, l'utilisation d'une unité différente accroît la protection en cas de panne système.

La valeur par défaut est `MQS_ROOT:[MQM.LOG]`.

Vous pouvez indiquer un autre nom de répertoire par la commande `crtmqm` et le paramètre `-ld`. Lorsqu'un gestionnaire de files d'attente est créé, un sous-répertoire est également créé dans le répertoire du gestionnaire de files d'attente, afin de contenir les fichiers journaux. Le nom de ce sous-répertoire est basé sur celui du gestionnaire de files d'attente. Cela garantit que le chemin des fichiers journaux est unique et qu'il est conforme aux limitations de longueur des noms de répertoires.

Si vous n'utilisez pas le paramètre `-ld` dans la commande `crtmqm`, c'est la valeur de l'attribut `LogDefaultPath` du fichier `mq5.ini` qui est utilisée.

**Remarque :** Sous MQSeries pour Compaq OpenVMS, l'ID utilisateur et le groupe `mqm` doivent avoir des droits d'accès complets aux fichiers de journalisation. Si vous modifiez les emplacements de ces fichiers, vous devez accorder ces droits vous-même. Si vous conservez les emplacements par défaut fournis avec le produit, ce n'est pas nécessaire.

## Strophe XAResourceManager

La strophe `XAResourceManager` identifie les gestionnaires de ressources qui sont impliqués dans les unités de travail globales coordonnées par le gestionnaire de files d'attente.

Une strophe `XAResourceManager` est nécessaire dans le fichier `qm.ini` pour chaque instance d'un gestionnaire de ressources participant à des unités de travail globales ; le fichier `mq5.ini` ne fournit aucune valeur par défaut.

Pour plus d'informations sur l'ajout d'attributs `XAResourceManager` à un fichier `qm.ini`, reportez-vous à la section «Coordination des bases de données» à la page 122.

## Modification du fichier de configuration de MQSeries

### **Name=*nom* (obligatoire)**

Identifie l'instance du gestionnaire de ressources.

La valeur Name doit comporter au maximum 31 caractères, et doit être unique dans le fichier `qm.ini`. Vous pouvez utiliser le nom du gestionnaire de ressources tel qu'il est défini dans sa structure de commutation XA.

Cependant, si vous utilisez plusieurs instances du même gestionnaire de ressources, vous devez créer un nom unique pour chaque instance. Afin de garantir son caractère unique, incluez le nom de la base de données dans la chaîne Name (par exemple).

MQSeries utilise la valeur Name dans les messages et dans la sortie de la commande **dspmqrn**.

Nous vous recommandons de ne pas modifier le nom d'une instance de gestionnaire de ressources, ni de supprimer son entrée du fichier `qm.ini`, une fois que le gestionnaire de files d'attente associé a démarré et que le nom du gestionnaire de ressources est utilisé.

### **SwitchFile=*nom* (obligatoire)**

Définit le nom complet du fichier de chargement contenant la structure de commutation XA du gestionnaire de ressources.

### **XAOpenString=*chaîne* (facultatif)**

Définit la chaîne de données à transmettre au point d'entrée `xa` du gestionnaire de ressources. Le contenu de cette chaîne dépend du gestionnaire de ressources. Par exemple, elle peut identifier la base de données à laquelle doit accéder cette instance du gestionnaire de ressources. Pour plus de détails, reportez-vous à la section «Ajout des données de configuration XAResourceManager pour Oracle» à la page 128 et consultez la section relative à la chaîne appropriée dans la documentation du gestionnaire de ressources.

### **XACloseString=*chaîne* (facultatif)**

Définit la chaîne de données à transmettre au point d'entrée `xa_close` du gestionnaire de ressources. Le contenu de cette chaîne dépend du gestionnaire de ressources. Pour plus de détails, reportez-vous à la section «Ajout des données de configuration XAResourceManager pour Oracle» à la page 128 et consultez la section relative à la chaîne appropriée dans la documentation du gestionnaire de ressources.

### **ThreadOfControl=THREAD | PROCESS**

La valeur de l'attribut ThreadOfControl est utilisée par le gestionnaire de files d'attente à des fins de sérialisation lorsqu'il doit appeler le gestionnaire de ressources à partir de l'un de ses processus à unités d'exécution multiples.

#### **THREAD**

Signifie que le gestionnaire de ressources est accessible à partir de plusieurs unités d'exécution ("thread aware"). Dans un processus MQSeries à unités d'exécution multiples, les appels de fonction XA peuvent être effectués vers le gestionnaire de ressources externe à partir de plusieurs unités d'exécution à la fois.

#### **PROCESS**

Signifie que le gestionnaire de ressources n'accepte qu'une seule connexion en même temps ("thread safe"). Dans un processus MQSeries à unités d'exécution multiples, un seul appel de fonction XA simultané peut être effectué vers le gestionnaire.

L'entrée ThreadOfControl ne s'applique pas aux appels de fonction XA émis par le gestionnaire de files d'attente dans un processus d'application à unités

## Modification du fichier de configuration de MQSeries

d'exécution multiples. En règle générale, une application qui dispose d'unités de travail concurrentes sur différentes unités d'exécution requiert que ce mode de fonctionnement soit pris en charge par chacun des gestionnaires de ressources.

### Strophe Channels

La strophe Channels contient des informations sur les canaux.

**MaxChannels=100** | *nombre*

Définit le nombre maximal de canaux autorisé. Valeur par défaut 100.

**MaxActiveChannels=***valeur* **\_MaxChannels**

Définit le nombre maximal de canaux autorisés à être actifs à un instant donné. La valeur par défaut est fixée par l'attribut MaxChannels.

**MaxInitiators=3** | *nombre*

Définit le nombre maximal d'initialisateurs.

**MQIBINDTYPE=FASTPATH** | **STANDARD**

Définit les liaisons entre applications.

#### FASTPATH

Les canaux se connectent en mode MQCONN FASTPATH, c'est-à-dire sans processus agent.

#### STANDARD

Les canaux se connectent en mode STANDARD.

**AdoptNewMCA=NO** | **SVR** | **SDR** | **RCVR** | **CLUSRCVR** | **ALL** | **FASTPATH**

Si MQSeries reçoit une demande de démarrage d'un canal mais détecte qu'un processus amqcrsta existe déjà pour ce canal, il est nécessaire d'arrêter le processus existant avant de lancer le nouveau. L'attribut AdoptNewMCA permet de contrôler l'arrêt d'un processus existant et le démarrage d'un nouveau processus pour le type de canal spécifié.

Si vous définissez l'attribut AdoptNewMCA pour un type de canal donné et que le nouveau canal ne démarre pas parce que le canal est déjà actif :

1. Le nouveau canal tente d'arrêter le précédent en l'invitant à s'arrêter.
2. Si le serveur du canal précédent ne répond pas dans l'intervalle défini par AdoptNewMCATimeout, le processus (ou l'unité d'exécution) du serveur de canal en question est supprimé(e).
3. Si le serveur du canal précédent n'a pas mis fin à son activité après l'étape 2 et que l'intervalle AdoptNewMCATimeout a expiré une seconde fois, MQSeries met fin au canal sur une erreur "CANAL EN SERVICE".

Indiquez une ou plusieurs valeurs, séparées par des virgules ou des espaces, parmi les valeurs de la liste suivante :

**NO** La fonction AdoptNewMCA n'est pas requise. Il s'agit du type d'arrêt par défaut.

**SVR** Adopte les canaux de type serveur

**SDR** Adopte les canaux de type émetteur

**RCVR** Adopte les canaux de type récepteur

**CLUSRCVR**

Adopte les canaux de type récepteur pour les clusters

**ALL** Adopte tous les types de canaux, à l'exception des canaux FASTPATH

**FASTPATH**

Adopte le canal s'il s'agit d'un canal FASTPATH. Ceci se produit uniquement lorsque vous spécifiez également le type de canal approprié ; par exemple, AdoptNewMCA=RCVR,SVR,FASTPATH

### Attention !

L'attribut AdoptNewMCA peut se comporter de manière imprévisible avec les canaux FASTPATH en raison de la structure interne du gestionnaire de files d'attente. Il convient donc de faire preuve d'une grande prudence lors de l'activation de l'attribut AdoptNewMCA avec des canaux FASTPATH.

### **AdoptNewMCATimeout=60 | 1—3600**

Définit en secondes le délai pendant lequel le nouveau processus attend l'arrêt de l'ancien. Indiquez une valeur en secondes comprise entre 1 et 3600. Valeur par défaut : 60.

### **AdoptNewMCACheck=QM | ADDRESS | NAME | ALL**

Permet de définir le type de vérification requis lors de l'activation de l'attribut AdoptNewMCA. Il est important d'effectuer les trois types de vérification suivants, si possible, pour éviter un arrêt par inadvertance ou malveillance de vos canaux. Au minimum, assurez-vous que les noms de canaux sont identiques.

Indiquez une ou plusieurs valeurs, séparées par des virgules ou des espaces, parmi les valeurs suivantes :

**QM** Signifie que le processus d'écoute vérifie la correspondance entre les noms de gestionnaires de files d'attente.

#### **ADDRESS**

Signifie que le processus d'écoute vérifie l'adresse de communication. Par exemple, l'adresse TCP/IP.

#### **NAME**

Signifie que le processus d'écoute vérifie la correspondance entre les noms de canaux.

**ALL** Signifie que le processus d'écoute vérifie la correspondance entre les noms de gestionnaires, l'adresse de communication et la correspondance entre les noms de canaux.

AdoptNewMCACheck=NAME,ADDRESS est la valeur par défaut pour FAP1, FAP2 et FAP3, tandis que AdoptNewMCACheck=NAME,ADDRESS,QM est la valeur par défaut pour FAP4 et suivant.

## Strophes LU62 et TCP

Ces strophes indiquent les paramètres de configuration du protocole réseau. Elles se substituent aux attributs par défaut des canaux.

**Remarque :** Seuls les attributs qui modifient des valeurs par défaut ont besoin d'être spécifiés.

### **LU62**

Les attributs suivants peuvent être définis :

#### **TPName**

Définit le nom du TP à lancer sur le site éloigné.

#### **LocalLU**

Définit le nom de l'unité logique à utiliser sur les systèmes locaux.



## Modification du fichier de configuration de MQSeries

### TCP

Les attributs suivants peuvent être définis :

**Port=1414** | *numéro\_de\_port*

Définit le numéro de port par défaut en notation décimale pour les sessions TCP/IP. Le numéro de port classique pour MQSeries est 1414.

**KeepAlive=YES** | **NO**

Permet d'activer et de désactiver la fonction KeepAlive. KeepAlive=YES indique à TCP/IP de vérifier à intervalles réguliers la disponibilité de l'autre extrémité de la connexion. Lorsque ce n'est pas le cas, le canal est fermé.

**ListenerBacklog=nombre**

Lors de la réception TCP/IP, le nombre maximal de demandes de connexion en attente est défini. Ces demandes attendent sur le port TCP/IP que le programme d'écoute les accepte. Les valeurs par défaut pour la liste des demandes destinées au programme d'écoute sont répertoriées dans le tableau 8.

Tableau 8. Valeurs par défaut pour le nombre maximal de demandes de connexion en attente (TCP)

Plateforme	Valeur ListenerBacklog par défaut
OS/390	255
OS/2 Warp	10
Windows NT Server	100
Windows NT Workstation	5
AS/400	255
Sun Solaris	100
HP-UX	20
AIX V4.2 ou suivant	100
AIX V4.1 ou suivant	10
Toutes les autres plateformes	5

Si le nombre de demandes en attente atteint les valeurs du tableau 8, la connexion TCP/IP est rejetée et le canal ne peut démarrer.

Pour les canaux MCA, ceci fait passer le canal en mode RETRY. Le canal tentera alors de se reconnecter ultérieurement.

Pour les connexions client, le client reçoit de MQCONN un code anomalie MQRC\_Q\_MGR\_NOT\_AVAILABLE. Il devra tenter de se reconnecter ultérieurement.

L'attribut ListenerBacklog permet de remplacer la valeur par défaut par votre valeur pour les demandes en attente destinées au programme d'écoute TCP/IP.

**Remarque :** Certains systèmes d'exploitation acceptent une valeur supérieure à la valeur par défaut indiquée. Au besoin, vous pouvez vous en servir pour éviter d'atteindre la limite de connexion.



## Strophe ExitPath

**ExitDefaultPath=chaîne**

L'attribut ExitDefaultPath définit l'emplacement des :

- exits de canal pour les clients
- exits de canal et de conversion de données pour les serveurs

Le chemin de l'exit est lu dans la strophe ClientExitPath du fichier mqs.ini pour les clients, et dans cette strophe (ExitPath) pour les serveurs.

## Exemples de fichiers mqs.ini et qm.ini

La figure 19 montre un exemple de fichier mqs.ini sous MQSeries pour Compaq OpenVMS.

```

*****#
#* Module Name: mqs.ini                               *#
#* Type      : MQSeries Configuration File           *#
#* Function  : Define MQSeries resources for the node *#
#*                                                  *#
#* Notes    :                                       *#
#* 1) This is an example MQSeries configuration file *#
#*                                                  *#
#*                                                  *#
AllQueueManagers:
#*                                                  *#
#* The path to the qmgrs directory, below which queue manager data *#
#* is stored                                       *#
#*                                                  *#
DefaultPrefix=mqs_root:[mqm]

ClientExitPath:
  ExitsDefaultPath=mqs_root:[mqm.exits]

LogDefaults:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=1024
  LogType=CIRCULAR
  LogBufferPages=17
  LogDefaultPath=mqs_root:[mqm.log]

QueueManager:
  Name=saturne.gest.files
  Prefix=mqs_root:[mqm]
  Directory=saturne$gest$files

DefaultQueueManager:
  Name=saturne.gest.files

QueueManager:
  Name=pluton.gest.files
  Prefix=mqs_root:[mqm]
  Directory=pluton$gest$files

```

Figure 19. Exemple de fichier de configuration MQSeries pour les systèmes MQSeries pour Compaq OpenVMS

La figure 20 à la page 190 représente les groupes d'attributs susceptibles d'être présents dans un fichier de configuration de gestionnaire de files d'attente sous MQSeries pour Compaq OpenVMS.

## Modification du fichier de configuration de MQSeries

```
#####  
** Module Name: qm.ini                **  
** 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 **  
**          **  
#####  
ExitPath:  
  ExitsDefaultPath=mqm_root:[mqm.exits]  
  
Service:  
  Name=AuthorizationService  
  EntryPoints=9  
  
ServiceComponent:  
  Service=AuthorizationService  
  Name=MQSeries.UNIX.auth.service  
  Module=amqzfu  
  ComponentDataSize=0  
  
Service:  
  Name=NameService  
  EntryPoints=5  
  
ServiceComponent:  
  Service=NameService  
  Name=MQSeries.DCE.name.service  
  Module=amqzfa  
  ComponentDataSize=0  
  
Log:  
  LogPrimaryFiles=3  
  LogSecondaryFiles=2  
  LogFilePages=1024  
  LogType=CIRCULAR  
  LogBufferPages=17  
  LogPath=mqm_root:[mqm.log.saturne$gest$files]  
  
XAResourceManager:  
  Name=Oracle Resource Manager Bank  
  SwitchFile=sys$share:oraswit0.exe  
  XAOpenString=MQBankDB  
  XACloseString=  
  ThreadOfControl=PROCESS  
  
CHANNELS:  
  MaxChannels = 20           ; Maximum number of Channels allowed.  
                             ; Default is 100.  
  MaxActiveChannels = 10    ; Maximum number of Channels allowed to be  
                             ; active at any time. The default is the  
                             ; value of MaxChannels.  
  
TCP:  
  KeepAlive = Yes           ; TCP/IP entries.  
                             ; Switch KeepAlive on
```

Figure 20. Exemple de fichier de configuration de gestionnaire de files d'attente pour MQSeries pour Compaq OpenVMS

## Modification du fichier de configuration de MQSeries

### Notes :

Le programme MQSeries installé sur le noeud utilise les emplacements par défaut pour les gestionnaires de files d'attente et les journaux.

saturne.gest.files est le gestionnaire de files d'attente par défaut sur le noeud. Le nom du répertoire contenant les fichiers associés à ce gestionnaire de files d'attente a été automatiquement transformé en nom de fichier admis par le système de fichiers.

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.



---

## Chapitre 14. Identification des incidents

Ce chapitre propose plusieurs méthodes qui vous permettront de résoudre les incidents que vous pourriez rencontrer au cours de l'utilisation de MQSeries pour Compaq OpenVMS.

Tous les incidents ne peuvent pas toujours être résolus immédiatement (par exemple, certains problèmes de performance peuvent être provoqués par les limitations de votre matériel). 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 196
- «Vérifications approfondies» à la page 197
- «Considérations sur le développement d'applications» à la page 199
- «Sortie incorrecte» à la page 201
- «Journaux d'erreurs» à la page 203
- «Files d'attente de rebut» à la page 208
- «Fichiers de configuration et identification des incidents» à la page 208
- «Traçage de MQSeries» à la page 208
- «First failure support technology (FFST)» à la page 209
- «Résolution des incidents liés aux clients» à la page 214

---

### Vérifications préalables

Les incidents rencontrés lors de l'utilisation de MQSeries peuvent être liés aux éléments suivants :

- MQSeries
- le réseau,
- l'application.
- le système d'exploitation.

Les sections qui suivent abordent des points essentiels à considérer lorsque vous cherchez la cause de l'incident.

### MQSeries a-t-il fonctionné correctement auparavant ?

Si MQSeries n'a jamais fonctionné, il est probable que vous ne l'avez pas encore configuré correctement. Reportez-vous au manuel *MQSeries pour Compaq OpenVMS Alpha - Mise en route, Version 5.1* pour vérifier que vous avez installé et configuré le produit 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 203.

## 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 du code de retour.

### L'incident peut-il être reproduit ?

Si oui, étudiez les conditions dans lesquelles il peut être reproduit :

- L'incident est-il provoqué par une commande ou une demande 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 ? L'opération a-t-elle échoué sur tous les systèmes MQSeries et tous les gestionnaires de files ou seulement sur certains ?
- 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.

### 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 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 ?

### 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.

Avant de répondre **Oui** à cette question, 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. Étudiez 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'opération 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 ?

### **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.

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 traduction, la compilation ou l'édition des liens de l'application dans 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 196 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 ?

## Vérifications préalables

### 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).

### 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.

### Une mise à jour est-elle nécessaire ?

MQSeries dépend du système d'exploitation sous-jacent (OpenVMS) et de divers éléments de mise en réseau tels que TCP/IP. Vérifiez auprès du constructeur du produit si vous avez effectué toutes les mises à jour nécessaires.

---

## 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. Certaines d'entre elles peuvent être à l'origine de l'incident constaté.

- 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*.



## Vérifications approfondies

A ce stade, les contrôles préliminaires vous ont peut-être permis de déterminer la cause de l'incident. Dans ce cas, vous devriez pouvoir le résoudre, peut-être avec l'aide d'autres manuels de la bibliothèque MQSeries (voir la section Bibliographie) et des bibliothèques d'autres programmes sous licence.

Si l'origine reste indéterminée, vous devez entamer une recherche plus approfondie.

Le but de cette section est de vous aider à identifier la cause de l'incident si les contrôles préliminaires ne vous ont pas permis de la découvrir.

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 199

Si aucune de ces descriptions ne semble correspondre au symptôme, l'incident est peut-être provoqué par un autre composant de votre système.

### Avez-vous obtenu une sortie incorrecte ?

Dans ce manuel, «sortie incorrecte» s'applique 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.

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.

Vérifiez dans le journal des erreurs si un message d'erreur MQSeries si un message d'erreur préfixé par «AMQ» a été généré. Consultez la section «Journaux d'erreurs» à la page 203 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 à ce sujet.

## Vérifications approfondies

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 203 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 d'application pour insérer des commandes dans la file SYSTEM.ADMIN.COMMAND.QUEUE, faut-il effectuer un point de synchronisation ?  
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 ou du système MQSeries dans son entier. Commencez par arrêter les gestionnaires de files d'attente un à un pour essayer d'isoler un gestionnaire défaillant. Si cette démarche n'aboutit pas, arrêtez et redémarrez MQSeries. Traitez les messages d'erreur consignés à cette occasion dans le journal des erreurs.

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.
  - 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 ?

## Vérifications approfondies

- 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 l'initiateur de canal 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 le manuel *MQSeries Intercommunication* pour obtenir des informations sur la définition des canaux.

---

## 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.

Pour plus d'informations sur le développement d'applications, reportez-vous au *Guide de programmation d'applications*.

## Considérations sur le développement d'applications

### Effet sur 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.

### Effet de la persistance d'un message

Les messages persistants sont journalisés. La journalisation des messages réduit les performances de votre application ; aussi, n'utilisez des messages persistants que pour les données essentielles. Si les données d'un message peuvent être supprimées lorsqu'un gestionnaire de files d'attente s'arrête ou échoue, utilisez un message non persistant.

### 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 mesuré 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 entre-temps.

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 OpenVMS.

### 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 et d'unités d'exécution 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 197.

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 ?

## sortie incorrecte

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 ?
- 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'ait 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 applicative 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.

- 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 l'ensemble de données n'est pas reconnu pour la conversion, l'exit de conversion est activé pour vous permettre d'effectuer cette opération avec vos propres sous-programmes.

La seule exception est lorsque vous envoyez des données à MQSeries pour MVS/ESA.

Consultez le manuel *MQSeries Intercommunication* pour plus d'informations sur la conversion de données.

---

## Journaux d'erreurs

MQSeries 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.

## Journaux d'erreurs

- Si le nom du gestionnaire de files d'attente est connu et que le gestionnaire est disponible :  
MQS\_ROOT: [MQM.QMGRS.NomGestFiles.ERRORS]AMQERR01.LOG
- Si le gestionnaire de files d'attente n'est pas disponible :  
MQS\_ROOT: [MQM.QMGRS.\$SYSTEM.ERRORS]AMQERR01.LOG
- Si une erreur s'est produite avec une application client :  
MQS\_ROOT: [MQM.ERRORS]AMQERR01.LOG
- First Failure Support Technology\* (FFST) – voir «Consultation des fichiers FFST» à la page 209.

**Remarque :** Dans le cas de clients, les erreurs sont stockées en racine.

## Fichiers journaux

Au moment de l'installation, un répertoire [MQM.QMGRS.\$SYSTEM.ERRORS] est créé dans le chemin du fichier QMGRS. Le sous-répertoire peut contenir trois fichiers de journalisation des erreurs appelés :

- AMQERR01.LOG
- AMQERR02.LOG
- AMQERR03.LOG

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 ont les mêmes noms que ceux de \$SYSTEM, c'est-à-dire AMQERR01, AMQERR02, et AMQERR03 et la taille de chacun est de 256 Ko. Les fichiers sont placés dans le sous-répertoire d'erreurs de chaque gestionnaire de files créé.

Les messages d'erreur générés sont enregistrés dans AMQERR01. Quand AMQERR01 dépasse 256 ko, ils sont copiés dans AMQERR02. Avant la copie, AMQERR02 est copié dans AMQERR03.LOG. Le contenu précédent de AMQERR03, s'il existe, est écrasé.

Les derniers messages d'erreur sont toujours placés dans AMQERR01, 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 sous-répertoire [MQM.QMGRS.\$SYSTEM.ERRORS].

Pour examiner le contenu d'un fichier journal d'erreurs, utilisez votre éditeur OpenVMS habituel.

## 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 un répertoire des erreurs créé au moment de l'installation dans le répertoire racine, mqm.



Si le fichier de configuration MQSeries est lisible, ainsi que l'attribut DefaultPrefix de la strophe AllQueueManagers, les erreurs sont consignées dans le répertoire DefaultPrefix[.errors].

Pour plus d'informations sur les fichiers de configuration, reportez-vous au «Chapitre 13. Configuration de MQSeries» à la page 173.

## Messages opérateur

Sous MQSeries pour Compaq OpenVMS, Les messages opérateur identifient les erreurs normales, en général provoquées directement par les utilisateurs indiquant, par exemple, des paramètres incorrects dans une commande. Les messages opérateur sont affichés dans la langue nationale (NLS) si les catalogues de messages sont installés aux emplacements standard.

Ces messages sont écrits dans la fenêtre associée, si elle existe, ainsi que dans le journal des erreurs AMQERR01.LOG du répertoire du gestionnaire de files d'attente. Par exemple :

```
MQS_ROOT: [MQM.QMGRS.QUEUE$MANAGER.ERRORS]
```

De plus, certains messages opérateur sont consignés dans le fichier AMQERR01.LOG du répertoire du gestionnaire de files d'attente et d'autres dans la copie du répertoire \$SYSTEM du journal des erreurs.

## Exemple de journal des erreurs

L'exemple qui suit est une partie d'un journal d'erreur MQSeries pour Compaq OpenVMS :

```
06/29/00 09:41:39 AMQ7467: The oldest log file required to start queue
manager BK1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may
be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the log
files required to recreate objects from their media images, you will
have to restore them to recreate the objects.
-----
06/29/00 09:41:39 AMQ7468: The oldest log file required to perform media
recovery of queue manager BK1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any
log files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory.
-----
06/29/00 09:42:05 AMQ7467: The oldest log file required to start queue
manager BK1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this
may be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the log
files required to recreate objects from their media images,
you will have to restore them to recreate the objects.
-----
06/29/00 09:42:05 AMQ7468: The oldest log file required to perform media
recovery of queue manager BK1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any
log files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory.
-----
```

## Journaux d'erreurs

```
06/29/00 09:42:06 AMQ8003: MQSeries queue manager started.

EXPLANATION: MQSeries queue manager BKM1 started.
ACTION: None.
-----
06/29/00 09:42:06 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this
may be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the
log files required to recreate objects from their media images,
you will have to restore them to recreate the objects.
-----
06/29/00 09:42:06 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images.
Any log files prior to this will not be accessed by media recovery
operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 09:46:27 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 : Aucune.
-----
06/29/00 09:46:43 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may be
required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the
log files required to recreate objects from their media images, you
will have to restore them to recreate the objects.
-----
06/29/00 09:46:43 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any
log files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 09:46:44 AMQ8004: MQSeries queue manager ended.

EXPLANATION: MQSeries queue manager BKM1 ended.
ACTION : Aucune.
-----
06/29/00 09:46:59 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this
may be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory. If you move any of the log files
required to recreate objects from their media images, you will have to
restore them to recreate the objects.
-----
06/29/00 09:47:00 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any log
files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
```

```

-----
06/29/00 09:47:08 AMQ7472: Object TEST1, type queue damaged.
EXPLANATION: Object TEST1, type queue has been marked as damaged. This
indicates that the queue manager was either unable to access the object in
the file system, or that some kind of inconsistency with the data in
the object was detected.
ACTION: If a damaged object is detected, the action performed depends on
whether the queue manager supports media recovery and when the damage
was detected. If the queue manager does not support media recovery,
you must delete the object as no recovery is possible. If the queue manager
does support media recovery and the damage is detected during the processing
performed when the queue manager is being started, the queue manager will
automatically initiate media recovery of the object. If the queue
manager supports media recovery and the damage is detected once the queue
manager has started, it may be recovered from a media image using the
rcrmqobj command or it may be deleted.
-----
06/29/00 09:47:09 AMQ8003: MQSeries queue manager started.
EXPLANATION: MQSeries queue manager BKM1 started.
ACTION: None.
-----
06/29/00 09:47:09 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.
EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may be
required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory. If you move any of the log files
required to recreate objects from their media images, you will have to
restore them to recreate the objects.
-----
06/29/00 09:47:10 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.
EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any log
files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 09:47:47 AMQ7081: Object TEST1, type queue recreated.
EXPLANATION: The object TEST1, type queue was recreated from its media
image.
ACTION: None.
-----
06/29/00 11:22:10 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.
EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may
be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory. If you move any of the log files
required to recreate objects from their media images, you will have
to restore them to recreate the objects.
-----
06/29/00 11:22:10 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.
EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any log files
prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 11:22:11 AMQ8004: MQSeries queue manager ended.
EXPLANATION: MQSeries queue manager BKM1 ended.
ACTION: None.
-----
...

```

## 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 (`amqsbcbg`) 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. Lorsque le gestionnaire de files d'attente est créé à l'aide de la commande `crtmqm`, une file d'attente de rebut appelée `SYSTEM.DEAD.LETTER.QUEUE` est automatiquement créée comme objet par défaut. Toutefois, cette file n'est pas définie comme file d'attente de rebut pour le gestionnaire. Consultez la section «Définition d'une file d'attente de rebut» à la page 48.

---

## 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, par exemple :

```
MQS.INI;1 MQM (RWED, RWED, RW, R)
(identifiant=MQM, ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
```
- Vérifiez que le fichier de configuration MQSeries référence les répertoires de journaux et de gestionnaires de files d'attente corrects.

---

## Traçage de MQSeries

MQSeries pour Compaq OpenVMS utilise les commandes suivantes dans le cadre du traçage :

- **strmqtrc** – reportez-vous à la section «strmqtrc (lancement de la trace MQSeries)» à la page 311
- **dspmqtrc** – reportez-vous à la section «dspmqtrc (affichage de la sortie de trace formatée MQSeries)» à la page 266
- **endmqtrc** – reportez-vous à la section «endmqtrc (arrêt de la trace MQSeries)» à la page 277

La fonction de trace utilise un fichier pour chaque entité tracée, les informations de traces étant enregistrées dans le fichier approprié.

Les fichiers associés à la trace sont créés dans le répertoire `MQS_ROOT:[MQM.TRACE]`.

Il comprennent des informations sur les gestionnaires de files d'attente, ainsi que tous les traçages précédents et traçages `$SYSTEM`.

## Fichiers de trace

Les noms de fichiers de trace sont construits selon le modèle suivant:

AMQppppppppp.TRC

où pppppppp désigne l'identificateur de processus (PID) produisant la trace.

### Remarques :

1. Sous MQSeries pour Compaq OpenVMS, l'identificateur de processus comprend toujours huit caractères.
2. Un fichier de trace est créé pour chaque processus faisant partie de l'entité en cours de traçage.

## Exemple de données de trace

L'exemple suivant est un extrait de trace OpenVMS :

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 21. Exemple de trace MQSeries pour Compaq OpenVMS

### 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)

Sous MQSeries pour Compaq OpenVMS, les informations FFST sont enregistrées dans un fichier situé dans le répertoire MQS\_ROOT:[MQM.ERRORS].

Ces erreurs, généralement graves et irrémédiables, indiquent soit un incident de configuration du système, soit une erreur interne MQSeries.

## Consultation des fichiers FFST

Les fichiers sont intitulés AMQnnnnnnnn\_mm.FDC, où:

nnnnnnnn

est l'ID du processus signalant l'erreur,

# FFST

*mm* est le numéro de séquence, normalement 0.

Lorsqu'un processus crée un fichier FFST, il écrit également une entrée dans le journal des erreurs système contenant le nom du fichier FFST, cet enregistrement participe au traçage automatique des incidents.

```
+-----+
MQSeries First Failure Symptom Report
=====
Date/Time      :- Monday January 29 21:32:03 GMT 2001
Host Name      :- CELERY (Unknown)
PIDS           :- 5697175
LVLS          :- 510
Product Long Name :- MQSeries for OpenVMS Alpha
Vendor         :- IBM
Probe Id       :- ZX005025
Application Name :- MQM
Component      :- zxcProcessChildren
Build Date     :- Jan  8 2001
Userid        :- [400,400] (SJACKSON)
Program Name   :- AMQZXMA0.EXE
Process        :- 202001DA
Thread         :- 00000001
QueueManager   :- JJJH
Major Errorcode :- zrcX_PROCESS_MISSING
Minor Errorcode :- OK
Probe Type     :- MSGAMQ5008
Probe Severity :- 2
Probe Description :- AMQ5008: An essential MQSeries process 538968541
cannot be found and is assumed to be terminated.
Arith1         :- 538968541 202001dd
VMS Errorcode  :- -SYSTEM-W-NONEXPR, nonexistent process (000008E8)

JPI Quota information:
=====
ASTCNT=247/250(98%) *          BIOCNT=500/500(100%) *
BYTCNT=183616/183616(100%) *  DIOCNT=250/250(100%) *
ENQCNT=4885/5000(97%) *      FILCNT=241/250(96%) *
PAGFILCNT=975280/1000000(97%) * TQCNT=246/250(98%) *
FREPTCNT=2147483647          APTCNT=0
GPGCNT=5808                  PPGCNT=5872
VIRTPEAK=203264              DFWSCNT=1392
WSAUTH=2784                  WSAUTHEXT=65536
WSEXTENT=65536               WSPEAK=11680
WSQUOTA=2784                 WSSIZE=15792
CPULIM=0                     MAXDETACH=0
MAXJOBS=0                    JOBPRCNT=2
PAGEFLTS=2895                PRCCNT=2/100(2%) +
(*) - % resource remaining, (+) - % resource used

Privilege and rights information:
=====
CURPRIV=bugchk detach netmbx prmgrbl sysgbl sysprv tmpmbx world
IMAGPRIV=bugchk prmgrbl sysgbl world
AUTHPRIV=bugchk detach netmbx prmgrbl sysgbl sysprv tmpmbx world
SJACKSON                      INTERACT
REMOTE                          MQM
SYS
IMAGE_RIGHTS=
SYS$NODE_CELERY

SYI information:
=====
ACTIVE CPU=1/1(100%) +          CLUSTER NODES=1
FREE_GBLPAGES=16000528/16174643(98%) * GBLPAGFIL=1000000
FREE_GBLSECTS=936/1550(60%) *      MEMSIZE=16384
PAGEFILE_FREE=16888/16888(100%) *  PAGE_SIZE=8192
SWAPFILE_FREE=936/936(100%) *      MAXPROCESSCNT=102
PROCSECTCNT=64                   BALSETCNT=100
WSMAX=65536                       NPAGEDYN=2269184
NPAGEVIR=9437184                  PAGEDYN=1597440
VIRTUALPAGECNT=2147483647          LOCKIDTBL_MAX=109437
PQL_DASTLM=24                     PQL_MASTLM=100
PQL_DBYOLM=32                     PQL_MBIOLM=100
PQL_DBYTLM=65536                  PQL_MBYTLM=100000
PQL_DCPULM=0                      PQL_MCPULM=0
+-----+
```

PQL_DDIOLM=32	PQL_MDIOLM=100
PQL_DFILLM=128	PQL_MFILLM=100
PQL_DPGFLQUOTA=65536	PQL_MPGFLQUOTA=32768
PQL_DPRCLM=32	PQL_MPRCLM=10
PQL_DTQELM=16	PQL_MTQELM=0
PQL_DWSDEFAULT=1392	PQL_MWSDEFAULT=1392
PQL_DWSQUOTA=2784	PQL_MWSQUOTA=2784
PQL_DWSEXTENT=65536	PQL_MWSEXTENT=65536
PQL_DENQLM=128	PQL_MENQLM=300
PQL_DJTQUOTA=4096	PQL_MJTQUOTA=0
CLISYMTBL=750	DEFMBXMXMSG=256
DEFMBXBUFQUO=1056	CHANNELCNT=5000
DLCKEXTRASTK=2560	PIOPAGES=575
CTLPAGES=256	CTLIMGLIM=35
(*) - % resource remaining, (+) - % resource used	

MQM Function Stack  
zxcProcessChildren  
xcsFFST

MQM Trace History

```

--> xllFreeSem
<-- xllFreeSem rc=OK
--> xcsFreeQuickCell
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
--> xstFreeCell
<-- xstFreeCell rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
<-- xcsFreeQuickCell rc=OK
<-- xcsCloseEventSem rc=OK
--> xcsFreeMemBlock
--> xstFreeMemBlock
--> xcsRequestThreadMutexSem
<-- xcsRequestThreadMutexSem rc=OK
--> xcsReleaseThreadMutexSem
<-- xcsReleaseThreadMutexSem rc=OK
--> xstFreeBlockFromSharedMemSet
--> xllSpinLockSlowRequest
<-- xllSpinLockSlowRequest rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
--> xstFreeBlockInExtent
--> xcsQueryMutexSem
<-- xcsQueryMutexSem rc=OK
--> xcsRequestMutexSem
--> xllSemReq
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemReq rc=OK
<-- xcsRequestMutexSem rc=OK
--> xclDeleteMutexMem
--> xllCSCloseMutex
--> xihHANDLEtoSUBPOOLFn
--> xihGetConnSPDetailsFromList
--> xihGetConnSPDetails
<-- xihGetConnSPDetails rc=OK
<-- xihGetConnSPDetailsFromList rc=OK
<-- xihHANDLEtoSUBPOOLFn rc=OK
--> xllSpinLockSlowRequest
<-- xllSpinLockSlowRequest rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
--> xllFreeSem
<-- xllFreeSem rc=OK
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
--> xcsFreeQuickCell
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
--> xstFreeCell
<-- xstFreeCell rc=OK
--> xllSpinLockRelease

```



## FFST

```
<-- xllSpinLockRelease rc=OK
<-- xcsFreeQuickCell rc=OK
<-- xllCSCloseMutex rc=OK
<-- xclDeleteMutexMem rc=OK
--> xstSerialiseExtent
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
<-- xstSerialiseExtent rc=OK
--> xstFreeChunk
--> xstDeleteChunk
<-- xstDeleteChunk rc=OK
--> xstInsertChunk
<-- xstInsertChunk rc=OK
<-- xstFreeChunk rc=OK
--> xstReleaseSerialisationOnExtent
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
<-- xstReleaseSerialisationOnExtent rc=OK
<-- xstFreeBlockInExtent rc=OK
<-- xstFreeBlockFromSharedMemSet rc=OK
<-- xstFreeMemBlock rc=OK
<-- xcsFreeMemBlock rc=OK
<-- zcpDeleteIPC rc=OK
--> xcsReleaseMutexSem
--> xllSemRel
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemRel rc=OK
<-- xcsReleaseMutexSem rc=OK
--> xcsFreeMemBlock
--> xstFreeMemBlock
--> xcsRequestThreadMutexSem
<-- xcsRequestThreadMutexSem rc=OK
--> xcsReleaseThreadMutexSem
<-- xcsReleaseThreadMutexSem rc=OK
--> xstFreeBlockFromSharedMemSet
--> xllSpinLockSlowRequest
<-- xllSpinLockSlowRequest rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
--> xstFreeBlockInExtent
--> xcsQueryMutexSem
<-- xcsQueryMutexSem rc=OK
--> xcsRequestMutexSem
--> xllSemReq
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemReq rc=OK
<-- xcsRequestMutexSem rc=OK
--> xclDeleteMutexMem
--> xllCSCloseMutex
--> xihHANDLEtoSUBPOOLFn
--> xihGetConnSPDetailsFromList
--> xihGetConnSPDetails
<-- xihGetConnSPDetails rc=OK
<-- xihGetConnSPDetailsFromList rc=OK
<-- xihHANDLEtoSUBPOOLFn rc=OK
--> xllSpinLockSlowRequest
<-- xllSpinLockSlowRequest rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
--> xllFreeSem
<-- xllFreeSem rc=OK
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
--> xcsFreeQuickCell
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
--> xstFreeCell
<-- xstFreeCell rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
<-- xcsFreeQuickCell rc=OK
<-- xllCSCloseMutex rc=OK
```



```

<-- xclDeleteMutexMem rc=OK
--> xstSerialiseExtent
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
<-- xstSerialiseExtent rc=OK
--> xstFreeChunk
--> xstDeleteChunk
<-- xstDeleteChunk rc=OK
--> xstInsertChunk
<-- xstInsertChunk rc=OK
<-- xstFreeChunk rc=OK
--> xstReleaseSerialisationOnExtent
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
<-- xstReleaseSerialisationOnExtent rc=OK
<-- xstFreeBlockInExtent rc=OK
<-- xstFreeBlockFromSharedMemSet rc=OK
<-- xstFreeMemBlock rc=OK
<-- xcsFreeMemBlock rc=OK
<-- zxcCleanupAgent rc=OK
--> xcsReleaseMutexSem
--> xllSemRel
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemRel rc=OK
<-- xcsReleaseMutexSem rc=OK
--> xcsCheckProcess
--> kill
<-- kill rc=OK
<-- xcsCheckProcess rc=OK
--> xcsCheckProcess
--> kill
<-- kill rc=OK
<-- xcsCheckProcess rc=OK
--> xcsRequestMutexSem
--> xllSemReq
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemReq rc=OK
<-- xcsRequestMutexSem rc=OK
--> xcsReleaseMutexSem
--> xllSemRel
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemRel rc=OK
<-- xcsReleaseMutexSem rc=OK
--> xcsCheckProcess
--> kill
<-- kill rc=Unknown(FFFF)
<-- xcsCheckProcess rc=xecP_E_INVALID_PID
--> xcsBuildDumpPtr
--> xcsGetMem
<-- xcsGetMem rc=OK
<-- xcsBuildDumpPtr rc=OK
<-- xcsBuildDumpPtr rc=OK
<-- xcsBuildDumpPtr rc=Unknown(4B)
--> xcsFFST

```

ECAnchor

6A91B0				5A584541	ZXEA
6A91C0	03000000	E8030000	03220000	0100DA01	...&#232;...&#218;..
6A91D0	2C05A500	D4040000	0A00DA01	01000000	..¥.&#212;.....&#218;.....
6A91E0	B0000000	0A00DA01	03000000	E8030000	°.....&#218;.....&#232;... ..
6A91F0	03220000	0100DA01	07000000	00000000	.".....&#218;.....
6A9200	283F9700	03000000	F0030000	08410000	(?~.....A..
6A9210	0300DA01	03000000	F0030000	08410000	..&#218;.....;...A..
6A9220	0300DA01	01000000	B4000000	0200DA01	..&#218;.....&#218;..
6A9230	03000000	E8030000	03220000	0100DA01	...&#232;.....".....&#218;..
6A9240	00000000	00000000	00000000	00000000	.....
6A9250	00000000	00000000	00000000	00000000	.....
6A9260	00000000	01000000	B4000000	0200DA01	.....&#218;..
6A9270	03000000	E8030000	03220000	0100DA01	...&#232;.....".....&#218;..
6A9280	B41F0000	0200DA01	01000000	B4000000	.....&#218;.....
6A9290	0200DA01	03000000	E8030000	03220000	..&#218;.....&#232;....."

## FFST

```
6A92A0 0100DA01 DA012020 00000000 EFCDD0300 ..&#218;.&#218;. ....&#205;..
6A92B0 00000000 00000000 00000000 00000000 .....
6A92C0 00000000 00000000 01000000 00000000 .....
6A92D0 44454641 554C5400 00000000 00000000 DEFAULT.....
6A92E0 00000000 00000000 00000000 00000000 .....
6A92F0 00000000 00000000 00000000 00000000 .....
6A9300 2F6D7173 5F726F6F 742F6D71 6D000000 /mqs_root/mqm...
6A9310 00000000 00000000 00000000 00000000 .....
6A9320 to 6A93F0 suppressed, lines same as above
6A9400 5A435048 01000000 B8000000 0400DA01 ZCPH....&#184;....&#218;.
6A9410 03000000 F0030000 08410000 0300DA01 .....A....&#218;.
6A9420 DC040000 0400DA01 01000000 B8000000 &#220;....&#218;....&#184;...
6A9430 0400DA01 03000000 F0030000 08410000 ..&#218;.....A..
6A9440 0300DA01 00000000 00000000 00000000 ..&#218;.....
6A9450 00000000 00000000 00000000 00000000 .....
6A9460 to 6A9470 suppressed, lines same as above
6A9480 00000000 00000000 5A435048 01000000 .....ZCPH....
6A9490 B8000000 0500DA01 03000000 F0030000 &#184;....&#218;.....
6A94A0 08410000 0300DA01 DC040000 0500DA01 .A....&#218;.&#220;....&#218;.
6A94B0 01000000 B8000000 0500DA01 03000000 ...&#184;....&#218;....
6A94C0 F0030000 08410000 0300DA01 4C4B0000 .....A....&#218;.LK..
6A94D0 0500DA01 01000000 B8000000 0500DA01 ..&#218;....&#184;....&#218;.
6A94E0 03000000 F0030000 08410000 0300DA01 .....A....&#218;.
6A94F0 07000000 00090000 50140000 0100DA01 .....P....&#218;.
6A9500 01000000 B0000000 0100DA01 03000000 ....°....&#218;....
6A9510 E8030000 03220000 0100DA01 98170000 &#232;...."....&#218;.
6A9520 0200DA01 01000000 B4000000 0200DA01 ..&#218;....&#218;.
6A9530 03000000 E8030000 03220000 0100DA01 ...&#232;...."....&#218;.
6A9540 00000000 00000000 08000000 3C0A0000 .....<...
6A9550 50140000 0100DA01 01000000 B0000000 P....&#218;....°...
6A9560 0100DA01 03000000 E8030000 03220000 ..&#218;....&#232;...."..
6A9570 0100DA01 08180000 0200DA01 01000000 ,&#218;....&#218;....
6A9580 B4000000 0200DA01 03000000 E8030000 .....&#218;....&#232;...
6A9590 03220000 0100DA01 00000000 00000000 ."....&#218;.....
6A95A0 00000000 00000000 00000000 00000000 .....
6A95B0 to 6A9650 suppressed, lines same as above
6A9660 01000000 00000000 78180000 0200DA01 ....,...x....&#218;.
6A9670 01000000 B4000000 0200DA01 03000000 ...&#218;....
6A9680 E8030000 03220000 0100DA01 09000000 &#232;...."....&#218;....
6A9690 780B0000 50140000 0100DA01 01000000 x...P....&#218;....
6A96A0 B0000000 0100DA01 03000000 E8030000 °....&#218;....&#232;...
6A96B0 03220000 0100DA01 00000000 00000000 ."....&#218;.....
6A96C0 00000000 00000000 00000000 00000000 .....
6A96D0 to 6A9750 suppressed, lines same as above
6A9760 00000000 00000000 DD012020 00000000 .....&#221;. ....
6A9770 00000000 00000000 01000000 09000000 .....
6A9780 00000000
```

IBM utilise Function Stack et Trace History pour l'identification des incidents. Dans la plupart des cas, l'administrateur système peut faire peu de chose quand un FFST est généré en dehors de soumettre les incidents aux centres d'assistance support.

Certains problèmes peuvent toutefois être résolus. Si le fichier FFST indique un «quota dépassé» ou un «espace insuffisant sur le disque» lors de l'appel d'une des fonctions internes, il est fort probable que la limite de paramètre SYSGEN a été dépassée.

Pour résoudre cet incident, ajustez les paramètres internes de façon à réduire les limites internes. Pour plus d'informations, consultez le «Chapitre 13. Configuration de MQSeries» à la page 173.

## Résolution des incidents liés aux clients

Les applications clientes MQI reçoivent des codes anomalie MQRC\_\* de la même façon que les applications MQI non clientes. Il existe toutefois des codes anomalie supplémentaires pour les conditions d'erreur associées à des clients. Par exemple :

- si le poste éloigné ne répond pas ;
- si une erreur de ligne de communication a été détectée ;

- si l'adresse du poste n'est pas correcte.

Les erreurs se produisent le plus fréquemment lorsqu'une application émet un appel MQCONN et reçoit la réponse MQRC\_Q\_MQR\_NOT\_AVAILABLE. Un message d'erreur, écrit dans le fichier journal du client, explique la cause de l'erreur. Les messages peuvent aussi être consignés sur le serveur, en fonction de la nature de la défaillance.

### Arrêt des clients

Même après l'arrêt d'un client, le processus sur le serveur peut garder ses files d'attente ouvertes. Normalement, cela ne dure qu'un court moment, le temps que la couche de communication détecte le départ de l'interlocuteur.

### Messages d'erreur sur les postes clients

Lorsqu'une erreur intervient au niveau d'un système client, les messages d'erreur sont insérés dans les fichiers d'erreurs associés au serveur, si possible. Si un message ne peut y être inséré, le code client tente de le placer dans un journal d'erreurs situé dans le répertoire racine du poste client.

#### Systèmes clients OS/2, UNIX, et OpenVMS

Les messages d'erreur des clients OS/2, UNIX, et OpenVMS sont placés dans les journaux des erreurs de leurs systèmes serveurs MQSeries correspondants. Habituellement ces fichiers se trouvent dans le répertoire MQS\_ROOT:[MQM.ERRORS] sur les systèmes OpenVMS et dans /var/mqm/errors sur les systèmes UNIX.

#### Clients DOS et Windows\*

L'emplacement du fichier journal AMQERR01.LOG est défini par la variable d'environnement MQDATA. L'emplacement par défaut, s'il n'est pas modifié par MQDATA, est l'unité :

C:\

Le fait de travailler en environnement DOS implique l'utilisation de la variable d'environnement MQDATA.

Il s'agit de la bibliothèque par défaut utilisée par le code client pour stocker les informations de trace et d'erreur ; elle contient également le nom du répertoire dans lequel est stocké le fichier qm.ini (nécessaire lors de la configuration de NetBIOS). Si rien n'est spécifié, l'emplacement par défaut est l'unité C.

Les noms des fichiers par défaut contenus dans cette bibliothèque sont les suivants :

#### AMQERR01.LOG

Pour les messages d'erreur.

#### AMQERR01.FDC

Pour les messages First Failure Data Capture.

## **résolution des incidents liés aux clients**

---

## Chapitre 15. Optimisation des performances

Ce chapitre présente de quelle façon adapter votre système OpenVMS afin d'optimiser les performances de MQSeries.

Pour un produit tel que MQSeries, il est bien évidemment impossible de définir des valeurs des paramètres OpenVMS qui conviennent à tous les cas. Les valeurs les plus efficaces sont déterminées en fonction de la charge de MQSeries et du système OpenVMS, considérés comme un ensemble. Bien que le manuel *MQSeries pour Compaq OpenVMS Alpha, Version 5.1 Quick Beginnings* propose des valeurs de paramètres qui donnent un point de départ raisonnable, il peut être nécessaire d'augmenter ces valeurs en fonction de la croissance de la charge du gestionnaire de file d'attente. Ce processus est appelé "adaptation".

L'optimisation des performances des systèmes OpenVMS en général est décrite dans le manuel *OpenVMS Performance Management*. Pour optimiser les performances de votre système, utilisez les informations contenues dans le manuel qui vient d'être cité, ainsi que les informations suivantes, particulières à MQSeries :

- Certains paramètres d'optimisation s'appliquent à l'ensemble du système (c'est le cas de GBLPAGES). Ces paramètres sont contrôlés par l'utilitaire SYSGEN, et sont donc souvent appelés paramètres SYSGEN. Utilisé régulièrement, le mécanisme AUTOGEN FEEDBACK surveille les ressources utilisées par le système et ajuste les paramètres SYSGEN pour les adapter à l'évolution de la charge. Ce mécanisme permet de réduire considérablement les interventions manuelles requises pour que le système reste optimisé. Il permet également d'éviter les erreurs dues à l'épuisement des ressources. Les paramètres SYSGEN en rapport avec MQSeries sont les suivants :

### **GBLPAGES, GBLSECTIONS et GBLPAGFIL**

Le gestionnaire de files d'attente fonctionne à l'aide d'un ensemble de processus collaboratifs, qui communiquent via la mémoire partagée (globale). Il faut par conséquent s'assurer que les valeurs des paramètres SYSGEN (GRLPAGES, GRLSECTIONS ET GRLPAGFIL) qui contrôlent cette mémoire globale sont suffisamment élevées. Le manuel *MQSeries pour Compaq OpenVMS Alpha, Version 5.1 Quick Beginnings* propose des valeurs de départ pour ces paramètres. Comme le besoin de mémoire globale augmente avec le nombre d'utilisateurs de MQSeries, il vous faudra également augmenter les paramètres SYSGEN correspondants.

### **CHANNELCNT**

Pour communiquer et se synchroniser, les processus des gestionnaires de files d'attente utilisent les boîtes aux lettres OpenVMS. Ces boîtes aux lettres sont accessibles par des canaux, et il faut s'assurer que la valeur du paramètre SYSGEN CHANNELCNT est suffisante. La valeur fixée lors de l'installation suffira dans la plupart des cas. Cependant, il faudra probablement l'augmenter pour un système très chargé, avec de nombreux processus MQSeries actifs.

Pour définir manuellement les paramètres SYSGEN, il vous faut modifier le fichier MODPARAMS.DAT, ainsi qu'il est décrit dans le manuel *OpenVMS Performance Management*.

## Optimisation des performances

- Certains paramètres OpenVMS s'appliquent à un nom d'utilisateur ou à un processus donné (c'est le cas de PGFLQUOTA). Ces paramètres sont généralement contrôlés (mais pas toujours) par l'utilitaire AUTHORIZE. Il n'existe pas de méthode pour automatiser l'ajustement de ces paramètres. Cependant, étant donné que ces paramètres limitent certaines ressources utilisées par un processus donné, vous pouvez décider de les régler pour une valeur supérieure au minimum indispensable, afin de prévoir les éventuels pics de charge. Les paramètres de processus en rapport avec MQSeries sont les suivants :

### PGFLQUOTA

Ce paramètre contrôle la partie du fichier d'échange (pagefile) qu'un processus peut utiliser. Les processus MQSeries gèrent souvent des messages de grande taille et/ou en grand nombre, et risquent donc d'utiliser une partie importante du fichier d'échanges.

### PRCLM

Ce paramètre contrôle le nombre de sous-processus qu'un processus peut créer. Étant donné que la plupart des processus MQSeries sont créés en tant que sous-processus du contrôleur d'exécution, la valeur de ce paramètre doit être élevée.

### ENQLM, ASTLM, TQELM

Comme nous l'avons indiqué plus haut, le gestionnaire de files d'attente fonctionne à l'aide d'un ensemble de processus collaboratifs. Ces processus synchronisent leurs activités grâce au Lock Manager d'OpenVMS, aux alertes systèmes asynchrones (ASTs) et aux temporisateurs. Les valeurs de ces trois paramètres, qui limitent l'utilisation des ressources, doivent être assez élevées pour répondre aux besoins du gestionnaire de files d'attente.

---

## Définition de la valeur des paramètres spécifiques aux processus

Généralement, c'est l'utilitaire AUTHORIZE qui sert à définir la valeur de ces trois paramètres, pour le nom d'utilisateur approprié (pour MQSeries, ce nom est généralement MQM).

Cependant, dans OpenVMS, certains quotas de processus sont partagés entre tous les processus de la même tâche, c'est-à-dire le processus parent et tous ses sous-processus. Les quotas de processus qui rentrent dans cette catégorie comprennent BYTLM, FILLM, PGFLQUOTA, PRCLM, TQELM et ENQLM. Ils sont appelés *quotas partagés*.

Comme la plupart des processus de gestionnaires de files d'attente sont créés en tant que sous-processus du contrôleur d'exécution, les quotas partagés sont communs à tous ces processus. C'est pourquoi il faut généralement les définir à des valeurs qui peuvent sembler excessives pour un seul processus. C'est tout particulièrement le cas du paramètre PGFLQUOTA, qui limite la quantité de mémoire virtuelle que **l'ensemble** des processus d'un gestionnaire de files d'attente peut créer. C'est pourquoi le contrôleur d'exécution ne définit pas ses valeurs initiales de quota à partir du fichier géré par AUTHORIZE, mais les définit lui-même, lors de son démarrage, à des valeurs raisonnables. En conséquence, AUTHORIZE ne peut pas servir à modifier ces valeurs. Pour les modifier, vous devez utiliser les noms logiques suivants :

```
MQS_ASTLM
MQS_BIOLM
MQS_BYTLM
MQS_DIOLM
```

## Définition de la valeur des paramètres spécifiques aux processus

MQS\_ENQLM  
MQS\_FILLM  
MQS\_PGFLQUOTA  
MQS\_PRCLM  
MQS\_TQELM

Pour définir la valeur de ces noms logiques, utilisez le fichier SYS\$MANAGER:MQS\_SYSTARTUP.COM. MQSeries propose un modèle de ce fichier, SYS\$MANAGER:MQS\_SYSTARTUP.TEMPLATE qui peut être modifié et renommé. Par exemple, pour changer la valeur du paramètre PGFLQUOTA :

1. Recopiez le fichier .TEMPLATE, MQS\_SYSTARTUP.TEMPLATE et changez son extension en .COM.
2. Modifiez ce fichier MQS\_SYSTARTUP.COM pour enlever les caractères de commentaire au début des lignes qui définissent les valeurs des noms logiques correspondant aux quotas de processus, par exemple mqs\_pgflquota.
3. Définissez les nouvelles valeurs.

Par exemple :

```
$! DEFINE/SYSTEM MQS_PGFLQUOTA 1000000
```

doit être changé en :

```
$ DEFINE/SYSTEM MQS_PGFLQUOTA 5000000
```

4. Appelez le fichier mqs\_systartup pour fixer les valeurs. Par exemple :

```
$ @sys$manager:mqs_systartup
```

Ceci se fait en général lors de la procédure de démarrage du système.

Le manque de quotas partagés est généralement en cause quand une nouvelle application client n'arrive pas à se connecter au gestionnaire de files d'attente, ou bien lorsqu'une autre application échoue parce que la nouvelle connexion a consommé trop de ressources.

Veuillez noter également que les paramètres SYSGEN PQL\_D\* ne s'appliquent pas au contrôleur d'exécution. En effet, le contrôleur d'exécution fixe lui-même la plupart des valeurs de ses quotas lors de son démarrage.





---

## Chapitre 16. MQSeries pour OpenVMS et le clustering

Les *clusters OpenVMS* et les *clusters de gestionnaires de files d'attente (GFA) MQSeries* sont deux choses différentes, indépendantes l'une de l'autre.

**Remarque :** Dans ce document, le terme *cluster* désigne un cluster de gestionnaires de files d'attente MQSeries. Un cluster OpenVMS est toujours désigné par le terme *cluster OpenVMS*.

Les clusters de GFA MQSeries n'utilisent pas forcément de protocoles d'intercommunication de cluster OpenVMS, le Distributed Lock Manager OpenVMS ou le système de fichiers de cluster OpenVMS. Toutes les communications entre les GFA et un cluster MQSeries s'effectuent par le biais des canaux MQSeries, à l'aide de l'un des protocoles pris en charge. Il est donc possible de configurer des clusters de GFA MQSeries, même si les GFA sont exécutés sur des systèmes OpenVMS ne faisant pas partie du même cluster OpenVMS.

Si un GFA MQSeries est configuré au sein d'un cluster OpenVMS, il ne peut fonctionner à la fois que sur un seul noeud OpenVMS (désigné par le terme noeud dans le reste de ce chapitre) du cluster OpenVMS. La fonction d'un GFA MQSeries ne peut être distribuée sur plusieurs noeuds d'un cluster OpenVMS. Une erreur est retournée en cas de tentative de démarrage d'un GFA MQSeries sur plus d'un noeud OpenVMS. Toutefois, plusieurs GFA MQSeries configurés dans un cluster OpenVMS peuvent être exécutés sur différents noeuds du cluster.

MQSeries V5.1 intègre une nouvelle fonctionnalité appelée groupe de reprise, qui améliore la disponibilité des GFA MQSeries dans un cluster OpenVMS. Elle permet de redémarrer automatiquement un gestionnaire de files d'attente sur un autre noeud du cluster OpenVMS en cas de panne. Elle peut être utilisée avec ou sans les clusters de GFA MQSeries (consultez la section «Groupe de reprise de cluster OpenVMS» à la page 222).

---

### Installation de MQSeries sur un cluster OpenVMS

L'installation de MQSeries pour Compaq OpenVMS Alpha, V5.1 sur un cluster OpenVMS est quasiment identique à l'installation sur un système OpenVMS autonome. Toutefois, avant de procéder à l'installation, veuillez prendre en compte les informations suivantes :

- Si le cluster OpenVMS est équipé de plusieurs disques système, MQSeries doit être installé sur chaque disque système chargé de l'amorçage d'un noeud *et* devant exécuter MQSeries. MQSeries doit être installé une seule fois par disque système, et non une fois par noeud.
- Le disque contenant la structure de répertoires MQS\_ROOT doit être monté sur tous les noeuds OpenVMS devant exécuter les GFA contenus dans la structure de répertoires. Il est possible d'avoir différentes structures de répertoires MQS\_ROOT pour chaque noeud. Toutefois, si vous devez configurer des groupes de reprise, chaque noeud OpenVMS d'un groupe doit renvoyer à la même structure de répertoires MQS\_ROOT. Lors de l'installation de MQSeries, vous devez indiquer le répertoire MQS\_ROOT (en réponse à la question "Indiquez l'unité racine pour les fichiers de données MQSeries :"), pour chaque installation.

## Installation d'un cluster OpenVMS

- Si le disque contenant les fichiers journaux du GFA est différent du disque contenant MQS\_ROOT, il doit alors être monté sur tous les noeuds du groupe de reprise.
- MQSeries utilise un compte MQM dont le répertoire par défaut est SYS\$SPECIFIC:[MQS\_SERVER]. Ce répertoire est créé uniquement pour le noeud sur lequel MQSeries est installé. Il doit être créé pour chaque noeud supplémentaire amorcé à partir du même disque système *et* devant exécuter MQSeries. Pour ce faire, exécutez les commandes DCL suivantes sur chaque noeud supplémentaire :

```
$create/directory sys$specific:[mqs_server]/owner=[mqs_server] -  
/protection=(s:rwed,o:rwed,g,w)  
$set sec/acl=(identifiant=mqm,options=default,access=r+w+e+d+c) -  
sys$specific:[000000]mqs_server.dir  
$set sec/acl=(identifiant=mqm,access=r+w+e+d+c) -  
sys$specific:[000000]mqs_server.dir
```

---

## Groupe de reprise de cluster OpenVMS

### Présentation des groupes de reprise de cluster OpenVMS

Les groupes de reprise de cluster OpenVMS sont une nouvelle fonctionnalité de MQSeries pour Compaq OpenVMS, V5.1. Ils permettent de redémarrer automatiquement les gestionnaires de files d'attente sur un autre noeud du cluster OpenVMS en cas de problème du GFA MQSeries. Les problèmes gérés par les groupes de reprise de cluster OpenVMS sont les suivants :

- Arrêt d'un noeud OpenVMS exécutant un GFA MQSeries
- Panne système d'un noeud OpenVMS exécutant un GFA MQSeries
- Arrêt d'un noeud OpenVMS exécutant un GFA MQSeries dont la fermeture ne s'est pas effectuée correctement
- Echec lors d'une procédure de Contrôleur d'exécution de GFA MQSeries

Les problèmes suivants ne sont pas gérés par les groupes de reprise de cluster OpenVMS :

- Erreur sur un noeud OpenVMS exécutant un GFA MQSeries et ne provoquant pas l'échec du noeud ou du GFA MQSeries.
- Echec d'une procédure GFA MQSeries (hors procédure de Contrôleur d'exécution). Un GFA MQSeries n'est jamais redémarré automatiquement sur le même noeud.
- Panne matérielle ou logicielle du disque contenant les données de journal et les fichiers GFA MQSeries.
- Altération des fichiers de file d'attente et données de journal du GFA MQSeries.

Les groupes de reprise de cluster OpenVMS sont uniquement pris en charge pour les GFA qui utilisent le protocole TCP/IP pour les canaux MQSeries. Les piles TCP/IP suivantes sont prises en charge :

- Digital\* TCP/IP Services pour OpenVMS V5.0A
- TCPware\* for OpenVMS V5.4 de Process Software
- Multinet\* for OpenVMS 4.3 de Process Software

### Concepts des groupes de reprise de cluster OpenVMS

Un groupe de reprise de cluster OpenVMS est un groupe de noeuds OpenVMS qui peuvent potentiellement exécuter un gestionnaire de files d'attente MQSeries. Un groupe de reprise de cluster OpenVMS peut comprendre entre un et quatre noeuds OpenVMS. Tous doivent être membres du même cluster OpenVMS. Un groupe de reprise est spécifique à un GFA MQSeries. Plusieurs groupes de reprise peuvent être configurés dans un cluster OpenVMS. Pour que l'utilisation des groupes de reprise soit possible, les noms de GFA doivent comprendre moins de 25 caractères.

La *reprise par transfert* ou "failover" est le processus qui permet le redémarrage d'un gestionnaire de files d'attente MQSeries sur un autre noeud OpenVMS, lorsqu'un problème pris en charge se produit. Une fois le processus terminé, on dit que le GFA MQSeries a basculé ou a été transféré.

La *reprise par restauration* ou "failback" est le procédé qui permet le redémarrage d'un gestionnaire de files d'attente MQSeries sur son noeud OpenVMS d'origine une fois le problème résolu. Les groupes de reprise ne prennent pas en charge la remise en ligne automatique, elle doit être effectuée manuellement. Une fois le processus terminé, on dit que le GFA MQSeries a été remis en ligne ou restauré.

Un *moniteur de reprise* est un processus exécuté sur chaque membre d'un groupe de reprise. Ces moniteurs exécutent toutes les fonctions de reprise par transfert. Ils coopèrent ensemble pour procurer ces fonctions. La commande **runmqfm** permet de démarrer un moniteur de reprise (pour plus d'informations sur cette commande, consultez la section «runmqfm (lancement d'un moniteur de reprise)» à la page 293.)

L'un des moniteurs de reprise est désigné comme *moniteur de reprise principal*. On dit alors qu'il est en *observation*. Le premier moniteur de reprise à démarrer dans un groupe de reprise est celui qui est chargé de l'observation. Un groupe de reprise devient *actif* lorsque le premier moniteur de reprise est démarré. Si le moniteur de reprise principal échoue, ou si le noeud sur lequel il est installé tombe en panne, un autre moniteur est automatiquement désigné pour prendre en charge l'observation. Son rôle est de vérifier le fonctionnement du GFA MQSeries et de procéder à la reprise par transfert si une panne supportée se produit. Toute opération devant être effectuée sur un autre noeud OpenVMS est transmise par le moniteur de reprise principal au moniteur de reprise du noeud qui se chargera de l'opération.

Les groupes de reprise de cluster OpenVMS sont administrés par la commande DCL **failover**. Elle peut être utilisée depuis n'importe quel noeud du groupe de reprise. Toutes les commandes sont envoyées au moniteur de reprise principal qui décide alors quel moniteur de reprise devra traiter la commande. Il la transmet si nécessaire à un autre moniteur de reprise.

Le fichier de configuration du groupe de reprise contient les informations détaillées sur le groupe de reprise du cluster OpenVMS. Il comprend entre autres le nombre et les noms des noeuds. Nommé **FAILOVER.INI**, ce fichier se trouve dans le répertoire **MQS\_ROOT:[MQM.QMGRS.nomGFA]**. Il s'agit d'un fichier texte modifiable à l'aide d'un éditeur de texte. Il doit être créé avant la mise en route du premier moniteur de reprise. Un modèle de fichier de configuration, nommé **FAILOVER.TEMPLATE**, figure dans le répertoire **MQS\_EXAMPLES**. Les paramètres qu'il contient ne peuvent pas être modifiés dynamiquement. Pour qu'une modification soit prise en compte, tous les moniteurs de reprise doivent être

## Concepts des groupes de reprise de cluster OpenVMS

arrêtés puis redémarrés. Cette procédure doit faire l'objet de beaucoup d'attention, car la reprise par transfert automatique du GFA MQSeries ne peut avoir lieu lorsque les moniteurs sont arrêtés.

Pour les GFA MQSeries d'un groupe de reprise, toutes les commandes MQSeries continuent de fonctionner normalement, à l'exception de **strmqm** et de **endmqm**. Ces deux commandes retournent une erreur lorsqu'un GFA MQSeries se trouve dans un groupe de reprise actif. La commande **failover** doit être utilisée pour démarrer et arrêter le GFA MQSeries.

La priorité de noeud OpenVMS est celle qui est donnée à chaque noeud OpenVMS du groupe de reprise. Elle sert à déterminer sur quel noeud OpenVMS le GFA doit être démarré en cas de panne. Le noeud OpenVMS dont la valeur de priorité numérique est la plus basse a la priorité la plus importante.

L'adresse TCP/IP du groupe de reprise du cluster OpenVMS est celle qui est affectée au groupe de reprise. Tous les canaux qui renvoient au GFA du groupe de reprise doivent être configurés de façon à indiquer cette adresse TCP/IP dans le nom de connexion. Chaque groupe de reprise d'un cluster OpenVMS doit utiliser une adresse TCP/IP unique. Tous les noeuds OpenVMS du groupe de reprise doivent avoir une adresse TCP/IP d'interface dans le même sous-réseau. L'adresse TCP/IP du groupe de reprise doit aussi faire partie de e même sous-réseau

## Préparation à la configuration d'un groupe de reprise dans un cluster OpenVMS

Procédez de la façon suivante avant de configurer un groupe de reprise :

1. Créez un GFA à l'aide de **crtmqm**, s'il n'existe pas déjà.
2. Définissez une adresse TCP/IP pour le groupe de reprise.
3. Créez ou modifiez les canaux MQSeries de façon à utiliser l'adresse TCP/IP du groupe de reprise.
4. Déterminez quels noeuds OpenVMS doivent figurer dans le groupe de reprise et définissez leurs priorités.
5. Veillez à ce que la variable logique MQS\_ROOT indique le même répertoire sur tous les noeuds OpenVMS du groupe de reprise, et que le disque soit monté sur tous les noeuds. Les disques contenant le répertoire MQS\_ROOT et les fichiers journaux ne doivent pas utiliser le protocole MSCP entre les noeuds d'un groupe de reprise, car si le noeud d'origine devient indisponible, le disque ne pourra plus être accédé par le noeud destination

## Configuration d'un groupe de reprise dans un cluster OpenVMS

Procédez de la façon suivante pour configurer un groupe de reprise de cluster OpenVMS :

1. Copiez le fichier MQS\_EXAMPLES:FAILOVER.TEMPLATE vers  
MQS\_ROOT:[MQM.QMGRS.*nomGFA*]FAILOVER.INI.
2. Editez le fichier MQS\_ROOT:[MQM.QMGRS.*nomGFA*]FAILOVER.INI et modifiez-le pour cette configuration de groupe de reprise. (Consultez la section «Edition du fichier de configuration FAILOVER.INI» à la page 225.)
3. Editez les procédures de commande **START\_QM.COM**, **END\_QM.COM** et **TIDY\_QM.COM**. (Consultez la section «Procédures de commande utilisées par les groupes de reprise» à la page 226.)

## Configuration des groupes de reprise de cluster OpenVMS

4. Définissez une sécurité ICC pour l'Association ICC utilisée par les moniteurs de reprise (consultez la section «Paramétrage de la sécurité pour les associations ICC» à la page 233.)
5. Démarrez un moniteur de reprise sur chaque noeud du groupe de reprise à l'aide de la commande **runmqfm -m nomGFA**.
6. Démarrez le GFA à l'aide de la commande **failover -m nomGFA -n nomnoeud -s**.
7. Modifiez la procédure d'arrêt spécifique au site pour :
  - Arrêter ou déplacer le GFA s'il s'exécute sur un noeud en cours d'arrêt.
  - Arrêter le moniteur de reprise.

## Tâches à effectuer après la configuration d'un groupe de reprise dans un cluster OpenVMS

Vous pouvez effectuer les tâches suivantes une fois votre groupe de reprise configuré :

- Editez le fichier de démarrage du système pour mettre en route le moniteur de reprise sur chaque noeud du groupe de reprise, à l'aide de la commande **runmqfm**. Exécutez la commande **runmqfm** après la commande de démarrage de MQSeries.
- Si vous souhaitez démarrer automatiquement le GFA, après le démarrage du système, et une fois le moniteur de reprise lancé, placez une commande dans le démarrage système du noeud concerné. La commande de démarrage du GFA sur un noeud est **failover -m nomGFA -n nomnoeud -s**.
- Modifiez la procédure d'arrêt du site pour stopper le moniteur de reprise lors de l'arrêt du système. De même, pour arrêter ou déplacer le GFA s'il est exécuté sur un noeud en cours d'arrêt.

## Edition du fichier de configuration FAILOVER.INI

Le fichier FAILOVER.INI doit être personnalisé pour chaque groupe de reprise. La signification de chaque champ est indiquée dans le tableau 9 à la page 225. Le modèle de configuration fourni dans MQS\_EXAMPLES figure dans l'«Annexe F. Modèles de groupe de reprise de cluster OpenVMS» à la page 333. Les lignes du fichier commençant par un caractère '#' sont ignorées lors de l'exécution du moniteur de reprise. La casse des caractères des noms de fichiers doit être la même que dans le fichier modèle. Chaque nom de fichier doit être suivi d'un caractère "=" et de la valeur associée. Tous les champs du modèle sont obligatoires : aucun champ ne doit donc être supprimé.

Tableau 9. Description des champs du fichier FAILOVER.INI

Nom du champ	Description
IpAddress	Adresse TCP/IP devant être utilisée par le groupe de reprise
PortNumber	Numéro de port TCP/IP utilisé par le programme d'écoute pour le GFA
TimeOut	Ce délai d'inactivité est transmis à la procédure EndCommand. Consultez la section «Procédures de commande utilisées par les groupes de reprise» à la page 226.
StartCommand	Commande utilisée pour démarrer le GFA
EndCommand	Commande utilisée pour arrêter le gestionnaire de file d'attente
TidyCommand	Commande utilisée pour se rattacher à un noeud, suite à une panne de GFA dans laquelle le noeud reste actif

## Edition de failover.ini

Tableau 9. Description des champs du fichier FAILOVER.INI (suite)

Nom du champ	Description
LogDirectory	Répertoire contenant les fichiers journaux créés par les procédures StartCommand, EndCommand et TidyCommand
NodeCount	Nombre de noeuds d'un groupe de reprise. Le nombre de triplets de paramètres de noeuds est défini d'après ce champ et doit y correspondre. Le nombre maximum de noeuds pris en charge est de quatre.
NodeName	Le nom de noeud du noeud. Il s'agit de la valeur précisée pour le paramètre système SCSNODE OpenVMS.
Interface	Nom d'interface TCP/IP du noeud lors de l'utilisation de Digital TCP/IP Services pour la pile TCP/IP OpenVMS. Il est affiché par la commande \$tcpip show interface. Ce champ n'est pas utilisé pour les piles TCP/IP TCPware pour OpenVMS ou Multinet pour OpenVMS mais la valeur par défaut de we0 doit cependant être précisée (ne supprimez pas le champ du fichier de configuration.)
Priority	Priorité donnée à un noeud au sein d'un groupe de reprise. La valeur doit être comprise entre 1 et 10. Une valeur égale à 1 est le niveau de priorité le plus élevé. Plusieurs noeuds peuvent avoir la même priorité. Si une panne se produit, ou si aucun noeud spécifique n'est précisé par une commande <b>failover -s</b> ou <b>-f</b> , le GFA est démarré sur le noeud disponible doté de la priorité la plus importante.

## Procédures de commande utilisées par les groupes de reprise

Les groupes de reprise utilisent trois procédures de commande pour remplir certaines fonctions. L'emplacement de ces commandes est indiqué dans le fichier de configuration FAILOVER.INI par les noms de champs StartCommand, EndCommand et TidyCommand. Les modèles de ces commandes (START\_QM.TEMPLATE, END\_QM.TEMPLATE et TIDY\_QM.TEMPLATE) figurent dans MQS\_EXAMPLES. Ces fichiers figurent dans l'«Annexe F. Modèles de groupe de reprise de cluster OpenVMS» à la page 333.

Cinq ou six paramètres sont transmis à ces commandes. Ils sont indiqués dans le tableau 10 à la page 226:

Tableau 10. Paramètres transmis aux procédures de commande

Paramètre	Valeur
P1	Nom du GFA
P2	Nom du répertoire du GFA
P3	Adresse TCP/IP du cluster
P4	Nom d'interface du noeud
P5	Numéro de port du programme d'écoute
P6	Délai d'expiration d'arrêt du GFA (procédure EndCommand seulement)

La procédure StartCommand est utilisée pour démarrer le GFA dans les cas suivants :

- Lorsque l'indicateur **-s** de la commande **failover** est précisé explicitement.
- Lorsque le GFA est déplacé vers un autre noeud OpenVMS à l'aide de l'indicateur **-f** de la commande **failover**.



- Lorsqu'il est redémarré automatiquement après une panne.

Par défaut, la procédure StartCommand configure l'adresse TCP/IP du groupe de reprise sur le noeud devant exécuter le GFA, puis démarre celui-ci à l'aide de la commande **strmqm -m nomGFA**. Selon les besoins du système, la procédure de commande peut être modifiée de la façon suivante :

- Modification de la commande **strmqm**
- Ajout de commandes pour lancer des processus MQSeries supplémentaires tels que le programme d'écoute
- Ajout de commandes de démarrage de processus applicatifs

La procédure StartCommand doit s'achever avec un état égal à 1 pour que le GFA puisse être surveillé une fois démarré.

La procédure EndCommand est utilisée pour arrêter un GFA dans les circonstances suivantes :

- Lorsque l'indicateur -e de la commande **failover** est précisé explicitement.
- Lorsque le GFA est déplacé vers un autre noeud OpenVMS à l'aide de l'indicateur -f de la commande **failover**.

Par défaut, la procédure EndCommand tente d'arrêter le GFA à l'aide de la commande **endmqm -i nomGFA**. Si le GFA ne s'est pas arrêté dans le délai précisé dans le fichier de configuration, la procédure EndCommand tente de l'arrêter à l'aide de la commande **endmqm -i nomGFA**. Si la file d'attente n'est toujours pas arrêtée à la fin d'un autre délai d'expiration, il est possible d'y mettre fin en supprimant le processus de Contrôleur d'exécution. Une fois le GFA arrêté, l'adresse TCP/IP du groupe de reprise est déconfigurée. Si le GFA est arrêté correctement par les commandes **endmqm**, un statut SS\$\_NORMAL est retourné. Si le GFA est arrêté en supprimant le Contrôleur d'exécution, le statut SS\$\_ABORT est retourné. Si le GFA n'est toujours pas arrêté au bout du troisième délai d'expiration, le statut SS\$\_TIMEOUT est indiqué. Ces statuts sont utilisés par le moniteur de reprise principal pour déterminer le résultat de la procédure EndCommand et définit l'état du groupe de reprise en conséquence. Selon les besoins du système, la procédure de commande peut être modifiée de la façon suivante :

- Ajout de commandes pour arrêter d'autres processus MQSeries tels que le programme d'écoute
- Ajout de commandes d'arrêt de processus applicatifs

La commande TidyCommand est utilisée pour rattacher un noeud OpenVMS si le GFA échoue mais que le noeud OpenVMS continue de fonctionner.

Par défaut, la procédure TidyCommand déconfigure l'adresse TCP/IP du groupe de reprise. Selon les besoins du système, la procédure de commande peut être modifiée de la façon suivante :

- Ajout de commandes d'arrêt des processus MQSeries encore en cours d'exécution, tels que le programme d'écoute
- Ajout de commandes d'arrêt de processus applicatifs encore en cours d'exécution

Les fichiers modèles sont configurés par défaut pour utiliser les commandes de Digital TCP/IP Services pour OpenVMS, afin de configurer et déconfigurer l'adresse TCP/IP. Si vous utilisez TCPware for OpenVMS ou MultiNet for

## Procédures de commande

OpenVMS, désactivez les commandes Digital TCP/IP Services for OpenVMS et activez les commandes TCPware for OpenVMS ou MultiNet for OpenVMS.

### Administration des groupes de reprise

Les groupes de reprise doivent être gérés depuis un compte SYSTEM ou un compte d'administration MQSeries. Ils sont gérés à l'aide de deux commandes DCL : **runmqfm** et **failover**. La commande **runmqfm** est utilisée pour démarrer les moniteurs de **reprise par transfert** et la commande **failover** traite toutes les autres tâches d'administration. Ces commandes sont décrites dans les sections «runmqfm (lancement d'un moniteur de reprise)» à la page 293 et «failover (gestion d'un groupe de reprise)» à la page 278.

### Démarrage des moniteurs de reprise

Les moniteurs de reprise sont démarrés sur un noeud OpenVMS en lançant la commande **runmqfm** depuis ce noeud. Par exemple, pour démarrer un moniteur de reprise du GFA TESTQM, utilisez la commande :

```
$ runmqfm -m TESTQM
```

Ceci a pour effet de créer un processus distinct avec un nom basé sur celui du GFA et finissant par **\_FM**. Dans cet exemple, le nom de processus est **TESTQM\_FM**. Ce processus figure dans un écran actif monmq.

Si un fichier journal est nécessaire, il suffit de le préciser en redirigeant le résultat de la commande **runmqfm**. Des informations de débogage supplémentaires peuvent être affichées dans le fichier journal en ajoutant l'indicateur **-d**. Par exemple :

```
$ runmqm -m TESTQM -d > sys$manager:fm.log
```

Notez que la commande **runmqfm** démarre les processus de moniteur de reprise, et non le GFA.

### Démarrage d'un GFA dans un groupe de reprise

Pour démarrer un GFA dans un groupe de reprise, au moins un moniteur de reprise doit fonctionner, dont celui du noeud sur lequel vous souhaitez démarrer le GFA. Le GFA est démarré à l'aide de l'indicateur **-s** de la commande **failover**. La commande peut être exécutée depuis n'importe quel noeud du groupe de reprise. Par exemple, si vous souhaitez démarrer le GFA TESTQM sur le noeud BATMAN, utilisez la commande suivante :

```
$ failover -m TESTQM -n BATMAN -s
```

Si vous devez démarrer le GFA sur le noeud OpenVMS ayant le niveau de priorité le plus élevé disponible, utilisez la commande sans l'indicateur **-n**. Par exemple :

```
$ failover -m TESTQM -s
```



## Démarrage des moniteurs de reprise

Notez qu'une fois qu'un moniteur de reprise est démarré pour un GFA (sur n'importe quel noeud), toute tentative d'utiliser la commande **strmqm** pour démarrer le GFA échoue. Cependant, une fois tous les moniteurs de reprise arrêtés pour un GFA, la commande **strmqm** peut être utilisée normalement.

## Arrêt d'un GFA dans un groupe de reprise

Pour arrêter un GFA dans un groupe de reprise, un moniteur de reprise doit être actif sur le noeud sur lequel est exécuté le GFA. Le GFA est arrêté à l'aide de l'indicateur **-e** de la commande **failover**. La commande peut être exécutée depuis n'importe quel noeud du groupe de reprise. Par exemple, si vous souhaitez arrêter le GFA TESTQM sur le noeud BATMAN, utilisez la commande suivante :

```
$ failover -m TESTQM -e
```

Notez qu'une fois qu'un moniteur de reprise est démarré pour un GFA (sur n'importe quel noeud), toute tentative d'utiliser la commande **endmqm** pour démarrer le GFA échoue. Toutefois, une fois tous les moniteurs de reprise arrêtés pour un GFA, la commande **endmqm** peut être utilisée normalement.

## Déplacement d'un GFA dans un groupe de reprise

Pour déplacer un GFA dans un groupe de reprise, il faut arrêter le GFA sur le noeud où il est actuellement exécuté puis le redémarrer sur un autre noeud du groupe de reprise. Pour déplacer un GFA dans un groupe de reprise, un moniteur de reprise doit être exécuté sur le noeud où le GFA est actif, et un autre sur le noeud sur lequel vous souhaitez déplacer le GFA.

Un GFA est déplacé à l'aide de l'indicateur **-f** de la commande **failover**. La commande peut être exécutée depuis n'importe quel noeud du groupe de reprise. Par exemple, si vous souhaitez déplacer le GFA TESTQM vers le noeud ROBIN, utilisez la commande suivante :

```
$ failover -m TESTQM -n ROBIN -f
```

Si vous devez déplacer le GFA vers le noeud OpenVMS ayant le niveau de priorité le plus élevé disponible, retirez l'indicateur **-n** de la commande. Par exemple :

```
$ failover -m TESTQM -f
```

## Affichage de l'état du groupe de reprise

Trois types d'états décrivent le statut global du groupe de reprise :

- L'état d'un GFA dans un groupe de reprise
- Les états des GFA des noeuds du groupe de reprise (un pour chaque noeud)
- Les états des moniteurs de noeud du groupe de reprise (un pour chaque noeud)

## Déplacement d'un GFA

Les valeurs possibles de ces états sont indiquées dans les trois tableaux suivants.

Tableau 11. Etat du GFA dans un groupe de reprise

Etat	Description
STOPPED	Le GFA n'a jamais été démarré dans le groupe de reprise ou a été arrêté proprement
STARTED	Le GFA a été démarré dans le groupe de reprise. Le groupe de reprise essaiera de redémarrer le GFA en cas de panne de ce dernier.

Tableau 12. Etats des GFA des noeuds du groupe de reprise

Etat	Description
AVAILABLE	Le noeud est prêt à démarrer le GFA si un problème se produit sur un autre noeud.
RUNNING	Le GFA fonctionne actuellement sur ce noeud.
EXCLUDED	Le GFA a été arrêté de façon incorrecte sur ce noeud, sans que celui-ci ne tombe en panne. Si un GFA tombe en panne sur un autre noeud, il ne sera pas redémarré sur ce noeud.

Tableau 13. Etats des moniteurs de noeud du groupe de reprise

Etat	Description
STARTED	Un moniteur de reprise par transfert est en cours d'exécution sur ce noeud, mais il ne s'agit pas de celui qui est chargé de la surveillance.
WATCHING	Le moniteur de reprise exécuté sur ce noeud procède à la surveillance du groupe de reprise.
STOPPED	Aucun moniteur de reprise n'est exécuté sur ce noeud

L'état d'un groupe de reprise s'affiche à l'aide de l'indicateur -q de la commande **failover**. Au moins un moniteur de reprise doit être exécuté. La commande peut être exécutée depuis n'importe quel noeud du groupe de reprise. Par exemple, pour afficher l'état d'un groupe de reprise pour le GFA TESTQM, utilisez la commande suivante :

```
$ failover -m TESTQM -q
```

Voici un exemple de résultat de commande :

```

83H8439, 5697-270 (C) Copyright IBM Corp. 1996. ALL RIGHTS RESERVED.

OpenVMS Cluster Failover Set - Configuration and State.

Queue Manager Name           : TESTQM
Sequence No                   : 11
TCP/IP Address                 : 10.20.30.40
Listener Port Number          : 1414
Timeout to end the Queue Manager : 30
Queue Manager state in Failover Set : STARTED

OpenVMS Node - Configuration and State

Node name                      : BATMAN
Priority                        : 2
TCP/IP Interface                : we0
Queue Manager state            : RUNNING
Failover Monitor state         : WATCHING

Node name                      : ROBIN
Priority                        : 1
TCP/IP Interface                : we0
Queue Manager state            : EXCLUDED
Failover Monitor state         : STARTED
    
```

## Paramétrage des symboles DCL pour indiquer l'état d'un groupe de reprise

Dans certains cas, il peut s'avérer nécessaire d'écrire des procédures de commande DCL pour contrôler les groupe de reprise. L'indicateur -l de la commande **failover** définit trois symboles DCL locaux indiquant l'état du groupe de reprise. Ces symboles peuvent alors être utilisés pour entamer des procédures conditionnelles en fonction de l'état du GFA. Au moins un moniteur de reprise doit être exécuté. La commande peut être exécutée depuis n'importe quel noeud du groupe de reprise. Les symboles définis sont indiqués dans le tableau 14.

Tableau 14. Symboles DCL et description

Nom du symbole DCL	Description
MQS\$QMGR_NODE	Défini sur le noeud OpenVMS qui exécute le GFA, ou une chaîne vide si aucun GFA n'est exécuté
MQS\$AVAILABLE_NODES	Défini sur la liste des noeuds OpenVMS disponibles pour exécuter le GFA. Il s'agit des noeuds qui sont dans l'état AVAILABLE et qui exécutent un moniteur de reprise.
MQS\$MONITOR_NODES	Paramétré sur la liste des noeuds OpenVMS qui exécutent un moniteur de reprise.

Par exemple, pour paramétrer les symboles sur l'état du groupe de reprise du GFA TESTQM, utilisez la commande suivante :

```
$ failover -m TESTQM -l
```

## Paramétrage des symboles DCL

Exemple de résultats du paramétrage de symboles :

```
MQS$AVAILABLE_NODES = ""
MQS$MONITOR_NODES = "BATMAN,ROBIN"
MQS$QMGR_NODE = "BATMAN"
```

## Arrêt d'un moniteur de reprise

Il est possible d'arrêter le moniteur de reprise d'un noeud OpenVMS à l'aide de l'indicateur -h de la commande **failover**. La commande peut être exécutée depuis n'importe quel noeud du groupe de reprise. Par exemple, pour arrêter le moniteur de reprise du GFA TESTQM du noeud BATMAN, utilisez la commande suivante :

```
$ failover -m TESTQM -n BATMAN -h
```

Si le moniteur de reprise que vous souhaitez arrêter est celui qui est chargé de l'observation, un autre moniteur de reprise se charge de la surveillance. Si le moniteur de reprise que vous arrêtez est le dernier moniteur du groupe de reprise, alors il n'y en aura plus aucun d'actif. Dans ce cas, le GFA peut être démarré et arrêté à l'aide des commandes **strmqm** et **endmqm**. L'indicateur -h de la commande **failover** n'arrête jamais un GFA. Si le GFA fonctionne sur le noeud OpenVMS sur lequel le moniteur de reprise est arrêté, il continue de fonctionner.

## Exécution de commandes pendant une mise à jour

Les commandes **failover** utilisant les indicateurs -s, -e, -f et -c sont considérées comme des mises à jour. Lorsque ces commandes sont en cours, le moniteur de reprise principal en fait état. Lorsque cet indicateur est positionné, toute autre commande de mise à jour et d'arrêt de reprise par transfert échoue car les mises à jour simultanées sont interdites. Les commandes qui ne sont pas considérées comme des mises à jour, telles que les indicateurs -q et -l continuent de fonctionner lorsqu'une mise à jour est en cours.

Dans de rares cas, une mise à jour échouée peut laisser l'indicateur de mise à jour en cours affiché. L'indicateur -u de la commande **failover** efface l'indicateur de mise à jour en cours. Cette commande doit être utilisée avec précaution. Par exemple, pour effacer l'indicateur de mise à jour en cours du GFA TESTQM, utilisez la commande :

```
$ failover -m TESTQM -u
```

## Modification de l'état d'un groupe de reprise

Dans certains cas, il peut être nécessaire de modifier l'état d'un groupe de reprise. Pour cela, utilisez l'indicateur -c de la commande **failover**. Le cas le plus probable où vous aurez à utiliser cet indicateur est lorsque un noeud est dans un état EXCLUDED, suite à une panne, et que vous souhaitez le remettre dans un état AVAILABLE après récupération du noeud. Par exemple, pour faire passer en AVAILABLE le GFA TESTQM du noeud BATMAN, utilisez la commande suivante :

```
$ failover -m TESTQM -n BATMAN -c -qmgr disponible
```

## Modification de l'état d'un groupe de reprise

Vous pouvez aussi empêcher temporairement un noeud d'être choisi pour exécuter le GFA en le faisant passer d'un état AVAILABLE à EXCLUDED. Par exemple, pour faire passer dans le mode EXCLUDED le GFA TESTQM du noeud BATMAN, utilisez la commande suivante :

```
$ failover -m TESTQM - n BATMAN -c -qmgr exclu
```

Il est aussi possible de modifier tous les autres états, mais le changement ne prendra effet que s'il est cohérent avec le système exécuté. Par exemple, si un moniteur de reprise fonctionne sur un noeud et que vous essayez de le faire passer sur STOPPED, la modification ne prendra pas effet. Sauf dans le cas où vous souhaitez faire passer l'état du GFA du noeud de EXCLUDED à AVAILABLE, il n'est pas nécessaire d'utiliser la commande de modification d'état, car le moniteur de reprise principal procède toutes les 30 secondes à un contrôle de l'intégrité, et change les états en cas d'incohérence avec le système exécuté.

## Paramétrage de la sécurité pour les associations ICC

Les programmes client et du groupe de reprise utilisent les appels OpenVMS Intra Cluster Communication (ICC) pour transmettre les messages. Pour empêcher les utilisateurs non autorisés d'envoyer des messages aux processus du moniteur de reprise, la sécurité des associations ICC doit être configurée dans la procédure de commande **SYS\$STARTUP:ICC\$SYSTARTUP.COM**.

Chaque groupe de reprise utilise deux noms d'association : l'un avec le nom du GFA utilisé pour communiquer avec le moniteur de reprise principal, l'autre avec le nom du GFA avec le suffixe **\_MQ\_FM**, utilisé pour communiquer avec chaque moniteur de reprise.

La figure 22 à la page 234 présente un exemple des entrées requises dans **ICC\$SYSTARTUP.COM** pour chaque noeud du groupe de reprise. Le groupe de reprise BATMAN et ROBIN comprend deux noeuds. Le nom du GFA est TESTQM.

## Paramétrage de la sécurité

```
#! ----- List Nodes with Special Actions -----
#!
$ nodeactions = "/BATMAN/ROBIN/"
$ if f$locate("/"+nodename+"/",nodeactions) .eq. f$length(nodeactions) -
then goto exit ! No action for this node
$ goto 'nodename' ! Go to action code for this node
#!
#! ----- Major Nodes -----
$BATMAN:
$ROBIN:
#!
#!
#! Place in here calls to @SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT and
#! @SYS$MANAGER:ICC$ADD_REGISTRY_TABLE that apply to FAilover odes dans le
#! cluster
#!
#!
$ @SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT ICC$::"TESTQM" -
"/owner=MQM/ac1=((id=MQM,access=open+access),(id=*,access=none))"
#!
$ @SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT 'nodename'::"TESTQM" -
"/owner=MQM/ac1=((id=MQM,access=open+access),(id=*,access=none))"
#!
$ @SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT 'nodename'::"TESTQM_MQ_FM" -
"/owner=MQM/ac1=((id=MQM,access=open+access),(id=*,access=none))"
#!
$ set security/class=logica_name_table icc$registry_table -
/ac1=(id=MQM,access=read+write)
#!
$ GOTO EXIT
#!
```

Figure 22. Exemple des entrées requises pour ICC\$SYSTARTUP.COM

Notez que les noms d'association ICC sont limités à 31 caractères. Par conséquent, la longueur maximale du nom de GFA MQSeries est de 25 caractères, dans le cadre d'un groupe de reprise. Pour plus d'informations sur la configuration de la sécurité pour les associations ICC, consultez le manuel *OpenVMS System Manager's Manual*.

## Résolution des incidents liés aux groupe de reprise

Lors de l'exécution des procédures **start\_qm.com**, **end\_qm.com** et **tidy\_qm.com**, un fichier journal est écrit dans le LogDirectory indiqué dans le fichier de configuration **failover.ini**. Les noms des fichiers journaux sont **qmgrname\_procedurename.log**. Par exemple, pour un GFA ,nommé TESTQM la procédure de commande **start\_qm.com** produira un fichier journal portant le nom **testqm\_start\_qm.log**.

Par défaut, le moniteur de reprise ne produit pas de fichier journal, mais il est possible d'en demander un à l'aide du paramètre de réacheminement dans la commande **runmqfm**. Des informations de débogage supplémentaires peuvent être écrites dans le fichier en spécifiant le paramètre **-d** dans la commande **runmqfm**.

Vérifiez si un fichier FDC a été généré dans **MQS\_ROOT:[MQM.ERRORS]**.

## Utilisation de MultiNet for OpenVMS avec des groupes de reprise

Pour utiliser Multinet for OpenVMS avec des groupes de reprise, le service d'alias de cluster doit être activé. Vous pouvez l'activer par la commande :

```
$ MULTINET CONFIGURE/SERVERS
SERVER-CONFIG> ENABLE CLUSTERALIAS
SERVER-CONFIG> EXIT
```

Les fichiers de commandes modèles partent du principe qu'il n'y a qu'une adresse d'alias de cluster, et qu'elle est utilisée par le groupe de reprise. Toutefois, si d'autres adresses d'alias de cluster sont utilisées, les procédures de commande devront être modifiées afin que les autres adresses restent dans le nom logique MULTINET\_CLUSTER\_IP\_ALIASES.

### Exemple d'utilisation des groupes de reprise

Dans l'exemple qui suit, deux noeuds d'un cluster OpenVMS, BATMAN et ROBIN, seront configurés en un groupe de reprise pour le gestionnaire de file d'attente TESTQM. L'adresse TCP/IP du groupe de reprise est 10.20.30.40, le port du programme d'écoute TCP/IP est 1414. Le noeud BATMAN est désigné comme noeud primaire et ROBIN comme noeud secondaire. Initialement le GFA sera démarré sur BATMAN et redémarré sur ROBIN en cas de panne. Si le GFA fonctionne sur ROBIN, il ne sera pas restauré sur BATMAN lors du redémarrage de ROBIN. En cas d'arrêt du noeud exécutant le GFA, celui-ci est à son tour stoppé ainsi que le moniteur de reprise. Si le noeud ne fonctionne pas, le GFA n'est pas arrêté et seul le moniteur de reprise est stoppé.

#### Personnalisation de failover.template

Le fichier failover.template doit d'abord être copié comme suit, puis modifié :

```
mqs_root: [mqm.qmgrs.testqm] failover.ini.
```

## Exemple de groupe de reprise

```
# FAILOVER.TEMPLATE
# Template for creating a FAILOVER.INI configuration file
# All lines beginning with a '#' are treated as comments
#
# OpenVMS Cluster Failover Set Configuration information
# -----
#
# The TCP/IP address used by the OpenVMS Cluster Failover Set
#
IpAddress=10.20.30.40
#
# The TCP/IP port number used by the MQSeries Queue Manager
#
PortNumber=1414
#
# The timeout used by the EndCommand command procedure
#
TimeOut=30
#
# The command procedure used to start the Queue Manager
#
StartCommand=@sys$manager:start_qm
#
# The command procedure used to end the Queue Manager
#
EndCommand=@sys$manager:end_qm
#
# The command procedure used to tidy up on a node after a
# Queue Manager failure but the OpenVMS node did not fail
#
TidyCommand=@sys$manager:tidy_qm
#
# The directory in which the log files for the start, end and
# tidy commands are written
#
LogDirectory=mqs_root:[mqm.errors]
#
# The number of nodes in the OpenVMS Cluster Failover Set. The
# number of nodes defined below must agree with this number
#
NodeCount=2
#
# The Name of the OpenVMS node
#
NodeName=BATMAN
#
# The TCP/IP interface name for the node
#Interface=we0
#
# The priority of the node
#
Priority=1
#
# The Name of the OpenVMS node
#
NodeName=ROBIN
#
# The TCP/IP interface name for the node
#
Interface=we0
#
# The priority of the node
#
Priority=2
```

Figure 23. Failover.template pour la création d'un fichier de configuration FAILOVER.INI



### Modification des procédures de commandes d'un groupe de reprise

Les procédures de commande sont copiées à partir des fichiers modèles. Les seules modifications sont que le démarrage du programme d'écoute, dans **start\_qm.com**, et son arrêt dans **end\_qm.com** sont activées (retrait des caractères de mise en commentaire). S'il s'avère nécessaire d'arrêter ou de démarrer certaines applications, les commandes appropriées peuvent être ajoutées aux procédures.

### Exemple de procédure de commande de démarrage de groupe de reprise : **start\_failover\_set.com**

La procédure de commande **start\_failover\_set.com** est utilisée pour démarrer le moniteur de reprise sur chaque noeud et démarrer le GFA. La procédure est appelée à partir du démarrage système, une fois la procédure de commande **MQS\_STARTUP.COM** exécutée. Deux paramètres sont transmis à la procédure : le nom du GFA et le nom du noeud primaire. Dans notre exemple, elle est appelée de la façon suivante :

```
$@start_failover_set testqm batman
```

La procédure de commande **start\_failover\_set.com** démarre le moniteur de reprise, puis utilise le paramètre **-l** de la commande **failover** pour déterminer l'état du groupe de reprise. Attention, il est possible que le moniteur de reprise ne soit pas totalement démarré au moment de l'exécution de la commande **failover**. Jusqu'à trois tentatives seront donc effectuées, chacune à une seconde d'intervalle. Ensuite, s'il s'agit du noeud principal, et que GFA n'est pas démarré, la mise en route est effectuée à l'aide du paramètre **-s** de la commande **failover**.

## Exemple de groupe de reprise

```
$on error then exit
$@sy$manager:mqs_symbols
$!
$! start_failover_set.com
$! -----
$! Command procedure to start a Failover Set Queue Manager during startup
$!
$! p1 = Queue Manager name
$! p2 = Primary Node name
$!
$! Check that the Queue Manager has been specified
$!
$if p1 .eqs ""
$then
$ Write sys$output "Queue Manager name omitted"
$ exit
$else
$ qmgr_name = p1
$endif
$!
$! Check that the primary node name has been specified
$!
$if p2 .eqs ""
$then
$ Write sys$output "Node name omitted"
$ exit
$else
$ primary_node = p2
$endif
$!
$! Get the node name of this node
$!
$this_node=f$getsyi("nodename")
$!
$! Start the Failover Monitor on this node
$!
$runmqfm -m 'qmgr_name'
$!
$! Check that the Failover Monitor has fully started
$! Wait up to 3 seconds
$!
$count = 0
$check_start:
$on error then continue
$!
$! Set the MQS$* symbols to the state of Failover Set
$! Wait up to 3 seconds
$!
```

Figure 24. Procédure de commande start\_failover\_set (Numéro 1 de 2)

## Exemple de groupe de reprise

```
$failover -m 'qmgr_name' -l
$!
$! If an error is returned wait a second and try again
$!
$if ( ($status/8) .and %xffff ) .ne. 0 then goto wait
$!
$! If this node is not listed as running a monitor wait a second and try again
$!
$if f$locate( this_node, mqs$monitor_nodes ) .ne. f$length( mqs$monitor_nodes )
$then
$ goto start_qm
$endif
$wait:
$on error then exit
$count = count + 1
$!
$! If we have waited 3 seconds display an error and exit
$!
$if count .ge. 3
$then
$ write sys$output "Failover Monitor not started"
$ exit
$else
$ wait 00:00:01
$ goto check_start
$endif
$start_qm:
$!
$! Only start the Queue Manager on the primary node
$!
$if this_node .nes. primary_node
$then
$ write sys$output "Queue Manager not started on Secondary node"
$ exit
$endif
$!
$! Start the Queue Manager on the primary node if it is not already running.
$!
$if mqs$qmgr_node .eqs. ""
$then
$ failover -m 'qmgr_name' -n 'this_node' -s
$else
write sys$output "Queue Manager already started"
$endif
$exit
```

Figure 24. Procédure de commande `start_failover_set` (Numéro 2 de 2)

## Exemple de procédure de commande d'arrêt de groupe de reprise, `end_failover_set.com`

La procédure de commande `end_failover_set.com` est utilisée pour arrêter sur chaque noeud le GFA puis le moniteur de reprise. La procédure est appelée depuis l'arrêt spécifique au site, avant l'exécution de la procédure de commande `MQS_SHUTDOWN.COM`. Un paramètre est transmis à la procédure, le nom du GFA. Dans notre exemple, la procédure est appelée de la façon suivante :

```
$@start_failover_set testqm
```

## Exemple de groupe de reprise

La procédure de commande **end\_failover\_set.com** obtient l'état du groupe de reprise grâce au paramètre **-l** de la commande **failover**. Puis, si le GFA fonctionne sur ce noeud, il est arrêté. Ensuite le moniteur de reprise est stoppé.

```
on error then exit
$@sys$manager:mqs_symbols
$!
$! end_failover_set.com
$! -----
$! Command procedure to end a Failover Set Queue Manager during shutdown
$!
$! p1 = Queue Manager name
$!
$! Check that the Queue Manager has been specified
$!
$if p1 .eqs ""
$then
$ write sys$output "Queue Manager name omitted"
$ exit
$else
$ qmgr_name = p1
$endif
$!
$! Get the node name of this node
$!
$this_node=f$getsyi("nodename")
$!
$! Set the MQS$* symbols to the state of the Failover Set
$!
$failover -m 'qmgr_name' -l
$!
$! If an error then exit
$!
$if ( ($status/8) .and %xffff ) .ne. 0
$then
$ write sys$output "Error querying Failover Set"
$ exit
$endif
$!
$! If the Queue Manager is not running on this node then exit
$!
$ if mqs$qmgr_node .nes. this_node
$then
$ write sys$output "Queue Manager not running on this node"
$ goto halt_fm
$endif
$!
$! End the Queue Manager
$!
$failover -m gjtest -e
$halt_fm:
$!
$! Halt the Failover Monitor
$!
$failover -m gjtest -n 'this_node' -h
```

Figure 25. Procédure de commande *end\_failover\_set*

---

## Partie 2. Section de référence

<b>Chapitre 17. Commandes de contrôle MQSeries</b>	243
Noms d'objets MQSeries . . . . .	243
Consultation des fichiers objets . . . . .	244
Lecture des diagrammes de syntaxe . . . . .	244
Aide sur la syntaxe . . . . .	245
Exemples . . . . .	246
Codes retour de MQSeries . . . . .	247
crtmqcvx (conversion de données) . . . . .	248
crtmqm (création d'un gestionnaire de files d'attente) . . . . .	250
dltmqm (suppression d'un gestionnaire de files d'attente) . . . . .	254
dmpmqlog (cliché du journal) . . . . .	257
dspmqaout (affichage des droits d'accès) . . . . .	259
dspmqrst (affichage du serveur de commandes)	263
dspmqrfls (affichage des fichiers MQSeries) . . . . .	264
dspmqrtrc (affichage de la sortie de trace formatée MQSeries) . . . . .	266
dspmqrtrn (affichage des transactions MQSeries)	268
endmqcvx (arrêt d'un serveur de commandes) . . . . .	270
endmqslr (arrêt du programme d'écoute) . . . . .	273
endmqm (arrêt d'un gestionnaire de files d'attente)	274
endmqtrc (arrêt de la trace MQSeries) . . . . .	277
failover (gestion d'un groupe de reprise) . . . . .	278
rcdmqimg (enregistrement d'une image de support) . . . . .	282
rcrmqobj (recréation d'un objet) . . . . .	284
rsvmqtrn (résolution des transactions MQSeries)	287
runmqchi (exécution de l'initialisateur de canal)	289
runmqchl (démarrage d'un canal) . . . . .	290
runmqdlq (lancement du gestionnaire de files d'attente de rebut) . . . . .	291
runmqfm (lancement d'un moniteur de reprise)	293
runmqslr (exécution du programme d'écoute) . . . . .	294
runmqsc (exécution des commandes MQSeries) . . . . .	296
runmqstmc (lancement du moniteur de déclenchement client) . . . . .	299
runmqstrm (lancement du moniteur de déclenchement) . . . . .	300
setmqaut (définition ou réinitialisation des droits d'accès) . . . . .	301
strmqcsv (lancement d'un serveur de commandes)	308
strmqm (lancement d'un gestionnaire de files d'attente) . . . . .	309
strmqtrc (lancement de la trace MQSeries) . . . . .	311



---

## 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 OpenVMS. Toutes les commandes de ce chapitre peuvent être émises depuis une ligne de commandes DCL.

Les noms de commandes et leurs indicateurs ne font pas la distinction entre les majuscules et les minuscules : vous pouvez les entrer en majuscules, en minuscules ou une combinaison des deux. Par contre, les paramètres permettant de contrôler les commandes (tels que les noms de files d'attente) font la distinction entre les majuscules et les minuscules. Pour plus d'informations, reportez-vous à la section «Utilisation des majuscules et minuscules dans les commandes de contrôle» à la page 25.

Avant de pouvoir utiliser n'importe quelle commande de contrôle, il faut avoir lancé `mqs_startup` au moins une fois depuis le dernier redémarrage.

---

### Noms d'objets MQSeries

En général, les noms d'objets MQSeries peuvent comporter jusqu'à 48 caractères. Cette règle s'applique à tous les objets suivants :

- Gestionnaires de files d'attente (attention, si ce gestionnaire est supporté par un groupe de reprise OpenVMS, la longueur maximale du nom est de 25 caractères).
- Files d'attente
- Définitions de processus
- Listes de noms
- Clusters

La longueur maximale des noms de canaux est de 20 caractères.

Les noms MQSeries peuvent être composés des caractères suivants :

- Majuscules de A à Z
- Minuscules de a à z
- Chiffres de 0 à 9
- Le point (.)
- Le souligné (\_)
- La barre oblique (/) (voir remarque 1)
- Le signe pourcentage (%) (voir remarque 1)

#### Remarques :

1. Les caractères barre oblique et pourcentage sont des caractères spéciaux. Un nom ne peut pas commencer par les caractères barre oblique ni pourcentage. 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.

## 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. Pour plus de détails, reportez-vous à la section «Signification des noms de fichiers MQSeries» à la page 23.

Pour savoir comment afficher le nom réel d'un objet ou d'un fichier, reportez-vous à la section «dspmqfls (affichage des fichiers MQSeries)» à la page 264.

## 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 :

Tableau 15. Syntaxe des commandes

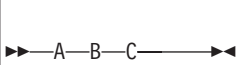

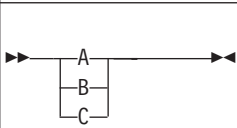
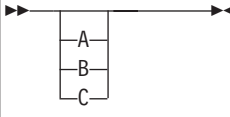
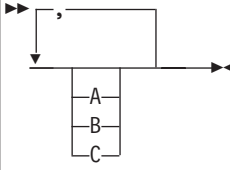
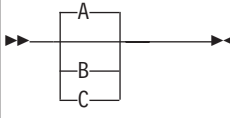
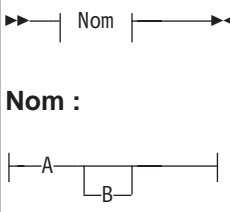
Convention	Signification
	Vous devez préciser les valeurs A, B et C. Les valeurs requises sont présentées sur la ligne principale du diagramme.
	Vous pouvez indiquer la valeur A. Les valeurs facultatives sont présentées sous la ligne principale du diagramme.
	Vous avez le choix entre les trois valeurs A, B ou C, et vous devez en préciser une.



Tableau 15. Syntaxe des commandes (suite)

Convention	Signification
	<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>Le fragment de syntaxe Nom est à l'écart du diagramme de syntaxe principal.</p>

## Aide sur la syntaxe

Vous pouvez obtenir une aide sur la syntaxe de n'importe quelle commande de ce chapitre en entrant le nom de la commande suivi d'un point d'interrogation. MQSeries répond en affichant la syntaxe requise pour la commande indiquée. La syntaxe présente tous les paramètres et variables associés à la commande. Différents types de parenthèses sont utilisés pour indiquer si un paramètre est obligatoire. Par exemple :

```
NomCmd[-x ParamOpt ] ( -c | -b ) { -p principal } argument
```

Où :

**NomCmd**

Nom de la commande pour laquelle l'aide est demandée.

**[-x ParamOpt ]**

Les crochets indiquent qu'il s'agit d'un ou plusieurs paramètres facultatifs.

## Aide sur la syntaxe

( -c | -b )

Il s'agit d'un paramètre obligatoire. Dans ce cas, vous devez choisir l'indicateur c ou b.

{ -p principal }

Une liste facultative de variables que vous pouvez préciser. Cependant, si cette représentation est utilisée, vous devez préciser au moins une variable en entrant la commande.

argument

Un argument obligatoire.

## Exemples

1. Résultat de la commande endmqm ?

```
endmqm [-z] [-c | -i | -p] NomGestFiles
```

2. Résultat de la commande rcdmqimg ?

```
rcdmqimg [-z] [-m NomGestFiles] -t TypeObj [NomObjGenerique]
```

## Codes retour de MQSeries

La plupart des commandes MQSeries, comme **crtmqm**, affichent une ligne d'état lors de leur fin, pour indiquer leur réussite ou leur échec.

Si l'état d'une commande doit être vérifié dans un fichier de commandes DCL, il peut être nécessaire d'interpréter la valeur retournée par un programme MQSeries.

Les codes retour de MQSeries sont définis dans un fichier message nommé :  
SYS\$MESSAGE:MQS\_MSG.EXE.

Pour récupérer le texte du message associé à un code retour, vous *devez* utiliser la commande CL SET MESSAGE. Cette commande charge les codes de message dans la table de messages de votre processus. Par exemple :

```
$ SET MESSAGE SYS$MESSAGE:MQS_MSG.EXE
```

Ensuite, vous pouvez utiliser la fonction lexicale F\$MESSAGE pour afficher le texte correspondant à un code de retour MQSeries. Par exemple :

```
$ strmqm )(*bad-qm-name&%$#
Le nom du gestionnaire de files d'attente est soit non valide soit inconnu
$ WRITE SYS$OUTPUT F$MESSAGE($STATUS)
%MQS-F-CSPRC_Q_MGR_NAM, Erreur de nom de gestionnaire de files d'attente
```

Pour convertir le code de retour OpenVMS en une valeur de code utilisée par MQSeries pour systèmes OS/2 ou UNIX, vous pouvez utiliser l'équation DCL suivante :

```
$ RC = $STATUS / 8 .AND. %xFFF
```

Par exemple :

```
$ crtmqm &*)*(
Le nom du gestionnaire de files d'attente est soit non valide soit inconnu
$ RC = $STATUS / 8 .AND. %xFFF
$ SHOW SYMBOL RC
RC = 72 Hex = 00000048 Octal = 00000000110
```

## 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 *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 source.tmp cible.c
```

Le fichier d'entrée, source.tmp ressemble à ceci :

```

/* 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.c, 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 MQSeries d'en-tête amqsvmha.h.

```

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);
}

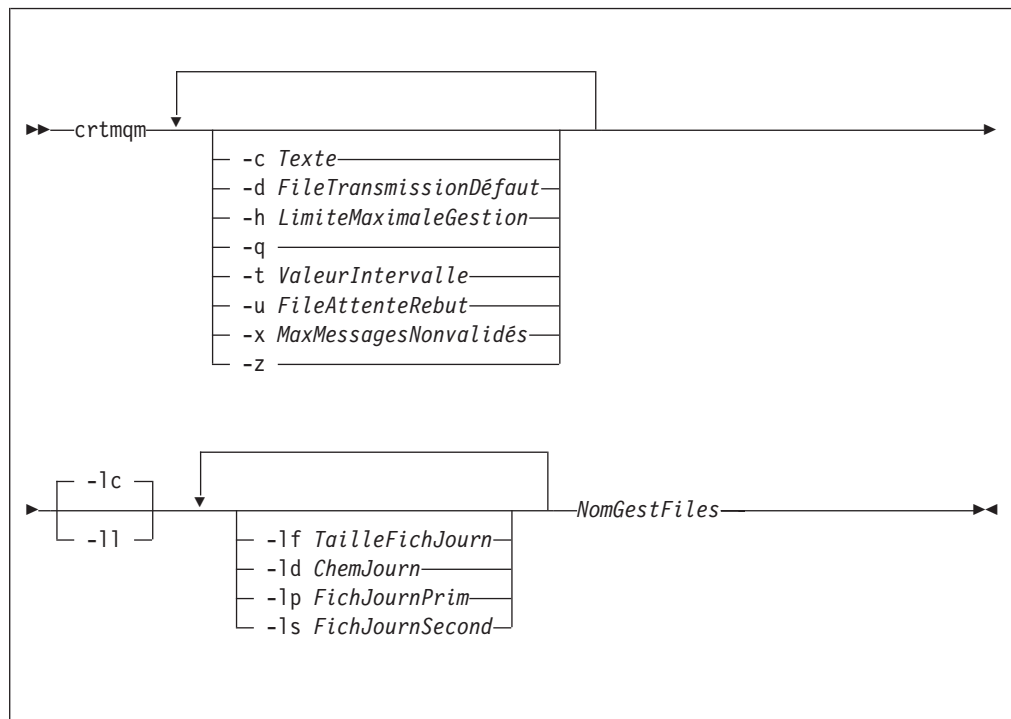
```

## 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.

### Syntaxe



### Paramètres obligatoires

#### *NomGestFiles*

Indique le 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*

Indique un texte descriptif pour ce gestionnaire de files d'attente. Par défaut, le texte reste blanc.

Vous disposez de 64 caractères au maximum. Si des caractères mixtes sont nécessaires, la description doit être placée entre guillemets.

#### *-d FileTransmissionDéfaut*

Indique le 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.

**-h** *LimiteMaximaleGestion*

Indique le 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.

- 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. Pour plus de détails sur les fichiers de configuration, reportez-vous au «Chapitre 13. Configuration de MQSeries» à la page 173.

**-t** *ValeurIntervalle*

Précise l'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 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*

Indique le nom de la file d'attente locale qui sera utilisée 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.

Si cet attribut est omis, aucune file de rebut n'est définie.

**-x** *MaximumMessagesAttente*

Indique le 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.

L'ensemble d'indicateurs suivant est utilisé pour définir le mode de consignation utilisé par le gestionnaire de files d'attente créé. Pour plus d'informations sur les journaux, consultez la section «Utilisation du journal pour la reprise» à la page 145.

**-lc** La réutilisation automatique doit être utilisée. Il s'agit de la méthode de journalisation par défaut.

**-ll** La journalisation linéaire doit être utilisée.

**-lf** *TailleFich Journ*

Indique la taille des fichiers journaux en unités de 4 Ko. La valeur minimale est de 64, et la valeur maximale 16384. La valeur par défaut est 1024, ce qui donne une taille de journal par défaut de 4 Mo.

**-ld** *Chem Journ*

Indique le répertoire à utiliser pour les fichiers journaux. La valeur par défaut est MQS\_ROOT:[MQM.LOG]. Elle peut aussi être modifiée lors de la personnalisation de MQSeries.

L'ID utilisateur mqm et le groupe mqm doivent avoir des droits d'accès complets aux fichiers de journalisation. Si vous modifiez les emplacements de ces fichiers, vous devez accorder ces droits vous-même. Ceci est fait automatiquement si les fichiers journaux gardent leurs emplacements par défaut.

**-lp** *Fich Journ Prim*

Indique le nombre de fichiers journaux primaires à attribuer. La valeur par défaut est 3, le minimum 2 et le maximum 62.

**-ls** *Fich Journ Second*

Indique le nombre de fichiers journaux secondaires à attribuer. La valeur par défaut est 2, le minimum 1 et le maximum 61.

**Remarque :** Le nombre total de fichiers journaux est limité à 63, quel que soit le nombre demandé.

Les limites indiquées dans la liste de paramètres précédente sont imposées par MQSeries. Les limites imposées par le système d'exploitation peuvent être inférieures à la valeur maximale indiquée.

## 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.
- 100 L'emplacement du journal est incorrect.
- 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.



115 Taille de journal incorrecte.

## Exemples

1. Cette commande crée un gestionnaire de files d'attente par défaut appelé `Paint.gest.files`, auquel est attribuée la description `Paint shop`. Elle indique également que la journalisation linéaire doit être utilisée

```
crtmqm -c "Paint shop" -ll -q "Paint.gest.files"
```

2. Cet exemple nécessite un certain nombre de fichiers journaux. Deux fichiers journaux primaires et secondaires sont indiqués.

```
crtmqm -c "Paint shop" -ll -lp 2 -ls 3 -q "Paint.gest.files"
```

3. Dans cet exemple, un autre gestionnaire de file d'attente, `travel`, 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 voyage
```

Une fois l'événement de déclenchement généré, les autres événements de déclenchement sont désactivés pendant cinq secondes.

## Commandes connexes

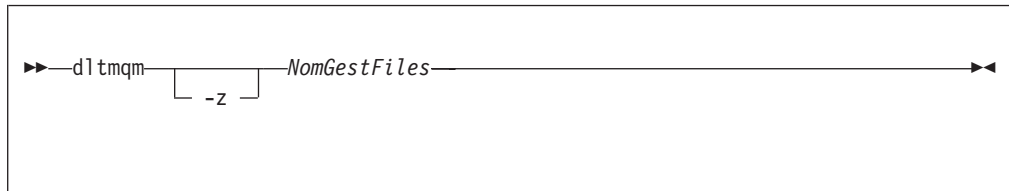
<b>strmqm</b>	lancement d'un gestionnaire de files d'attente
<b>endmqm</b>	arrêt d'un gestionnaire de files d'attente
<b>dltmqm</b>	suppression d'un gestionnaire de files d'attente

## 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.
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.
100	L'emplacement du journal est incorrect.
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**

<b>crtmqm</b>	création d'un gestionnaire de files d'attente
<b>strmqm</b>	lancement d'un gestionnaire de files d'attente
<b>endmqm</b>	arrêt d'un gestionnaire de files d'attente

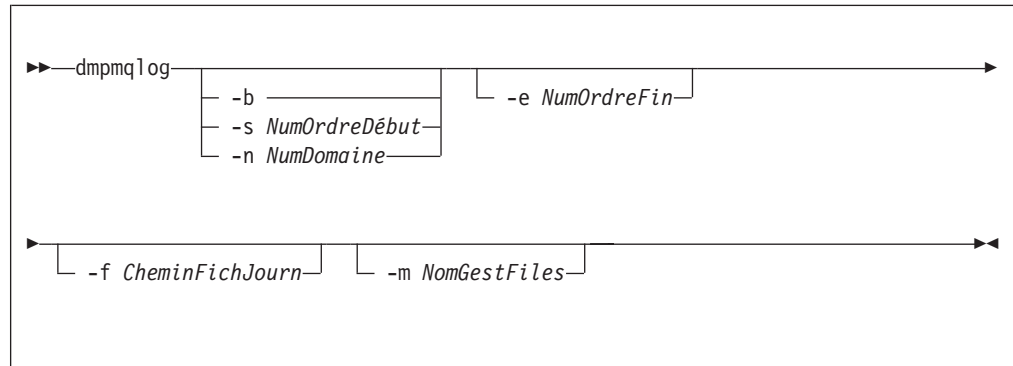
## dmpmqlog (cliché du journal)

### Objet

La commande **dmpmqlog** permet de cliquer une version formatée du journal système MQSeries.

Le journal à cliquer doit avoir été créé sur le même type de système d'exploitation que celui sur lequel la commande est émise.

### Syntaxe



### Paramètres facultatifs

#### Point de départ du cliché

Utilisez l'un des paramètres suivants pour spécifier le numéro d'ordre de journal (LSN) auquel le cliché doit démarrer. Si aucun point de départ n'est spécifié, le cliché commence par défaut à partir du numéro d'ordre de journal du premier enregistrement de la portion active du journal.

**-b** Indique que le cliché doit commencer à partir du numéro d'ordre de base de journal. Le numéro d'ordre de base de journal identifie le début du domaine de journal contenant le début de la portion active du journal.

#### **-s** *NumOrdreDébut*

Indique que le cliché doit commencer à partir du numéro d'ordre de journal spécifié. Le numéro d'ordre est spécifié sous le format `nnnn:nnnn:nnnn:nnnn`.

Si vous utilisez la journalisation circulaire, la valeur de ce numéro d'ordre de journal doit être supérieure ou égale à la valeur du numéro d'ordre de base du journal.

#### **-n** *NumDomaine*

Indique que le cliché doit commencer à partir du numéro de domaine spécifié. Le numéro de domaine doit être compris entre 0-9 999 999.

Ce paramètre n'est valable que pour les gestionnaires de files d'attente dont la valeur du paramètre *LogType* (indiqué dans le fichier de configuration `qm.ini`) est `LINEAR`.

#### **-e** *NumOrdreFin*

Indique que le cliché doit s'arrêter au numéro d'ordre de journal spécifié. Le numéro d'ordre est spécifié sous le format `nnnn:nnnn:nnnn:nnnn`.

## dmpmqlog

### **-f** *CheminFichJour*

Indique le nom absolu du chemin de répertoire des fichiers journaux au lieu du nom relatif. Le répertoire spécifié doit contenir le fichier d'en-tête du journal (`amqh1ct1.lfh`) ainsi qu'un sous-répertoire appelé `active`. Ce sous-répertoire doit contenir les fichiers journaux. Par défaut, on suppose que les fichiers journaux se trouvent dans les répertoires spécifiés dans les fichiers `mqs.ini` et `qm.ini`. Si cette option est utilisée, les noms de files d'attente, associés à des identificateurs de files d'attente, ne seront indiqués dans le cliché que si un nom de gestionnaire de files d'attente est spécifié de manière explicite pour l'option `-m` et si le fichier catalogue figure dans le chemin de répertoire de ce gestionnaire.

Sur un système prenant en charge les noms de fichiers longs, ce fichier est appelé `qmqmobjcat` pour mettre en correspondance les identificateurs de files d'attente et les noms de files d'attente, il doit s'agir du fichier utilisé lors de la création des fichiers journaux. Par exemple, dans le cas d'un gestionnaire de files d'attente appelé `qm1`, le fichier catalogue d'objets est situé dans le répertoire `MQS_ROOT:[MQM.QMGRS.QM1.QMANAGER]`. Pour effectuer ce mappage, vous devrez peut-être créer un gestionnaire de files d'attente temporaire, appelé par exemple `tmpq`, remplacer son catalogue d'objets par celui associé aux fichiers journaux donnés, puis lancer la commande `dmpmqlog` en spécifiant `-m tmpq` et `-f` avec le nom absolu du chemin de répertoire des fichiers journaux.

### **-m** *NomGestFiles*

Nom du gestionnaire de files d'attente. Si ce paramètre est omis, le nom du gestionnaire de files d'attente par défaut est utilisé.

Le gestionnaire de files d'attente spécifié ou utilisé par défaut ne doit pas être actif lorsque la commande **dmpmqlog** est émise. De même, le gestionnaire de files d'attente ne doit pas être lancé tant que la commande **dmpmqlog** s'exécute.

## dspmqaout (affichage des droits d'accès)

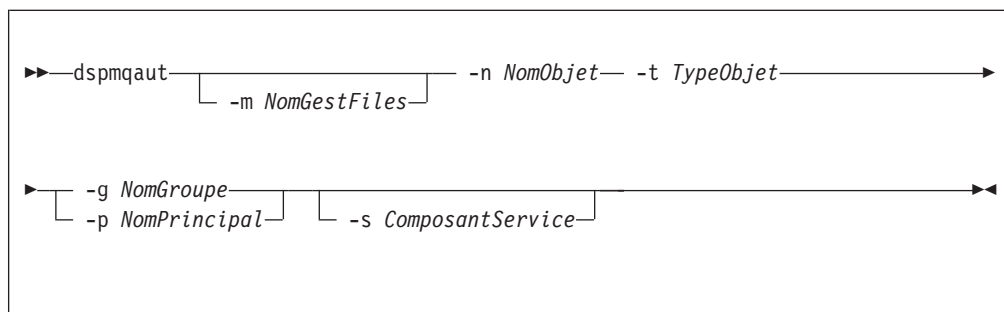
### Objet

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

#### -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).

Vous devez préciser le nom du gestionnaire de files d'attente, de la file d'attente ou de la définition de procédure.

#### -t *TypeObjet*

Indique le type de l'objet sur lequel porte l'interrogation. Les valeurs possibles sont :

<b>queue ou q</b>	Une ou plusieurs files correspondant au paramètre TypeObjet
<b>qmgr</b>	Objet gestionnaire de files d'attente
<b>process ou prcs</b>	Processus
<b>namelist ou nl</b>	Liste de noms

### Paramètres facultatifs

#### -m *NomGestFiles*

Indique le nom du gestionnaire de files d'attente sur lequel porte l'interrogation.

#### -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.

## dspmqaout

### -p *NomPrincipal*

Indique le nom d'un utilisateur dont les droits d'accès à l'objet indiqué doivent être affichés.

### -s *ComposantService*

Ce paramètre ne s'applique que si vous utilisez des services d'autorisation optionnels.

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 d'accès est adressée au premier composant optionnel du service

## 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 16 présente les droits qui peuvent être accordés suivant les différents types d'objets.

Tableau 16. 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	✓	✓	✓	✓
clr	✓			
connect			✓	
cpy	✓	✓	✓	✓
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.



<b>alladm</b>	Exécution de toutes les opérations d'administration concernant cet objet.
<b>allmqi</b>	Utilisation de tous les appels MQI concernant cet objet.
<b>altusr</b>	Indication d'un ID utilisateur secondaire pour un appel MQI.
<b>browse</b>	Extraction d'un message d'une file par l'émission d'un appel MQGET avec l'option BROWSE.
<b>chg</b>	Modification des attributs de l'objet indiqué à l'aide du jeu de commandes approprié.
<b>clr</b>	Effacement d'une file (commande PCF Clear queue seulement).
<b>connect</b>	Connexion de l'application au gestionnaire de files d'attente indiqué par l'émission d'un appel MQCONN.
<b>cpy</b>	Copie la définition d'un objet (par exemple la commande PCF Copy queue).
<b>crt</b>	Création des objets du type indiqué à l'aide du jeu de commandes approprié.
<b>dlt</b>	Suppression de l'objet indiqué à l'aide du jeu de commandes approprié.
<b>dsp</b>	Affichage des attributs de l'objet indiqué à l'aide du jeu de commandes approprié.
<b>get</b>	Extraction d'un message d'une file par l'émission d'un appel MQGET.
<b>inq</b>	Interrogation sur une file donnée par l'émission d'un appel MQINQ.
<b>passall</b>	Transmission de tout le contexte.
<b>passid</b>	Transmission du contexte d'identité.
<b>put</b>	Insertion d'un message dans une file d'attente donnée par l'émission d'un appel MQPUT.
<b>set</b>	Définition des attributs d'une file à partir de MQI par l'émission d'un appel MQSET.
<b>setall</b>	Définition de tout le contexte d'une file d'attente.
<b>setid</b>	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.

## dspmqaout

- 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.

## 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 :

```
L'entité personnel dispose des droits d'accès suivants à  
l'objet :  
    get  
    browse  
    put  
    inq  
    set  
    connect  
    altusr  
    passid  
    passall  
    setid
```

## Commandes connexes

**setmqaout** Définition ou réinitialisation des droits d'accès

## dspmqcsv (affichage du serveur de commandes)

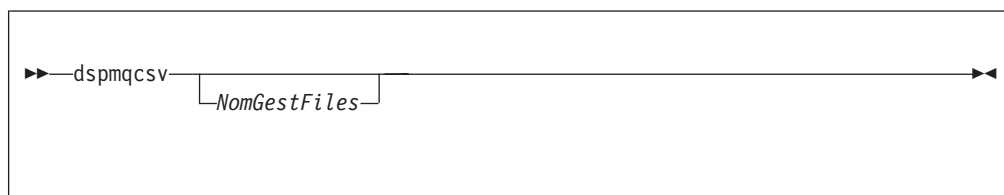
### Objet

Utilisez la commande **dspmqcsv** 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



### Paramètres facultatifs

*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` :

```
dspmqcsv "gest.files.vénus"
```

### Commandes connexes

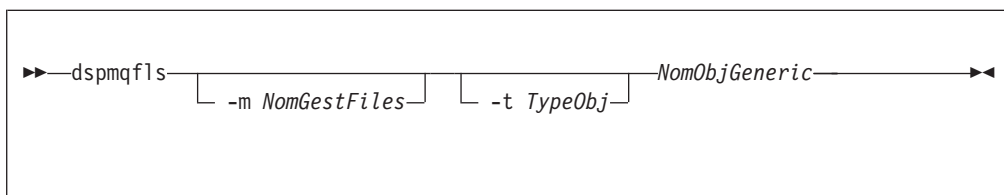
- |                 |                                     |
|-----------------|-------------------------------------|
| <b>strmqcsv</b> | lancement d'un serveur de commandes |
| <b>endmqcsv</b> | arrêt d'un serveur de commandes     |

## dspmqls (affichage des fichiers MQSeries)

### Objet

Utilisez la commande **dspmqls** pour afficher le nom du système de fichiers réel 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 «Signification des noms de fichiers MQSeries» à la page 23.

### Syntaxe



### Paramètres obligatoires

#### *NomObjGeneric*

Indique le nom de l'objet MQSeries. Le nom est une chaîne de caractères sans indicateur et 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 **dspmqfls** affiche le répertoire contenant la file d'attente, mais *pas* le nom de cette file.

**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.

```
dspmqfls 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.

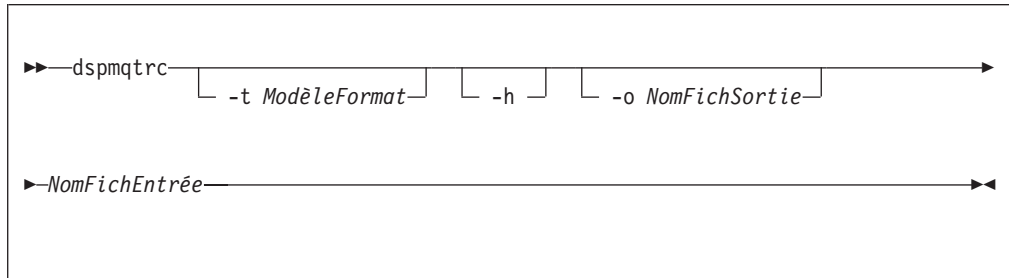
```
dspmqfls -m RAYON -t prcs PROC*
```

## dspmqrtrc (affichage de la sortie de trace formatée MQSeries)

### Objet

La commande **dspmqrtrc** permet d'afficher une sortie de trace formatée MQSeries.

### Syntaxe



### Paramètres obligatoires

*NomFichEntrée*

Indique le nom du fichier contenant les traces non formatées. Par exemple :  
MQS\_ROOT:[MQM.TRACE]AMQ20202345.TRC.

### Paramètres facultatifs

**-t** *ModèleFormat*

Indique le nom du fichier modèle contenant les détails d'affichage de la trace.  
La valeur par défaut est SYS\$SHARE:AMQTRC.FMT.

**-h** Permet d'omettre les informations de l'en-tête du rapport.

**-o** *NomFichSortie*

Nom du fichier dans lequel les données formatées doivent être écrites.

### Exemples

1. La commande suivante affiche la redirection de la sortie :

```
dspmqrtrc mqs_root:[mqm.trace]amq20202345.trc > mqs_root:[mqm.trace]amq20202345.fmt
```

## **Commandes connexes**

<b>endmqtrc</b>	Arrêt de la trace MQSeries
<b>strmqtrc</b>	Lancement de la trace MQSeries

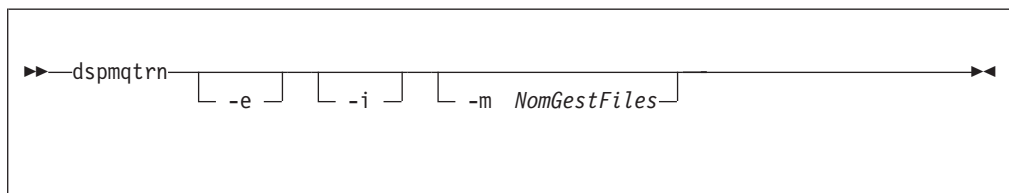
## dspmqrn (affichage des transactions MQSeries)

### Objet

La commande **dspmqrn** permet d'afficher les transactions en cours de préparation dans une procédure de validation en deux phases et connues du gestionnaire de files d'attente (lisez la note Attention ci-dessous).

Pour chaque transaction, un numéro de transaction (identificateur de transaction lisible par l'utilisateur), l'état de la transaction et l'ID transaction sont affichés. La longueur des ID de transactions peut être comprise entre 1 et 128 caractères, d'où le besoin d'un numéro de transaction.

### Syntaxe



**Attention :** Vous ne serez amené à utiliser cette commande que si vous utilisez un gestionnaire de transaction externe et des procédures de validation à deux phases. Si vous n'utilisez pas la validation à deux phases, n'appliquez pas cette commande. Utilisez-la seulement si le gestionnaire de points de synchronisation n'est pas parvenu à effectuer la transaction.

### Paramètres facultatifs

#### **-m** *NomGestFiles*

Indique le nom du gestionnaire de files d'attente dont les transactions doivent être examinées. Si aucun nom n'est indiqué, la commande s'applique au gestionnaire de files d'attente par défaut.

- e** Demande les détails des transactions en attente coordonnées en externe. Ces transactions sont celles pour lesquelles MQSeries a reçu une demande de préparation de validation sans avoir encore reçu d'informations sur l'issue de la transaction.
- i** Demande les détails des transactions en attente coordonnées en interne. Ces transactions sont celles pour lesquelles chaque gestionnaire de ressources a reçu une demande de préparation de validation, mais MQSeries n'a pas encore informé les gestionnaires de ressources de l'issue de la transaction.

Des informations sont affichées sur l'état supposé de la transaction dans chacun des gestionnaires de ressources y participant. Ces informations peuvent vous aider à évaluer les effets de l'échec d'un gestionnaire de ressources déterminé.

**Remarque :** Si vous ne précisez ni -e ni -i, les détails des transactions en attente coordonnées en externe et en interne sont affichées.

### 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.
- 102 Aucune transaction n'a été trouvée.

## **Commandes connexes**

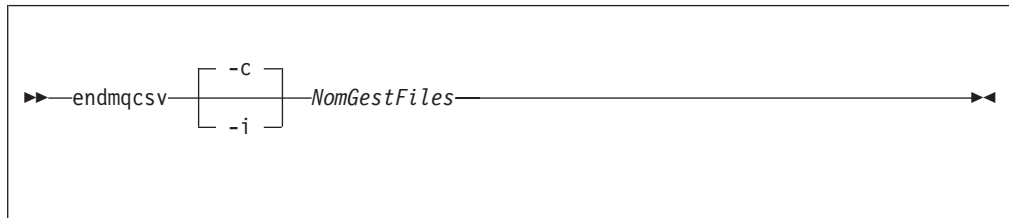
**rsvmqtrn** Résolution des transactions MQSeries

## 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 type d'arrêt par défaut.

**-i** Indique que le serveur de commandes doit être arrêté immédiatement. Les actions associées au 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"
```

**endmqcsv**

## **Commandes connexes**

<b>strmqcsv</b>	lancement d'un serveur de commandes
<b>dspmqcsv</b>	affichage du serveur de commandes

---

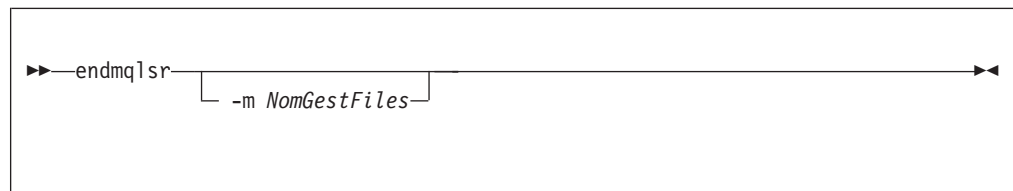
## endmq1sr (arrêt du programme d'écoute)

### Objet

La commande **endmq1sr** arrête tous les processus d'écoute pour le gestionnaire de files d'attente spécifié.

Le gestionnaire de files d'attente doit être arrêté avant de lancer la commande **endmq1sr**.

### Syntaxe



### Paramètres facultatifs

**-m** *NomGestFiles*

Définit le nom du gestionnaire de files d'attente. Si aucun nom n'est indiqué, la commande s'exécute sur le gestionnaire par défaut.

### 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.

## endmqm (arrêt d'un gestionnaire de files d'attente)

### Objet

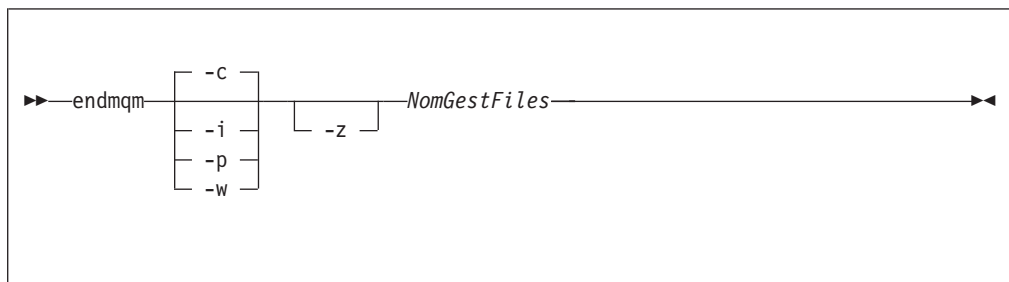
Utilisez la commande **endmqm** pour arrêter un gestionnaire local de files d'attente. Cette commande s'exécute selon l'un des trois modes suivants :

- 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 obligatoires

*NomGestFiles*

Indique le nom du gestionnaire de files d'attente de messages à arrêter.

### Paramètres facultatifs

**-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.

Vous reprenez la main immédiatement et vous n'êtes pas averti de l'arrêt du gestionnaire de files d'attente.

**-w** Arrêt avec attente.

Ce type d'arrêt équivaut à un arrêt contrôlé, excepté que vous ne reprenez la main qu'une fois le gestionnaire de files d'attente arrêté. Vous recevez le message "Attente de l'arrêt du gestionnaire de files d'attente *NomGestFiles*" pendant le processus d'arrêt.

**-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. Cela peut donner des résultats imprévisibles pour les applications MQSeries. Tous les processus du gestionnaire de files d'attente qui n'a pas réussi à s'arrêter sont terminés dans les 30 secondes suivant l'émission de la commande.

**Remarque :** Après un arrêt forcé, ou si le gestionnaire de files d'attente échoue, il se peut que ce dernier s'arrête sans avoir nettoyé sa mémoire partagée. Ceci peut provoquer des problèmes de redémarrage. Pour des informations sur l'utilisation de MONMQ pour effectuer ce nettoyage après un arrêt brutal de ce type, consultez la section «Gestion de la mémoire partagée avec MONMQ» à la page 364.

-z Supprime les messages d'erreur de la commande.

## Codes retour

0	Le gestionnaire de files d'attente est arrêté.
3	Le gestionnaire de files d'attente est en cours de création.
16	Le gestionnaire de files d'attente n'existe pas.
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.

## 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 appelé `mercure.gest.files`. Toutes les applications actuellement connectées ont la possibilité de se déconnecter.

```
endmqm "mercure.gest.files"
```

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"
```

**endmqm**

## **Commandes connexes**

<b>crtmqm</b>	Création d'un gestionnaire de files d'attente.
<b>strmqm</b>	Lancement d'un gestionnaire de files d'attente.
<b>dltmqm</b>	Suppression d'un gestionnaire de files d'attente.

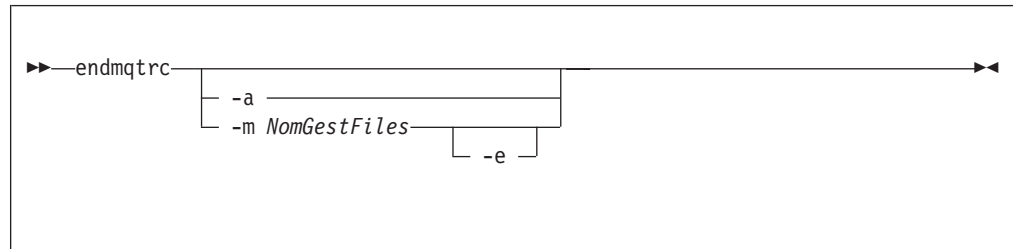


## endmqtrc (arrêt de la trace MQSeries)

### Objet

La commande **endmqtrc** permet d'arrêter le traçage pour le ou les éléments spécifiés.

### Syntaxe



### Paramètres facultatifs

#### **-m** *NomGestFiles*

Nom du gestionnaire de files d'attente dont la trace doit être arrêtée.

Un seul indicateur **-m** et le nom du gestionnaire de files correspondant peuvent être fournis dans la commande.

Cet indicateur **-m** et le nom du gestionnaire peuvent être spécifiés dans la même commande que l'indicateur **-e**.

**-e** Lorsque cet indicateur est spécifié, la trace initiale s'arrête.

**-a** Lorsque cet indicateur est spécifié, toute trace s'arrête.

Cet indicateur *doit* être spécifié seul.

### 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

**dspmqtrc**

Affichage d'une sortie de trace formatée.

**strmqtrc**

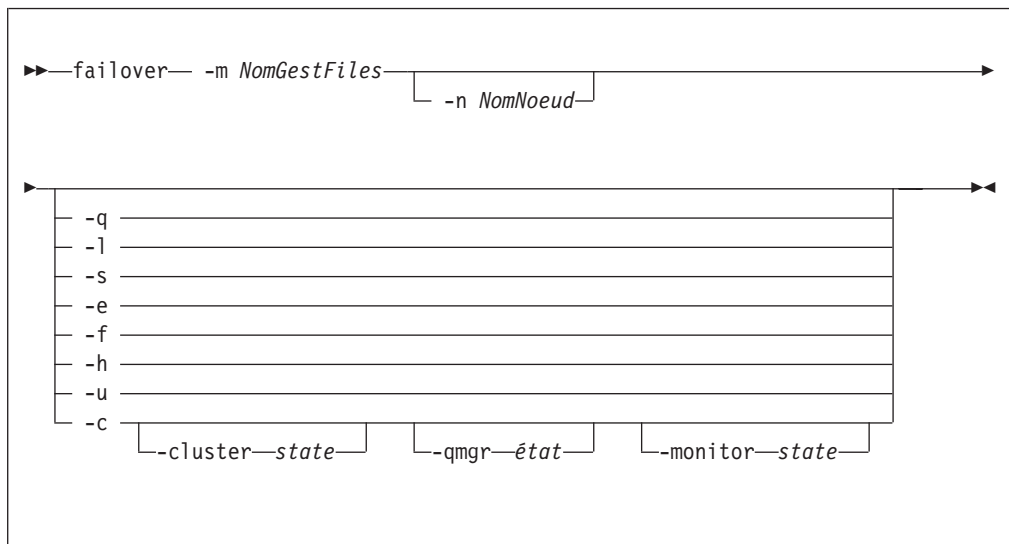
Lancement de la trace MQSeries

## failover (gestion d'un groupe de reprise)

### Objet

Utilisez la commande **failover** pour gérer un groupe de reprise. La commande **failover** comprend des paramètres de mise à jour et de requête. Elle peut être exécutée à partir de n'importe quel noeud OpenVMS du groupe de reprise.

### Syntaxe



### Paramètres obligatoires

**-m** *NomGestFiles*

Indique le nom du gestionnaire de files d'attente pour lequel la commande **failover** doit être appliquée. Le *NomGestFiles* doit faire moins de 25 caractères.

**-n** *NomNoeud*

Indique le nom du noeud OpenVMS auquel s'applique la commande. Cet attribut est nécessaire pour les paramètres -h et -c.

### Paramètres facultatifs

**-q** Interroge l'état du groupe de reprise et affiche le résultat.

**-l** Interroge l'état du groupe de reprise et définit les symboles DCL suivants :

Nom du symbole DCL	Description
MQS\$QMGR_NODE	Paramétré sur le noeud OpenVMS qui exécute le gestionnaire de files d'attente, et une chaîne vide si aucun GFA n'est exécuté.
MQS\$AVAILABLE_NODES	Paramétré sur la liste des noeuds OpenVMS disponibles pour exécuter le GFA. Il s'agit des noeuds qui sont dans l'état AVAILABLE et qui exécutent un moniteur de reprise.
MQS\$MONITOR_NODES	Paramétré sur la liste des noeuds OpenVMS qui exécutent un moniteur de reprise.

**-s** Démarre le gestionnaire de files d'attente du groupe de reprise. Si le paramètre

-n est précisé, le gestionnaire de files d'attente est démarré sur le noeud OpenVMS indiqué. Dans le cas contraire, il est démarré sur le noeud doté de la priorité la plus importante.

- e Arrête le gestionnaire de files d'attente du groupe de reprise.
- f Déplace le gestionnaire de files d'attente sur un autre noeud du groupe de reprise. Si le paramètre -n est précisé, le gestionnaire de files d'attente est déplacé sur le noeud indiqué. Dans le cas contraire, il est démarré sur le noeud doté de la priorité la plus importante.
- h Arrête le moniteur de reprise exécuté sur le noeud défini avec le paramètre -n.
- u Efface l'indicateur de mise à jour en cours.
- c Modifie l'état du groupe de reprise. Les états modifiés sont déterminés par les trois paramètres suivants. Les modifications prennent effet uniquement si elles sont cohérentes avec l'état d'exécution du groupe de reprise.
  - cluster started | stopped  
Utilisé avec le paramètre -c pour modifier l'état général du groupe de reprise.
  - qmgr available | running | excluded  
Utilisé avec le paramètre -c pour modifier l'état du gestionnaire de files d'attente du noeud défini avec le paramètre -n.
  - monitor started | stopped | watching  
Utilisé avec le paramètre -c pour modifier l'état du moniteur de reprise du noeud défini avec le paramètre -n.

## Codes retour

0	La commande a abouti.
5	Le gestionnaire de file d'attente est en cours d'exécution
36	Les arguments indiqués dans une commande ne sont pas valides
326	Le gestionnaire de files d'attente MQseries n'est pas exécuté actuellement
1925	Aucun moniteur de reprise n'est démarré pour le gestionnaire de files d'attente
1926	Mise à jour du groupe de reprise en cours
1937	Aucun noeud disponible pour démarrer le gestionnaire de files d'attente
1939	Arrêt forcé du gestionnaire de files d'attente
1940	Expiration de l'arrêt du gestionnaires de files d'attente avant achèvement

### Codes d'erreur OpenVMS

36	%SYSTEM-F-NOPRIV, droits insuffisants ou violation de la protection de l'objet
652	%SYSTEM-F-NOSUCHNODE, noeud distant inconnu
660	%SYSTEM-F-REJECT, objet de connexion au réseau rejeté

## Exemples

1. L'exemple qui suit présente l'interrogation de l'état du groupe de reprise pour un gestionnaire de files d'attente appelé testqm.

```
failover -m "testqm" -q
```

## failover

2. L'exemple qui suit démarre le gestionnaire de files d'attente appelé testqm sur le noeud batman.

```
failover -m "testqm" -n batman -s
```

3. Cet exemple déplace le gestionnaire de files d'attente appelé testqm vers le noeud doté du niveau de priorité le plus haut.

```
failover -m "testqm" -f
```

## Commandes connexes

`runmqfm` Démarrage d'un moniteur de reprise

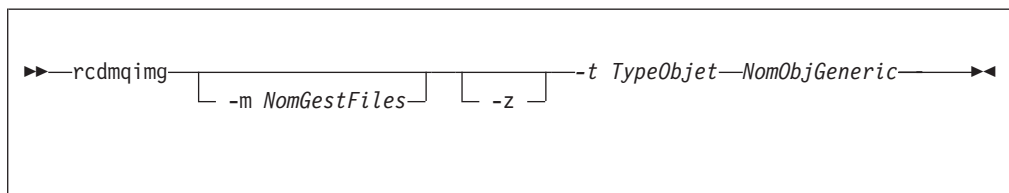
## rcdmqimg (enregistrement d'une image de support)

### Objet

Utilisez la commande **rcdmqimg** pour écrire une image d'un objet MQSeries ou d'un groupe d'objets, dans le journal afin qu'elle puisse être utilisée lors d'une récupération de support. Utilisez la commande associée **rcrmqobj** pour recréer l'objet à partir de l'image.

Cette commande est utilisée avec un gestionnaire de files d'attente actif. L'activité du gestionnaire est ensuite consignée si bien que le journal reflète toutes les modifications de l'objet, bien que l'image elle-même devienne périmée.

### Syntaxe



### Paramètres obligatoires

#### *NomObjGeneric*

Indique le nom de l'objet qui doit être enregistré. Ce paramètre peut comporter un astérisque de fin pour indiquer que tout objet dont le nom correspond à la portion de nom précédant l'astérisque doit être enregistré.

Ce paramètre est obligatoire *sauf si* vous enregistrez un objet gestionnaire de files d'attente ou le fichier de synchronisation de canaux. Si vous spécifiez un nom d'objet pour le fichier de synchronisation de canaux, celui-ci sera ignoré.

#### **-t** *TypeObjet*

Indique les types des objets dont les images doivent être enregistrées. Les types d'objets corrects sont :

#### **prcs** ou **process**

Processus

**q** ou **queue** Tous les types de files

**ql** ou **qlocal** Files d'attente locales

**qa** ou **qalias** Files d'attente alias

**qr** ou **qremote** Files d'attente éloignées

#### **qm** ou **qmodel**

Files d'attente modèles

**qmgr** Gestionnaire de files d'attente

**syncfile** Fichier de synchronisation du canal.

#### **nl** ou **namelist**

Listes de noms

#### **ctlg** ou **catalog**

Catalogue d'objets

**\* or all** Tout ce qui précède

## Paramètres facultatifs

**-m** *NomGestFiles*

Indique le nom du gestionnaire de files d'attente pour lequel des images doivent être enregistrées. Si aucun nom n'est indiqué, la commande s'applique au gestionnaire de files d'attente par défaut.

**-z** Supprime les messages d'erreur.

## 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.
68	La reconstitution du support n'est pas prise en charge.
69	La mémoire n'est pas disponible.
71	Erreur inattendue.
72	Le nom du gestionnaire de files d'attente est erroné.
119	Utilisateur non autorisé.
128	Aucun objet traité.
131	Incident lié aux ressources.
132	Objet détérioré.
135	Un objet temporaire ne peut pas être enregistré.

## Exemples

La commande suivante enregistre une image de l'objet gestionnaire de files d'attente `saturne.gest.files` dans le journal.

```
rcdmqimg -t qmgr -m "saturne.gest.files"
```

## Commandes connexes

`rcrmqobj`      Recréation d'un objet gestionnaire de files d'attente

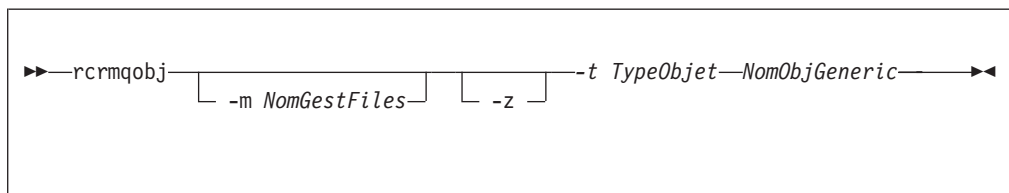
## rcrmqobj (recréation d'un objet)

### Objet

Utilisez la commande **rcrmqobj** pour recréer un objet ou un groupe d'objets à partir de l'image contenue dans le journal. Utilisez la commande associée **rcdmqimg**, pour enregistrer les images des objets dans le journal.

Cette commande doit être utilisée avec un gestionnaire de files d'attente actif. Toutes les activités du gestionnaire postérieures à l'enregistrement de l'image sont consignées dans le journal. Pour recréer un objet, vous devez relire le journal pour recréer les événements qui se sont produits après la capture de l'image.

### Syntaxe



### Paramètres obligatoires

#### *NomObjGeneric*

Indique le nom de l'objet qui doit être recréé. Ce paramètre peut comporter un astérisque de fin pour indiquer que tout objet dont le nom correspond à la portion de nom précédant l'astérisque doit être recréé.

Ce paramètre est obligatoire *sauf* si le type d'objet est le fichier de synchronisation de canaux ; si un nom d'objet est fourni pour ce type, il est ignoré.

#### **-t** *TypeObjet*

Indique le type des objets à recréer. Les types d'objets corrects sont :

#### **prcs** ou **process**

Processus

**q** ou **queue** Tous les types de files

**ql** ou **qlocal** Files d'attente locales

**qa** ou **qalias** Files d'attente alias

**qr** ou **qremote** Files d'attente éloignées

#### **qm** ou **qmodel**

Files d'attente modèles

#### **nl** ou **namelist**

Listes de noms

#### **ctlg** ou **catalog**

Catalogue d'objets

**\*** ou **all** Tout ce qui précède

**syncfile** Fichier de synchronisation des canaux



**Remarque :** L'utilisation de cet indicateur provoque la génération du fichier de synchronisation de canal pour le gestionnaire de files d'attente spécifié. Ceci est nécessaire car le fichier n'est pas enregistré par la commande **rcdmqimg**.

## Paramètres facultatifs

**-m** *NomGestFiles*

Indique le nom du gestionnaire de files d'attente pour lequel des objets doivent être recréés. Si aucun nom n'est indiqué, la commande s'applique au gestionnaire de files d'attente par défaut.

**-z** Supprime les messages d'erreur.

## 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.
66	L'image du support n'est pas disponible.
68	La reconstitution du support n'est pas prise en charge.
69	La mémoire n'est pas disponible.
71	Erreur inattendue.
72	Le nom du gestionnaire de files d'attente est erroné.
119	Utilisateur non autorisé.
128	Aucun objet traité.
135	Un objet temporaire ne peut pas être récupéré.
136	L'objet est en cours d'utilisation.

## Exemples

1. La commande suivante recrée toutes les files d'attente locales du gestionnaire de files d'attente par défaut :

```
rcrmqobj -t ql *
```

2. La commande suivante recrée toutes les files d'attente éloignées associées au gestionnaire de files d'attente magasin :

```
rcrmqobj -m "magasin" -t qr *
```

**rcrmqobj**

## **Commandes connexes**

**rcdmqimg**      enregistrement d'une image de support

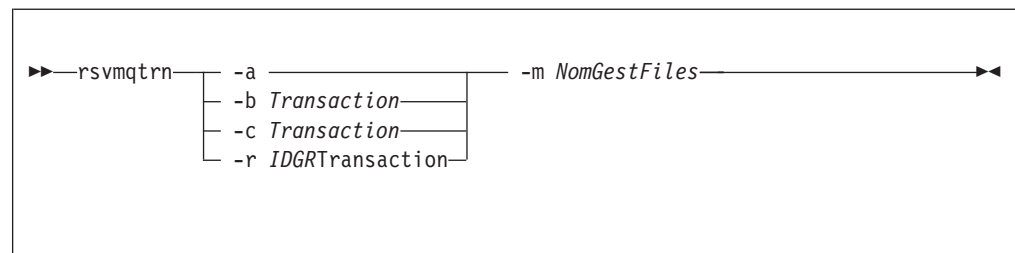
## rsvmqtrn (résolution des transactions MQSeries)

### Objet

Utilisez la commande **rsvmqtrn** pour valider ou annuler des transactions en attente coordonnées en interne ou en externe.

Cette commande ne doit être utilisée que si vous êtes certain que les transactions ne peuvent pas être résolues par les protocoles normaux. L'émission de cette commande peut compromettre l'intégrité de la transaction entre les gestionnaires de ressources dans le cas d'une transaction répartie.

### Syntaxe



### Paramètres obligatoires

**-m** *NomGestFiles*

Définit le nom du gestionnaire de files d'attente. Ce paramètre est obligatoire.

### Paramètres facultatifs

- a** Indique que le gestionnaire de files d'attente doit tenter de résoudre toutes les transactions en attente coordonnées en interne (soit toutes les unités de travail globales).
- b** Indique que la transaction indiquée doit être annulée. Cet indicateur est valable uniquement pour les transactions coordonnées en externe (c'est-à-dire les unités de travail externes).
- c** Indique que la transaction indiquée doit être validée. Cet indicateur est valable uniquement pour les transactions coordonnées en externe (c'est-à-dire les unités de travail externes).

**-r** *RMI d*

Indique le gestionnaire de ressources auquel la décision de validation ou d'annulation s'applique. Cet indicateur est valable uniquement pour les transactions coordonnées en interne, et pour les gestionnaires de ressources qui ne sont plus configurés dans le fichier *qm.ini* du gestionnaire de files d'attente. Le résultat fourni correspondra à la décision de MQSeries quant à la transaction.

*Transaction*

Indique le numéro de la transaction concernée. Ce paramètre peut être déterminé à l'aide de la commande **dspmqtrn**. Il est requis pour les paramètres *RMI d* -b, -c et -r.

### Codes retour

0 Opération réussie.

## **rsvmqtrn**

- 32 Les transactions n'ont pas pu être résolues.
- 34 Gestionnaire de ressources non reconnu.
- 35 Le gestionnaire de ressources n'est pas constamment disponible.
- 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.
- 85 Transactions inconnues.

## **Commandes connexes**

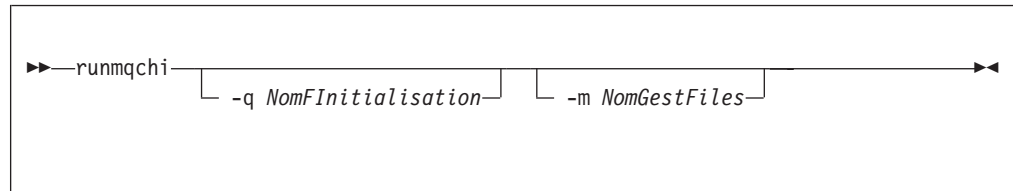
**dspmqtrn**      affichage des transactions MQSeries

## 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 utilisé.

#### **-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 donnant 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 ici 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 203.

---

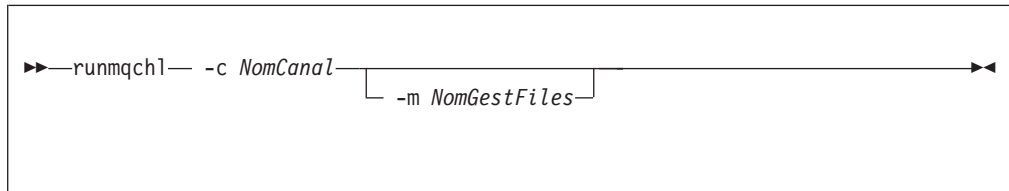
## runmqchl (démarrage d'un canal)

### Objet

Utilisez la commande **runmqchl** pour démarrer un canal émetteur (SDR), ou demandeur (RQSTR).

Le canal fonctionne en mode synchrone. Pour l'arrêter, émettez 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 car c'est ici 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

Utilisez la commande **runmqdlq** pour 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.

Le gestionnaire de files d'attente de rebut peut être utilisé pour effectuer diverses actions sur des messages sélectionnés en indiquant un ensemble de règles qui permettent à la fois de sélectionner un message et de définir l'action à exécuter sur ce dernier.

La commande **runmqdlq** est entrée via SYS\$INPUT. Une fois la commande traitée, les résultats et un résumé sont placés dans un état envoyé à SYS\$OUTPUT.

En saisissant SYS\$INPUT à partir du clavier, vous pouvez entrer les règles **runmqdlq** de façon interactive.

En redirigeant l'entrée à partir d'un fichier, vous pouvez appliquer une table de règles à la file indiquée. La table de règles doit comporter au moins une règle.

Si le gestionnaire DLQ est utilisé sans redirection de SYS\$INPUT à partir d'un fichier (la table de règles), le gestionnaire :

- lit ses entrées à partir du clavier,
- ne commence pas à traiter la file d'attente spécifiée avant de recevoir un caractère de fin de fichier (ctrl-Z).

Pour plus de détails sur les tables de règles et leur élaboration, reportez-vous à la «règle du gestionnaire DLQ» à la page 104.

### Syntaxe



### Paramètres facultatifs

Les règles MQSC concernant les lignes de commentaire et la jonction de lignes s'appliquent aussi aux paramètres d'entrée du gestionnaire DLQ.

#### *NomF*

Indique le nom de la file à traiter.

Si aucun nom n'est indiqué, la commande s'exécute sur la file d'attente de rebut du gestionnaire local. Si un ou plusieurs espaces ( ' ') sont utilisés, la file d'attente de rebut du gestionnaire de files d'attente local est explicitement affectée.

Un gestionnaire DLQ peut être utilisé pour sélectionner des messages particuliers dans une file d'attente de rebut en vue d'un traitement spécial. Par exemple, il peut s'agir de rediriger les messages vers différentes files d'attente

## runmqdlq

de rebut. Une instance ultérieure du gestionnaire DLQ peut alors traiter les messages, suivant une table de règles différente.

### *NomGestFiles*

Nom du gestionnaire de files d'attente propriétaire de la file à traiter.

Si aucun nom n'est indiqué, le gestionnaire de files d'attente par défaut de l'installation est utilisé. Si un ou plusieurs espaces ( ' ') sont utilisés, le gestionnaire de files d'attente par défaut de l'installation est explicitement affecté.

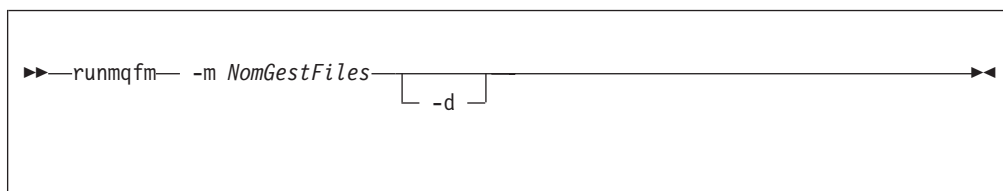


## runmqfm (lancement d'un moniteur de reprise)

### Objet

La commande **runmqfm** permet de lancer un moniteur de reprise sur un noeud OpenVMS. Le moniteur de reprise est exécuté sur le noeud OpenVMS sur lequel la commande **runmqfm** est émise.

### Syntaxe



### Paramètres obligatoires

**-m** *NomGestFiles*

Indique le nom du gestionnaire de files d'attente pour lequel la commande **runmqfm** doit être lancée. Le *NomGestFiles* doit faire moins de 25 caractères.

### Paramètres facultatifs

**-d** Indique que des informations de débogage complémentaires doivent être consignées dans le fichier journal.

### 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

Dans l'exemple qui suit, un moniteur de mise hors ligne pour le gestionnaire de files d'attente `testqm` est démarré. Il écrit des informations de débogage dans un fichier journal appelé `test.log`.

```
runmqfm -m "testqm" -d > test.log
```

### Commandes connexes

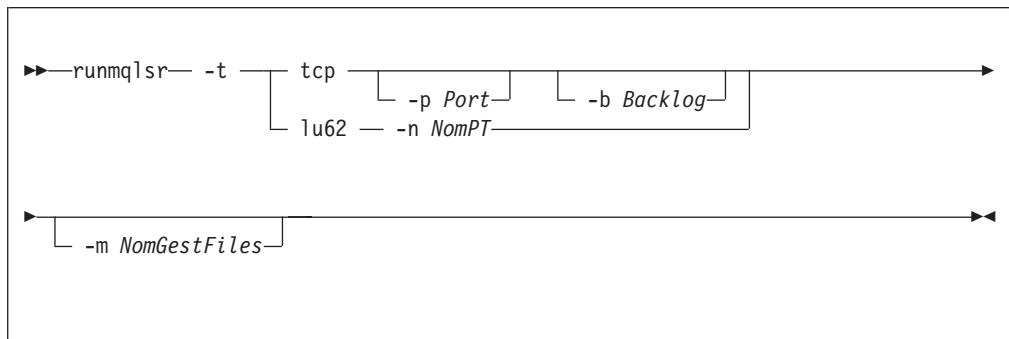
**failover** Gestion d'un groupe de reprise

## runmqlsr (exécution du programme d'écoute)

### Objet

La commande **runmqlsr** permet d'exécuter un processus d'écoute.

### Syntaxe



### Paramètres obligatoires

- t Définit le protocole de transmission à utiliser :
  - tcp** TCP/IP (Transmission Control Protocol / Internet Protocol)
  - lu62** SNA LU 6.2. (Les dernières informations concernant l'utilisation de ce paramètre se trouvent dans `sys$help:mqseries0510.release_notes.`)

### Paramètres facultatifs

- p *Port*  
Numéro de port pour TCP/IP. Cet indicateur n'est valable que pour TCP et UDP. Si aucune valeur n'est indiquée, elle est extraite du fichier de configuration du gestionnaire de files d'attente ou des valeurs par défaut définies dans le programme. Valeur par défaut : 1414.
- n *NomPT*  
Nom du programme transactionnel LU 6.2. Ce paramètre est valable pour le protocole de transmission LU 6.2 uniquement. Si aucune valeur n'est indiquée, elle est extraite du fichier de configuration du gestionnaire de files d'attente. Si la valeur n'existe pas, 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.
- b *Backlog*  
Indique le nombre de demandes de connexion simultanées prises en charge par le programme d'écoute. Pour obtenir une liste des valeurs par défaut ainsi que d'autres informations, reportez-vous à la section «Strophes LU62 et TCP» à la page 187.

### 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 ci-après exécute un programme d'écoute sur le gestionnaire de files d'attente par défaut avec le protocole TCP/IP. Elle précise que le programme d'écoute doit utiliser le port 4321.

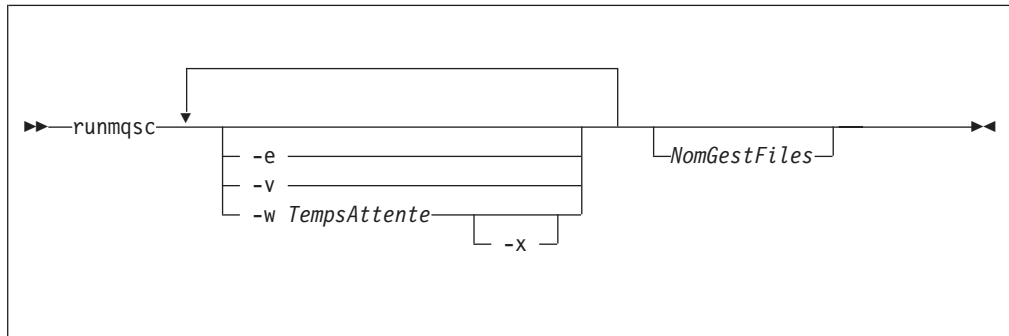
```
runmqsr -t tcp -p 4321
```

## 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 *Guide de programmation d'applications*.

### Syntaxe



### Description

Vous pouvez appeler la commande **runmqsc** suivant 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.

**Remarque :** L'ID utilisateur qui exécute le gestionnaire de files d'attente doit exister en local et disposer des droits d'accès adéquats.

La commande **runmqsc** est entrée via SYS\$INPUT. Une fois les commandes traitées, les résultats et un résumé sont placés dans un état envoyé à SYS\$OUTPUT.

En saisissant SYS\$INPUT à partir du clavier, vous pouvez entrer les commandes de façon interactive.

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.

## 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.
- v Indique le mode vérification (vérification des commandes indiquées sans exécution). Ce mode n'est disponible qu'en mode 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 68 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 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

- 0 Le traitement du fichier de commandes MQSC a abouti.
- 10 Le fichier de commandes MQSC a été traité avec des erreurs ; le rapport contient les raisons des échecs des commandes.
- 20 Erreur— le fichier de commandes MQSC n'a pas été exécuté.

## Exemples

1. Saisissez cette commande à l'invite OpenVMS :

```
runmqsc
```

## runmqsc

Vous pouvez maintenant saisir directement les commandes MQSC à l'invite OpenVMS. Aucun gestionnaire de files d'attente n'a été indiqué. Par conséquent, les commandes MQSC sont traitées sur le gestionnaire de files d'attente par défaut.

2. Utilisez cette commande pour indiquer que les commandes MQSC sont seulement vérifiées :

```
runmqsc -v BANQUE < DKA0:[USERS]FICHCOMM.IN
```

Cette commande vérifie le fichier de commandes MQSC `COMMFILE.IN` dans le répertoire `DKA0:[USERS]`. Le nom du gestionnaire de files d'attente est `BANK`. La sortie s'affiche dans la fenêtre en cours.

3. Cette commande exécute le fichier de commande MQSC `MQS_ROOT:[MQM.MQSC]MQSCFILE.IN` sur le gestionnaire de files d'attente par défaut.

```
runmqsc < MQS_ROOT:[MQM.MQSC]MQSCFILE.IN > MQS_ROOT:[MQM.MQSC]MQSCFILE.OUT
```

Dans cet exemple, la sortie est dirigée vers le fichier `MQS_ROOT:[MQM.MQSC]MQSCFILE.OUT`.

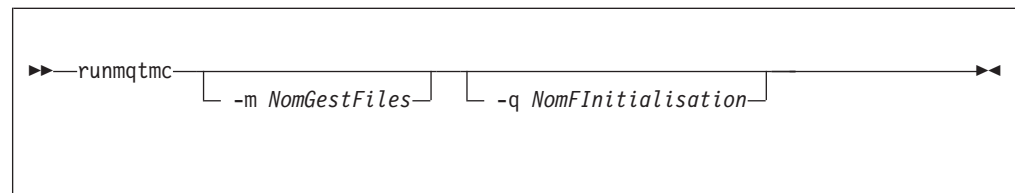
## runmqtmc (lancement du moniteur de déclenchement client)

### Objet

Utilisez la commande **runmqtmc** pour appeler un moniteur de déclenchement pour un client. Pour plus d'informations sur les moniteurs de déclenchement, reportez-vous au *Guide de programmation d'applications*.

**Remarque :** Cette commande est disponible *uniquement* sur les clients OpenVMS, OS/2, et AIX.

### Syntaxe



### Paramètres facultatifs

**-m** *NomGestFiles*

Indique le nom du gestionnaire de files d'attente sur lequel fonctionne le moniteur de déclenchement client. Si le nom est omis, le gestionnaire par défaut est utilisé.

**-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

- 0 Inutilisé. Le moniteur de déclenchement client est défini pour un fonctionnement continu et ne peut donc s'arrêter. Cette valeur est réservée.
- 10 Le moniteur de déclenchement client a été interrompu par une erreur.
- 20 Erreur— le moniteur de déclenchement client ne s'exécute pas.

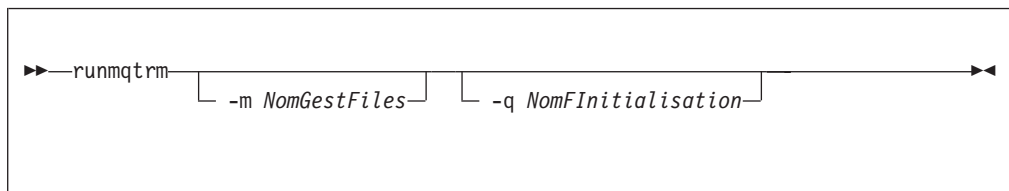
---

## 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 *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

- 0 Inutilisé. Le moniteur de déclenchement est défini pour fonctionner continuellement et, par conséquent, ne peut s'arrêter. Cette valeur est réservée.
- 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

►► `setmqaut` — `-m NomGestFiles` — `-n NomObjet` — `-t TypeObjet` —————►

—————►  
   | `-s ComposantService` |           | `-p NomPrincipal` |  
   |                           |           | `-g NomGroupe` |

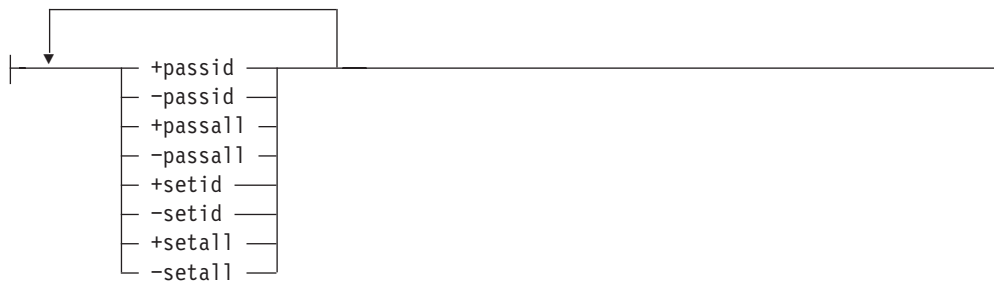
—————►  
Droits d'accès MQI		Droits d'accès MQI
Droits de contexte		Droits de contexte
Droits d'administration		Droits d'administration
Droits génériques		Droits génériques

#### Droits d'accès MQI :

—————►  
`+get`	
`-get`	
`+browse`	
`-browse`	
`+put`	
`-put`	
`+inq`	
`-inq`	
`+set`	
`-set`	
`+connect`	
`-connect`	
`+altusr`	
`-altusr`	

#### Droits de contexte :

## setmqaut



### 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 et principaux dans une seule commande.

**Remarque :** si vous spécifiez un ensemble de droits d'accès pour un principal, les mêmes droits d'accès sont accordés à tous les principaux du même groupe primaire.

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

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

### -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.

### -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.

### -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**

## Paramètres facultatifs

### -p *NomPrincipal*

Indique le nom du principal dont les droits d'accès doivent être modifiés.

Vous devez indiquer au moins un principal ou un groupe.

### -g *NomGroupe*

Indique le nom de l'identificateur de droits représentant le groupe d'utilisateurs 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 -g.

### -s *ComposantFonction*

Ce paramètre ne s'applique que si vous utilisez des fonctions de droits d'accès optionnelles.

## setmqaut

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 17 présente les droits qui peuvent être accordés suivant les différents types d'objets.

Tableau 17. 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	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

### Droits d'accès MQI

**altusr** Utilisation d'un autre ID utilisateur dans un message.

Reportez-vous au manuel *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.

- 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** La commande a abouti.
- 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.

## 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
```

3. Dans cet exemple, la liste des droits d'accès indique que le groupe valse a le droit de créer et de supprimer le gestionnaire de files d'attente saturne.gest.files :

```
setmqaut -m "saturne.gest.files" -t qmgr -g "waltz" +crt +dlt
```

## **Commandes connexes**

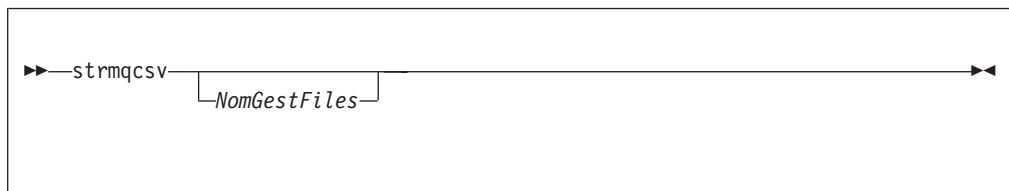
**dspmqaut**      Affichage des droits d'accès

## 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



### Paramètres facultatifs

#### *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



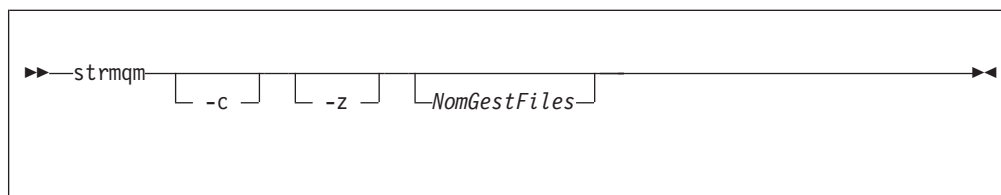
## strmqm (lancement d'un gestionnaire de files d'attente)

### Objet

Utilisez la commande **strmqm** pour lancer un gestionnaire de files d'attente local.

**Remarque :** Avant d'utiliser la commande **strmqm**, ou toute autre commande de contrôle, `mqs_startup` doit avoir été exécuté une fois après la dernière réinitialisation avant que le gestionnaire de files d'attente puisse être démarré et exécuté.

### 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.
23	Le journal 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.
100	L'emplacement du journal est incorrect.

## strmqm

### Exemples

La commande suivante lance le gestionnaire de files d'attente comptes :

```
strmqm "comptes"
```

### Commandes connexes

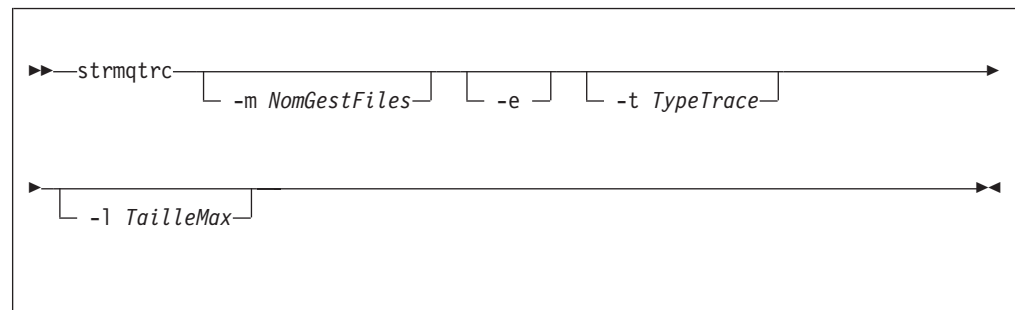
<b>crtmqm</b>	Création d'un gestionnaire de files d'attente.
<b>dltmqm</b>	Suppression d'un gestionnaire de files d'attente.
<b>endmqm</b>	Arrêt d'un gestionnaire de files d'attente.

## 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.

### Syntaxe



### Description

Vous pouvez demander un niveau de détail de trace plus ou moins important. Pour chaque valeur TypeTrace de flot spécifiée, y compris -t all, précisez -t parms ou -t detail pour tout type de trace particulier. Seul le niveau de détail de trace par défaut sera généré pour ce type de trace.

La section «Traçage de MQSeries» à la page 208 présente des exemples de données de trace générées par cette commande.

### Paramètres facultatifs

#### -m *NomGestFiles*

Nom du gestionnaire de files d'attente dont la trace doit être exécutée.

Cet indicateur -m et un nom du gestionnaire peuvent être spécifiés 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.

L'indicateur -m et le nom du gestionnaire sont obligatoires, sauf si vous spécifiez l'indicateur -e.

- e Lorsque cet indicateur est spécifié, la trace initiale est demandée. Il est donc possible de tracer la création ou le démarrage d'un gestionnaire de files d'attente. 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.

#### -t *TypeTrace*

Définit les points à tracer lors du traitement. Si cet indicateur est omis, tous les points de trace sont activés et la trace complète est générée.

Vous pouvez également fournir une ou plusieurs options de la liste suivante.

**Remarque :** Si plusieurs types de trace sont fournis, chaque type *doit* avoir son propre indicateur -t. Vous pouvez indiquer n'importe quel nombre d'indicateurs -t, pourvu que chacun soit associé à un type de trace autorisé.

Vous pouvez indiquer le même type de trace pour plusieurs indicateurs -t.

<b>all</b>	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.
<b>api</b>	Données de sortie des points de trace associés au MQI et aux composants principaux du gestionnaire de files d'attente.
<b>comms</b>	Données de sortie des points de trace associés aux données transmises par des réseaux de communication.
<b>csflows</b>	Données de sortie des points de trace associés au flot de traitement des fonctions communes.
<b>lqmflores</b>	Données de sortie des points de trace associés au flot de traitement du gestionnaire de files d'attente local.
<b>remoteflores</b>	Données de sortie des points de trace associés au flot de traitement du composant de communication.
<b>otherflores</b>	Données de sortie des points de trace associés au flot de traitement d'autres composants.
<b>csdata</b>	Données de sortie des points de trace associés aux tampons de données internes des fonctions communes.
<b>lqmdata</b>	Données de sortie des points de trace associés aux tampons de données internes du gestionnaire de files d'attente local.
<b>remotedata</b>	Données de sortie des points de trace associés aux tampons de données internes du composant de communication.
<b>otherdata</b>	Données de sortie des points de trace associés aux tampons de données internes d'autres composants.
<b>versiondata</b>	Données de sortie des points de trace associés à la version de MQSeries en cours d'exécution.
<b>commentary</b>	Données de sortie des points de trace associés aux commentaires des composants MQSeries.

#### **-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.

## **Codes retour**

<b>AMQ7024</b>	Ce message est généré si des arguments incorrects sont fournis dans la commande.
<b>AMQ8304</b>	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

<b>dspmqtrc</b>	Affichage d'une sortie de trace formatée.
<b>endmqtrc</b>	Arrêt de la trace MQSeries



---

## Partie 3. Annexes





---

## Annexe A. MQSeries pour Compaq OpenVMS, généralités

---

### Programme et référence

- 5724-A38 MQSeries pour Compaq OpenVMS, Alpha Version 5.1, référence 0790997.

---

### Caractéristiques matérielles requises

Les serveurs MQSeries peuvent être toute machine Compaq Alpha disposant d'au moins 128 Mo d'espace sur le disque système.

---

### Caractéristiques logicielles requises

Les caractéristiques logicielles requises sont les mêmes pour les environnements serveur et client Compaq OpenVMS, sauf indication contraire.

La version minimale est :

- Compaq OpenVMS Alpha Version 7.2-1.

---

### Connectivité

MQSeries pour Compaq OpenVMS est compatible avec les protocoles et matériels réseaux suivants :

Protocoles réseau :

- SNA LU6.2
- TCP/IP
- DECnet Phase V

Et tous les matériels de communications compatibles avec DECnet, TCP/IP ou la passerelle DIGITAL DECnet/SNA.

**Dans le cas de liaisons DECnet :**

- DECnetPLUS Version 7.1 for OpenVMS Version 7.2-1

**Dans le cas de liaisons TCP/IP :**

- DIGITAL TCP/IP Services for OpenVMS AlphaV5.0.a et V5.1
- TCPWare V5.4 de Process Software
- Multinet V4.3 de Process Software

**Dans le cas de liaisons SNA** , le logiciel et la licence SNA APPC LU6.2 doivent être installés. Il doit également avoir accès à une passerelle SNA configurée et convenable.

- DECnet SNA Gateway ST V1.3
- DECnet SNA LU6.2 API V2.4

---

### Sécurité

MQSeries pour Compaq OpenVMS utilise les fonctionnalités de sécurité d'OAM (Object Authority Manager) pour MQSeries pour Compaq OpenVMS.

## Sécurité

Toutes les ressources MQSeries fonctionnent avec l'identificateur de droits VMS MQM. Cet identificateur de droits est créé lors de l'installation de MQSeries. Vous devez lui affecter cet attribut de ressources, pour tous les utilisateurs qui ont besoin de contrôler des ressources MQSeries.

---

## Fonctions de maintenance

MQSeries fonctionne avec :

- L'interface de ligne de commande `runmqsc`.
- 

## Compatibilité

MQI pour MQSeries pour Compaq OpenVMS Alpha, V5.1 est compatible avec les applications qui fonctionnent sous la Version 2.2.1.1.

### Compilateurs pris en charge

Les programmes peuvent être écrits en C, C++, COBOL, ou Java.

- Les programmes en C peuvent utiliser le compilateur C de DEC
  - Les programmes en C++ peuvent utiliser le compilateur C++ de DEC
  - Les programmes en COBOL peuvent utiliser le compilateur COBOL de DEC
  - Les programmes en Java peuvent utiliser le compilateur Java
- 

## Choix de la langue

Un fichier texte des messages est codé dans le jeu de caractères à 7 bits du système d'exploitation OpenVMS.

---

## Compatibilité internationale

Avec MQSeries pour Compaq OpenVMS, il est possible de préciser le CCSID lors de la création de l'instance du gestionnaire de files d'attente. La valeur par défaut du CCSID du gestionnaire de files d'attente est 819. MQSeries pour Compaq OpenVMS 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 OpenVMS, y compris ceux qui supportent le caractère euro.

---

## Annexe B. Objets système et objets par défaut

Lorsque vous créez un gestionnaire de files d'attente à l'aide de la commande de contrôle **crtmqm**, ses objets système et objets par défaut sont automatiquement créés.

- 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 18. 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.
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.
SYSTEM.CLUSTER.TRANSMIT.QUEUE	File d'attente de transmission de tous les messages destinés aux clusters. .

## Objets système et objets par défaut

Tableau 19. 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 20. Objets système et objets par défaut - Listes de noms

Nom d'objet	Description
SYSTEM.DEFAULT.NAMELIST	Liste de noms par défaut.

Tableau 21. Objets système et objets par défaut - Processus

Nom d'objet	Description
SYSTEM.DEFAULT.PROCESS	Définition de processus par défaut.

## Annexe C. Structure de répertoires

La figure 26 montre la structure générale des données et des répertoires de journaux associés à un gestionnaire de files d'attente particulier. Les répertoires présentés s'appliquent à l'installation par défaut. Si vous avez installé MQSeries dans d'autres répertoires, le répertoire principal est modifié en conséquence.

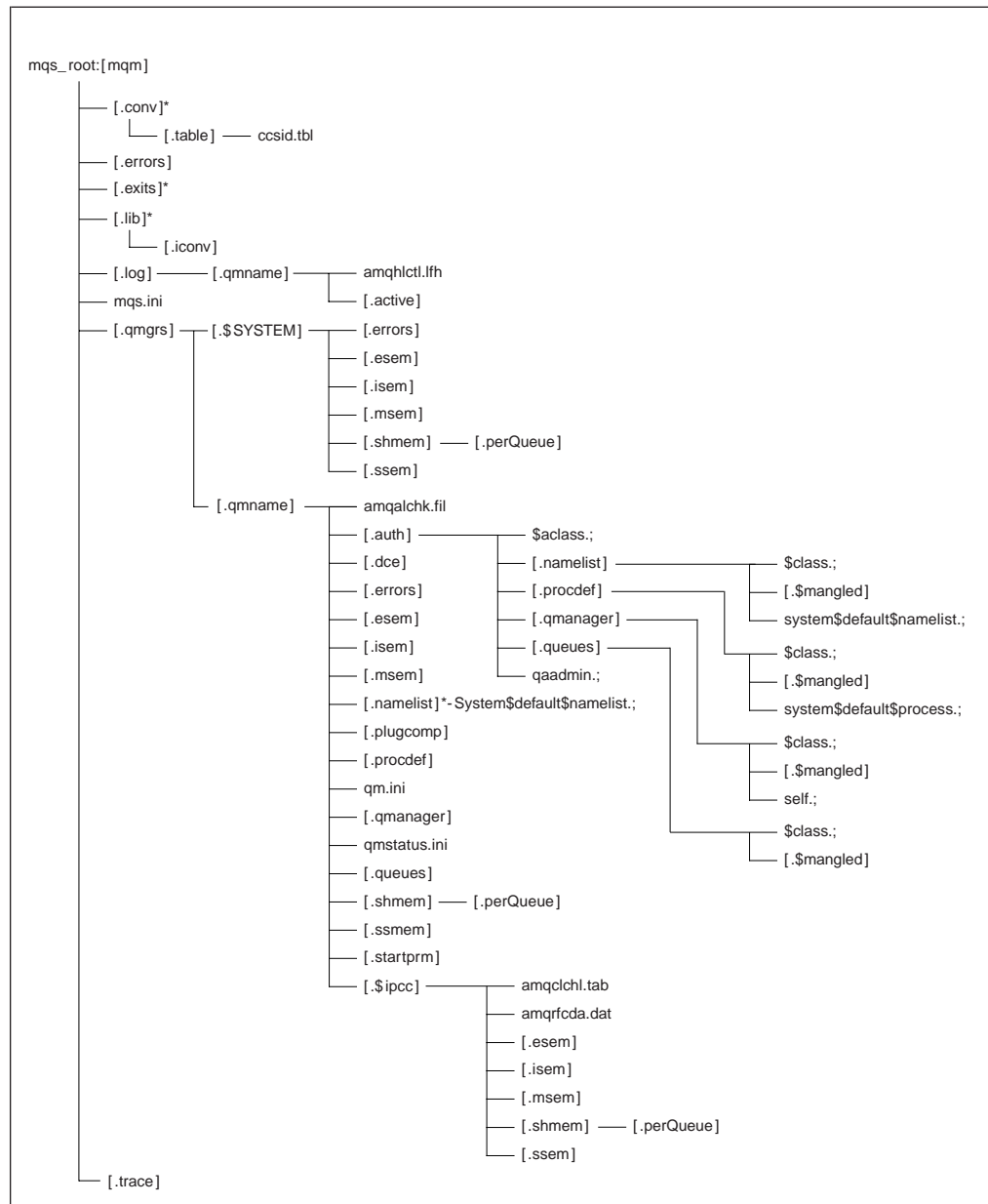


Figure 26. Arborescence de fichiers par défaut après le démarrage d'un gestionnaire de files d'attente

La figure 26, présente la structure de MQSeries après l'utilisation d'un gestionnaire de files d'attente pendant une certaine durée. La structure réelle dépend des opérations qui se sont produites sur le gestionnaire de files d'attente.

### Répertoire et fichiers dans MQS\_ROOT:[MQM]

Par défaut, les répertoires et fichiers suivants se trouvent dans le répertoire MQS\_ROOT:[MQM]:

- .conv** Répertoire contenant tous les fichiers utilisés pour la conversion des données.
- .table** Répertoire contenant le fichier ccsid.tbl.
- .errors** Répertoire contenant les fichiers de messages opérateur, classés du plus récent au plus ancien :
  - AMQERR01.LOG
  - AMQERR02.LOG
  - AMQERR03.LOG
- .exits** Répertoire vide contenant les exits écrits par les utilisateurs.
- .lib** Répertoire contenant le sous-répertoire .iconv. Celui-ci contient toutes les tables de conversion de jeux de codes.
- .iconv** Répertoire contenant les tables de conversion de jeux de codes (telles que les tables 002501B5.TBL à 44B031A8.TBL).
- .log** Les sous-répertoires et fichiers suivants existent une fois que vous avez installé MQSeries, créé et démarré un gestionnaire de files d'attente, et utilisé ce dernier pendant un certain temps.
  - amqhlctl.lfh**  
Fichier de contrôle de journal.
  - active** Répertoire contenant les fichiers journaux numérotés comme suit :
    - S0000000.LOG
    - S0000001.LOG
    - S0000002.LOG
    - ... et ainsi de suite.
- mqs.ini**  
Fichier de configuration MQSeries
- .qmgrs**  
Répertoire contenant un sous-répertoire `.$system` et un sous-répertoire `.qmname` pour chaque gestionnaire de files d'attente. Le répertoire `.$system` contient les fichiers et répertoires utilisés en interne par MQSeries. Pour plus d'informations sur le sous-répertoire `.qmname` consultez la section «`Répertoires et fichiers du sous-répertoire MQS_ROOT:[MQM.QMGRS.QMNAME]`».
- .trace** Ce répertoire contient les fichiers de trace créés à l'aide de la commande `strmqtrc`.

---

### Répertoires et fichiers du sous-répertoire MQS\_ROOT:[MQM.QMGRS.QMNAME]

Par défaut, les répertoires et fichiers suivants se trouvent dans le répertoire MQS\_ROOT:[MQM.QMGRS.QMNAME]. Le fichier `.QMNAME` est généré pour chaque gestionnaire de files d'attente créé et exécuté sur le système.

- amqalchk.fil**  
Fichier contenant des informations sur le dernier point de contrôle.

- .auth** Répertoire contenant des sous-répertoires et des fichiers associés aux droits d'accès.
- \$aclass;**  
Fichier contenant les strophes de droits d'accès de toutes les classes.
- .namelist**  
Répertoire contenant un fichier pour chaque liste de noms. Chaque fichier contient les strophes de droits d'accès pour la liste de noms associée.
- \$class;**  
Fichier contenant les strophes de droits d'accès de la classe des listes de noms.
- .\$mangled**  
Lorsque des noms de listes de noms contiennent des caractères non pris en charge par OpenVMS, ils sont automatiquement convertis en noms OpenVMS valides. Les noms autorisés par OpenVMS sont indiqués dans ce fichier. Consultez la section «Signification des noms de fichiers MQSeries» à la page 23.
- system\$default\$namelist**  
Fichier contenant les strophes de droits d'accès de la liste de noms par défaut.
- .procdef**  
Chaque définition de processus MQSeries est associée à un fichier de ce répertoire.
- \$class;**  
Fichier contenant les strophes de droits d'accès de la classe de définition de processus.
- .\$mangled**  
Lorsque des noms de définition de processus contiennent des caractères non pris en charge par OpenVMS, ils sont automatiquement convertis en noms OpenVMS valides. Les noms autorisés par OpenVMS sont indiqués dans ce fichier. Consultez la section «Signification des noms de fichiers MQSeries» à la page 23.
- .system\$default\$process;**  
Fichier contenant les strophes de droits d'accès des processus par défaut.
- .qmanager**  
Répertoire contenant un fichier pour chaque gestionnaire de file d'attente. Chaque fichier contient les strophes de droits d'accès du gestionnaire de files d'attente associé.
- \$class;**  
Fichier contenant les strophes de droits d'accès de la classe du gestionnaire de files d'attente.
- .\$mangled**  
Lorsque des gestionnaires de files d'attente contiennent des caractères non pris en charge par OpenVMS, ils sont automatiquement convertis en noms OpenVMS valides. Les

## Structure de répertoires

noms autorisés par OpenVMS sont indiqués dans ce fichier. Consultez la section «Signification des noms de fichiers MQSeries» à la page 23.

**self.;** Fichier contenant les strophes de droits d'accès de l'objet gestionnaire de files d'attente.

### **.queues**

Répertoire contenant un fichier pour chaque file d'attente. Chaque fichier contient les strophes de droits d'accès de la file d'attente associée.

### **\$CLASS**

Fichier contenant les strophes de droits d'accès de la classe de file d'attente.

### **.\$mangled**

Lorsque des noms de files d'attente contiennent des caractères non pris en charge par OpenVMS, ils sont automatiquement convertis en noms OpenVMS valides. Les noms autorisés par OpenVMS sont indiqués dans ce fichier. Consultez la section «Signification des noms de fichiers MQSeries» à la page 23.

### **Fichiers de définition de la file d'attente**

Chaque fichier correspond à un objet prédéfini pour le gestionnaire de files d'attente.

```
system$admin$channel$event.;
system$admin$command$queue.;
system$admin$perfm$event.;
system$admin$qmgr$event.;
system$channel$initq.;
system$channel$syncq.;
system$cics$initiation$queue.;
system$cluster$command$queue.;
system$cluster$repository$queue.;
system$cluster$transmit$queue.;
system$dead$letter$queue.;
system$default$alias$queue.;
system$default$initiation$queue.;
system$default$local$queue.;
system$default$model$queue.;
system$default$remote$queue.;
system$mqsc$reply$queue.;
```

### **.qaadmin.;**

Fichier utilisé en interne pour le contrôle des droits d'accès.

**.dce** Répertoire vide réservé à l'usage du support DCE.

**.errors** Répertoire contenant les fichiers de messages opérateur, classés du plus récent au plus ancien :

```
amqerr01.log
amqerr02.log
amqerr03.log
```

**.esem** Répertoire contenant des fichiers utilisés en interne.

**.isem** Répertoire contenant des fichiers utilisés en interne.

**.msem** Répertoire contenant des fichiers utilisés en interne.



### **.namelist**

Répertoire contenant les listes de noms pour chaque gestionnaire de file d'attente.

### **.plugcomp**

Répertoire vide réservé à l'usage des services optionnels.

### **.procdef**

Chaque définition de processus MQSeries est associée à un fichier de ce répertoire. Le nom du fichier correspond au nom de la définition de processus.

**qm.ini** Fichier de configuration du gestionnaire de files d'attente.

### **.qmanager**

Objet gestionnaire de files d'attente.

### **qmstatus.ini**

Fichier contenant une description du statut du gestionnaire de files d'attente.

### **.queues**

Chaque file d'attente dispose d'un répertoire contenant une seule file d'attente appelée 'q'.

Le nom de fichier correspond au nom de file d'attente. Il est soumis à certaines restrictions décrites dans la section «Signification des noms de fichiers MQSeries» à la page 23.

### **.shmem**

#### **.perQueue**

Répertoire contenant des fichiers utilisés en interne.

**.ssem** Répertoire contenant des fichiers utilisés en interne.

### **.startprm**

Répertoire contenant des fichiers temporaires utilisés en interne.

### **.\$ipcc**

#### **amqclchl.tab**

Fichier contenant la table du canal client.

#### **amqrfcda.dat**

Fichier contenant la table du canal.

**.esem** Répertoire contenant des fichiers utilisés en interne.

**.isem** Répertoire contenant des fichiers utilisés en interne.

**.msem** Répertoire contenant des fichiers utilisés en interne.

#### **.shmem**

#### **.perQueue**

Répertoire contenant des fichiers utilisés en interne.

**.ssem** Répertoire contenant des fichiers utilisés en interne.

## Structure de répertoires

---

## Annexe D. Comparaison des jeux de commandes

Les tableaux suivants comparent les fonctions disponibles depuis les différents jeux de commandes d'administration :

- «Commandes d'administration des gestionnaires de files d'attente»
- «Commandes d'administration des serveurs de commandes»
- «Commandes d'administration des files d'attente» à la page 328
- «Commandes d'administration des processus» à la page 328
- «Commandes d'administration des canaux» à la page 329
- «Autres commandes de contrôle» à la page 329

**Remarque :** Seules les commandes MQSC applicables à MQSeries pour Compaq OpenVMS sont indiquées.

---

### Commandes d'administration des gestionnaires de files d'attente

Tableau 22. Commandes d'administration des gestionnaires de files d'attente

Commandes PCF	Commandes 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
<b>Remarque :</b> * Commande non disponible sous PCF.		

---

### Commandes d'administration des serveurs de commandes

Tableau 23. Commandes d'administration des serveurs de commandes

Description	Commande de contrôle
Affichage du serveur de commandes	dspmqcsv
Lancement du serveur de commandes	strmqcsv
Arrêt du serveur de commandes	endmqcsv
<b>Remarque :</b> Les fonctions de ce groupe ne sont disponibles qu'en tant que commandes de contrôle. Il n'existe pas de commandes MQSC ou PCF équivalentes.	

## Commandes d'administration des files d'attente

Tableau 24. Commandes d'administration des files d'attente

Commandes PCF	Commandes MQSC
Change Queue	ALTER QLOCAL ALTER QALIAS ALTER QMODEL ALTER QREMOTE
Clear Queue	CLEAR QUEUE
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
<b>Remarque :</b> Il n'existe pas de commandes de contrôle équivalentes dans ce groupe.	

## Commandes d'administration des processus

Tableau 25. 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
<b>Remarque :</b> Il n'existe pas de commandes de contrôle équivalentes dans ce groupe.	

## Commandes d'administration des canaux

Tableau 26. Commandes d'administration des canaux

Commandes PCF	Commandes 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	–
Ping Channel	PING CHANNEL	–
Reset Channel	RESET CHANNEL	–
Resolve Channel	RESOLVE CHANNEL	–
Start Channel	START CHANNEL	runmqchl
Start Channel Initiator	START CHINIT	runmqchi
Start Channel Listener	–	runmqslr
Stop Channel	STOP CHANNEL	–

## Autres commandes de contrôle

Tableau 27. Autres commandes de contrôle

Description	Commande de contrôle
Création de l'exit de conversion MQSeries	crtmqcvx
Affichage des droits d'accès	dspmqaout
Affichage des fichiers utilisés par les objets	dspmqlfs
Affichage d'une sortie de trace MQSeries formatée.	dspmqrtrc
Arrêt de la trace MQSeries	endmqtrc
Gestion d'un groupe de reprise	failover
Enregistrement d'images de support	rcdmqimg
Recréation d'objet de support	rcrmqobj
Résolution des transactions MQSeries	rsvmqtrn
Exécution de commandes MQSC	runmqsc
Lancement du moniteur de déclenchement	runmqtrm
Lancement du moniteur de déclenchement client	runmqtrmc
Définition ou réinitialisation des droits d'accès	setmqaut
Démarrage d'un moniteur de mise hors ligne	runmqfm
Lancement de la trace MQSeries	strmqtrc
<b>Remarque :</b> Les fonctions de ce groupe ne sont disponibles qu'en tant que commandes de contrôle. Il n'existe pas de commandes de contrôle équivalentes directes dans ce groupe.	

## Comparaison des jeux de commandes

---

## Annexe E. Modèles de programmes MQI et de fichiers MQSC

MQSeries pour Compaq OpenVMS propose un ensemble de modèles de programmes MQI et de fichiers de commande MQSC que vous pouvez utiliser pour vous familiariser avec le produit. Ceux-ci sont décrits dans les sections suivantes :

- «Modèles de fichiers de commandes MQSC»
- «Modèles de programmes C et COBOL»
- «Outils divers» à la page 332

---

### Modèles de fichiers de commandes MQSC

Le tableau 28 indique les modèles de fichiers de commandes MQSC proposés. Il s'agit de simples fichiers texte ASCII contenant des commandes MQSC. Vous pouvez appliquer la commande **runmqsc** sur chaque fichier, de façon à créer les objets qui y sont indiqués. Consultez la section «Exécution des fichiers de commandes MQSC fournis» à la page 45.

Ces fichiers se situent par défaut dans le répertoire MQS\_EXAMPLES:

Tableau 28. Fichiers de commandes MQSC

Nom du fichier	Rôle
AMQSCOS0.TST	Crée un ensemble d'objets MQI à utiliser avec les modèles de programmes C et COBOL.

---

### Modèles de programmes C et COBOL

Le tableau 29 indique les modèles de fichiers source MQI. Ces fichiers se trouvent par défaut dans le répertoire MQS\_EXAMPLES: et les versions compilées dans le répertoire [.BIN] sous MQS\_EXAMPLES:. Pour plus d'informations sur le rôle et l'utilisation des programmes, reportez-vous au *Guide de programmation d'applications*.

Tableau 29. Modèles de programmes - fichiers source

C	COBOL	Rôle
AMQSBCG0.C	-	Lit puis génère une sortie pour les zones de descripteur et de contexte de messages, pour tous les messages d'une file d'attente précise.
AMQSECHA.C	AMQVECHX.COB	Répercute un message d'une file d'attente de messages à une file d'attente de réponses. Peut être lancé comme un programme d'application déclenché .
AMQSGBR0.C	AMQ0GBR0.COB	Inscrit les message d'une file d'attente dans SYS\$OUTPUT en laissant les messages dans la file. Utilise MQGET avec l'option de consultation.
AMQSGET0.C	AMQ0GET0.COB	Supprime les messages de la file d'attente nommée (à l'aide de MQGET) et les inscrit dans SYS\$OUTPUT.
AMQSINQA.C	AMQVINQX.COB	Procède à la lecture de la file d'attente déclenchée ; chaque requête a la forme d'un nom de file d'attente. Répond en fournissant les informations relatives à la file d'attente.
AMQSPUT0.C	AMQ0PUT0.COB	Copie SYS\$INPUT dans un message et met celui-ci sur une file d'attente spécifiée.

## Modèles

Tableau 29. Modèles de programmes - fichiers source (suite)

C	COBOL	Rôle
AMQSREQ0.C	AMQ0REQ0.COB	Place les messages de requête dans une file d'attente spécifiée, puis affiche les messages de réponse.
AMQSSETA.C	AMQVSETX.COB	Empêche l'enregistrement sur une file d'attente nommée et répond en indiquant le résultat. Peut être lancé comme une application déclenchée.
AMQSTRG0.C	–	Moniteur de déclenchement qui lit une file d'attente d'initialisation nommée puis démarre le programme associé à chaque message de déclenchement. Procure une partie des fonctions de déclenchement de la commande <b>runmqtrm</b> .
AMQSVFCX.C	–	Modèle de squelette C d'une routine d'exit de conversion de données.

## Outils divers

Ces fichiers outils sont destinés à prendre en charge le module de formatage et le conversion de code.

Tableau 30. Fichiers divers

Nom du fichier	Emplacement	Rôle
AMQTRC.FMT	SYS\$LIBRARY	Définit les formats de trace MQSeries.
CCSID.TBL	MQS_ROOT:[MQM.CONV.TABLE]	Editez ce fichier pour ajouter toute nouvelle valeur CSSID prise en charge pour votre système MQSeries. Pour plus d'informations sur les CCSID, consultez la documentation CDRA (Character Data Representation Architecture).



---

## Annexe F. Modèles de groupe de reprise de cluster OpenVMS

Cette annexe contient les modèles de groupes de reprise suivants :

- «Modèle de fichier de configuration FAILOVER.TEMPLATE»
- «Modèle de procédure StartCommand START\_QM.TEMPLATE» à la page 335
- «Modèle de procédure EndCommand END\_QM.TEMPLATE» à la page 336
- «Modèle de procédure TidyCommand TIDY\_QM.TEMPLATE» à la page 339

---

### Modèle de fichier de configuration FAILOVER.TEMPLATE

```
#####  
#*                                                                    *#  
#* Statement:      Licensed Materials - Property of IBM              *#  
#*                                                                    *#  
#*                33H2205, 5622-908                                  *#  
#*                33H2267, 5765-623                                  *#  
#*                29H0990, 5697-176                                  *#  
#*                (C) Copyright IBM Corp. 2000, 2001                *#  
#*                                                                    *#  
#####  
#  
# FAILOVER.TEMPLATE  
# Template for creating a FAILOVER.INI configuration file  
# All lines beginning with a '#' are treated as comments  
#  
# OpenVMS Cluster Failover Set Configuration information  
# -----  
#  
# The TCP/IP address used by the OpenVMS Cluster Failover Set  
#  
IpAddress=n.n.n.n  
#  
# The TCP/IP port number used by the MQSeries Queue Manager  
#  
PortNumber=1414  
#  
# The timeout used by the EndCommand command procedure  
#  
TimeOut=30  
#  
# The command procedure used to start the Queue Manager  
#  
StartCommand=@sys$manager:start_qm  
#  
# The command procedure used to end the Queue Manager  
#  
EndCommand=@sys$manager:end_qm  
#  
# The command procedure used to tidy up on a node after a  
# Queue Manager failure but the OpenVMS node did not fail  
#  
TidyCommand=@sys$manager:tidy_qm  
#  
# The directory in which the log files for the start, end and  
# tidy commands are written  
#  
LogDirectory=mqs_root:[mqm.errors]  
#  
# The number of nodes in the OpenVMS Cluster Failover Set. The  
# number of nodes defined below must agree with this number
```

## FAILOVER.TEMPLATE

```
#
NodeCount=2
#
# The Name of the OpenVMS node
#
NodeName=BATMAN
#
# The TCP/IP interface name for the node
#
Interface=we0
#
# The priority of the node
#
Priority=1
#
# The Name of the OpenVMS node
#
NodeName=ROBIN
#
# The TCP/IP interface name for the node
#
Interface=we0
#
# The priority of the node
#
Priority=2
```

*Figure 27. Fichier de configuration modèle : failover.template*

## Modèle de procédure StartCommand START\_QM.TEMPLATE

```

$ on error then exit
$!*****
$!* Statement:      Licensed Materials - Property of IBM      *
$!*                33H2205, 5622-908                          *
$!*                33H2267, 5765-623                          *
$!*                29H0990, 5697-176                          *
$!*                (C) Copyright IBM Corp. 2000, 2001        *
$!*****
$! Template command procedure used by Failover Sets to start the
$! queue manager
$! Parameters :
$! P1 = Queue Manager Name
$! P2 = Queue Manager Directory Name
$! P3 = TCP/IP address
$! P4 = TCP/IP interface name
$! P5 = Listener port number
$!
$ @sys$startup:mqs_symbols
$ set def mqs_root:[mqm.qmgrs.'p2'.errors]
$ define sys$scratch mqs_root:[mqm.qmgrs.'p2'.errors]
$!
$! Digital TCP/IP Services for OpenVMS commands
$!
$ @sys$startup:tcpip$define_commands
$!
$! Configure the IP address
$!
$ ifconfig 'p4' alias 'p3'
$!
$! TCPware for OpenVMS commands
$!
$! @tcpware:tcpware_commands
$!
$! Configure the IP address
$!
$! netcu add secondary 'p3'
$!
$! MultiNet for OpenVMS commands
$!
$! Configure the IP address
$!
$! define/sys/exec multinet_ip_cluster_aliases "'p3'"
$!
$! Restart the Multinet server
$!
$! @multinet:start_server
$! $! Start the queue manager
$!
$ strmqm 'p1'
$!
$! Start the listener
$!
$! runmqlsr -t tcp -p 'p5' -m 'p1'
$!
$! Insert commands to start any applications
$!
$exit

```

Figure 28. Modèle de procédure StartCommand : Start\_QM.template

## Modèle de procédure EndCommand END\_QM.TEMPLATE

```

$ on error then exit
$!
$!*****
$!*
$!* Statement:      Licensed Materials - Property of IBM      *
$!*
$!*              33H2205, 5622-908                          *
$!*              33H2267, 5765-623                          *
$!*              29H0990, 5697-176                          *
$!*              (C) Copyright IBM Corp. 2000, 2001         *
$!*
$!*****
$!
$! Template Command procedure used by Failover Sets to end the
$! queue manager
$!
$! Parameters :
$!
$! P1 = Queue Manager Name
$! P2 = Queue Manager Directory Name
$! P3 = TCP/IP address
$! P4 = TCP/IP interface name
$! P5 = Listener port number
$! P6 = End Queue Manager Timeout
$!
$ @sys$startup:mqs_symbols
$ check_qm==$sys$system:mqcheckqm
$ set def mqs_root:[mqm.qmgrs.'p2'.errors]
$ define sys$scratch mqs_root:[mqm.qmgrs.'p2'.errors]
$ SS$ _NORMAL=1
$ SS$ _ABORT=44
$ SS$ _TIMEOUT=556
$!
$! Insert commands to shutdown any applications prior to ending MQSeries
$!
$! Get the timeout period for each operation seconds
$!
$ timeout = 'p6'
$!
$! Initialise the outer loop
$!
$ out_count = 0
$!
$! Initialise the complete flag
$!
$ complete = 0
$!
$ out_next:
$ if (out_count .gt. 2) .or. (complete .eq. 1) then goto out_finish
$!
$ if out_count .eq. 0
$ then
$!
$! End the queue manager gracefully first
$!
$ spawn/nowait $endmqm -i 'p1'
$ else
$ if out_count .eq. 1
$ then
$!
$! End the queue manager abruptly
$!
$ spawn/nowait $endmqm -p 'p1'
$ else
$!

```

```

$! Stop/id the execution controller
$!
$ check_qm -m 'p1'
$ if ( mqs$ec_pid .nes. "" ) then $stop/id='mqs$ec_pid'
$ endif
$ endif
$!
$ in_start:
$!
$! Initialise the outer loop
$!
$ in_count = 0
$!
$ in_next:
$!
$! Inner loop
$!
$ if ( ( in_count .ge. timeout ) .and. ( timeout .ne. 0 ) ) -
    .or. ( complete .eq. 1 ) then goto in_finish
$!
$! Check if the execution controller is still running
$!
$ check_qm -m 'p1'
$ if mqs$ec_pid .eqs. ""
$ then
$!
$! The Execution controller is no longer running so we are finished
$!
$ complete = 1
$ goto in_finish
$ endif
$!
$! Wait a second and go round again
$!
$ wait 00:00:01
$ in_count = in_count + 1
$ goto in_next
$ in_finish:
$!
$! End of the inner loop
$!
$ out_count = out_count + 1
$ goto out_next
$ out_finish:
$!
$! End of the outer loop
$!
$! Digital TCP/IP Services for OpenVMS commands
$!
$ @sys$startup:tcpip$define_commands
$!
$! De-configure the IP address
$!
$ ifconfig 'p4' -alias 'p3'
$!
$! TCPware for OpenVMS commands
$!
$! @tcpware:tcpware_commands
$!
$! De-configure the IP address
$!
$! netcu remove secondary 'p3'
$!
$! MultiNet for OpenVMS commands
$!
$! De-configure the IP address
$!

```

## END\_QM.TEMPLATE

```
#!/ deass/sys/exec multinet_ip_cluster_aliases
#!/
#!/ Restart the Multinet server
#!/
#!/ @multinet:start_server
#!/
#!/
#!/ If the Queue Manager was shutdown successfully set the status
#!/ to SS$ _NORMAL. If it was necessary to STOP/ID the Execution
#!/ controller set the status to SS$ _ABORT and if the Execution
#!/ controller is still running set the status to SS$ _TIMEOUT to
#!/ indicate an error
#!/
$ if ( complete .eq. 1 )
$ then
$!
$! End the listener process
$!
$! endmq1sr -m 'p1'
$!
$ if ( out_count .eq. 3 )
$ then
$ exit SS$ _ABORT
$ else
$ exit SS$ _NORMAL
$ endif
$ else
$ exit SS$ _TIMEOUT
$ endif
```

Figure 29. Modèle de procédure EndCommand : END\_QM.template

## Modèle de procédure TidyCommand TIDY\_QM.TEMPLATE

```

$ on error then exit
$!*****
$!* Statement:      Licensed Materials - Property of IBM      *
$!*
$!*                33H2205, 5622-908                          *
$!*                33H2267, 5765-623                          *
$!*                29H0990, 5697-176                          *
$!*                (C) Copyright IBM Corp. 2000, 2001        *
$!*****
$! Template Command procedure used by Failover Sets to tidy up after
$! a queue manager failure
$!
$! Parameters :
$! P1 = Queue Manager Name
$! P2 = Queue Manager Directory Name
$! P3 = TCP/IP address
$! P4 = TCP/IP interface name
$! P5 = Listener port number
$!
$ @sys$startup:mqs_symbols
$ set def mqs_root:[mqm.qmgrs.'p2'.errors]
$ define sys$scratch mqs_root:[mqm.qmgrs.'p2'.errors]
$!
$! Insert commands to do any tidying up after a queue manager has failed
$!
$! Digital TCP/IP Services for OpenVMS commands
$!
$ @sys$startup:tcpip$define_commands
$!
$! De-configure the IP address
$!
$ ifconfig 'p4' -alias 'p3'
$!
$! TCPware for OpenVMS commands
$!
$! @tcpware:tcpware_commands
$!
$! De-configure the IP address
$!
$! netcu remove secondary 'p3'
$!
$! MultiNet for OpenVMS commands
$!
$! De-configure the IP address
$!
$! deass/sys/exec multinet_ip_cluster_aliases
$!
$! Restart the Multinet server
$!
$! @multinet:start_server
$!
$exit

```

Figure 30. Modèle de procédure TidyCommand : TIDY\_QM.template





## Annexe G. Jeux de codes acceptés par MQSeries pour Compaq OpenVMS

MQSeries pour Compaq OpenVMS accepte la plupart des jeux de codes utilisés par les paramètres nationaux (c'est à dire les sous ensembles de l'environnement utilisateur qui définissent les conventions propres à une culture) et qui sont intégrés en standard sur MQSeries pour Compaq OpenVMS.

Si aucune variable de valeur d'environnement n'est définie, le CCSID utilisé est 819 - le jeu de codes ISO8859-1.

L'identificateur de jeu de caractères codés (CCSID - Coded Character Set Identifier) utilisé dans MQSeries pour identifier le jeu de codes utilisé dans les messages et en-têtes de messages est obtenu en analysant la catégorie LC\_CTYPE de la configuration locale.

Le tableau 31 indique les paramètres nationaux et CCSID enregistrés pour le jeu de codes utilisé dans l'environnement local.

Tableau 31. Paramètres nationaux et CCSID

Paramètres nationaux	Langue	Jeu de codes	CCSID
C	Anglais	ISO8859-1	819
CS_CZ_ISO8859-2	Tchèque	ISO8859-2	912
DA_DK_ISO8859-1	Danois	ISO8859-1	819
DE_DE_ISO8859-1	Allemand	ISO8859-1	819
DE_CH_ISO8859-1	Allemand - Suisse	ISO8859-1	819
EL_GR_ISO8859-7	Grec	ISO8859-7	813
EN_GB_ISO8859-1	Anglais - Royaume Uni	ISO8859-1	819
EN_US_ISO8859-1	Anglais - Etats Unis	ISO8859-1	819
ES_ES_ISO8859-1	Espagnol	ISO8859-1	819
FI_FI_ISO8859-1	Finois	ISO8859-1	819
FR_FR_ISO8859-1	Français - France	ISO8859-1	819
FR_BE_ISO8859-1	Français - Belgique	ISO8859-1	819
FR_CA_ISO8859-1	Français - Canada	ISO8859-1	819
FR_CH_ISO8859-1	Français - Suisse	ISO8859-1	819
HU_HU_ISO8859-2	Hongrois	ISO8859-2	912
IS_IS_ISO8859-1	Islandais	ISO8859-1	819
IT_IT_ISO8859-1	Italien - Italie	ISO8859-1	819
IW_IL_ISO8859-8	Hébreu	ISO8859-8	916
JA_JP_EUCJP	Japonais	eucJP	954
JA_JP_SDECKANJI	Japonais	SDECKANJI	954**
JA_JP_SJIS	Japonais	SJIS	932
KO_KR_DECKOREAN	Coréen	DECKOREAN	970**

## Jeux de codes acceptés

Tableau 31. Paramètres nationaux et CCSID (suite)

Paramètres nationaux	Langue	Jeu de codes	CCSID
NL_NL_ISO8859-1	Néerlandais - Pays Bas	ISO8859-1	819
NL_BE_ISO8859-1	Néerlandais - Belgique	ISO8859-1	819
NO_NO_ISO8859-1	Norvège	ISO8859-1	819
PL_PL_ISO8859-2	Polonais	ISO8859-2	912
PT_PT_ISO8859-1	Portugais	ISO8859-1	819
SK_SK_ISO8859-2	Slovaque	ISO8859-2	912
RU_RU_ISO8859-5	Cyrillique	ISO8859-5	915
SV_SE_ISO8859-1	Suédois	ISO8859-1	819
TR_TR_ISO8859-9	Turc	ISO8859-9	920
ZH_CN_DECHANZI	Chinois - Simplifié	DECHANZI	1383**
ZH_HK_DECHANZI	Chinois - Simplifié	DECHANZI	1383**
ZH_HK_EUCTW	Chinois - Traditionnel	eucTW	964
ZH_HK_EUCTW	Chinois - Traditionnel	eucTW	964
ZH_HK_DECHANYU	Chinois - Traditionnel	DECHANYU	964**
ZH_TW_DECHANYU	Chinois - Traditionnel	DECHANYU	964**
ZH_HK_BIG5	Chinois - Traditionnel	big5	950
ZH_TW_BIG5	Chinois - Traditionnel	big5	950
<b>Remarque :</b>			
** L'identificateur de jeu de caractères codés utilisé est le CCSID IBM enregistré le plus proche.			

Pour plus d'informations sur la prise en charge inter-plateforme de ces paramètres locaux, consultez le manuel *MQSeries Application Programming Reference*.

---

## Annexe H. Utilitaire de diagnostic MONMQ

L'utilitaire MONMQ est un outil d'aide au diagnostic et à l'identification des incidents rencontrés avec MQSeries pour Compaq OpenVMS. Il peut être utilisé de façon interactive, depuis une ligne de commande ou un script DCL.

En général, l'utilitaire MONMQ est utilisé pour :

- Gérer la mémoire partagée,
- Rassembler des informations sur l'utilisation des ressources OpenVMS,
- Générer une sortie de trace depuis un gestionnaire de files d'attente en cours d'exécution.

MONMQ dispose d'une aide à l'utilisation des paramètres et peut exécuter un script de commandes MONMQ. Lors du démarrage de MONMQ, un script par défaut `sys$manager:mqs_trace_startup.mqt` est exécuté pour définir la configuration initiale.

```
$monmq
ok - trace mailbox 0 opened as default

MQT> help
Help can be used to display information about available commands or parameters
Help [ <verb> || <parameter/variable name> || commands || parameters || examples]

Valid trace commands are in the format:
Verb [<parameters>] [<variable = expression>][;][optional second command]
```

---

### Généralités

La fonction de trace de MQSeries sous OpenVMS utilise des sections globales et des boîtes aux lettres. Jusqu'à dix sections de trace (LU) peuvent cohabiter sur n'importe quel noeud sur lequel est installé MQSeries. Cependant, il est fortement recommandé de n'utiliser qu'une seule LU à la fois par session de trace. Il n'est pas non plus conseillé d'avoir la même LU ouverte plusieurs fois simultanément. Ces deux actions pourraient avoir des résultats imprévisibles.

Chaque section partagée (LU) contient les définitions de canaux et la définition de la LU. Chaque définition de canal contient les détails de l'unité d'exécution connectée ainsi que la pile privée et la mémoire tampon circulaire des unités d'exécution. De plus, la section partagée contient un ensemble d'indicateurs utilisés pour les communications interprocessus entre MONMQ et les unités d'exécution connectées.

A chaque LU est associée une boîte aux lettres, destinée à recevoir les messages de trace en temps réel. Pour effectuer un traçage en temps réel, un processus client doit être lancé à l'aide de la commande **TRACE START**. Ce processus détaché lit, formate et affiche chaque message dès son arrivée dans la boîte aux lettres de la LU. Chaque unité d'exécution connectée écrit à la même boîte aux lettres, ce qui permet de voir physiquement l'intercommunication entre les processus/unités d'exécution MQSeries.

L'utilisation correcte de MONMQ offre une méthode complète de diagnostic des incidents liés aux délais d'exécution interprocessus, à l'insuffisance des ressources du système d'exploitation ou à des problèmes de code.

## Variables utilisées dans MONMQ

Les commandes MONMQ sont décrites dans cette annexe.

---

## Variables utilisées dans MONMQ

De nombreuses commandes de MONMQ utilisent des variables. Une variable utilise une valeur par défaut, définie par la commande **set** si aucune autre n'est précisée dans la commande. Lorsqu'une variable est utilisée avec une autre commande que la commande **set**, la valeur par défaut de cette variable n'est pas modifiée.

Les variables peuvent contenir :

- Des nombres entiers (décimaux ou hexadécimaux).  
Les valeurs hexadécimales peuvent être saisies en les commençant par 0x, ou bien en tapant une valeur contenant les lettres de a à f, si un nombre hexadécimal est attendu.
- Du texte, saisi entre deux caractères " .
- Un intervalle, représenté par la notation minimum:maximum.  
Un intervalle permet par exemple d'appliquer une commande à une série de canaux.

Par exemple :

```
MQT> set lu=2
MQT> set pid=0x223
MQT> set pid=2fa
MQT> set buffile="filename.buf"
MQT> set chl=0:20
```

Vous pouvez afficher la valeur par défaut actuelle des variables en utilisant la commande **variables**.

```
MQT> variables
defined variables
lu=0:0
chl=0:20
mask=0 (HEX)
function=0 (HEX)
resource=0
listfile=(null)
active=0 (BOOL)
post=0 (BOOL)
nochls=20
component=0 (HEX)
pid=0 (HEX)
div=0
wait=1 (BOOL)
buffile=(null)
fname=(null)
buffer=1000
line=0
node=(null)
depth=32
timestamp=0 (BOOL)
step=0 (BOOL)
delay=100

defined constants
fent=1 (HEX)
fxxx=8 (HEX)
spl=40 (HEX)
prc=200 (HEX)
log=2000 (HEX)
mutex=4
microseconds=2
fout=2 (HEX)
dgn=10 (HEX)
evt=80 (HEX)
msc=400 (HEX)
shl=4000 (HEX)
mailbox=5
milliseconds=3
ferr=4 (HEX)
shm=20 (HEX)
mtx=100 (HEX)
inf=800 (HEX)
memory=3
nanoseconds=1
seconds=4
```

## Variables utilisées dans MONMQ

Nous vous recommandons de définir des valeurs par défaut pour simplifier les commandes. Par exemple, les séquences de commandes suivantes sont fonctionnellement identiques :

```
MQT> open lu=0 buffer=1000 nochls=20
MQT> open lu=1 buffer=1000 nochls=20
MQT> show channels lu=0 chl=1:10
MQT> show channels lu=1 chl=1:10
```

ou

```
MQT> set nochls=20 chl=0:10 buffer=1000 lu=0
MQT> open
MQT> open lu=1
MQT> show channels
MQT> show channels lu=1
```

La frappe des commandes MONMQ peut être réduite au nombre minimal de caractères nécessaires à les différencier. Cette série de commandes pourrait être réduite à :

```
MQT> se noc=20 ch=0:10 buffe=1000 lu=0
MQT> op
MQT> op lu=1
MQT> sh ch
MQT> sh ch lu=1
```

MONMQ peut effectuer des calculs arithmétiques sur les variables afin que des commandes telles que  $lu=lu+1$  soient possibles.

De nouvelles variables peuvent être déclarées à l'aide de la commande **declare**. Les paramètres sont les suivants :

- Nom de la variable
- Type de la variable
- Texte d'aide. La commande help peut alors afficher le texte d'aide.

```
MQT> declare ec int "channel number for execution controller"
MQT> set ec = 4
MQT> show channel chl=ec
Chl  Pid  Mailbox  Stack  Active  Post  Time  Mask  Process Name
 4   2c1f  7ee70290  4      0      0    0   ffffffff  AMQZXA00.EXE
MQT> help ec

VARIABLE ec:
channel number for execution controller

MQT>
```

## Variables utilisées dans MONMQ

### Attribution des valeurs par défaut

DEFAULT variable=<expression> [variable=expression] .

Cette commande permet d'attribuer des valeurs par défaut à toutes les variables définies dans MONMQ. Une fois défini, le nom de la variable par défaut peut être omis de la ligne de commande. Par exemple :

```
MQT>default lu=2 ch1=3:6
```

Cette commande attribue la valeur par défaut 2 à la variable lu, et les valeurs 3 à 6 inclus à la variable de canal. Désormais, lorsque des commandes comme **show channels** seront utilisées, les canaux 3 à 6 inclus de la lu 2 seront affichés.

Les valeurs par défaut sont utilisées uniquement lorsque la variable est omise dans la ligne de commande. Toutes les valeurs par défaut sont définies dans le fichier de script de démarrage MQS\_TRACE\_STARTUP.MQT. Vous pouvez modifier ce fichier en fonction de vos besoins.

---

## Ouverture et création d'une section de trace et de sa boîte aux lettres

OPEN [lu=number] [nochls=number] [buffer=number]

Cette commande permet d'ouvrir et de créer une section de trace et sa boîte aux lettres. La commande **open** crée la ressource de base nécessaire au traçage des processus MQSeries. Une section partagée et une boîte aux lettres sont associées à chaque LU pour communiquer avec les processus MQSeries.

Cette commande comprend trois paramètres optionnels. Le premier paramètre [LU] est le numéro attribué à la section de trace/boîte aux lettres. Il sert de référence pour la plupart des autres commandes MONMQ. Deux LU maximum peuvent être créées sur un seul noeud. La valeur par défaut est zéro. Si la LU indiquée existe déjà, MONMQ se connecte à la section de trace existante. Si aucune section n'existe, une nouvelle est créée.

Le second paramètre [nochls] indique le nombre de canaux attribués à cette LU. Chaque canal représente une seule connexion de processus/unité d'exécution MQSeries. La valeur par défaut est 20.

Le troisième paramètre [buffer] indique la taille maximale de la mémoire tampon de l'historique de trace pour chaque canal. La valeur par défaut est 1000.

Avant de pouvoir appliquer les autres commandes MONMQ, il faut qu'au moins une LU soit ouverte.

---

## Affichage de la définition de l'unité logique

SHOW SEGMENT [lu=range]

Cette commande affiche la définition de l'unité logique (LU). Le texte ci-dessous est un exemple de sortie obtenue par la commande **show segment lu=0**. Chaque zone est accompagnée d'une brève description.

```

Trace LU          : 0          /* The LU number as specified in the OPEN [lu] parameter.
Mailbox name     : MQS_TRC_MBX_0 /* The permanent mailbox name assigned to this LU
Device name      : MBA1065:    /* The device name of the mailbox
Status           : Disabled    /* The current status of the mailbox ie.
Mailbox channel  : 352        /* The mailbox channel number assigned to the MONMQ process
History buffer size: 1000     /* The maximum number of message entries in the history
/* circular buffer (as specified by the OPEN [buffer] parameter)
Threads mapped # : 1          /* The number of processes/threads mapped to this LUs global
/* section (MONMQ always attached)
Time stamping    : Enabled    /* Global timestamp flag (not yet implemented)
Max channels #   : 20        /* Number of channels defined for this LU as specified by the
/* OPEN [nochls] parameter.
Display depth    : 0          /* The stack display depth. Default (0) is to display all stack entries.
Text filename    :           /* The client text trace file
Binary filename  :           /* The client binary trace file
Last status      : 1          /* Last status of mailbox Qio activity (useful if VMS low on
/* resources and MONMQ fails)
Connection map[0] : 0        /* A bit map of all connected channels (maximum no. of channels is 128)

```

---

## Fermeture et suppression d'une LU

CLOSE [lu=*number*]

Cette commande a l'effet inverse de la commande **OPEN** : elle ferme et supprime la LU indiquée. La fermeture de la LU est parfaitement contrôlée : chaque processus connecté est averti d'avoir à se déconnecter, chaque canal est réinitialisé, la boîte aux lettres de trace est désattribuée, et enfin la section partagée est effacée. Cette commande ne doit être exécutée que lorsqu'une session de trace est terminée.

---

## Affichage des informations sur les canaux

SHOW CHANNELS [full] [connected] [chl=*range*]

Cette commande affiche les informations sur les canaux indiqués. Le paramètre [connected] affichera uniquement les canaux auxquels des unités d'exécution sont connectées. Par exemple, la commande show channels connected affiche les informations suivantes :

Chl	Pid/Tid	Mailbox	Stack	History	RTime	Time	Mask	Process Name
0	00000245/1	800b0330	4	0	0	0	fffffff	AMQZLAA0.EXE
1	00000244/1	800b0200	8	0	0	0	fffffff	RUNMQCHI.EXE
2	00000243/1	800b01e0	10	0	0	0	fffffff	AMQRRMFA.EXE
3	00000242/1	800b01c0	4	0	0	0	fffffff	AMQZLLP0.EXE
4	00000241/1	800b0220	5	0	0	0	fffffff	AMQHASM0.EXE
5	00000240/1	800b03f0	4	0	0	0	fffffff	AMQZXMA0.EXE

## SHOW CHANNELS

Le paramètre [full] affiche la définition complète des canaux indiqués. Par exemple, la commande **show channels full connected chl=0:3** affiche les informations suivantes :

```
Pid/Tid      : 0000024b/1      /* Connected threads process id and thread sequence number
Status       : *** Connected *** /* Current status of channel (thread is connected)
Process name  : AMQRRMFA.EXE   /* Process name of connected thread
Assigned LU   : 0              /* This channels associated LU
Channel no.   : 2              /* Allocated channel number within the LU
Mailbox channel : 800b01e0     /* Connected threads mailbox channel number for the trace mailbox
Current stack depth : 10      /* Threads current stack depth
Circular logging : Disabled    /* History enabled flag
Next log entry : 0            /* Next history buffer slot number
Realtime tracing : Disabled    /* Real time enable flag (needs client to read messages)
Time stamping : Disabled      /* Enables timestamping for this threads messages
Trace mask    : ffffffff      /* Hexadecimal format of trace mask for this thread (see show mask command)
Step mode     : Off           /* Not yet implemented
No Wait      : 0n            /* Forces threads qio activity to wait for a resource if not available
Last QIO status : 0         /* Threads last qio call status
Mapped address : 9a2000-c9ffff /* The virtual mapped address range of the LU global section for this thread
```

---

## Affichage du masque de trace actuel d'un canal

SHOW MASK [chl=range]

Cette commande permet d'afficher le masque de trace actuel d'un canal. Les lignes en caractères gras indiquent que le bit de masque btrace. Par exemple, la commande **show mask chl=1** affiche les informations suivantes :

Trace Mask for Channel 1

<b>Bit 00 - (fent) function entry</b>	<b>Function entry messages</b>
<b>Bit 01 - (fout) function exit</b>	<b>Function exit messages</b>
Bit 02 - (ferr) function exit with error	Function exit with error return status
Bit 03 - (fxx) missing function exit	Unbalanced function entry/exit message (see note below)
Bit 04 - (dgn) diagnostic messages	Diagnostic messages
Bit 05 - (shm) shared memory	OVMS shared memory messages
<b>Bit 06 - (spl) spinlocks</b>	<b>OVMS spinlock messages</b>
<b>Bit 07 - (evt) events</b>	<b>OVMS event messages</b>
Bit 08 - (mtx) mutexes	OVMS mutex messages
Bit 09 - (prc) process msgs	OVMS thread messages
Bit 10 - (msc) miscellaneous	OVMS kernal miscellaneous messages
Bit 11 - (inf) informational	Internal data messages as requested by show command
Bit 12 -	Reserved for user defined messages

Cet exemple montre que, pour cette unité d'exécution, seuls les messages function entry, function exit, spinlocks et event feront l'objet d'un traçage. Tous les autres types de messages sont bloqués.

---

## Affichage du contenu des piles d'unités d'exécution cibles

SHOW STACK [chl=range]

Cette commande permet d'afficher le contenu de la pile d'unités d'exécution cibles. Par exemple, la commande **show stack chl=0:1** affiche les informations suivantes :

```
0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 12:36:20.18 03 - 02 ---->| zcpReceiveOnLink
0003 - 12:36:20.81 03 - 03 ---->| xcsWaitEventSem
0004 - 12:36:20.83 03 - 04 ---->| vms_evt

0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 12:36:20.18 03 - 02 ---->| zcpSendOnLink
0003 - 12:36:20.81 03 - 03 ---->| xcsPostEventSem
0004 - 12:36:20.83 03 - 04 ---->| vms_evt
```



## Affichage des processus MQSeries actifs et de l'utilisation de la mémoire

### SHOW PROCESSES

Cette commande affiche tous les processus MQSeries actifs du noeud ainsi que l'utilisation de la mémoire. Par exemple, la commande **show process** affiche les informations suivantes :

PID	Proc_Name	Image	Process	WS_Size	WS_Peak	Virt_Peak	Gbl_Pg_Cnt	Prc_Pg_Cnt	Total_Mem
0000023D	BKM3_AG	AMQZLAA0	Agent	23152	16576	203776	3616	12960	16576
0000023C	BKM3_CI	RUNMQCHI	Run Chan Init	8752	6208	180832	1840	4368	6208
0000023B	BKM3_RM	AMQRRMFA	Repository Mgr	11152	8144	185360	2224	5920	8144
0000023A	BKM3_CP	AMQZLLP0	Checkpoint	8752	6384	185952	1920	4464	6384
00000239	BKM3_LG	AMQHASM	Logger	8752	6288	182016	2080	4208	6288
00000238	BKM3_EC	AMQZXMA0	EC	20752	15232	203792	3536	11680	15216
00000128	_FTA4:	MONMQ	MONMQ Utility	8400	8528	198736	2224	3584	5808

## Affichage de tous les messages d'un canal

### SHOW HISTORY [chl=range]

Cette commande affiche tous les messages contenus dans le tampon circulaire d'historique d'un canal. Chaque message est formaté, et mis en retrait sur la sortie à l'écran en fonction de sa position dans la pile. Par exemple, la commande **show history chl=3** affiche les informations suivantes :

```

0215 - 12:35:44.52 03 - 02 ---<| zxcProcessChildren
0216 - 12:35:44.55 03 - 02 --->| zxcStartWLMserver
0217 - 12:35:44.57 03 - 02 ---<| zxcStartWLMserver
0218 - 12:35:44.59 03 - 02 --->| zcpReceiveOnLink
0219 - 12:35:44.61 03 - 03 ----->| xcsRequestMutexSem
0220 - 12:35:44.63 03 - 04 ----->| xllSemReq
0221 - 12:35:44.66 03 - 05 ----->| vms_mtx
0222 - 12:35:44.66 03 - 05 .....>| vms_mtx :- Locking BKM3/@ipcc_m_1_10 - timeout: -1
0223 - 12:35:44.70 03 - 06 ----->| vms_get_lock
0224 - 12:35:44.72 03 - 06 -----<| vms_get_lock
0225 - 12:35:44.74 03 - 05 -----<| vms_mtx
0226 - 12:35:44.76 03 - 04 -----<| xllSemReq
0227 - 12:35:44.79 03 - 03 -----<| xcsRequestMutexSem
0228 - 12:35:44.81 03 - 03 ----->| xcsResetEventSem
0229 - 12:35:44.83 03 - 04 ----->| vms_evt
0230 - 12:35:44.83 03 - 04 .....>| vms_evt Reset on mailbox BKM3/@ipcc_e_1_2 : tout = -1
0231 - 12:35:44.87 03 - 05 ----->| vms_get_mbx_chan
0232 - 12:35:44.87 03 - 05 .....>| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0233 - 12:35:44.87 03 - 05 .....>| vms_get_mbx_chan Returning key 1a0
0234 - 12:35:44.94 03 - 05 -----<| vms_get_mbx_chan
0235 - 12:35:44.83 03 - 04 .....>| vms_evt rc = 0
0236 - 12:35:44.98 03 - 04 -----<| vms_evt
0237 - 12:35:45.00 03 - 03 -----<| xcsResetEventSem
0238 - 12:35:45.03 03 - 03 ----->| xcsReleaseMutexSem
0239 - 12:35:45.05 03 - 04 ----->| xllSemRel
0240 - 12:35:45.07 03 - 05 ----->| vms_mtx
0241 - 12:35:45.07 03 - 05 .....>| vms_mtx :- Unlocking BKM3/@ipcc_m_1_10 - timeout: -1
0242 - 12:35:45.11 03 - 06 ----->| vms_get_lock
0243 - 12:35:45.13 03 - 06 -----<| vms_get_lock
0244 - 12:35:45.16 03 - 05 -----<| vms_mtx

```

Cet exemple de sortie indique le numéro de ligne dans le tampon d'historique, l'heure à laquelle le message a été généré, le numéro du canal, la position du message dans la pile et le nom de la fonction. Le nombre maximum de messages d'historique a été défini au moment de l'ouverture de la LU. Lorsque ce nombre est atteint, MONMQ retourne au premier message et l'écrase, et ainsi de suite. Si un FFST est généré en cours du traçage, celui-ci est désactivé à partir de ce point, pour l'unité d'exécution défectueuse. Ceci permet d'éviter d'encombrer le tampon

## SHOW HISTORY

avec des messages de trace générés par des routines d'erreur. Le dernier message affiché dans le tampon d'historique est donc le point auquel le FFST a été généré.

---

## Affichage de toutes les sections globales MQSeries d'un noeud

SHOW GLOBALS

Cette section affiche toutes les sections globales MQSeries du noeud actuel.

MQS1_shm_00000000 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=6128/383
MQS1_shm_01300010 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=1904/119
MQS1_shm_012c000f (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=464/87
MQS1_shm_012c000e (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=4112/514
MQS1_shm_012c000d (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=240/30
MQS1_shm_012c000c (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=528/132
MQS1_shm_012c000b (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=272/51
MQS1_shm_012c000a (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=4112/771
MQS1_shm_012c0009 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=272/51
MQS1_shm_012c0008 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=144/27
MQS1_shm_012c0007 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=528/99
MQS1_shm_012c0006 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=16/2
MQS1_shm_012c0005 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=128/24
MQS1_shm_012c0004 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=1968/492
MQS1_shm_012c0003 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=1904/476
MQS1_shm_012c0002 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=304/76
MQS1_shm_012c0001 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=1904/595
MQS1_shm_01280000 (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=16/6
MQS1_shm_fffffffe (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=16/0
MQS1_shm_fffffff (00000000)	WRT	DZRO PRM SYS	Pgltcnt/Refcnt=144/63

---

## Ordre d'envoi de la table mutex au processus de traçage par l'unité d'exécution cible

SHOW MUTEX [chl=range]

Cette commande indique à l'unité d'exécution cible d'envoyer le contenu de sa table mutex au processus de traçage client. Il est essentiel que les bits de masque de trace adéquats soient définis, pour que le client puisse afficher ce type de données. Les bits INF et DGN doivent être activés dans le masque de trace de ce canal (reportez-vous à la section «Activation ou désactivation d'un bit de masque» à la page 359). Par exemple, la commande **show mutex chl=2** affiche les informations suivantes :

Mutex Utilisation for Process AMQZXMA0.EXE - Pid 248 \*\*\*

```

0960 - Lock ID: 0100013c - Name: BKM3/@ipcc_m_1_24
0961 - Lock ID: 020006ca - Name: BKM3/@ipcc_m_1_23
0962 - Lock ID: 0b00061e - Name: BKM3/@ipcc_m_1_22
0963 - Lock ID: 0900068e - Name: BKM3/@ipcc_m_1_21
0964 - Lock ID: 0b00032f - Name: BKM3/@ipcc_m_1_20
0965 - Lock ID: 210006eb - Name: BKM3/@ipcc_m_1_19
0966 - Lock ID: 07000742 - Name: BKM3/@ipcc_m_1_18
0967 - Lock ID: 1e000075 - Name: BKM3/@ipcc_m_1_17
0968 - Lock ID: 0c0004dd - Name: BKM3_m_1_45
0969 - Lock ID: 0d00035a - Name: BKM3/@ipcc_m_1_16
0970 - Lock ID: 190000a1 - Name: BKM3/@ipcc_m_1_15
0971 - Lock ID: 1a0005a3 - Name: BKM3/@ipcc_m_1_14
0972 - Lock ID: 14000628 - Name: BKM3/@ipcc_m_1_13
0973 - Lock ID: 130005f3 - Name: BKM3/@ipcc_m_1_12
0974 - Lock ID: 0f0000dc - Name: BKM3_m_1_43
0975 - Lock ID: 02000095 - Name: BKM3_m_1_42
0976 - Lock ID: 2200053e - Name: BKM3_m_1_41
0977 - Lock ID: 020000fc - Name: BKM3_m_1_40
0978 - Lock ID: 31000113 - Name: BKM3_m_1_39
0979 - Lock ID: 02000555 - Name: BKM3_m_1_38
0980 - Lock ID: 2e000389 - Name: BKM3_m_1_37
0981 - Lock ID: 2300011f - Name: BKM3_m_1_36
0982 - Lock ID: 02000109 - Name: BKM3_m_1_35
0983 - Lock ID: 02000327 - Name: BKM3_m_1_34
0984 - Lock ID: 020004a8 - Name: BKM3_m_1_33
0985 - Lock ID: 02000453 - Name: BKM3_m_1_32
0986 - Lock ID: 260007ad - Name: BKM3_m_1_31
0987 - Lock ID: 0200060c - Name: BKM3_m_1_30

```

Les informations affichées sont le numéro de ligne du fichier d'historique, le nom mutex et l'identificateur de verrouillage du système.

---

## Ordre d'envoi de la table interne des événements au processus de traçage par l'unité d'exécution cible

SHOW EVENTS [chl=*range*]

Cette commande indique à l'unité d'exécution cible d'envoyer le contenu de sa table d'événements interne au processus de traçage client. Il est essentiel que les bits de masque de trace adéquats soient définis, pour que le client puisse afficher ce type de données. Les bits INF et DGN doivent être activés dans le masque de trace de ce canal (reportez-vous à la section «Activation ou désactivation d'un bit de masque» à la page 359). Par exemple, la commande **show events chl=2** affiche les informations suivantes :

## SHOW EVENTS

Event Utilisation for Process AMQZXMA0.EXE - Pid 248 \*\*\*

```
1037 - Channel: 000003e0 - Name: BKM3/@ipcc_e_1_19
1038 - Channel: 000003d0 - Name: BKM3/@ipcc_e_1_18
1039 - Channel: 000003c0 - Name: BKM3/@ipcc_e_1_17
1040 - Channel: 000003b0 - Name: BKM3/@ipcc_e_1_14
1041 - Channel: 000003a0 - Name: BKM3/@ipcc_e_1_13
1042 - Channel: 00000390 - Name: BKM3/@ipcc_e_1_12
1043 - Channel: 00000380 - Name: BKM3/@ipcc_e_1_10
1044 - Channel: 00000370 - Name: BKM3/@ipcc_e_1_9
1045 - Channel: 00000360 - Name: BKM3/@ipcc_e_1_8
1046 - Channel: 00000330 - Name: BKM3/@ipcc_e_1_7
1047 - Channel: 00000300 - Name: BKM3/@ipcc_e_1_6
1048 - Channel: 000002f0 - Name: BKM3/@ipcc_e_1_5
1049 - Channel: 000002b0 - Name: BKM3_e_1_11
1050 - Channel: 000002a0 - Name: BKM3_e_1_10
1051 - Channel: 00000290 - Name: BKM3_e_1_9
1052 - Channel: 00000280 - Name: BKM3_e_1_8
1053 - Channel: 00000270 - Name: BKM3_e_1_7
1054 - Channel: 00000260 - Name: BKM3_e_1_6
1055 - Channel: 00000250 - Name: BKM3_e_1_5
1056 - Channel: 00000240 - Name: BKM3_e_1_4
1057 - Channel: 00000230 - Name: BKM3_e_1_3
1058 - Channel: 00000220 - Name: BKM3_e_1_2
1059 - Channel: 00000210 - Name: BKM3_e_1_1
1060 - Channel: 00000200 - Name: BKM3_e_1_0
1061 - Channel: 000001c0 - Name: BKM37/@ipcc_e_1_4
1062 - Channel: 000001b0 - Name: BKM3/@ipcc_e_1_3
1063 - Channel: 000001a0 - Name: BKM3/@ipcc_e_1_2
1064 - Channel: 00000190 - Name: BKM3/@ipcc_e_1_1
1065 - Channel: 00000180 - Name: BKM3/@ipcc_e_1_0
1066 - *** End of data ***
```

Les informations affichées sont le numéro de ligne du fichier d'historique, le numéro de canal de la boîte aux lettres et le nom de l'événement.

---

## Ordre d'envoi de la table interne de mémoire partagée mappée au processus de traçage par l'unité d'exécution cible

SHOW MEMORY [chl=*range*]

Cette commande indique à l'unité d'exécution cible d'envoyer le contenu de sa table de mémoire partagée mappée au processus de traçage client. Il est essentiel que les bits de masque de trace adéquats soient définis, pour que le client puisse recevoir ce type de données. Les bits INF et DGN doivent être activés dans le masque de trace de ce canal. (reportez-vous à la section «Activation ou désactivation d'un bit de masque» à la page 359). Par exemple, la commande **show memory chl=2** affiche les informations suivantes :

\*\*\* Shared Memory Utilisation for Process AMQZXMA0.EXE - pid/tid 248-1 \*\*\*

```
0942 - ShmId: 0248000f - Addr: 011d8000/01211fff - Perm: 950 - Size: 00038530 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/AMQ
0943 - ShmId: 0248000e - Addr: 00fbc000/011bdfff - Perm: 950 - Size: 002005f8 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/AMQ
0944 - ShmId: 0248000d - Addr: 00f1c000/00f39fff - Perm: 950 - Size: 0001d478 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/WLM
0945 - ShmId: 0248000c - Addr: 00cba000/00cfbfff - Perm: 944 - Size: 000405f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/HMEMSET.0
0946 - ShmId: 0248000b - Addr: 00c98000/00cb9fff - Perm: 944 - Size: 000205f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon005.0
0947 - ShmId: 0248000a - Addr: 00a96000/00c97fff - Perm: 944 - Size: 00200584 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon004.0
0948 - ShmId: 02480009 - Addr: 00a74000/00a95fff - Perm: 944 - Size: 000205f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon003.0
0949 - ShmId: 02480008 - Addr: 00a62000/00a73fff - Perm: 944 - Size: 000105f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon002.0
0950 - ShmId: 02480007 - Addr: 00a20000/00a61fff - Perm: 944 - Size: 000405f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon001.0
0951 - ShmId: 02480006 - Addr: 00908000/00909fff - Perm: 950 - Size: 00001664 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/PLU
0952 - ShmId: 02480005 - Addr: 008f8000/00907fff - Perm: 950 - Size: 0000fff8 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/IPC
0953 - ShmId: 02480004 - Addr: 00802000/008f7fff - Perm: 950 - Size: 000f4838 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/IPC
0954 - ShmId: 02480003 - Addr: 00714000/00801fff - Perm: 950 - Size: 000ec718 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/SUB
0955 - ShmId: 02480002 - Addr: 006ee000/00713fff - Perm: 944 - Size: 000253a8 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/zutSESSAN
0956 - ShmId: 02480001 - Addr: 00600000/006edfff - Perm: 944 - Size: 000ec710 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/SUBPOOL.0
0957 - ShmId: 01280000 - Addr: 005f8000/005f9fff - Perm: 950 - Size: 00000454 Name: /var/mqm/errors
0958 - *** End of data ***
```

Les informations affichées sont le numéro de ligne dans le fichier d'historique, l'identificateur de mémoire partagée, la plage d'adresses mappée virtuelle, les indicateurs utilisés pour la création/le mappage de la section et le nom MQSeries interne donné à la section.

---

## Affichage des composants MQSeries actifs, par nom et identificateurs hexadécimaux

### SHOW COMPONENTS

Cette commande permet d'afficher tous les composants MQSeries actifs, par nom et identificateurs hexadécimaux associés. Ces identificateurs hexadécimaux seront utilisés dans d'autres commandes d'affichage MONMQ telles que show functions et select component. Par exemple, la commande **show components** affiche les informations suivantes :

```

00000001 - Data hardening
00000002 - Log management
00000003 - Object Catalogue
00000004 - Queue management
00000005 - Transaction Management
00000006 - Mobile Component
00000007 - Mobile Component
00000008 - Communications
0000000a - Object Authority Manager
0000000b - Logger
0000000d - LQM Kernal
0000000f - Administration App
00000010 - Administration App
00000013 - Command Server
00000014 - Remote queue processor
00000015 - XA Transaction Manager
00000016 - Data Conversion
00000017 - Common Services
00000018 - Common Services (overflow)
00000019 - Application Interface
0000001a - IPCC
0000001b - DCE Support
0000001c - Pluggable Services
0000001d - Agent
0000001e - XA Transaction Manager
0000001f - C++ Layer
00000020 - CLI
00000021 - Z Utilities
00000022 - Execution Controller
00000023 - App. Bindings
00000024 - Service Component
00000025 - Publish/Subscribe
00000026 - MMC Snap-in for Admin
00000027 - Web Administration
00000028 - KYG Services
00000029 - OVMS MQ kernel

```

---

## Affichage des fonctions du composant spécifié

### SHOW FUNCTIONS [comp=*hex*]

Cette commande affiche toutes les fonctions du composant indiqué. Celui-ci doit être saisi au format hexadécimal. Utilisez la commande SHOW COMPONENT pour afficher tous les composants MQSeries actifs. Par exemple, la commande **show functions component=0x1f** affiche les informations suivantes :

## ONSTARTUP start

```
00000000 - ImqBinary::copyOut
00000001 - ImqBinary::pasteIn
00000002 - ImqCache::operator =
00000003 - ImqCache::moreBytes
00000004 - ImqCache::read
00000005 - ImqCache::resizeBuffer
00000006 - ImqCache::setDataOffset
00000007 - ImqCache::setMessageLength
00000008 - ImqCache::useEmptyBuffer
00000009 - ImqCache::write
0000000a - ImqDeadLetterHeader::pasteIn
0000000b - ImqDistributionList::openInfoPrepare
0000000c - ImqItem::structureIdIs
0000000d - ImqQueueManager::backout
0000000e - ImqQueueManager::begin
0000000f - ImqQueueManager::commit
00000010 - ImqQueueManager::connect
00000011 - ImqQueueManager::disconnect
00000012 - ImqMessageTracker::setAccountingToken
00000013 - ImqMessageTracker::setCorrelationId
00000014 - ImqMessageTracker::setGroupId
00000015 - ImqMessageTracker::setMessageId
00000016 - ImqObject::close
00000017 - ImqObject::closeTemporarily
00000018 - ImqObject::inquire
00000019 - ImqObject::open
0000001a - ImqObject::openFor
.....
```

---

## Activation du traçage dès le lancement d'un processus

ONSTARTUP [start] [lu=*number*] [chl=*range*]

Cette commande active le traçage dès le début d'un processus. Lorsqu'il est exécuté, un nom logique MQS\_DEF\_TRACE est défini dans la table de noms logiques du système ; il a un nom équivalent, au format : *lu channel*. Au démarrage de tout processus MQSeries, ce nom logique est contrôlé dans la routine d'initialisation des processus. S'il y figure, la connexion à la LU et au canal indiqués est effectuée. Si l'identificateur du canal est déjà attribué, le canal disponible suivant est utilisé. Cette commande est utile si vous souhaitez réaliser un traçage dès les premières phases de la création d'une unité d'exécution ou d'un processus MQSeries.

---

## Désactivation du traçage dès le lancement d'un processus MQSeries

ONSTARTUP [stop]

Cette commande retire l'affectation de la logique MQS\_DEF\_TRACE de la table de noms logiques système. Elle permet ainsi d'éviter que le processus MQSeries ne soit tracé dès son démarrage.

---

## Connexion de l'unité d'exécution cible au canal indiqué

CONNECT pid *number* [tid=*number*] [chl=*range*]

Cette commande indique à l'unité d'exécution cible de se connecter au canal précisé. Si aucun canal n'est spécifié, le premier canal disponible est utilisé.

---

## Déconnexion de l'unité d'exécution cible du canal indiqué

DISCONNECT [chl=*range*]

Cette commande indique à l'unité d'exécution cible de se déconnecter du canal précisé.

---

## Affichage des messages de trace en temps réel destinés à la boîte aux lettres de trace de la LU

TRACE START [node=*string*]

Cette commande lance un processus de trace client. Il affiche les messages de trace en temps réel destinés à la boîte aux lettres de trace de la LU. Le paramètre de noeud (optionnel) crée une fenêtre sur ce noeud et redirige les sorties vers cette fenêtre.

---

## Arrêt d'un processus client en cours

TRACE STOP

Cette commande détache le processus client en cours de la boîte aux lettres de trace et l'arrête. Toutes les unités d'exécution écrivant dans cette boîte aux lettres sont désactivées.

MQT> trace stop

```
Circular buffering has been disabled for process 24d thread 1
Circular buffering has been disabled for process 24c thread 1
Circular buffering has been disabled for process 24b thread 1
Circular buffering has been disabled for process 248 thread 1
Disconnecting thread pid : 24d, tid : 1 from channel 0 ..... OK
Disconnecting thread pid : 24c, tid : 1 from channel 1 ..... OK
Disconnecting thread pid : 24b, tid : 1 from channel 2 ..... OK
Disconnecting thread pid : 248, tid : 1 from channel 3 .....OK
```

\*\*\* Trace ended - no processes connected \*\*\*

La sortie ci-dessus s'affiche dans la fenêtre de trace du client.

---

## Indication des données de trace

SELECT [component] AND/OR [function] OR [*fname*]

Cette commande permet d'indiquer jusqu'à huit combinaisons composant/fonctions devant être tracées. Toutes les autres données de trace sont ignorées. Vous pouvez indiquer un nom de fonction, un composant, ou un couple composant/fonction. S'il est valide, le couple composant/fonction est écrit dans la table de filtre. Si aucune entrée ne correspond, le traçage est réalisé par défaut pour TOUS les couples composant/fonctions.

La saisie de la commande **SELECT** sans paramètre affiche le contenu de la table de filtre. Sur chaque ligne de la sortie, figure l'entrée d'index de la table, le composant, la fonction au format hexadécimal et son nom. Si vous saisissez seulement un composant, toutes ses fonctions seront tracées. Les informations affichées sont alors 0xffff suivi de la valeur de la fonction.



## SELECT

Par exemple, la commande **SELECT** toute seule affiche :

```
Ch1:0 - Cmp/fnc selection criteria
ALL component/functions
```

Les commandes suivantes :

```
MQT>select fname="kill"
MQT>select comp=0x1f
MQT>select comp=0x20 func=0x3
MQT>select
```

affichent :

```
Ch1:0 - Cmp/fnc selection criteria
Idx: 0 - Cmp: 00000029 - Fnc: 00000005 - Name - kill
Idx: 1 - Cmp: 00000019 - Fnc: 0000ffff - Name -
Idx: 2 - Cmp: 00000016 - Fnc: 00000003 - Name - vqiAddCacheEntry
```

---

## Suppression d'une entrée donnée de la table de filtre de trace

DESELECT INDEX=<0:7>

Cette commande supprime une entrée donnée de la table de filtre de trace, selon le paramètre d'index de table. Tous les couples composants/fonctions sont tracés lorsque les huit entrées sont vides. Par exemple, une commande **select** affiche les entrées suivantes avec leurs index :

```
Ch1:0 - Cmp/fnc selection criteria
Idx: 0 - Cmp: 00000029 - Fnc: 00000005 - Name - kill
Idx: 1 - Cmp: 00000019 - Fnc: 0000ffff - Name -
Idx: 2 - Cmp: 00000016 - Fnc: 00000003 - Name - vqiAddCacheEntry
```

Les commandes **deselect** suivantes suppriment les processus ou fonctions indiquées :

```
MQT> dese1 index=0
MQT> dese1 index=2

MQT>select
Ch1:0 - Cmp/fnc selection criteria
Idx: 1 - Cmp: 00000019 - Fnc: 0000ffff - Name -
-----
MQT> deselect index=1
MQT> select

Ch1:0 - Cmp/fnc selection criteria
ALL component/functions
-----
```

---

## Ecriture de messages de trace dans un fichier binaire par un processus client

OPEN BINARY [filename=*string*]



Cette commande ouvre un fichier binaire de message de trace, et déclenche l'écriture dans ce fichier, par le processus client, des messages en temps réel. Ce fichier peut ensuite être utilisé pour analyser les performances des applications MQSeries. Le nom de fichier par défaut est `mqs_root:[mqm.errors]mqs_buffer_xx.bin` (où *xx* est le numéro de LU).

---

## Fermeture d'un fichier binaire de messages de trace

CLOSE BINARY

Cette commande ferme le fichier binaire de trace de LU indiqué.

---

## Écriture des messages de trace dans un fichier texte

OPEN TEXT [filename=*string*]

Cette commande ouvre un fichier texte et déclenche l'écriture, par le processus client, des messages de trace binaires formatés. Ce fichier pourra être consulté par la suite simplement par une commande DCL type ou edit. Le nom de fichier par défaut est `mqs_root:[mqm.errors]mqs_buffer_xx.lis` (où *xx* est le numéro de LU). L'avantage de ce fichier de sortie est qu'il est lisible directement. Toutefois, il occupe plus d'espace disque qu'un fichier binaire.

---

## Fermeture du fichier texte de messages de trace

CLOSE TEXT

Cette commande permet de fermer le fichier texte de trace de LU spécifié.

---

## Horodatage des messages

ENABLE TIMESTAMP [chl=*range*]

Cette commande active l'indicateur d'horodatage dans la table de définition des canaux. Utilisez-la pour obliger les processus MQSeries à indiquer l'heure dans chaque message. Cet indicateur doit être défini afin de pouvoir utiliser un fichier de trace binaire pour analyser les performances.

---

## Désactivation de l'horodatage des messages

DISABLE TIMESTAMP [chl=*range*]

Cette commande désactive l'indicateur d'horodatage dans la table de définition de canaux. (reportez-vous à la section «Horodatage des messages».)

---

## Activation du traçage

ENABLE TRACE [chl=*range*]

Cette commande active l'indicateur de trace RTime dans la table de définition de canaux. Utilisez cet indicateur pour activer ou désactiver l'envoi de messages de trace à un client. Lorsqu'une unité d'exécution est connectée et qu'un client de trace est présent, vous pouvez utiliser TRACE START pour activer ou désactiver le canal au lieu de déconnecter l'unité d'exécution.

## DISABLE TRACE

---

### Désactivation du traçage

DISABLE TRACE [chl=*range*]

Cette commande désactive l'indicateur de trace RTime de la table de définition de canaux. (reportez-vous à la section «Activation du traçage» à la page 357.)

---

### Enregistrement de l'historique des messages

ENABLE HISTORY [chl=*range*]

Cette commande active l'indicateur d'historique dans la table de définition de canaux, pour les canaux indiqués. Cette commande indique à l'unité d'exécution connectée d'écrire des messages de trace dans le tampon circulaire de la LU. L'écriture du message étant réalisée par le processus tracé, il n'est pas nécessaire qu'un processus client existe. La taille du tampon circulaire de trace est définie au cours de la création de la LU, par la commande **open**. Dans ce tampon circulaire, l'écriture reprend au début lorsque la dernière entrée est enregistrée. La zone Next Log de la table de définition des LU précise à quel endroit doit être écrit l'enregistrement suivant dans la mémoire tampon.

---

### Désactivation de l'historique de message

DISABLE HISTORY [chl=*range*]

Cette commande désactive l'indicateur d'historique de la table de définition de canaux, pour les canaux indiqués. Consultez la section «Enregistrement de l'historique des messages».

---

### Suppression de l'historique des messages

DELETE HISTORY [chl=*range*]

Cette commande permet de supprimer tous les messages du tampon d'historique circulaire. Cette commande peut être exécutée même si des processus sont en train d'écrire dans le tampon. Il n'est donc pas nécessaire de désactiver l'historique avant de procéder à sa suppression.

---

### Définition de la profondeur de l'historique

SET [depth]

Cette commande contrôle la position maximale dans la pile, affichée dans la fenêtre de trace du client. Les messages dont la position dépasse cette valeur n'apparaîtront pas dans la fenêtre. Toutefois, ils figureront dans le fichier de trace binaire et dans le tampon d'historique, le cas échéant. La valeur par défaut, définie sur zéro, affiche tous les messages, quelle que soit leur position dans la pile. Il est conseillé de définir cette valeur sur un très petit chiffre (par exemple 1) lorsque vous écrivez des données d'analyse dans le fichier binaire. Le fait d'afficher toute la pile affectera les performances du processus client.

---

### Réinitialisation des données de la pile et d'historique pour un canal

SET [free] [chl=*range*]

Cette commande permet de réinitialiser le canal indiqué. Toutes les données existantes sur la pile et l'historique sont effacées : l'attribution du canal est ainsi annulée et il peut être réutilisé.

## Activation ou désactivation d'un bit de masque

```
SET [mask=var] [chl=range]
```

Cette commande active ou désactive un bit de masque dans la zone de masque de contrôle des unités d'exécution connectées. Chaque bit représente un type de message généré par un processus MQSeries. Vous pouvez utiliser cette commande pour filtrer les types de messages que vous souhaitez tracer. Les types de messages sont les suivants :

```
MQT>set mask = 0xffffffff chl=1
MQT>show mask chl=1
```

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

Pour définir une combinaison de ces types de messages, délimitez chaque type de masque avec un symbole OR. Exemple :

```
MQT>set mask =0x0 chl=1 MQT>show mask chl=1
```

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

```
MQT>set mask = mtx | evt | fent chl=1
MQT>show mask chl=1
```

## SET mask

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

Chaque masque comprend huit types de masques, dont vous pouvez modifier l'état pour activer ou désactiver un type de message particulier. Exemple : si vous êtes uniquement intéressés par les points d'entrée des fonctions, entrez la commande **set mask = fent**.

---

## Attribution d'une couleur à un canal

```
SET COLOR [chl=range]
```

Cette commande permet d'associer une couleur à un canal. Toutes les sorties associées à ce canal apparaissent dans cette couleur, jusqu'à ce que la couleur soit changée ou le canal réinitialisé. Ceci s'avère particulièrement utile pour mettre en évidence ou faire la distinction entre les messages de différentes unités d'exécution, dans une même sortie. Par exemple, les commandes :

```
MQT> set color=yellow chl=2
MQT> set color=blue chl=0
MQT> sho chan chl=0:3 connected
```

affichent :

Chl	Pid/Tid	Mailbox	Stack	History	RTime	Time	Mask	Process Name
0	00000245/1	800b0330	4	0	0	0	fffffff	AMQZLAA0.EXE
1	00000244/1	800b0200	8	0	0	0	fffffff	RUNMQCHI.EXE
2	00000243/1	800b01e0	10	0	0	0	fffffff	AMQRRMFA.EXE
3	00000242/1	800b01c0	4	0	0	0	fffffff	AMQZLLP0.EXE

où le canal 0 est bleu et le canal 2 jaune.

---

## Réacheminement de la sortie vers un fichier

```
SET OUTPUT [filename=string]
```

Cette commande achemine toute sortie vers le fichier indiqué et désactive l'affichage à l'écran. La commande **output**, lorsqu'elle est utilisée en tant que paramètre avec d'autres commandes, est applicable à cette commande uniquement. Remarque : les erreurs continuent de s'afficher à l'écran, non dans le fichier. Seules les données de trace valides sont écrites dans le fichier indiqué.

---

## Analyse du fichier de trace binaire

```
ANALYSE [component] [function] [unit=xx]
```

## ANALYSE

Cette commande analyse le contenu du fichier de trace binaire utilisé précédemment au cours d'une session de trace. Vous pouvez préciser le composant ou la fonction à analyser, toutefois vous n'obtiendrez un résultat que si le fichier contient des données sur ce composant ou cette fonction. Par exemple, si lorsque vous avez généré le fichier, vous avez précisé un masque de trace ou sélectionné un composant spécifique, seuls ces éléments seront trouvés dans le fichier binaire. Les composants ou fonctions qui ne répondent pas à ce critère ne peuvent donc pas être utilisés dans l'analyse.

**Remarque :** Si vous utilisez la commande complète, faites attention à bien utiliser ANALYSE avec un *S* majuscule. Ne confondez pas cette commande MONMQ avec la commande ANALYZE d'OpenVMS, elles ont toutes deux un rôle différent.

Le paramètre unit permet de préciser l'unité de temps utilisée dans l'analyse. Il peut avoir l'une des valeurs suivantes : (xx) secondes, millisecondes, microsecondes, nanosecondes. L'unité par défaut est la milliseconde.

Par exemple, pour afficher des informations en microsecondes, utilisez la commande **analyse unit=micro:**. Les informations suivantes sont alors affichées :

## ANALYSE

```

=====
                        COMPONENT :- Common Services
=====
Calls  Minimum      Average      Maximum      Total      Function
42     0.00         66.41       311.78      2789.15    xcsRequestMutexSem
42     0.00         96.98       222.64      4072.98    xcsReleaseMutexSem
6      0.00        138.50       286.11       831.00     xcsResetEventSem
5      0.00       10112.60    10159.51    50563.01   xcsWaitEventSem
6      0.00         564.74     1029.23     3388.45    xcsCheckExtendMemory
42     0.00          51.32       266.86     2155.41    xllSemReq
42     0.00          73.05       159.17     3068.16    xllSemRel
=====
                        COMPONENT :- Common Services (overflow)
=====
Calls  Minimum      Average      Maximum      Total      Function
36     0.00         41.15       122.06     1481.35    xcsCheckProcess
6      0.00         37.92       68.35      227.52     xihGetConnSPDetailsFromList
6      0.00         65.26       114.25     391.58     xihHANDLEtoSUBPOOLFn
6      0.00         12.69       23.44     1267.49    xihGetNextSetConnDetailsFromList
6      0.00         12.53       22.46       75.19     xcsRequestThreadMutexSem
6      0.00         12.37       22.46       74.21     xcsReleaseThreadMutexSem
=====
                        COMPONENT :- IPCC
=====
Calls  Minimum      Average      Maximum      Total      Function
5      0.00       10589.97    11029.84    52949.85    xcpReceiveOnLink
=====
                        COMPONENT :- CLI
=====
Calls  Minimum      Average      Maximum      Total      Function
6      0.00          1.95        1.95        11.72      zapInquireStatus
=====
                        COMPONENT :- Execution Controller
=====
Calls  Minimum      Average      Maximum      Total      Function
6      0.00         954.46     3275.18    11726.79    zxcProcessChildren
6      0.00         12.69       22.46       76.17     zxcStartWLMServer
=====
                        COMPONENT :- OVMS MQ kernel
=====
Calls  Minimum      Average      Maximum      Total      Function
36     0.00         15.49       99.60       557.58     kill
6      0.00          0.65        3.91         3.91     vms_mapgbl
84     0.00         11.17       88.16       938.68     vms_get_lock
84     0.00         42.41      221.94     3562.54     vms_mtx
12     0.00         44.51      149.40       534.14     vms_get_mbx_chan
11     0.00       4651.77    10137.05    51169.42    vms_evt
6      0.00          1.63        4.88         9.76     vms_check_health
=====

```

Signification des colonnes :

<b>Calls</b>	Nombre d'entrées dans la fonction pendant la session de trace.
<b>Minimum</b>	Temps le plus court passé dans la fonction.
<b>Average</b>	Durée moyenne des appels passés à la fonction (durée totale divisée par le nombre d'appels).
<b>Maximum</b>	Temps le plus long passé dans la fonction.
<b>Total</b>	Durée totale passée dans cette fonction, pour tous les appels.
<b>Function</b>	Nom de la fonction.

**Remarque :** L'analyse porte sur le contenu du fichier de trace binaire. Il revient à l'utilisateur de définir les limites de l'analyse, en ouvrant le fichier de trace binaire et en activant/désactivant la trace au moment souhaité.

---

## Affichage de l'état actuel des unités d'exécution MQSeries

FFST [chl=*range*]

Cette commande oblige l'unité d'exécution connectée au canal cible à effectuer un FFST. Cette commande n'a AUCUN EFFET sur le chemin d'exécution des unités d'exécution cibles. Elle réalise une image instantanée de l'état en cours de toute unité d'exécution MQSeries. Le rapport FFST fournit des informations sur l'utilisation des ressources des unités d'exécution, les droits d'accès et autres données système utiles. Il est clairement indiqué dans l'en-tête que le FFST a été créé par MONMQ (voir ci-dessous), et qu'il N'EST PAS le résultat d'un incident.

Voici un exemple de sortie écran :

```
MQSeries First Failure Symptom Report
=====
Date/Time       :- Wednesday November 12  10:59:38 GMT 2000
Host Name       :- CATWMN (Unknown)
PIDS            :- 5697175
LVLS           :- 510
Product Long Name :- MQSeries for OpenVMS Alpha
Vendor         :- IBM
Probe Id       :- VM026000
Application Name :- MQM
Component      :- vms_evt
Build Date     :- Oct 22 2000 (Collector)
Userid        :- [400,400] (SYSTEM)
Program Name   :- AMQZXMA0.EXE
Process       :- 00000248
Thread        :- 00000001
QueueManager   :- BKM3
Major Errorcode :- xecF_E_UNEXPECTED_SYSTEM_RC
Minor Errorcode :- OK
Probe Type     :- MSGAMQ6119
Probe Severity  :- 2
Probe Description :- AMQ6119: An internal MQSeries error has occurred
                  (** FORCED FFST BY USER **)
Comment1       :- *** FORCED FFST BY USER ***
Comment2       :- -SYSTEM-S-NORMAL, normal successful completion

etc...
```

---

## Fermeture d'une trace et sortie de MONMQ

EXIT

Cette commande permet de fermer la trace et de quitter MONMQ.

---

## Sortie de MONMQ sans fermeture de la trace

QUIT

Cette commande permet de quitter MONMQ sans fermer la trace, qui continue à s'exécuter. La session de trace précédente sera reprise à la prochaine activation de MONMQ.

## Gestion de la mémoire partagée avec MONMQ

Dans des circonstances inhabituelles, par exemple en cas de panne d'un gestionnaire de files d'attente ou d'un arrêt forcé effectué avec la commande OpenVMS stop /id, il se peut que le gestionnaire de files d'attente ne supprime pas automatiquement les segments de mémoire partagée de MQSeries. Si ce problème se produit, il est impossible de redémarrer le gestionnaire de files d'attente car **strmqm** indique qu'il fonctionne déjà.

MONMQ peut lister la mémoire partagée MQ (sections globales) existante, et supprimer ces sections de mémoire partagées.

**Remarque :** Assurez-vous que tous les gestionnaires de files d'attente sont arrêtés avant d'utiliser MONMQ pour supprimer les segments de mémoire partagée. Dans le cas contraire, un incident de gestionnaire de files d'attente se produit, entraînant un risque d'altération des files d'attente.

Vous pouvez utiliser la commande MONMQ SHOW PROCESS pour vous assurer qu'aucun processus MQ n'est en cours. Si des processus sont en cours sur un gestionnaire de files d'attente en panne, et que vous ne parvenez pas à l'arrêter avec la commande endmqm, utilisez la commande DCL OpenVMS stop /id=<pid>.

Vérifiez qu'aucun processus MQSeries n'est en cours d'exécution à l'aide de la commande suivante :

```
MQT> show process
```

```
MQ Processes
PID      Proc Name      Image      Process      WS Size      WS Peak      Virt Peak      Gbl Pg Cnt Prc Pg Cnt Total Mem
-----
List the shared memory global sections that currently exist

MQT> show globals
MQS1_shm_2695000a (00000000) WRT      DZRO PRM SYS Pgltcnt/Refcnt=4112/514
MQS1_shm_26950009 (00000000) WRT      DZRO PRM SYS Pgltcnt/Refcnt=272/34
MQS1_shm_ffffffff (00000000) WRT      DZRO TMP SYS Pgltcnt/Refcnt=144/63
MQS1_shm_00000000 (00000000) WRT      DZRO TMP SYS Pgltcnt/Refcnt=6304/394
```

Vous obtiendrez la liste des sections globales de mémoire partagée existantes à l'aide de la commande :

```
MQT> show globals
```

Pour supprimer les sections globales suivantes :



```
MQT> delete
Deleted global section: MQS1_shm_2695000a
Deleted global section: MQS1_shm_26950009
Deleted global section: MQS1_shm_ffffffff
sys$delgbl - unable to delete section MQS1_shm_00000000
```

L'erreur relative à la section MQS\_shm\_00000000 était prévisible puisqu'elle est utilisée par MONMQ. Vous pouvez maintenant quitter MONMQ avec la commande :

```
MQT> exit
```

Vous pouvez utiliser la commande delete depuis un script, si vous êtes sûr que tous les processus de gestionnaires de files d'attente sont arrêtés :

```
$ monmq delete
Deleted global section: MQS1_shm_2695000a
Deleted global section: MQS1_shm_26950009
Deleted global section: MQS1_shm_ffffffff
sys$delgbl - unable to delete section MQS1_shm_00000000
```

---

## Scripts et macros dans MONMQ

Vous pouvez exécuter un script de commandes depuis MONMQ ou à partir d'une invite de commande. Des scripts peuvent être utiles pour rassembler des ensembles de données ou pour configurer l'environnement MONMQ. Au démarrage de MONMQ, un script est exécuté depuis SYS\$MANAGER:MQS\_TRACE\_STARTUP.MQT pour configurer les variables de trace dans MONMQ.

**Remarque :** Si le script ne figure pas dans le répertoire courant, le chemin complet doit être indiqué entre guillemets. Par exemple :

```
MQT> ! "sys$manager:test.mqt"
```

Vous pouvez aussi définir une macro pour accélérer les tâches communes ou répétitives. Une macro se compose de trois parties :

1. Le nom de la macro. Il doit s'agir d'un nom de commande unique.
2. Le corps de la macro. Il peut comprendre plusieurs lignes, et consiste en une liste de commandes MQSeries. Le corps de la macro est délimité par { et }.  
Lorsque vous déclarez une macro composée de plusieurs lignes, l'invite MONMQ devient **\*\*MACRO>**. Tout élément \$n, où n est un nombre à un seul chiffre, est remplacé par le paramètre n de la ligne de commande de la macro.
3. Un court texte d'aide, affiché par la commande help <macroname>. Le texte d'aide doit être mis entre guillemets.

Vous devez tenir compte des éventuels problèmes de délais lorsque vous déclarez une macro. Une macro s'exécute très rapidement, mais certaines commandes MONMQ demandent à un processus distant de réaliser une tâche. Cette tâche doit être terminée avant que la commande de macro suivante ne démarre.

C'est pour cette raison qu'il faut parfois ajouter des temporisations. Pour définir une temporisation, utilisez la commande **sleep**. Elle applique un paramètre de délai indiqué en dixièmes de seconde.

## Scripts et macros

Les commandes suivantes permettent de créer une macro qui déconnecte un canal, réinitialise le masque de trace et libère le canal.

**Remarque :** Vous pouvez placer plusieurs commandes MONMQ sur une ligne en utilisant le caractère “;” comme séparateur.

```
MQT> declare tmpchl intrange "variable to hold a chl range temporarily"
MQT> macro remove { set tmpchl = chl ; dis chl= $1 ; sleep delay=5
**MACRO> set mask=0xffffffff chl= $1 ; set free chl = $1
**MACRO> set chl=tmpchl
**MACRO> } "A macro to disconnect and free channels Param: chl number"
MQT> help remove

VERB remove:
A macro to disconnect and free channels Param: chl number

Macro text:
set tmpchl = chl ; dis chl= $1 ; sleep delay=5 ; set mask=0xffffffff chl=$1
; set free chl = $1 ; set chl=tmpchl
MQT>remove 4
ok - process disconnected process 282 from channel 4
```

## Exemple de session de trace

Cette section décrit une session de trace type, en indiquant chaque commande MONMQ de la séquence. Cet exemple décrit le traçage du contrôleur d'exécution d'un gestionnaire de files d'attente en cours d'exécution, ainsi que de l'unité d'exécution principale des agents qui lui sont associés.

Avant de commencer le traçage, les conditions suivantes doivent être réunies :

- Démarrez le gestionnaire de files d'attente faisant l'objet du traçage - STRMQM BKM3
- Vérifiez que sys\$manager:mqs\_trace\_startup.mqt n'a pas d'autres commandes que celles qui sont préinstallées par défaut.
- Vérifiez que la logique MQS\_DEF\_TRACE N'EST PAS définie. Dans le cas contraire, appliquez la commande **ONSTARTUP END** de MONMQ.

Start momq.

```
>monmq
```

L'invite MONMQ s'affiche :

```
MQT>
```

Ouvrez une LU unique avec zéro comme identificateur, dix canaux et un tampon d'historique de 100 messages (remarque : 100 messages seraient insuffisants pour un traçage normal. 1000 est habituellement le nombre qui convient).

```
MQT> open lu=0 nochls=10 buffer=100
ok - LU:0 opened
```

Affichez la définition LU1 :

## Exemple de session de trace

```
MQT> show seg lu=1
```

```
Trace LU           : 1
Mailbox name       : MQS_TRC_MBX_1
Device name        : MBA431:
Status             : Disabled
Mailbox channel    : 384
History buffer size : 100
Threads mapped #   : 1
Time stamping      : Enabled
Max channels #     : 10
Display depth      : 0
Text filename      :
Binary filename    :
Last status        : 1
Connection map[0]  : 0
=====
```

Affichez les processus MQSeries :

```
MQT> show process
```

PID	Proc_Name	Image	Process	WS_Size	WS_Peak	Virt_Peak	Gbl_Pg_Cnt	Prc_Pg_Cnt	Total_Mem
2A00023D	BKM1_AG	AMQZLAA0	Agent	23152	16576	203776	3616	12960	16576
2A00023C	BKM1_CI	RUNMQCHI	Run Chan Init	8752	6208	180832	1840	4368	6208
2A00023B	BKM1_RM	AMQRRMFA	Repository Mgr	11152	8144	185360	2224	5920	8144
2A00023A	BKM1_CP	AMQZLLP0	Checkpoint	8752	6384	185952	1920	4464	6384
2A000239	BKM1_LG	AMQHASMx	Logger	8752	6288	182016	2080	4208	6288
2A000238	BKM1_EC	AMQZXMA0	EC	20752	15232	203792	3536	11680	15216
2A000128	_FTA4:	MONMQ	MONMQ Utility	8400	8528	198736	2224	3584	5808
									-----
									52112

Identifiez le contrôleur d'exécution et le processus agent, puis connectez-les aux canaux un et deux respectivement :

```
MQT>connect pid=0x238 tid=1 lu=1 chl=1
MQT>connect pid=0x23D tid=1 lu=1 chl=2
```

Contrôlez les données de connexion :

```
MQT>show channel full connected lu=1
```

## Exemple de session de trace

```
Pid/Tid          : 2a000bc/1
Status           : *** Connected ***
Process name     : AMQZXMA0.EXE
Assigned LU      : 1
Channel no.      : 1
Mailbox channel  : 800c03f0
Current stack depth : 4
Circular logging  : Disabled
Next log entry   : 0
Realtime tracing : Disabled
Time stamping    : Disabled
Trace mask       : ffffffff
Step mode        : Off
No Wait          : On
Last QIO status  : 0
Mapped address   : 1242000-126bfff
=====
```

```
Pid/Tid          : 2a000c1/1
Status           : *** Connected ***
Process name     : AMQZLAA0.EXE
Assigned LU      : 1
Channel no.      : 2
Mailbox channel  : 800c03f0
Current stack depth : 4
Circular logging  : Disabled
Next log entry   : 0
Realtime tracing : Disabled
Time stamping    : Disabled
Trace mask       : ffffffff
Step mode        : Off
No Wait          : On
Last QIO status  : 0
Mapped address   : 1376000-139ffff
=====
```

Définissez les valeurs par défaut de chl, lu et tid, pour ne pas avoir à les entrer dans les commandes suivantes.

```
MQT>default chl=1:2 lu=1 tid=1
```

Définissez les couleurs des canaux de façon à identifier les différents messages de trace. Notez que le paramètre chl est précisé dans ces deux commandes, car si la valeur par défaut (1:2) avait été utilisée, les couleurs jaune puis cyan auraient été attribuées aux deux canaux.

```
MQT>set chl=1 color=yellow
MQT>set chl=2 color=cyan
```

Précisez maintenant les canaux.

```
MQT>show channels
```

Chl	Pid/Tid	Mailbox	Stack	History	RTime	Time	Mask	Process Name
1	2a0000bc/1	800c03f0	4	0	0	0	fffffff	AMQZXMA.EXE
2	2a0000c1/1	800c0360	4	0	0	0	fffffff	AMQZLAA0.EXE

## Exemple de session de trace

Les processus sont maintenant connectés aux canaux. Vous pouvez à présent examiner leurs piles.

```
MQT>show stacks
```

```
0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 00:00:00.00 03 - 02 ---->| zcpReceiveOnLink
0003 - 00:00:00.00 03 - 03 ----->| xcsWaitEventSem
0004 - 00:00:00.00 03 - 04 ----->| vms_evt

0001- 00:00:00.00 03 - 01 -->| zlaMain
0002 - 00:00:00.00 03 - 02 ---->| zcpReceiveOnLink
0003 - 00:00:00.00 03 - 03 ----->| xcsWaitEventSem
0004 - 00:00:00.00 03 - 04 ----->| vms_evt
```

Grâce à l'horodatage de ces deux canaux, il est possible de voir si le processus a échoué ou non.

```
MQT>enable timestamp
MQT>show stack
```

```
0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 12:36:20.18 03 - 02 ---->| zcpReceiveOnLink
0003 - 12:36:20.81 03 - 03 ----->| xcsWaitEventSem
0004 - 12:36:20.83 03 - 04 ----->| vms_evt

0001- 00:00:00.00 03 - 01 -->| zlaMain
0002 - 12:36:20.18 03 - 02 ---->| zcpReceiveOnLink
0003 - 12:36:20.81 03 - 03 ----->| xcsWaitEventSem
0004 - 12:36:20.83 03 - 04 ----->| vms_evt
```

Le fait que l'heure soit à présent indiquée pour certains messages, prouve que les deux processus sont actifs. Dans cet exemple, les deux événements font partie d'une boucle dont le délai d'expiration est de 10 secondes. Vous pouvez comparer le délai d'expiration et l'heure du message en exécutant en continu la commande **show stack** jusqu'à ce qu'une modification des données d'horodatage ait lieu.

L'activation de l'historique permet à présent d'obliger chaque processus à écrire ses messages de trace dans le tampon circulaire.

```
MQT>enable history
MQT> show history
```

Ecrivez à présent tous les messages de trace dans le tampon. Vous pouvez vérifier cette procédure en affichant le masque de trace et la table des composants/fonctions.

```
MQT>show mask
```

## Exemple de session de trace

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

```
Trace Mask for Channel 2
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

```
MQT> select
```

```
Ch1:1 - Cmp/fnc selection criteria
ALL component/functions
```

```
-----
Ch1:2 - Cmp/fnc selection criteria
ALL component/functions
-----
```

## Exemple de session de trace

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

```
Trace Mask for Channel 2
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

Concentrez-vous maintenant sur un type de messages précis. Supposons par exemple que vous soyez intéressé uniquement par les messages de diagnostic de mémoire partagée.

```
MQT>set mask=shm
MQT>show mask
```

Les deux processus vont maintenant écrire uniquement les messages de diagnostic de mémoire dans le tampon. Supprimez donc ce tampon, et après quelques secondes d'attente, examinez de nouveau son contenu de la mémoire tampon.

```
MQT>clear history

(wait a few seconds)

MQT> show history
```

## Exemple de session de trace

\*\*\* Trace History Ch1:1 \*\*\*

```
0990 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0991 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_fffffffe mapped at 1510000-1511fff
0992 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0993 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_fffffffe mapped at 1510000-1511fff
0994 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0995 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_fffffffe mapped at 1510000-1511fff
0996 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0997 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_fffffffe mapped at 1510000-1511fff
0998 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0999 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_fffffffe mapped at 1510000-1511fff
```

\*\*\* End of buffer \*\*\*

\*\*\* Trace History Ch1: \*\*\*

\*\*\* End of buffer \*\*\*

A présent, affichez également les messages de diagnostic d'événement dans la sortie de trace. Une attente de quelques secondes est nécessaire une fois le masque défini.

```
MQT>set mask=evt | shm
```

```
(wait a few seconds)
```

```
MQT>show history
```



## Exemple de session de trace

\*\*\* Trace History Ch1:1 \*\*\*

```
0977 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0978 - 00:00:00.00 00 - 04 ..... | vms_evt Event BKM3/@ipcc_e_1_2 TIMEOUT
0979 - 00:00:00.00 00 - 04 ..... | vms_evt rc = 1
0980 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0981 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0982 - 00:00:00.00 00 - 04 ..... | vms_evt Reset on mailbox BKM3/@ipcc_e_1_2 : tout = -1
0983 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0984 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Returning key 1a0
0985 - 00:00:00.00 00 - 04 ..... | vms_evt rc = 0
0986 - 00:00:00.00 00 - 04 ..... | vms_evt Wait on mailbox BKM3/@ipcc_e_1_2 : tout = 10000
0987 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0988 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Returning key 1a0
0989 - 00:00:00.00 00 - 04 ..... | vms_evt Event BKM3/@ipcc_e_1_2 TIMEOUT
0990 - 00:00:00.00 00 - 04 ..... | vms_evt rc = 1
0991 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0992 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0993 - 00:00:00.00 00 - 04 ..... | vms_evt Reset on mailbox BKM3/@ipcc_e_1_2 : tout = -1
0994 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0995 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Returning key 1a0
0996 - 00:00:00.00 00 - 04 ..... | vms_evt rc = 0
0997 - 00:00:00.00 00 - 04 ..... | vms_evt Wait on mailbox BKM3/@ipcc_e_1_2 : tout = 10000
0998 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0999 - 00:00:00.00 00 - 05 ..... | vms_get_mbx_chan Returning key 1a0
```

\*\*\* End of buffer \*\*\*

\*\*\* Trace History Ch1:2 \*\*\*

```
0982 - 00:00:00.00 01 - 04 ..... | vms_evt Event BKM3/@ipcc_e_1_7 TIMEOUT
0983 - 00:00:00.00 01 - 04 ..... | vms_evt rc = 1
0984 - 00:00:00.00 01 - 04 ..... | vms_evt Reset on mailbox BKM3/@ipcc_e_1_7 : tout = -1
0985 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0986 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Returning key 1c0
0987 - 00:00:00.00 01 - 04 ..... | vms_evt rc = 0
0988 - 00:00:00.00 01 - 04 ..... | vms_evt Wait on mailbox BKM3/@ipcc_e_1_7 : tout = 10000
0989 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0990 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Returning key 1c0
0991 - 00:00:00.00 01 - 04 ..... | vms_evt Event BKM3/@ipcc_e_1_7 TIMEOUT
0992 - 00:00:00.00 01 - 04 ..... | vms_evt rc = 1
0993 - 00:00:00.00 01 - 04 ..... | vms_evt Reset on mailbox BKM3/@ipcc_e_1_7 : tout = -1
0994 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0995 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Returning key 1c0
0996 - 00:00:00.00 01 - 04 ..... | vms_evt rc = 0
0997 - 00:00:00.00 01 - 04 ..... | vms_evt Wait on mailbox BKM3/@ipcc_e_1_7 : tout = 10000
0998 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0999 - 00:00:00.00 01 - 05 ..... | vms_get_mbx_chan Returning key 1c0
```

\*\*\* End of buffer \*\*\*

```
MQT>disable history
MQT>clear history
```

Procédez maintenant au traçage d'une fonction spécifique. Grâce aux commandes **SHOW COMPONENT** et **SHOW FUNCTION**, vous pouvez identifier les éléments qui feront l'objet du traçage. Dans cet exemple, le traçage réalisé est celui de la fonction de services communs 'xcsRequestMutexSem'. Le composant est 0x17 et le code de fonction 0x1b. Il existe deux façons de procéder :

## Exemple de session de trace

```
MQT>select comp=0x17 func=0x1b
```

ou

```
MQT>select fname="xcsRequestMutexSem"
```

Si vous activez maintenant l'historique, vous constatez qu'aucune sortie n'apparaît dans le tampon. Ceci est dû au fait qu'il faut réinitialiser les bits de masque de trace sur all.

```
MQT>enable history
MQT>show history
```

```
*** Trace History Ch1:1 ***
```

```
*** End of buffer ***
```

```
*** Trace History Ch:2 ***
```

```
*** End of buffer ***
```

```
MQT>set mask=0xffffffff
MQT>show history
```

## Exemple de session de trace

\*\*\* Trace History Ch1:1 \*\*\*

```
0973 - 00:00:00.00 01 - 02 --->| xcsRequestMutexSem
0974 - 00:00:00.00 01 - 03 ---->| x11SemReq
0975 - 00:00:00.00 01 - 04 ----->| vms_mtx
0976 - 00:00:00.00 01 - 04 .....| vms_mtx :- Locking BKM3_m_1_45 - timeout: -1
0977 - 00:00:00.00 01 - 05 ----->| vms_get_lock
0978 - 00:00:00.00 01 - 05 -----<| vms_get_lock
0979 - 00:00:00.00 01 - 04 -----<| vms_mtx
0980 - 00:00:00.00 01 - 03 -----<| x11SemReq
0981 - 00:00:00.00 01 - 02 ----<| xcsRequestMutexSem
0982 - 00:00:00.00 01 - 03 ---->| xcsRequestMutexSem
0983 - 00:00:00.00 01 - 04 ----->| x11SemReq
0984 - 00:00:00.00 01 - 05 ----->| vms_mtx
0985 - 00:00:00.00 01 - 05 .....| vms_mtx :- Locking BKM3_m_1_6 - timeout: -1
0986 - 00:00:00.00 01 - 06 ----->| vms_get_lock
0987 - 00:00:00.00 01 - 06 -----<| vms_get_lock
0988 - 00:00:00.00 01 - 05 -----<| vms_mtx
0989 - 00:00:00.00 01 - 04 -----<| x11SemReq
0990 - 00:00:00.00 01 - 03 ----<| xcsRequestMutexSem
0991 - 00:00:00.00 01 - 03 ---->| xcsRequestMutexSem
0992 - 00:00:00.00 01 - 04 ----->| x11SemReq
0993 - 00:00:00.00 01 - 05 ----->| vms_mtx
0994 - 00:00:00.00 01 - 05 .....| vms_mtx :- Locking BKM3/@ipcc_m_1_18 - timeout: -1
0995 - 00:00:00.00 01 - 06 ----->| vms_get_lock
0996 - 00:00:00.00 01 - 06 -----<| vms_get_lock
0997 - 00:00:00.00 01 - 05 -----<| vms_mtx
0998 - 00:00:00.00 01 - 04 -----<| x11SemReq
0999 - 00:00:00.00 01 - 03 ----<| xcsRequestMutexSem
```

\*\*\* End of buffer \*\*\*

Vous constatez à présent que seule la fonction précisée et ses fonctions filles sont tracées pour les deux processus. La commande **select** permet de tracer simultanément au maximum huit composants et fonctions. Pour activer le traçage en temps réel, il est nécessaire de créer un processus client qui affichera les messages d'une LU spécifique. Pour ce faire, lancez une commande **TRACE**.

Elle a pour effet de démarrer un processus client sur le noeud précisé et d'attendre les messages de trace entrants. Les sessions de trace affichées sur le client peuvent toujours être contrôlées à l'aide de **MONMQ**.

```
MQT>trace start node="mihell"
```

Activez maintenant le processus client et affichez les messages à mesure qu'ils arrivent.

```
MQT>enable trace
```

Vous pouvez ajouter et retirer des unités d'exécution MQSeries de la sortie de trace à votre gré. Les unités d'exécution peuvent rester connectées tandis que leurs données de trace sont désactivées afin que le traçage n'ait pas d'effet négatif sur les performances.

Il est possible de lancer le traçage dès le démarrage d'un processus ou d'une unité d'exécution. Pour ce faire, utilisez la commande **ONSTARTUP** : tous les nouveaux processus MQSeries seront tracés dès leur lancement.

## Exemple de session de trace

Pour arrêter une session de trace, désactivez tous les canaux actifs et arrêtez le processus client à l'aide de la commande **close**.

Si vous souhaitez que le traçage continue de s'exécuter, utilisez la commande **quit** dans MONMQ et arrêtez ensuite le traçage au moment voulu.

Les bits de masque de trace et les sélections composant/fonction sont deux choses différentes. Les bits de masque de trace contrôlent la sortie des types de messages de trace. L'entrée de trace et la sortie de trace sont des exemples de types de messages. Si vous les désactivez, tous les paramètres que vous effectuerez à l'aide de la commande **select** resteront sans effet, car la sélection composant/fonction dépend de la définition de ces bits de masque.

---

## Annexe I. Exits utilisateur

MQSeries pour Compaq OpenVMS prend en charge les programmes d'exit de canal et de conversion de données. Pour plus d'information 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 *Guide de programmation d'applications* et *MQSeries Application Programming Reference*.

Cette annexe contient des informations spécifiques à l'utilisation des programmes d'exit sous MQSeries pour Compaq OpenVMS.

---

### Exits de canal et de pondération de charge

Sous MQSeries pour Compaq OpenVMS il n'est pas nécessaire de lier une version d'exit tournant dans une unité d'exécution distincte.

---

### Exits de pondération de charge de cluster MQSeries

Si vous êtes amené à lier un exit de pondération de charge sous OpenVMS, indiquez les informations suivantes dans le fichier d'options de liaison :

```
sys$share:mqm/share  
sys$share:mqt1/share  
SYMBOL_VECTOR=(c1w1Function=PROCEDURE,MQStart=PROCEDURE)
```

Un nom logique d'exécution système est nécessaire pour référencer l'image d'exit. Par exemple, si le nom de l'exit est SYS\$SHARE:AMQSWLM.EXE, définissez le nom logique suivant :

```
$DEFINE/SYSTEM/EXEC AMQSWLM SYS$SHARE:AMQSWLM
```

Il n'est pas nécessaire d'indiquer l'extension de fichier .EXE dans le nom logique.

Pour définir ce nom logique au démarrage, créez-le dans SYS\$MANAGER:MQS\_SYSTARTUP.COM.



---

## Annexe J. Applications sécurisées

Si vous attachez beaucoup d'importance aux performances de votre environnement, et que celui-ci est stable, vous pouvez alors considérer que vos applications utilisateur, canaux et programmes d'écoute sont "sécurisés", c'est-à-dire qu'ils utilisent l'édition des liens. (Le temps passé au traitement des appels MQPUT et MQGET de messages non persistants peut être réduit de jusqu'à 400% sur les systèmes OpenVMS.)

Dans une application sécurisée, l'application MQSeries et l'agent de gestion de la file d'attente locale font partie du même processus. L'application se connecte directement aux ressources du gestionnaire de files d'attente et devient en fait une extension de ce dernier. Cette option peut compromettre l'intégrité d'un gestionnaire de files d'attente puisqu'il n'existe aucune protection contre l'écrasement de la mémoire.

De plus, les applications partagées peuvent être amenées à créer certaines ressources telles que de la mémoire partagée. Il se peut que l'accès à ces ressources par un autre processus de files d'attente soit nécessaire ; elles doivent donc faire partie du même UIC. Les processus de gestionnaire de files d'attente s'exécutent tous dans un compte MQM. Par conséquent, les applications sécurisées doivent elles aussi être exécutées dans le cadre de ce compte.

Prenez en compte les points ci-dessous avant d'utiliser des applications sécurisées.

---

### Applications utilisateur

Il n'est pas nécessaire d'exécuter votre application directement depuis le compte MQM. Après connexion à un gestionnaire de files d'attente, MQSeries modifie automatiquement le profil de sécurité de l'unité d'exécution active afin que cette dernière identifie le compte MQM. L'unité d'exécution retrouve son identité naturelle après l'appel de déconnexion du gestionnaire de files d'attente.

Ayez conscience du fait qu'une application sécurisée connectée à un gestionnaire de files d'attente, est exécutée sous le compte MQM. S'il s'avère nécessaire de remplacer l'identité de l'unité d'exécution par une autre UIC pendant qu'elle est connectée à un gestionnaire de files d'attente, veillez à la redéfinir sur MQM avant de lancer l'appel MQI suivant.

### Configuration des applications sécurisées

Pour exécuter une application sécurisée sous MQSeries pour OpenVMS, paramétrez le type de liaison dans la zone Options de l'appel MQCONNX sur MQCNO\_FASTPATH\_BINDING. (Pour une liaison standard, utilisez l'option MQCNO\_STANDARD\_BINDING.) Si aucune option n'est indiquée (MQCNO\_NONE), la valeur par défaut utilisée est STANDARD\_BINDING.

Vous pouvez également utiliser le nom logique MQ\_CONNECT\_TYPE à la place du type de liaison indiqué pour l'appel MQCONNX. Si le nom logique est défini, vous devez avoir le choix entre FASTPATH et STANDARD pour le type de liaison. Toutefois, la liaison FASTPATH est utilisée uniquement si l'option de connexion est correctement indiquée dans l'appel MQCONNX. Ce nom logique permet

## Configuration des applications sécurisées

d'exécuter une application avec la liaison STANDARD\_BINDING en cas d'incident avec la liaison FASTPATH\_BINDING, sans avoir à reconstruire l'application.

Vous avez donc deux possibilités pour exécuter une application sécurisée :

- Spécifiez l'option MQCNO\_FASTPATH\_BINDING dans l'appel MQCONNX et définissez le nom logique MQ\_CONNECT\_TYPE sur FASTPATH

ou

- Spécifiez l'option MQCNO\_FASTPATH\_BINDING dans l'appel MQCONNX et ne précisez pas le nom logique MQ\_CONNECT\_TYPE.

Pour plus d'informations sur l'utilisation des applications sécurisées, consultez le manuel *MQSeries Intercommunication*.

---

## Exécution de canaux et de programmes d'écoute en tant qu'applications sécurisées

Les programmes de canaux lancés à l'aide de la commande **runmqsc start channel** s'exécutent sous le compte MQM. Il en est de même pour les programmes récepteurs de canal démarrés par des requêtes TCP entrantes (ou une connexion DECnet).

Les commandes **runmqchl** et **runmqslr** créent un processus autonome qui s'exécute sous le compte MQM. La combinaison du nom logique MQ\_CONNECT\_TYPE et du MQIBindType dans la strophe des canaux d'un fichier `qm.ini` d'un gestionnaire de files d'attente indique si un canal ou un programme d'écoute doit être exécuté de façon sécurisée.

Pour définir un canal ou un programme d'écoute sécurisé, appliquez l'une des méthodes suivantes :

- Indiquez MQIBindType=FASTPATH dans le fichier `qm.ini` et définissez le nom logique sur FASTPATH

ou

- Indiquez MQIBindType=FASTPATH dans le fichier `qm.ini` et ne complétez pas le nom logique.

## Messages non persistants rapides

L'attribut de canal NPMSPEED (vitesse de message non persistant) peut être utilisé pour préciser la vitesse à laquelle des messages non persistants doivent être envoyés. Vous pouvez indiquer une vitesse normale ou rapide. La vitesse par défaut est rapide, ce qui implique que les messages non persistants d'un canal doivent attendre le point de synchronisation avant de pouvoir être extraits. L'extraction de ces messages devient possible bien plus rapidement, mais ils peuvent se perdre en cas de panne de transmission ou d'arrêt du canal pendant que les messages sont en transit. Pour plus d'informations sur l'exécution de canaux et de programmes d'écoute en tant qu'applications sécurisées ainsi que sur les messages non persistants, consultez le manuel *MQSeries Intercommunication*.



---

## Annexe K. Informations utiles pour la configuration

Cette annexe comprend toutes les informations complémentaires dont vous pourrez avoir besoin pour configurer MQSeries pour Compaq OpenVMS.

Ces informations seront insérées dans les manuels concernés lors de leur prochaine mise à jour.

---

### Guide de programmation d'applications

Les informations relatives à la programmation sous OpenVMS seront modifiées pour prendre en compte le fait que les appels d'interface de file d'attente de messages (MQI) ne peuvent pas être réalisées depuis une routine AST.

Ceci est dû au fait que MQSeries utilise lui-même des routines AST et que celles-ci ne peuvent s'exécuter lorsqu'une autre routine AST est déjà active.

### Déclenchement d'application

Le fichier de commande MQTRIGGER.COM est fourni comme exemple du fichier de commande pour prendre les paramètres envoyés par le moniteur de déclenchement MQSeries (RUNMQTRM) et les répartir dans les zones de la structure MQTMC2.

Le fichier de commande s'attend à ce que le premier paramètre soit l'image ou le fichier de commande appelés avec certaines zones de la structure MQTMC2.

MQTRIGGER transmet les zones suivantes de la structure MQTMC2 à l'image ou au fichier de commande appelé :

Paramètre	Zone MQTMC2
1	QName
2	ProcessName
3	TriggerData
4	ApplType
5	UserData
6	QMgrName

### Exemples

1. Pour déclencher l'image amqsech :

La zone ApplcId de la définition de processus de déclenchement est spécifiée de la façon suivante :

```
APPLICID('@mqs_examples:mqtrigger $mqbin:amqsech')
```

## Informations utiles pour la configuration

Cet exemple suppose que le répertoire logique MQBIN a été défini comme suit :

```
SYS$SYSROOT:[SYSHLP.EXAMPLES.MQSERIES.BIN]
```

2. Pour appeler un fichier de commande dka200:[user]cmd.com:

La zone AppliId de la définition de processus de déclenchement est spécifiée de la façon suivante :

```
APPLICID('@mqs_examples:mqtrigger @dka200:[user]cmd')
```

---

## Annexe L. 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 Director of Licensing  
IBM Europe Middle-East Africa  
Tour Descartes  
La Défense 5  
2 avenue Gambetta  
92066 - Paris La Défense CEDEX  
FRANCE

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.

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.

## Remarques

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	NetView
CICS	OS/2
First Failure Support Technology	VSE/ESA
OS/390	BookManager

UNIX est une marque enregistrée dans certains pays, dont seule la société X/Open Limited Company peut concéder la licence.

DIGITAL, OpenVMS, Compaq, DecNet et Alpha sont des marques de Compaq Corporation.

Intel est une marque d'Intel.

Microsoft, Windows, et le logo Windows sont des marques de Microsoft.

MultiNet et TCPware sont des marques de Process Software.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

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 « gamme » MQSeries, concernent l'ensemble des produits MQSeries de Niveau 2. Les derniers produits MQSeries de Niveau 2 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 Tandem NonStop Kernel, version 2.2.0.1
- 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
- *Introduction à MQSeries*, 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*, 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
- *Guide de programmation d'applications*, SC11-1087
- *MQSeries Application Programming Reference*, SC33-1673
- *MQSeries Programming Interfaces Reference Summary*, SX33-6095
- *MQSeries Using 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 d'une plateforme pour chaque produit MQSeries.

#### MQSeries pour AIX, version 5.2

*MQSeries pour AIX, version 5.0 - Mise en route*, GC11-1213

#### MQSeries for AS/400, version 5.2

*MQSeries pour AS/400 V5.1 - Mise en route*, GC11-1588

*MQSeries for AS/400 V5.1 System Administration*, SC34-5558

*MQSeries for AS/400 V5.1 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 Tru64 UNIX, version 5.1

*MQSeries pour Compaq Tru64 UNIX, version 5.1 - Mise en route*, GC11-1697

#### MQSeries pour HP-UX, version 5.2

*MQSeries pour HP-UX, version 5.0 - Mise en route*, GC11-1215

## Bibliographie

### 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, version 5.0-*  
*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 pour R/3 Version 1.2 -*  
*Guide de l'utilisateur,* 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, version 5.0 -*  
*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 Tandem NonStop Kernel, version 2.2.0.1

*MQSeries for Tandem NonStop Kernel*  
*System Management Guide,* GC33-1893

### 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 pour Windows V2.0 - Guide de*  
*l'utilisateur,* GC11-1170

### MQSeries pour Windows, version 2.1

*MQSeries pour Windows V2.1 - Guide de*  
*l'utilisateur,* GC11-1289

### MQSeries pour Windows NT et Windows 2000, version 5.2

*MQSeries pour Windows NT, version 5.0*  
*- 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

- *Compaq OpenVMS Performance Management*, janvier 1999  
 Ce manuel fournit les informations nécessaires à l'optimisation des performances des systèmes sous OpenVMS.
- *Compaq OpenVMS System Management Utilities 2 volumes*, janvier 1999  
 Ces manuels contiennent des informations de référence sur les utilitaires de gestion système pour OpenVMS.
- *Character Data Representation Library, Character Data Representation Architecture, Reference and Registry*, SC09-2190-00  
 Ce manuel de référence présente l'architecture CDRA (Character Data Representation Architecture) ainsi que les éléments qui la composent.
- *DecNet SNA Gateway for Synchronous Transport Installation (OpenVMS)*, novembre 1993  
 Ce manuel fournit des instructions d'installation et de configuration pour la passerelle DecNet SNA Gateway.
- *Digital SNA APPC/LU6.2 Programming Interface for OpenVMS*, mai 1996  
 Ce manuel fournit des instructions d'installation et de configuration pour SNA APPC/LU6.2.
- *Digital TCP/IP Services for OpenVMS Installation and Configuration*, janvier 1999  
 Ce manuel fournit des instructions d'installation et de configuration pour Digital TCP/IP.
- *Guidelines for OpenVMS Cluster Configurations*, janvier 1999

## Publications connexes

Ce manuel explique comment optimiser la disponibilité et l'évolutivité d'un cluster OpenVMS.

- *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.

---

## 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, 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*.

**adaptateur** : Interface entre MQSeries pour OS/390 et TSO, IMS, CICS, ou des espaces adresse par lot. Un adaptateur est un élément de liaison permettant aux applications d'accéder aux services MQSeries.

**adresse relative en octets (RBA)** : Adresse en octets d'un enregistrement ou d'un intervalle de contr<sup>o</sup>le par rapport au début de l'espace mémoire alloué au programme auquel il appartient.

**afficheur d'événements** : Outil fourni avec Windows NT, permettant d'examiner et de gérer les fichiers journaux.

**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.

**allié** : Espace adresse OS/390 connecté à MQSeries pour OS/390.

**analyse et suppression d'image-mémoire (DAE)** : Service OS/390 permettant à un programme d'installation de supprimer les images-mémoire SVC et ABEND SYSUDUMP qui ne sont plus utiles car elles reproduisent des images-mémoire déjà écrites.

**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. S'oppose à *validation*.

**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*).

**appel du superviseur (SVC)** : Instruction OS/390 qui interrompt un programme en cours d'exécution et passe la main au superviseur, pour que ce dernier exécute le service spécifié par l'instruction.

**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*. Type d'arrêt de l'adaptateur CICS dans le cadre duquel ce dernier se déconnecte de MQSeries seulement après l'achèvement de toutes les tâches actives en cours. S'oppose à *arrêt forcé*.

**arrêt contr<sup>o</sup>lé** : Voir *arrêt avec mise au repos*.

**arrêt forcé** : Type d'arrêt de l'adaptateur CICS dans le cadre duquel ce dernier se déconnecte immédiatement de MQSeries pour OS/390, quel que soit l'état des tâches actives. S'oppose à *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*.

**ASID** : Identificateur d'espace adresse.

**attribut** : Élément définissant les caractéristiques d'un objet MQSeries.

## B

**bloc de contrôle des tâches (TCB)** : Bloc de contrôle OS/390 utilisé pour communiquer des informations sur les tâches d'un espace adresse qui sont connectées à un sous-système OS/390 tel que MQSeries pour OS/390 ou CICS.

**BMS** : Sigle de Basic Mapping Support (voir *module de structuration des entrées-sorties*).

**BSDS** : Jeu de données d'amorçage.

## 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. Également 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.

**CI** : Sigle de Control Interval (voir *intervalle de contrôle*).

**CL** : Sigle de Control Language (voir *langage de commande*).

**classe d'archivage** : Sous MQSeries pour OS/390, définit l'ensemble de pages devant contenir les messages d'une file d'attente donnée. La classe d'archivage est définie lors de la configuration de la file d'attente.

**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 MQSeries*.

**client MQSeries** : Partie de MQSeries pouvant être installée sur un système sans installer la totalité du gestionnaire de files d'attente. Le client MQSeries accepte les appels MQI provenant des applications et communique avec un gestionnaire de files d'attente sur un 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.

**code de fin anormale** : Code hexadécimal sur 4 octets qui désigne de manière unique un incident subi par MQSeries pour OS/390. Une liste complète des codes de fin anormale de MQSeries pour OS/390 complétée,

pour chacun d'eux, de l'explication correspondante, est fournie dans le manuel *MQSeries for OS/390 Messages and Codes*.

**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 de contr"le** : Sous MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, commande pouvant être tapée interactivement à partir de la ligne de commande du système d'exploitation. Ce type de commande nécessite uniquement l'installation de MQSeries. Il n'est pas nécessaire d'installer un programme spécial d'exécution.

**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 ou d'une connexion TCP/IP.

**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 administrateur** : Commandes MQSeries permettant de gérer les objets MQSeries (tels que des files d'attente, des processus et des listes de noms).

**commandes de contr"le 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)*.

**commutation de tâches** : Chevauchement des opérations d'E-S et du traitement de plusieurs tâches. Dans MQSeries pour OS/390, le commutateur de tâches optimise les performances en autorisant certains appels MQI à s'exécuter dans le cadre de sous-tâches plut"t que dans celui du bloc de contr"le des tâches (TCB) principal de CICS.

**connecter (se)** : 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.

**connexion différée** : Événement en instance, activé lorsqu'un sous-système CICS tente de se connecter à MQSeries pour OS/390 avant que MQSeries pour OS/390 ait démarré.

**consignation circulaire** : 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 journaux. La consignation est effectuée dans le premier fichier de la chaîne, puis, une fois ce dernier plein, le système passe au suivant. La procédure se poursuit jusqu'à ce que tous les fichiers soient pleins. La consignation est alors à nouveau effectuée dans le premier fichier de la chaîne si de l'espace y a été libéré ou si les enregistrements qu'il contient ne sont plus utiles. Cette technique est utilisée durant la reprise de redémarrage et permet d'utiliser les fichiers journaux pour retrouver et annuler les transactions en cours lors de l'arrêt du système. S'oppose à *consignation linéaire*.

**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*.

**consignation simple** : Méthode d'enregistrement des activités MQSeries pour OS/390 dans laquelle chaque modification est placée dans un seul fichier. S'oppose à *double consignation*.

**contexte** : Informations concernant l'origine d'un message.

**contr"le d'autorisation** : Contr"les de sécurité effectués lorsqu'un utilisateur essaye 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.

**CPF** : Sigle de Command Prefix (voir *préfixe de commande*).

**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

**DAE** : Sigle de Dump Analysis and Elimination (voir *analyse et suppression d'image-mémoire*).



**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*).

**déchargement** : Dans MQSeries pour OS/390, processus automatique grâce auquel le journal actif d'un gestionnaire de files d'attente est transféré vers son archive.

**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.

**DLQ** : Sigle de Dead-Letter Queue (voir *file d'attente de rebut*).

**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*.

**double consignation** : Méthode d'enregistrement des activités MQSeries pour OS/390 consistant à enregistrer chaque modification en double, de telle manière que si un redémarrage est nécessaire et que l'un des jeux de données est illisible, il soit possible d'utiliser le second. S'oppose à *consignation linéaire*.

**DQM** : Sigle de Distributed Queue Management (voir *gestion répartie des files d'attente*).

## E

**éditeur de la base de registres** : Dans Windows NT, programme permettant l'édition de la base de registres.

**enregistrement d'annulation/rétablissement** : Enregistrement de consignation utilisé lors d'un processus de reprise. La partie rétablissement de l'enregistrement décrit une modification qui doit être apportée à un objet MQSeries, tandis que sa partie annulation décrit comment annuler la modification si les données n'ont pas encore été validées.

**ensemble de pages** : Ensemble de données VSAM utilisé lorsque MQSeries pour OS/390 déplace des données (par exemple, files d'attente et messages) de mémoires tampons en mémoire principale vers une mémoire auxiliaire permanente (disque).

**en-tête d'événement** : Dans un message d'événement, partie des données identifiant le type du code anomalie de l'événement.

**environnement** : Voir *environnement d'application*.

**environnement d'application** : Logiciels accessibles par un programme d'application. Sous OS/390, CICS et IMS sont des environnements d'application.

**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 local** : Sur les systèmes UNIX, sous-ensemble de l'environnement utilisateur composé de conventions propres à une culture ou à un pays donné, telles que le format de l'heure, de la date, des nombres, des devises, les critères de discrimination, de classement et de conversion des caractères. Le CCSID du gestionnaire de files d'attente est dérivé de l'ID utilisateur qui l'a créé dans l'environnement local.

**ESM** : Sigle de External Security Manager (voir *gestionnaire de sécurité externe*).

**espace adresse** : Zone de mémoire virtuelle disponible pour un travail particulier.

**espace adresse allié** : Voir *allié*.

**ESTAE** : Sigle de Extended Specify Task Abnormal Exit (voir *spécification de tâche sur fin anormale*).

**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. Exemple : une file d'attente est indisponible.
- 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, des événements outils sont disponibles pour la gestion des définitions de ressources des gestionnaires de files d'attente, des performances et des canaux. Ils peuvent être utilisés par un programme d'analyse écrit par l'utilisateur dans une application d'administration pour afficher les événements sur l'écran de l'opérateur système. Ils permettent également aux applications de jouer un rôle d'agents pour les autres réseaux d'administration en gérant les rapports et en générant 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** : Dans MQSeries pour UNIX, fichier contenant les définitions de sécurité d'un objet, d'une classe d'objets ou de toutes les classes d'objets.

**fichier de configuration** : Sous MQSeries pour UNIX, MQSeries pour OS/2 Warp, et MQSeries pour Windows NT et Windows 2000, fichier contenant les informations nécessaires à la configuration, par exemple, des journaux, des communications ou des fonctions optionnelles. Synonyme de *fichier .ini*. Voir aussi *strophe*.

**fichier de contrôle du journal** : Sous MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, fichier contenant les informations nécessaires à la gestion des fichiers journaux (par exemple, taille et emplacement, nom du fichier disponible suivant).

**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*.

**fichier journal** : Sous MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, fichier dans lequel sont consignées les modifications importantes apportées aux données gérées par un gestionnaire de files d'attente. MQSeries alloue des fichiers journaux secondaires lorsque les fichiers principaux sont pleins.

**fichier journal de reprise** : Dans MQSeries pour OS/390, fichier contenant des informations nécessaires à la récupération des messages, des files d'attente et du sous-système MQSeries. MQSeries pour OS/390 écrit chaque enregistrement dans un fichier de données appelé *fichier journal actif*. Lorsque ce dernier est plein, son contenu est transféré dans un fichier sur disque ou sur bande appelé *journal d'archivage*. Synonyme de *journal*.

**FIFO** : Sigle de First-In-First-Out (voir *premier entré, premier sorti*).

**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 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** : File d'attente locale créée lors de l'ouverture d'un objet file d'attente modèle par un programme. 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 d'autorisation d'accès (SAF)** : Fonction OS/390 par le biais de laquelle MQSeries pour OS/390 communique avec un gestionnaire de sécurité externe tel que RACF.

**fonction de contr<sup>le</sup> 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 contr<sup>le</sup> des canaux.

**fonction de trace** : Sous MQSeries, fonction permettant d'enregistrer les activités MQSeries. Les

traces peuvent être placées dans GTF et dans les fonctions de gestion du système (SMF). Reportez-vous également à *trace globale* et *trace de performance*.

**fonction de trace générale (GTF)** : Programme OS/390 qui enregistre les événements système significatifs (par exemple, interventions du superviseur et opérations d'E-S) pour faciliter le diagnostic des incidents.

**fonction interactive de productivité du système (ISPF)** : Programme IBM tenant lieu de gestionnaire de dialogue et d'éditeur plein écran. Ce programme permet de rédiger des programmes d'application, de générer des écrans standard et des boîtes de dialogue interactives entre le programmeur d'application et l'utilisateur du terminal.

**fonctions optionnelles** : Sous MQSeries pour UNIX, 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*.

**FRR** : Sigle de Functional Recovery Routine (voir *sous-programme de reprise fonctionnelle*).

## G

**GCPC** : Sigle de Generalized command preprocessor (voir *préprocesseur général de commandes*).

**gestionnaire d'autorisation externe (ESM)** : Utilitaire de sécurité appelé par l'outil OS/390 System Authorization. RACF est un ESM.

**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 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 de fin et de reprise (RTM)** : Programme traitant toutes les fins normales et anormales de tâches en passant la main à un sous-programme de reprise lié à la fonction de fin.

**gestionnaire de ressources** : Application, programme ou transaction qui gère l'accès aux ressources partagées telles que les mémoires tampons et les fichiers. MQSeries, CICS et IMS sont des gestionnaires de ressources.

**gestionnaire des droits d'accès aux objets (OAM)** : Sous MQSeries pour UNIX, MQSeries pour AS/400 et MQSeries pour Windows NT et Windows 2000, fonction de définition des droits d'accès par défaut aux commandes et aux 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.

**GTF** : Generalized Trace Facility (voir *fonction de trace générale*).

## I

**ID de session** : Dans MQSeries pour OS/390, identificateur unique et propre à CICS qui désigne la liaison qu'un agent de transfert de messages doit utiliser pour transférer des messages d'une file d'attente de transmission à la liaison.

**ID de transaction** : Dans CICS, nom spécifié lors de la définition d'une transaction et utilisé ensuite pour appeler cette dernière.

**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'espace adresse (ASID)** : Identificateur unique affecté par le système à un espace adresse.

**identificateur d'objet** : Identificateur ou marque par le biais duquel un programme accède à un objet MQSeries qu'il doit utiliser.

**image de support** : Dans MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, séries d'enregistrements de fichier journal contenant l'image d'un objet, qui peut ainsi être recréé si nécessaire.

**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 (TMI)** : 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. 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.

**interruption sur erreur machine** : Interruption due à un incident ou un dysfonctionnement lié à un

équipement. Une interruption sur erreur machine est soit remédiable par matériel, soit remédiable par logiciel, soit irrémédiable.

**intervalle de contr"le** : Zone à longueur fixe située dans la mémoire à accès direct dans laquelle VSAM stocke les enregistrements et crée les espaces libres distribués. L'intervalle de contr"le est l'unité d'information que VSAM transmet à la mémoire à accès direct ou reçoit de cette dernière.

**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.

**IPCS** : Sigle de Interactive Problem Control System (voir *système interactif de maintenance du logiciel*).

**ISPF** : Sigle de Interactive System Productivity Facility (voir *fonction interactive de productivité du système*).

**jeu de données d'amorçage BSDS** : Jeu de données VSAM contenant :

- Une liste de tous les fichiers journaux actifs et d'archivage sous MQSeries pour OS/390.
- Une liste en boucle de toutes les activités récentes sous MQSeries pour OS/390.

Le jeu de données BSDS est nécessaire au redémarrage de MQSeries pour OS/390. Il est également appelé fichier d'amorçage BSDS.

**jeu de données d'initialisation en entrée** : Jeu de données utilisé par MQSeries pour OS/390 lors du démarrage.

**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.

**journal actif** : Voir *journal de reprise*.

**journal Application** : Sous Windows NT, journal consignait les événements importants liés aux applications.

**journal d'archivage** : Voir *journal de reprise*.

**journal des événements** : Voir *journal Application*.

## L

**langage de commande (CL)** : Sous MQSeries pour AS/400, langage qui permet de lancer des commandes à partir de la ligne de commande ou en écrivant un programme CL.

**liste de noms** : Objet MQSeries contenant une liste de noms (de files d'attente, par exemple).

## M

**MCA** : Sigle de Message Channel Agent (voir *agent MCA*).

**MCI** : Sigle de Message Channel Interface (voir *interface MCI*).

**mémoire tampon du fichier journal en sortie** : Sous MQSeries pour OS/390, mémoire tampon contenant les enregistrements du fichier journal de reprise avant leur écriture dans le journal d'archivage.

**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. S'oppose à *message de réponse* et à *message de rapport*.

**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. S'oppose à *message de réponse* et à *message de demande*.

**message de réponse** : Message utilisé pour répondre à un message de demande. S'oppose à *message de demande* et à *message de rapport*.

**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 *messaging synchrone*.

**messaging sans contrainte de temps** : Voir *messaging asynchrone*.

**messaging 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 *messaging asynchrone*.

**message sauvegardé sur disque** : Message écrit en mémoire auxiliaire (c'est-à-dire sur disque dur) afin d'éviter la perte en cas d'incident système. Voir aussi *message persistant*.

**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.

**mode double** : Voir *double consignment*.

**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 d'alertes** : Dans MQSeries pour OS/390, il s'agit d'un composant de l'adaptateur CICS qui traite les événements non prévus qui découlent des demandes de connexion à MQSeries pour OS/390.

**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 (voir *interface de file d'attente de messages*).

**MQSC** : Commandes MQSeries.

**MQSeries** : Famille de programmes IBM permettant de gérer des files d'attente de messages.

## N

**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.

**notification** : Sous MQSeries pour OS/390 et MQSeries pour Windows 2.1, fonction qui permet au système d'exploitation d'avertir un programme de l'arrivée d'un message en file d'attente.

**notification de fin** : Événement en instance, activé lorsqu'un sous-système CICS se connecte avec succès à MQSeries pour OS/390.

**NSI** : Sigle de Name Service Interface (voir *interface de service annuaire*).

**NTFS** : Sigle de New Technology File System (voir *Système de fichiers de technologie nouvelle*).

**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

**OAM** : Sigle de Object Authority Manager (voir *gestionnaire des droits d'accès aux objets*).

**objet** : Sous MQSeries, on appelle objet un gestionnaire de files d'attente, une file d'attente, une définition de processus, un canal, une liste de noms ou une classe d'archivage (OS/390 uniquement).

**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.

**parcourir** : 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 *récupérer*.

**PCF** : Sigle de Programmable Command Format (voir *format de commande programmable*).

**percolation** : Dans le cadre de la reprise sur incident, passage de la main, selon un chemin préétabli, d'un sous-programme de reprise à un sous-programme de reprise de plus haut niveau.

**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 reprise** : Sous MQSeries pour OS/390, terme utilisé pour décrire un ensemble de copies de sauvegarde d'ensembles de pages MQSeries pour OS/390 et les fichiers journaux correspondants nécessaires pour reconstituer ces ensembles. Ces copies constituent un point de reprise potentiel en cas de perte de l'ensemble de pages (par exemple, erreur d'E-S de l'ensemble de pages).

**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.

**pool de mémoires tampons** : Zone de la mémoire principale utilisée pour les définitions des files d'attente, des messages et des objets pool de mémoires tampons. Voir aussi *ensemble de pages*.

**préfixe de commande (CPF)** : Dans MQSeries pour OS/390, chaîne de caractères identifiant le gestionnaire de files d'attente vers lequel les commandes MQSeries pour OS/390 sont envoyées et duquel proviennent les messages opérateur MQSeries pour OS/390.

**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. (A)

**préprocesseur général de commandes (GCPC)** : Composant MQSeries pour OS/390 traitant et exécutant les commandes MQSeries.

**principal** : Sous MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, terme utilisé pour un ID utilisateur. Utilisé par le gestionnaire des droits d'accès aux objets pour vérifier les droits d'accès aux ressources système.

**processeur de commandes** : Composant MQSeries chargé de traiter les commandes.

**profil de commutation** : Dans MQSeries pour OS/390, profil RACF utilisé lors du démarrage de MQSeries ou lors de l'émission d'une commande d'actualisation de la sécurité. Chaque profil de commutation détecté par MQSeries entraîne la désactivation du contrôle de la ressource spécifiée.

**programme d'écoute** : Dans le cadre de la mise en files d'attente réparties, programme qui gère les informations concernant les connexions réseau entrantes.

**programme de transmission** : Voir *agent MCA*.

**PTF** : Sigle de Program Temporary Fix (voir *modification provisoire du logiciel*).



## Q

**qualificatif de haut niveau de bibliothèque destinataire (thlqual)** : Qualificatif de haut niveau pour les noms d'ensembles de données OS/390 cibles.

## 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*).

**recupérer** : 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*.

**registre** : Dans Windows NT, base de données sécurisée centralisant les informations de configuration du système et des applications.

**Registre de la ruche** : Sous Windows NT, structure des données stockées dans la base de registres.

**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.

**RESLEVEL** : Sous MQSeries pour OS/390, option qui contrôle le nombre d'ID utilisateur CICS dont les droits d'accès aux ressources API sous MQSeries pour OS/390 sont vérifiés.

**ressource** : Toute fonctionnalité du système de calcul ou d'exploitation nécessaire à l'exécution d'un travail ou d'une tâche. Sous MQSeries pour OS/390, les ressources peuvent être des pools de mémoires tampons, des fichiers de pagination, des fichiers journaux, des files d'attente et des messages.

**resynchronisation** : Dans MQSeries, option permettant de demander à un canal de démarrer et de convertir tout message de statut en attente de validation sans pur autant relancer la transmission des messages.

**RRS** : Sigle de Resource Recovery Services (voir *services de reprise de ressources*).

**RTM** : Sigle de Recovery Termination Manager (voir *gestionnaire de fin et de reprise*).

## S

**SAF** : Sigle System Authorization Facility (voir *fonction d'autorisation d'accès*).

**SDWA** : Sigle de System Diagnostic Work Area (voir *zone de travail des diagnostics système*).

**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.

**service annuaire** : Dans MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, utilitaire qui détermine le nom du gestionnaire propriétaire d'une file d'attente déterminée.

**service d'autorisation** : Sous MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, service vérifiant les droits d'accès d'un ID utilisateur associé à une commande ou à un appel MQL.

**service d'identification (UIS)** : Sous MQSeries pour OS/2 Warp, outil permettant aux applications MQI d'associer un ID utilisateur, autre que l'ID par défaut, aux messages MQSeries.

**services de reprise de ressources (RRS)** : Outil OS/390 permettant la prise en charge des points de synchronisation en 2 phases entre des gestionnaires de ressources participants.

**SIT** : Sigle de System Initialization Table (voir *table d'initialisation du système*).

**sous-programme de reprise fonctionnelle (FRR)** : Gestionnaire OS/390 de reprise et d'interruption permettant à un sous-programme de reprise de reprendre la main en cas d'interruption du programme.

**sous-système** : Sous OS/390, groupe de modules dépendants de OS/390. MQSeries pour OS/390 est un exemple de sous-système OS/390.

**spécification de tâche étendue sur fin anormale (ESTAE)** : Macro OS/390 apportant une capacité de reprise et donnant la main à l'exit spécifié, afin d'effectuer un traitement, de faire le diagnostic d'une fin anormale ou de spécifier une adresse de relance.

**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 pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, un fichier de configuration (.ini) peut contenir de nombreuses strophes.

**SVC** : Sigle de Supervisor Call (voir *appel du superviseur*).

**SYS1.LOGREC** : Aide au service contenant des informations sur les erreurs de programme et de matériel.

**système de fichiers de technologie nouvelle (NTFS)** : Système de fichiers de Windows NT assurant la sécurisation des fichiers.

**système interactif de maintenance du logiciel (IPCS)** : Composant de OS/390 qui permet la gestion en ligne des incidents, le diagnostic interactif, le débogage en ligne d'images-mémoire sur fin anormale conservées sur disque, le suivi et le relevé des incidents.

## 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.

**table d'initialisation du système (SIT)** : Table contenant les paramètres que doit utiliser CICS au démarrage.

**TACL** : Sigle de Tandem Advanced Command Language.

**TCB** : Sigle de Task Control Block (voir *bloc de contrôle des tâches*).

**thlqual** : Sigle de Target library high-level qualifier (voir *qualificatif de haut niveau de bibliothèque destinataire*).

**TMI** : Trigger Monitor Interface (interface de moniteur de déclenchement).

**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.

**trace globale** : Option de trace MQSeries pour OS/390 dans le cadre de laquelle les données de trace proviennent de l'intégralité du sous-système MQSeries pour OS/390.

**tranid** : Voir *ID de transaction*.

**transformation du nom** : Dans MQSeries pour UNIX, MQSeries pour OS/2 Warp et MQSeries pour Windows NT et Windows 2000, processus interne modifiant le nom d'un gestionnaire de files d'attente de façon, si nécessaire, à ce qu'il soit valide et unique sur le système utilisé. Extérieurement, le nom du gestionnaire apparaît inchangé.

**type de canal de connexion client** : Type de définition de canal MQI associée à un client MQSeries. 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

**UIS** : Sigle de User Identifier Service (voir *service d'identification*).

**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 reprise en attente de validation** : Sous MQSeries, état d'une unité de reprise pour laquelle un point de synchronisation a été demandé mais n'a pas encore été confirmé.

**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.

**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** : Opération qui applique 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. S'oppose à *annulation*.

**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*.

**zone de travail des diagnostics système (SDWA)** : Données enregistrées dans une entrée SYS1.LOGREC et décrivant une erreur de programme ou de matériel.





# Index

## A

accès non autorisé (nécessité de protection) 81  
activation  
  événements 119  
  sécurité 85  
administration  
  droits 95  
  éloignée 67  
    canaux 68  
    files d'attente de transmission 68  
    objets 65  
  ensembles de commandes 21  
    commandes de contrôle 21  
    commandes MQSeries (MQSC) 22  
    commandes programmables (PCF) 23  
  locale 37  
administration éloignée  
  des objets 65  
  incidents 73  
  serveur de commandes 63  
administration locale 37  
affichage  
  attributs du gestionnaire de files d'attente 41  
  commande de sortie de trace formatée MQSeries 266  
  commandes de fichiers MQSeries 264  
  commandes de transactions MQSeries 268  
  définitions de processus 60  
  droit d'accès 259  
  état du serveur de commandes 63  
  identificateurs hexadécimaux des composants 353  
  pile d'unités d'exécution cibles 348  
  processus MQSeries actifs 349  
  serveur de commandes 263  
  table mémoire 352  
agent MCA  
  attribut AdoptNewMCA 186  
  canal en mode RETRY 188  
  description 66  
aide sur la syntaxe 245  
aide Windows 389  
algorithmes d'extraction de messages 9  
alias  
  files d'attente de réponse 77  
  gestionnaire de files d'attente 77  
alias de files d'attente de réponse 77  
amqsdllq (gestionnaire DLQ exemple) 104  
appels de gestionnaires de files d'attente  
  Voir aussi cluster 15  
application  
  asynchrone 7  
  connexion à un gestionnaire de files d'attente local 73  
  considération sur le développement 199

application (*suite*)  
  données 8  
  environnement client-serveur 17  
  erreurs de programmation (exemples) 196  
  prise en charge de l'administration MQI 37  
  sécurisée 379  
application sécurisée 379  
liaisons 379  
applications asynchrones 7  
arrêt  
  gestionnaire de files d'attente 33  
  contrôlé 33, 34  
  forcé 34  
  immédiat 34  
  mise au repos 33  
arrêt contrôlé 33  
arrêt d'un gestionnaire de files d'attente 34  
arrêt de la trace MQSeries 277  
Arrêt forcé du gestionnaire de files d'attente 34  
arrêt progressif (gestionnaire de files d'attente) 33  
attribut AdoptNewMCA 186  
attribut LIKE 50  
attribut ListenerBacklog 188  
attribut MaxMsgLength 10  
attribut REPLACE (commandes DEFINE) 44  
attributs  
  affichage du gestionnaire de files d'attente 41  
  attribut ALL 49  
  comparaison entre MQSC et PCF 23  
  files d'attente 11  
  gestionnaire de files d'attente  
    affichage 41  
    modification 42  
  modification 42, 50  
  valeur par défaut 49

## B

Bases de données relationnelles conformes à la norme XA 122  
binaire  
  fermeture d'un fichier binaire de trace 357  
  ouverture d'un fichier de trafic binaire 356  
BookManager 389

## C

canal  
  administration éloignée 68  
  affichage de l'historique des messages 349

canal (*suite*)  
  commande d'exécution de l'initialisateur 289  
  commande de démarrage 290  
  commandes 90  
  conditions de sécurité de la commande 90  
  connexion de l'unité d'exécution cible au 354  
  déconnexion de l'unité d'exécution cible du 355  
  définition 69  
  définition automatique 71  
  définition de canal émetteur 65  
  définition de canal récepteur 65  
  description 15, 65  
  droits des commandes d'échappement 95  
  événements 117  
  fastpath 187  
  identification parmi des gestionnaires 12  
  informations show channel (MONMQ) 347  
  lancement 70  
  mise en file d'attente éloignée 65  
  sécurisé 380  
  sécurité 90, 91  
  show mask (MONMQ) 348  
  strophe Channels (qm.ini) 186  
canal de connexion serveur (définition automatique) 71  
canal fastpath 187  
canal récepteur (définition automatique) 71  
caractère NL (conversion EBCDIC vers ASCII) 176  
CCSID 341  
  conversion de données 78  
  pris en charge par MQSeries pour Compaq OpenVMS 318  
  redémarrage du gestionnaire de files d'attente 79  
ccsid.tbl 78  
cellule, environnement DCE et files d'attente 171  
cliché  
  cliché des enregistrements d'un journal (commande dmpmqlog) 151  
  cliché du contenu d'un journal de reprise 151  
  commande de version formatée du journal système (dmpmqlog) 257  
clients 16  
  commande de lancement du moniteur de déclenchement 299  
  identification des incidents 214  
  liaison des applications 17  
  messages d'erreur sur DOS et Windows 215

- clients 16 (*suite*)
  - préparation des canaux 17
- cluster
  - attributs de la strophe
    - ExitProperties 177
  - de gestionnaires de files d'attente 9
  - description 66
  - gestionnaire de files d'attente
    - description 15
    - différence avec le cluster
      - OpenVMS 15
    - exit de pondération de charge 18, 377
    - file d'attente de transmission 13
    - utilisation des listes de noms 15
  - mise en file d'attente éloignée 65
  - OpenVMS
    - différence avec le cluster de
      - gestionnaire de files d'attente 221
      - groupe de reprise 222
      - installation de MQSeries 221
- clusters OpenVMS 221
- code anomalie 11, 103, 107
- code retour 194
  - commande crtmqcvx 248
  - commande crtmqm 252
  - commande dltnqm 254
  - commande dspmqaut 261
  - commande dspmqcsv 263
  - commande dspmqfls 265
  - commande dspmqtrn 268
  - commande endmqcsv 270
  - commande endmqslr 273
  - commande endmqm 275
  - commande endmqtrc 277
  - commande failover 279
  - commande rcdmqimg 283
  - commande rcrmobj 285
  - commande rsvmqtrn 287
  - commande runmqchi 289
  - commande runmqchl 290
  - commande runmqfm 293
  - commande runmqslr 294
  - commande runmqsc 297
  - commande runmqmtc 299
  - commande runmqtrm 300
  - commande setmqaut 305
  - commande strmqcsv 308
  - commande strmqm 309
  - commande strmqtrc 312
- codes retour
  - interprétation des 247
- commande analyse trace (MONMQ) 360
- commande close binary (MONMQ) 357
- commande close LU (MONMQ) 347
- commande close text (MONMQ) 357
- commande connect (MONMQ) 354
  - commande crtmqcvx
    - code retour 248
    - exemples 248
    - paramètres 248
  - commande crtmqm 250
    - code retour 252
    - commandes connexes 253
    - exemples 253
    - paramètres 250
- commande de lancement de la trace
  - MQSeries 311
- commande de reprise par transfert 223
- commande default variable (MONMQ) 346
- commande delete history (MONMQ) 358
- commande deselect index (MONMQ) 356
- commande disable history (MONMQ) 358
- commande disable timestamp (MONMQ) 357
- commande disable trace (MONMQ) 358
- commande disconnect (MONMQ) 355
- commande dltnqm 254
  - code retour 254
  - commandes connexes 256
  - exemples 254
  - paramètres 254
- commande dspmqaut 259
  - code retour 261
  - commandes connexes 262
  - exemples 262
  - paramètres 259
  - utilisation 85, 87
- commande dspmqcsv 263
  - code retour 263
  - commandes connexes 263
  - exemples 263
  - paramètres 263
- commande dspmqfls 264
  - code retour 265
  - exemples 265
  - paramètres 264
- commande dspmqtrc 266
  - commandes connexes 267
  - exemples 266
  - paramètres 266
- commande dspmqtrn 268
  - code retour 268
  - commandes connexes 269
  - paramètres 268
- commande enable history (MONMQ) 358
- commande enable timestamp (MONMQ) 357
- commande enable trace (MONMQ) 357
- commande endmqcsv 270
  - code retour 270
  - commandes connexes 272
  - exemples 270
  - paramètres 270
- commande endmqslr (arrêt du programme d'écoute)
  - code retour 273
  - format 273
  - objet 273
  - paramètres 273
- commande endmqm 33, 274
  - code retour 275
  - commandes connexes 276
  - exemples 275
  - paramètres 274
- commande endmqtrc 277
  - code retour 277
  - commandes connexes 277
- commande endmqtrc 277 (*suite*)
  - exemples 277
  - paramètres 277
- commande exit (MONMQ) 363
- commande failover 226, 278
  - code retour 279
  - commandes connexes 281
  - exemples 279
  - paramètres 278
- commande FFST (MONMQ) 363
- commande onstartup start (MONMQ) 354
- commande onstartup stop (MONMQ) 354
- commande open (MONMQ) 346
- commande open binary (MONMQ) 356
- commande open text (MONMQ) 357
- commande programmable (PCF)
  - administration with 23
  - attributs 23
  - commandes (description) 23
  - commandes PCF d'échappement 23
  - droits d'accès 91
  - sécurité requise 90
- commande quit (MONMQ) 363
- commande rcdmqimg 282
  - code retour 283
  - commandes connexes 283
  - exemples 283
  - paramètres 282
- commande rcrmobj 284
  - code retour 285
  - commandes connexes 286
  - exemples 285
  - paramètres 284
- commande rsvmqtrn 287
  - code retour 287
  - commandes connexes 288
  - paramètres 287
- commande runmqchi 289
  - code retour 289
  - paramètres 289
- commande runmqchl 290
  - code retour 290
  - paramètres 290
- commande runmqdlq 103
- commande runmqfm 223, 293
  - code retour 293
  - commandes connexes 293
  - exemples 293
  - paramètres 278, 293
- commande runmqslr (lancement du programme d'écoute)
  - code retour 294
  - exemple 295
  - format 294
  - objet 294
  - paramètres 294
- commande runmqsc 296
  - code retour 297
  - exemples 297
  - paramètres 297
  - redirection des données en entrée et en sortie 39
- commande runmqmtc 299
  - code retour 299
  - paramètres 299

- commande runmqtrm 300
  - code retour 300
  - paramètres 300
- commande select (MONMQ) 355
- commande set color (MONMQ) 360
- commande set depth (MONMQ) 358
- commande set free (MONMQ) 358
- commande set mask (MONMQ) 359
- commande set output (MONMQ) 360
- commande setmqaut 301
  - code retour 305
  - commandes connexes 307
  - exemples 306
  - paramètres 303
  - services optionnels 87
  - utilisation 85, 86
- commande show channels (MONMQ) 347
- commande show components (MONMQ) 353
- commande show events (MONMQ) 351
- commande show functions (MONMQ) 353
- commande show globals (MONMQ) 350
- commande show history (MONMQ) 349
- commande show mask (MONMQ) 348
- commande show memory (MONMQ) 352
- commande show mutex (MONMQ) 350
- commande show processes (MONMQ) 349
- commande show segment (MONMQ) 346
- commande show stack (MONMQ) 348
- commande strmqcsv 308
  - code retour 308
  - commandes connexes 308
  - exemples 308
  - paramètres 308
- commande strmqm 309
  - code retour 309
  - commandes connexes 310
  - exemples 310
  - paramètres 309
- commande strmqtrc 311
  - code retour 312
  - commandes connexes 313
  - exemples 312
  - paramètres 311
- commande trace start (MONMQ) 355
- commande trace stop (MONMQ) 355
- commandes
  - affichage d'une sortie de trace formatée MQSeries (dspmqtrc) 266
  - affichage des droits d'accès (dspmqaut) 259
  - affichage des fichiers MQSeries (dspmqfls) 264
  - affichage des transactions MQSeries (dspmqtrn) 268
  - affichage du serveur de commandes (dspmqcsv) 263
  - aide sur la syntaxe 245
  - arrêt d'un gestionnaire de files d'attente (endmqm) 274
  - arrêt d'un serveur de commandes (endmqcsv) 270
- commandes (*suite*)
  - arrêt de la trace MQSeries (endmqtrc) 277
  - commande d'arrêt du programme d'écoute (endmqslr) 273
  - commande d'exécution du programme d'écoute (runmqslr) 294
  - commande de cliché du journal (dmpmqlog) 257
  - commande programmable (PCF) 23
  - commandes de sécurité
    - dspmqaut 87
    - setmqaut 85
  - commandes MQSeries (MQSC) 22
  - comparaison de jeux de commandes 327
  - contrôle 21
  - conversion de données (crtmqcvx) 248
  - création d'un gestionnaire de files d'attente (crtmqm) 250
  - définition/réinitialisation des droits d'accès (setmqaut) 86, 301
  - démarrage d'un canal (runmqchl) 290
  - démarrage du gestionnaire de files d'attente de rebut (runmqdlq) 103
  - enregistrement d'une image de support (rcdmqimg) 282
  - exécution de l'initialisateur de canal (runmqchi) 289
  - exécution des commandes MQSeries (runmqsc) 296
  - fichiers de commandes MQSC
    - entrée 43
    - rapports de sortie 44
  - gestion de reprise (failover) 278
  - gestionnaire de files d'attente de rebut 291
  - lancement d'un gestionnaire de files d'attente (strmqm) 309
  - lancement d'un moniteur de reprise (runmqfm) 293
  - lancement d'un serveur de commandes (strmqcsv) 308
  - lancement de la trace MQSeries (strmqtrc) 311
  - lancement du moniteur de déclenchement (runmqtrm) 300
  - lancement du moniteur de déclenchement client (runmqtmc) 299
- MQSC
  - ALTER QLOCAL 50
  - ALTER QREMOTE 76
  - DEFINE CHANNEL 69
  - DEFINE QALIAS 55
  - DEFINE QLOCAL 50
  - DEFINE QLOCAL LIKE 50
  - DEFINE QLOCAL REPLACE 50
  - DEFINE QMODEL 57
  - DEFINE QREMOTE 74
  - DELETE QLOCAL 51
  - DELETE QREMOTE 76
  - DISPLAY QREMOTE 75
- commandes (*suite*)
  - MQSeries (MQSC)
    - utilisation 22
    - vérification 45
  - recréation d'un objet (rcrmqobj) 284
  - résolution des transactions MQSeries (rsvmqtrn) 287
  - runmqsc 39
  - suppression d'un gestionnaire de files d'attente (dlmqm) 254
- commandes de contrôle 21
  - distinction entre majuscules et minuscules 25
  - runmqsc 39
  - utilisation 27
- commandes MQSC
  - ALTER QLOCAL 50
  - ALTER QREMOTE 76
  - DEFINE CHANNEL 69
  - DEFINE QALIAS 55
  - DEFINE QLOCAL 50
  - DEFINE QLOCAL LIKE 50
  - DEFINE QLOCAL REPLACE 50
  - DEFINE QMODEL 57
  - DEFINE QREMOTE 74
  - DELETE QLOCAL 51
  - DELETE QREMOTE 76
  - DISPLAY QREMOTE 75
  - distinction entre majuscules et minuscules 26
  - émission interactive 39
  - longueur maximale des lignes 44
  - utilisation 22
- commandes opérateur (sans réponse) 197
- commandes PCF
  - automatisation des tâches d'administration à l'aide de commandes PCF 61
  - commandes PCF d'échappement 62
  - MQAI (utilisation à des fins de simplification) 62
- Commandes PCF
  - attributs dans les commandes MQSC et PCF 62
- commandes PCF d'échappement 23, 62
- commandes pour MQSeries 21
  - composant
    - show functions (MONMQ) 353
  - composant optionnel
    - gestionnaire des droits d'accès aux objets (OAM) 83
    - service annuaire 171
- configuration
  - base de données (qm.ini) 184
  - édition 173
  - exemple d'un fichier qm.ini 189
  - failover.ini 225
  - fichier de configuration du gestionnaire de files d'attente, qm.ini 175
  - gestionnaires de bases de données 123
  - groupe de reprise de cluster OpenVMS 225
  - implémentation des modifications 174

- configuration (*suite*)
  - logs 182
  - modification 173
  - mqs.ini (description) 174
  - Oracle 125
  - priorités 174
  - strophe AllQueueManagers (mqs.ini) 175
  - strophe Channels (qm.ini) 186
  - Strophe ClientExitPath, mqs.ini 177
  - Strophe DefaultQueueManager (mqs.ini) 177
  - strophe Exitpath (qm.ini) 189
  - strophe ExitProperties (mqs.ini) 177
  - strophe Log (qm.ini) 182
  - strophe LogDefaults, mqs.ini 178
  - strophe LU62 (qm.ini) 187
  - strophe QueueManager (mqs.ini) 180
  - strophe Service (qm.ini) 181
  - strophe ServiceComponent (qm.ini) 181
  - strophe TCP (qm.ini) 187
  - strophe XAResourceManager (qm.ini) 184
- considérations sur les performances
  - lors du traçage 208
- contrôle des gestionnaires de files d'attente 116
- conventions de dénomination (support de langue nationale) 243
- conversion de données 77
  - attribut ConvEBCDICNewline (strophe AllQueueManagers) 176
  - commande crtmqcvx 248
  - conversion de données par défaut 78
  - conversion de formats de message définis par l'utilisateur 79
  - conversion du caractère EBCDIC NL vers ASCII 176
- conversion de données par défaut 78
- conversion du caractère EBCDIC NL vers ASCII 176
- CorrelId (effet sur les performances) 200
- création
  - commande crtmqm 250
  - définitions de processus 59
  - files d'attente 11
  - gestionnaire de files d'attente 28, 32
  - objets 10

## D

- DCE
  - cellule 171
  - configuration 172
  - partage de files d'attente 171
  - sécurité 19
- débogage
  - contrôles secondaires 197
  - erreurs courantes de programmation 196
  - vérifications préalables 193
- déclenchement 381
  - définition 7
  - événements comparés aux événements outils 118

- déclenchement 381 (*suite*)
  - file d'attente d'application (définition) 58
  - files d'attente d'événements 118
  - gestion des objets 58
  - messages d'une file d'attente d'initialisation 13
  - moniteur
    - commande de lancement 300
    - description 13
    - utilisation des listes de noms 15
  - moniteur de reprise
    - commande de lancement 293
  - définition automatique de canaux 71
  - définition de canal émetteur 65
  - définition de canal récepteur 65
  - définition dynamique de canaux 71
  - définitions de processus
    - affichage 60
    - création 59
    - description 14
  - délai des réponses des commandes MQSC 72
  - démarrage
    - gestionnaire de files d'attente 32
    - GFA dans un groupe de reprise 228
  - désactivation des événements 119
  - désactivation du gestionnaire des droits d'accès aux objets 85
  - diagrammes linéaires (lecture) 244
  - Digital TCP/IP Services pour OpenVMS 227
  - distinction entre majuscules et minuscules 25
    - commandes de contrôle 25
    - commandes MQSC 26
    - noms de gestionnaires de files d'attente 24
  - droits
    - administration 95
  - droits d'accès
    - commande de définition ou réinitialisation 301
    - commande dspmqaut 87
    - commande setmqaut 87
    - commandes 87
    - contexte 89
    - identificateurs de droits 85
    - listes 86
    - MQI 92
    - services optionnels 87
    - utilisateur secondaire 89
  - droits d'accès d'utilisateur secondaire 89
  - droits d'accès du groupe primaire 84
  - droits d'accès MQOPEN 92
  - droits d'accès MQPUT 92
  - droits de contexte 89
  - droits de traitement 84

## E

- édition de liens 379
- éloignée
  - administration 67
  - définition d'une file d'attente (création) 73
  - émission de commandes MQSC 71

- éloignée (*suite*)
  - gestion de la sécurité 90
  - mise en file d'attente
    - recommandations 73
  - objet de file d'attente (utilisation) 77
- éloignées
  - files d'attente
    - alias de files d'attente de réponse 77
    - en tant que gestionnaire de files d'attente alias 77
  - émission de commandes MQSeries 38
  - en-tête de rebut (MQDLH) 103
  - ensemble de commandes
    - administration 21
  - environnement d'exploitation 317
  - environnement local 341
  - équilibre de la charge de travail
    - redirection des appels MQI 55
  - erreurs de programmation (exemples) 196
  - événement outil
    - activation 119
    - description 115
    - messages 119
    - types 117
    - utilisation 116
  - événements
    - canal 117
    - déclenchement 118
    - files d'attente 118
    - outil
      - activation et désactivation 119
      - définition 115
      - description 115
      - message 119
      - types 117
      - utilisation 116
    - show events (MONMQ) 351
    - types 117
  - événements Performances 117
  - exemple
    - données de trace 209
  - exemples
    - commande crtmqcvx 248
    - commande crtmqm 253
    - commande dltnmqm 254
    - commande dspmqaut 262
    - commande dspmqcvs 263
    - commande dspmqfls 265
    - commande dspmqtrc 266
    - commande endmqcvs 270
    - commande endmqm 275
    - commande endmqtrc 277
    - commande failover 279
    - commande rcdmqimg 283
    - commande rcrmqobj 285
    - commande runmqfm 293
    - commande runmqfsr 295
    - commande runmqsc 297
    - commande setmqaut 306
    - commande strmqcvs 308
    - commande strmqm 310
    - commande strmqtrc 312
    - création d'une file d'attente de transmission 76
    - erreurs de programmation 196

- exemples (*suite*)
  - fichier mqseries.ini (MQSeries pour Compaq OpenVMS) 189
  - fichier qm.ini 189
  - journal des erreurs 205
- exigences
  - logiciel 317
  - matériel 317
- exit
  - exit de canal 18
  - exit de pondération de charge du cluster 18
  - exit utilisateur 18
  - pondération de charge du cluster 377
- exit de pondération de charge du cluster 377
- exit utilisateur 377
  - description 18
  - exit de canal 18
  - exit de conversion de données 18
  - pondération de charge du cluster 18, 377

## F

- failover.template 235
- FFST
  - FFST (MONMQ) 363
- FFST (consultation) 209
- fichier de configuration de failover.ini 235
- fichier de configuration failover.ini
  - édition 225
- fichier de configuration qm.ini
  - définition 175
  - modification 173
  - priorités 174
  - strophe Channels 186
  - strophe Exitpath 189
  - strophe Log 182
  - strophe LU62 187
  - strophe Service 181
  - strophe ServiceComponent 181
  - strophe TCP 187
  - strophe XAResourceManager 184
- fichier MODPARAMS.DAT 217
- fichiers
  - configuration
    - identification des incidents 208
    - configuration de MQSeries 174
    - configuration du gestionnaire de files d'attente 175
  - contrôle de journal 136
  - droits d'accès
    - accès aux 101
    - chemins 98
    - classe 100
    - contenu 99
    - gestion 101
    - informations 98
    - toutes les classes 101
  - signification des noms 23
- fichiers de commandes 43
- fichiers de commutation (création) 124
- fichiers de configuration
  - failover.template 333
- fichiers de configuration (*suite*)
  - gestionnaire de files d'attente (qm.ini)
    - contenu 175
    - désactivation du gestionnaire des droits d'accès aux objets 85
    - sommaire 189
    - strophe Channels 186
    - strophe ExitPath 189
    - strophe Log 182
    - strophe Service 181
    - strophe ServiceComponent 181
    - strophe XAResourceManager 184
    - strophes LU62 et TCP 187
  - groupe de reprise de cluster
    - OpenVMS 225
  - modification 173
  - MQSeries (mqseries.ini)
    - chemin 46
    - contenu 174
    - strophe AllQueueManagers 175
    - Strophe ClientExitPath 177
    - strophe
      - DefaultQueueManager 177
      - strophe ExitProperties 177
      - strophe LogDefaults 178
      - strophe QueueManager 180
- fichiers de droits d'accès
  - accès 101
  - chemins 98
  - classe 100
  - gestion 101
  - informations 98
  - répertoires 98
  - toutes les classes 101
- Fichiers de droits d'accès
  - contenu 99
- fichiers journaux 204
- file d'attente cible
  - définition 74
  - messages non transmis 103
  - nom du gestionnaire de files d'attente propriétaire 75
- file d'attente d'initialisation
  - définition 59
  - description 13
- file d'attente de commandes 14
- file d'attente de messages non transmis 103
- file d'attente de rebut (DLQ)
  - définition 30
  - description 14
  - gestionnaire, voir aussi Gestionnaire DLQ 103
  - lancement du gestionnaire de files d'attente de rebut (runmqdlq) 291
- file d'attente de réponses 14
- file d'attente de transmission
  - administration éloignée 68
  - cluster 13
  - création 76
  - définition 69
  - description 13, 66
  - identification dans un gestionnaire 12
  - indication du nom 75
  - valeur par défaut 13, 31, 76
- file d'attente de transmission de cluster
  - description 13
- file d'attente de transmission par défaut 76
- file d'attente des événements 14
- files
  - objets
    - modèle 12
- files d'attente
  - alias 12
  - alias (manipulation) 55
  - application (définition pour le déclenchement) 58
  - attributs 11
    - modification 50
  - commande 14
  - définition 11
  - description 8
  - droits d'accès 88
  - dynamiques 9
  - éloignées 11, 12
    - création 73
    - gestionnaire de files d'attente
      - alias 77
      - utilisation 77
  - événement 14
  - file d'attente de réponses 14, 77
  - incident 198
  - initialisation
    - définition 59
    - messages de déclenchement 13
  - locales 11, 12
    - copie 50
    - définition 48
    - nettoyage 51
    - suppression 51
    - valeur par défaut 16
  - manipulation 47
  - message non transmis 14
    - définition 30
  - modèle 12
    - définition 57
    - manipulation 56
  - notification des événements 118
  - objets
    - alias 12
    - éloignés 12
    - locales 12
  - partagées entre plusieurs gestionnaires de files d'attente 171
  - pour les applications MQSeries 37
  - prédéfinies 9
  - rebut 14
    - définition 30
  - réparties (sortie incorrecte) 203
  - tâches de configuration
    - partagées 171
  - temporaires 9
  - transmission 13
    - administration éloignée 68
    - création 76
    - définition 69
    - valeur par défaut 31, 76
  - transmission de cluster 13
  - visualisation 51
- files d'attente alias 55
- description 12



- files d'attente alias 55 (*suite*)
  - droits d'accès 88
- files d'attente dynamiques 9
  - description 9
  - droits d'accès 88
- files d'attente éloignées
  - description 11, 12
  - droits d'accès 88
- files d'attente locales
  - commande 14
  - copie de définitions 50
  - définition 48
  - description 11, 12
  - initialisation 13
  - message non transmis 14
  - nettoyage 51
  - rebut 14
  - suppression 51
  - transmission 13
- files d'attente modèles
  - définition 57
  - description 12
  - manipulation 56
- files d'attente partagées
  - tâches de configuration 171
- files d'attente permanentes 9
- files d'attente prédéfinies 9
- files d'attente temporaires 9
- fin des commandes interactives sur MQSC 41
- fonctions
  - affichage dans un composant 353
- format PDF (Portable Document Format) 388
- format PostScript 389
- formats de messages définis par l'utilisateur 79

## G

- gestion de l'accès 83
- gestion de la mémoire partagée 364
- gestion des fichiers journaux 143
- gestion des groupes de reprise 278
- gestion des objets pour le déclenchement 58
- gestionnaire de files d'attente
  - administration éloignée 65
  - administration locale 37
  - alias avec file d'attente éloignée 77
  - arrêt 33
    - contrôlé 33
    - mise au repos 33
  - arrêt dans un groupe de reprise 229
  - arrêt forcé 34
  - arrêt immédiat 34
  - cliché du contenu d'un journal de reprise 151
  - cluster 13, 15
  - commande crtmqm 250
  - commande de cliché de la version formatée du journal système (dmpmqlog) 257
  - commande dltmlqm 254
  - commande endmqm 274
  - contrôle 116
  - conversion des messages 78

- gestionnaire de files d'attente (*suite*)
  - création 28, 32
  - démarrage 32
  - démarrage dans un groupe de reprise 228
  - déplacement dans un groupe de reprise 229
  - description 11
  - droits d'accès 88
  - enregistrement d'images de support 146
  - événements 117
  - fichiers de configuration
    - définition 31
  - fichiers qm.ini 175
  - gestionnaire des droits d'accès aux objets
    - désactivation 85
  - gestionnaire des droits d'accès aux objets.
    - description 83
  - journalisation circulaire (reprise) 138
  - journalisation linéaire 137
  - journaux 135
  - nom unique 29
  - nombre de 29
  - objets
    - appels MQI 11
  - redémarrage après modification du CCSID 79
  - redémarrer 35
  - répertoires 88
  - répertoires de droits d'accès 98
  - serveur de commandes 63
  - sous MVS/ESA 72
  - spécification dans runmqsc 42
  - suppression 35
  - transformation de nom 24
  - valeur par défaut 29
    - modification 33
    - modification par erreur 33
    - suppression accidentelle 251
- gestionnaire de files d'attente cible
  - affectation d'un alias 77
  - création d'une file de transmission 69
  - création de canaux 69
  - définition 74
  - lancement des canaux 70
  - ne faisant pas partie du cluster 14
- gestionnaire de files d'attente de rebut
  - exemple (amqsdllq) 104
  - lancement 103
  - règles 104
- gestionnaire de files d'attente MVS/ESA 72
- gestionnaire des droits d'accès aux objets
  - commande setmqaut 85
  - fonctionnement 83
  - principaux 83
- gestionnaire des droits d'accès aux objets. 83
  - commande dspmqaut 87
  - commande setmqaut 86
  - désactivation 85
  - identificateurs de droits par défaut 84

- gestionnaire des droits d'accès aux objets. 83 (*suite*)
  - opérations sensibles 88
- gestionnaires de bases de données
  - commande dspmqtrn (vérification des unités de travail en attente) 131
  - commande rsvmqtrn (resynchronisation explicite d'unités de travail) 132
  - configuration 123
  - connexions 123
  - coordination 122
  - définition des gestionnaires de bases de données dans qm.ini 124
  - fichiers de commutation (création) 124
  - instances de gestionnaire de bases de données (suppression) 134
  - modification des données de configuration 133
  - restrictions (coordination des bases de données) 123
  - unités de travail en attente de validation 130
- Glossaire 391
- groupe de reprise
  - exemple de configuration 235
  - fichier de configuration 223
  - identification des problèmes 234
  - modification de l'état 232
  - utilisant Intra Cluster Communication (ICC) 233
  - utilisation de MultiNet for OpenVMS 235
- Groupe de reprise
  - description 222
- groupe de reprise (failover set)
  - administration 228
  - affichage de l'état 229
  - arrêt d'un GFA 229
  - configuration 224
  - démarrage d'un GFA 228
  - déplacement d'un GFA 229
  - étapes préalables à la configuration 224
  - procédures de commande 226
- groupe de reprise de cluster OpenVMS 224

## H

- historique
  - delete history (MONMQ) 358
  - des messages d'un canal 349
  - disable history (MONMQ) 358
  - enable history (MONMQ) 358
  - réinitialisation d'un canal (MONMQ) 358
  - set history (MONMQ) 358
- horodatage
  - disable timestamp (MONMQ) 357
  - enable timestamp (MONMQ) 357
- HTML (Hypertext Markup Language) 388
- Hypertext Markup Language (HTML) 388

**I**

- ID d'utilisateur connecté OpenVMS 88
- ID utilisateur
  - appartenant au groupe nobody 84
  - droits d'accès 81, 88
  - pour les droits d'accès 88
  - principaux 83
  - utilisateur connecté OpenVMS 88
- identificateur de droits
  - MQM 81
  - valeur par défaut pour les droits d'accès 84
  - valeurs par défaut (nobody) 84
- identificateur de droits OpenVMS
  - MQM 81
  - valeurs par défaut (nobody) 84
- identificateur de jeu de caractères codés
  - identificateur 341
- identificateurs de droits (droits d'accès) 83
- identificateurs de droits primaires (droits d'accès) 83
- identificateurs hexadécimaux
  - affichage des composants 353
- identification des incidents 193
  - clients 214
  - commandes sans réponse 197
  - éléments à vérifier en premier lieu 193, 196
  - erreurs de programmation 196
  - fichiers de configuration 208
  - sortie incorrecte
    - avec les files d'attente réparties 203
    - messages contenant des informations imprévues 202
    - n'apparaissant pas dans les files 201
  - trace 208
  - vérifications approfondies 197
- images de support
  - commande d'enregistrement 282
  - description 145
  - enregistrer 146
  - reconstitution 146
- incidents
  - exécution de commandes MQSC 46
  - reprise 145
  - utilisation de MQSC à distance 73
  - utilisation locale de MQSC 46
- informations renvoyées par les commandes MQSC 40
- informations utiles pour la configuration 381
- interface de file d'attente de messages (MQI) 7
- Intra Cluster Communication (ICC) 233

**J**

- jeu de codes 341
- jeu de commandes
  - comparaison 327
- jeux de caractères codés 77

**journal**

- cliché des enregistrements d'un journal (commande dmpmqlog) 151
- cliché du contenu 151
- configuration 182
- erreur 203
- erreur (exemple) 205
- fichier
  - contrôle 136
  - emplacement 144
  - réutilisation 138
- gestion 142
- gestionnaire de files d'attente 135
- paramètres 31
- sortie de la commande dmpmqlog 152
- strophe Log (qm.ini) 182
- taille 136
- utilisation pour la reprise 145

**journal de cliché**

- format 257
- objet 257
- paramètres 257

**journal des erreurs**

- erreur produites avant sa création 204
- exemple 205
- généralités 203

**journalisation**

- circulaire 136
- linéaire 137
- points de contrôle 138
- reconstitution de support 146
- types 136

**journalisation circulaire 136**

**journalisation linéaire 137**

**L**

**lancement**

- canaux 70

**lancement d'un gestionnaire de files d'attente. 309**

**liaisons**

- des applications sécurisées 379

**liste de noms**

- description 15
- utilisée avec le cluster de gestionnaire de files d'attente 15

**longueur actuelle de la file d'attente 50**

**longueur de file d'attente**

- actuelle 50
- détermination 50

**longueur maximale des lignes pour les commandes MQSC 44**

**M**

**macros**

- MONMQ 365

**manuels sous forme électronique 388**

**masque**

- set mask (MONMQ) 359

**maximum**

- longueur de file d'attente 11

**maximum (suite)**

- longueur des noms pour les objets MQSeries 10
- nombre de messages 9, 11
- taille de file d'attente 8
- taille de message 8, 10

**MAXMSGL 10**

**mémoire globale**

- contrôle à l'aide des paramètres sysgen 217

**message**

- algorithmes d'extraction 9
- considérations sur les performances
  - longueur 200
  - persistant 200
- contenant des informations imprévues 202
- d'événements outil 119
- descripteur 8, 62
- description 8
- erreurs sur clients DOS et Windows 215
- longueur variable 200
- mise en file d'attente 7
- n'apparaissant pas dans les files 201
- non transmis 208
- opérateur 205
- recherche d'un message
  - particulier 200
- taille maximale 8
- vitesse des messages non persistants 380

**message non persistant 380**

**messages d'erreur 40**

**messages d'erreur des client Windows 215**

**messages d'erreur des clients DOS 215**

**messages opérateur 205**

**mise en file d'attente de messages 7**

**mise en file d'attente éloignée 65**

**mise en file d'attente répartie**

- file d'attente de rebut 14
- file d'attente des messages non transmis 14
- sortie incorrecte 203

**mode file d'attente (runmqsc) 72**

**modèle**

- fichiers MQSC 331
- programmes (utilisation) 331

**modèles de groupe de reprise 333**

**modification des attributs d'un gestionnaire de files d'attente 42**

**modification des attributs de file d'attente 50**

**moniteur de groupe de reprise**

- observation 223

**moniteur de reprise**

- arrêt 232
- démarrage 228
- description 223

**moniteur de reprise principal**

- arrêt 232
- description 223

**mot clé APPLIDAT (règles) 106**

**mot clé APPLNAME (règles) 106**

**mot clé APPLTYPE (règles) 106**

**mot clé DESTQ (règles) 106**

- mot clé DESTQM (règles) 107
- mot clé FEEDBACK (règles) 107
- mot clé FORMAT (règles) 107
- mot clé FWDQ (règles) 108
- mot clé FWDQM (règles) 108
- mot clé HEADER (règles) 108
- mot clé INPUTQ (table des règles) 105
- mot clé INPUTQM (règles) 105
- mot clé MSGTYPE (règles) 107
- mot clé PERSIST (règles) 107
- mot clé PUTAUT (règles) 108
- mot clé REASON (règles) 107
- mot clé REPLYQ (règles) 107
- mot clé REPLYQM (règles) 107
- mot clé RETRY (règles) 109
- mot clé RETRYINT (règles) 205
- mot clé USERID (règles) 107
- mot clé WAIT (règles) 105
- mots clés d'action (règles) 107
- mots clés de modèles (table des règles) 106
- MQAI
  - description 62
- MQDATA 215
- MQDLH (en-tête de rebut) 103
- MQI
  - appels du gestionnaire de files d'attente 11
  - description 7
  - droits d'accès 91, 92
  - prise en charge de l'administration locale 37
- MQM
  - ID utilisateur 88
  - identificateur de droits 81
- MQPUT et MQPUT1 (considérations de performance) 201
- mqs.ini
  - chemin vers 46
  - définition 173
  - modification 173
  - priorités 174
  - strophe AllQueueManagers 175
  - Strophe ClientExitPath 177
  - strophe DefaultQueueManager 177
  - strophe ExitProperties 177
  - strophe LogDefaults 178
  - strophe QueueManager 180
- MQSC
  - attributs 23
  - commandes 22
  - commandes PCF d'échappement 23
  - délai des réponses des commandes 72
  - distinction entre majuscules et minuscules 26
  - émission à distance 71
  - émission de commandes 38
  - émission de commandes de façon interactive 39
  - exécution des commandes à partir de fichiers texte 42
  - fichiers de commandes
    - entrée 43
    - exécution 45
    - rapports de sortie 44
  - fin de l'entrée interactive 41

- MQSC (*suite*)
  - incidents
    - à distance 73
    - locaux 46
  - longueur maximale des lignes 44
  - modèles de fichiers 331
  - redirection des données en entrée et en sortie 39
  - sécurité des canaux 91
  - utilisation des commandes 42
  - vérification des commandes 45
- MQSC interactive
  - fin 41
  - informations renvoyées 40
  - utilisation 39
- MQSeries
  - identificateur de droits (MQM) 81
  - OpenVMS (généralités) 317
- MQSeries pour Compaq OpenVMS (généralités) 317
- MQZAO (constantes et droits d'accès) 92
- MsgId (effet sur les performances) 200
- MultiNet pour OpenVMS 235
- MultiNet pour OpenVMS
  - modèle de configuration 227

## N

- nettoyage d'une file d'attente locale 51
- nobody (identificateurs de droits par défaut) 84
- nom logique (désactivation de la sécurité) 85
- nom logique du système d'exploitation (désactivation de la sécurité) 85
- nom logique MQSNOAUT 85
- noms
  - autorisés pour les objets 243
  - nombre maximum de caractères 10
  - objets 10
- notification des événements 118

## O

- OAM 83
- objet
  - transformation de nom 24
- objets
  - accès aux 81
  - administration éloignée 65
  - attributs 10
  - commande de recréation 284
  - conventions de dénomination 243
  - définition de processus 14
  - fichier représentant un objet 244
  - file d'attente 12
  - gestion 10
  - gestionnaire de files d'attente
    - appels MQI 11
  - image de support 145
  - liste de noms 15
  - noms 10, 39
  - objets système par défaut 16, 319
  - pour le déclenchement 58
  - reconstitution 146

- objets (*suite*)
  - types 9
  - valeur par défaut
    - attributs 49
- OpenVMS
  - généralités 317
  - logiciel requis 317
  - matériel requis 317
- optimisation des performances 217
- oracle
  - configuration 125
  - fichier de commutation (création) 126
  - ORACLE\_HOME (variable d'environnement) 125
  - ORACLE\_SID (variable d'environnement) 125
  - paramétrage des variables d'environnement, vérification 125
  - paramètres de configuration (modification) 129
  - prise en charge XA Oracle (activation) 125
  - strophe XAResourceManager (ajout au gestionnaire de file d'attente qm.ini) 128
  - versions minimales requises 125

## P

- paramètres sysgen 217
- partage de files d'attente 171
- PDF (Portable Document Format) 388
- point de synchronisation (considérations de performance) 200
- principaux
  - doté de plusieurs identificateurs de droits d'accès 84
  - gestion de l'accès 83
- prise en charge transactionnelle
  - prise en charge transactionnelle 121
- procédure de commande End 226
- procédure de commande Start 226
- procédure de commande Tidy 226
- procédure EndCommand
  - modèle 336
- procédure StartCommand
  - modèle 335
- procédure TidyCommand
  - modèle 339
- procédures de commande 226
  - exemples 237
  - modification 237
- processus
  - début d'une trace 354
- processus (géré par messages) 7
- processus géré par messages 7
- processus gérés par messages 7
- programme d'écoute
  - commande d'arrêt du programme d'écoute (endmqlsr) 273
  - runmqlsr (exécution du programme d'écoute) 294
  - sécurisé 380
- programmes (modèles fournis) 331
- publications connexes 389
- publications MQSeries 387



## Q

quotas partagés 218

## R

reconstitution d'images de support 146  
redémarrage d'un gestionnaire de files  
d'attente 35  
redirection des données en entrées et en  
sortie (commandes MQSC) 39  
rééquilibrage de la charge de travail  
à l'aide de clusters 9  
règles, gestionnaire DLQ 104  
règles (gestionnaire DLQ)  
exemple 113  
modèles et actions (règles) 106  
mot clé ACTION 107  
mot clé APPLIDAT 106  
mot clé APPLNAME 106  
mot clé APPLTYPE 106  
mot clé DESTQ 106  
mot clé DESTQM 107  
mot clé FEEDBACK 107  
mot clé FORMAT 107  
mot clé FWDQ 108  
mot clé FWDQM 108  
mot clé HEADER 108  
mot clé MSGTYPE 107  
mot clé PERSIST 107  
mot clé PUTAUT 108  
mot clé REASON 107  
mot clé REPLYQ 107  
mot clé REPLYQM 107  
mot clé RETRY 109  
mot clé USERID 107  
saisie de données de contrôle 105  
mot clé INPUTQ 105  
mot clé INPUTQM 105  
mot clé RETRYINT 105  
mot clé WAIT 105  
syntaxe 109  
traitement 110  
remarques sur les performances  
avantages de MQPUT1 201  
Correlld 200  
longueur de message variable 200  
Msgld 200  
persistance d'un message 200  
point de synchronisation 200  
taille de message 200  
répertoires  
droits d'accès 98  
gestionnaire de files d'attente 88  
reprise (journalisation circulaire) 138  
reprise par restauration  
description 223  
reprise par transfert (failover)  
description 223  
ressources  
nécessité de les protéger 81  
ressources  
protégées 84  
ressources protégées 84  
restrictions  
accès aux objets MQM 81

restrictions (*suite*)

coordination des bases de  
données 123  
noms d'objet 243  
runmqsc  
émission de commandes MQSC 38  
fin 41  
incidents 46  
mode file d'attente 72  
résultat 40  
spécification d'un gestionnaire de files  
d'attente 42  
utilisation 42  
utilisation interactive 39  
vérification 45

## S

SAFEGUARD 317  
scénarios de reprise  
incidents d'unité de disque 149  
objet gestionnaire de files d'attente  
altéré 150  
simple objet altéré 150  
scripts  
MONMQ 365  
sections globales 350  
sécurité 81  
activation 85  
éloignée 90  
moniteur de groupe de reprise 233  
utilisation des commandes 85, 87  
serveur de commandes  
administration éloignée 63  
affichage 263  
affichage de l'état 63  
arrêt d'un serveur de commande 64  
commande d'arrêt 270  
commande de lancement 308  
lancement d'un serveur de  
commandes 63  
service annuaire 171  
service d'alias de cluster 235  
services optionnels  
désactivation du gestionnaire des  
droits d'accès aux objets  
désactivation 85  
gestionnaire des droits d'accès aux  
objets. 83  
service annuaire 171  
sortie  
réacheminement 360  
sortie incorrecte 201  
strophe AllQueueManagers  
(mqqs.ini) 175  
strophe Channels (qm.ini) 186  
Strophe ClientExitPath (mqqs.ini) 177  
strophe DefaultQueueManager  
(mqqs.ini) 177  
strophe Exitpath (qm.ini) 189  
strophe ExitProperties (mqqs.ini) 177  
strophe Log (qm.ini) 182  
strophe LogDefaults (mqqs.ini) 178  
strophe LU62 (qm.ini) 187  
strophe QueueManager (mqqs.ini) 180  
strophe Service (qm.ini) 181  
strophe ServiceComponent (qm.ini) 181

strophe TCP (qm.ini) 187  
strophe XAResourceManager  
(qm.ini) 184  
strophes  
AllQueueManagers (mqqs.ini) 175  
Channels (qm.ini) 186  
ClientExitPath (mqqs.ini) 177  
DefaultQueueManager (mqqs.ini) 177  
ExitPath (qm.ini) 189  
Log (qm.ini) 182  
LogDefaults (mqqs.ini) 178  
LU62 (qm.ini) 187  
QueueManager (mqqs.ini) 180  
service (qm.ini) 181  
strophe ExitProperties (mqqs.ini) 177  
Strophe ServiceComponent  
(qm.ini) 181  
TCP (qm.ini) 187  
XAResourceManager (qm.ini) 184  
structure de répertoires 321  
Support de l'euro 318  
support de la langue nationale  
conversion du caractère EBCDIC NL  
vers ASCII 176  
support de langue nationale  
conventions de dénomination 243  
SupportPac 389  
suppression  
commande dltmqm 254  
file d'attente locale 51  
gestionnaire de files d'attente 35  
syntaxe  
aide 245  
diagrammes (lecture) 244  
erreur (dans les commandes  
MQSC) 40  
SYS\$INPUT, dans runmqsc 42  
SYS\$OUTPUT, dans runmqsc 42  
système  
objets par défaut 16, 319

## T

table de processus mutex 350  
table mémoire 352  
taille de message (diminution) 51  
taille des journaux 136  
TCP/IP  
administration éloignée 16  
configuration 187  
configuré par la procédure  
StartCommand 227  
connexion rejetée 188  
définition des canaux 68  
définition du protocole de  
transmission 294  
Digital TCP/IP Services pour  
OpenVMS 227  
lancement des canaux 70  
modules pris en charge 317  
obligatoire au fonctionnement du  
cluster OpenVMS 224  
prise en charge des clusters  
OpenVMS 222  
Terminologie utilisée dans le  
manuel 391

- trace
  - considérations sur les performances 208
  - début d'un processus 354
  - disable trace (MONMQ) 358
  - enable trace (MONMQ) 357
  - exemple de données 209
  - exemple de session MONMQ 366
  - indication d'un composant ou d'une fonction 355
  - sortie d'une trace (MONMQ) 363
  - sortie de MONMQ 363
  - start trace (MONMQ) 355
  - stop trace (MONMQ) 355
  - utilisation de MONMQ 343
- transaction en attente de validation
  - gestionnaires de bases de données 130
- transactions
  - affichage des commandes MQSeries 268
  - commande de résolution MQSeries 287
- types d'événements 117
- types d'objets 9

## U

- unité de travail
  - définition 121
  - résultats incohérents 132
  - resynchronisation explicite (commande rsvmqtrn) 132
- unité de travail locale
  - définition 121
- unité logique
  - affichage à l'aide de show segment (MONMQ) 346
  - close (MONMQ) 347
- unités de travail globales
  - ajout de la strophe XAResourcemanager à qm.ini (Oracle) 128
  - définition 121
- Utilitaire AUTHORIZE 218
- utilitaire monmq
  - commandes
    - activation de l'historique 358
    - activation de l'horodatage 357
    - activation du traçage 357
    - affichage d'un segment 346
    - affichage de l'historique 349
    - affichage de la mémoire 352
    - affichage de la table mutex 350
    - affichage des composants 353
    - affichage des événements 351
    - affichage des fonctions 353
    - affichage des processus 349
    - affichage des sections globales 350
    - affichage du contenu des piles 348
    - affichage du masque 348
    - affichages des informations sur les canaux 347
    - analyse de la trace 360
    - arrêt d'une trace 355

- utilitaire monmq (*suite*)
  - commandes (*suite*)
    - connexion 354
    - déconnexion 355
    - définition de la profondeur de l'historique 358
    - définition du masque 359
    - désactivation de l'historique 358
    - désactivation de l'horodatage 357
    - désactivation du traçage 358
    - désélection index 356
    - fermeture d'un fichier binaire 357
    - fermeture d'une lu 347
    - fermeture du fichier texte 357
    - FFST 363
    - lancement d'une trace 355
    - lancement de trace 343
    - onstartup start 354
    - onstartup stop 354
    - ouverture 346
    - ouverture d'un fichier binaire 356
    - ouverture d'un fichier texte 357
    - réacheminement de la sortie 360
    - réinitialisation 358
    - sélection 355
    - set color 360
    - sortie 363
    - suppression de l'historique 358
    - variable par défaut 346
  - généralités 343
  - gestion de la mémoire partagée 364
  - traçage des processus MQSeries
    - exemple de session de trace 366
    - scripts et macros dans MONMQ 365
    - variables utilisées dans MONMQ 344

## V

- valeur par défaut
  - attributs des objets 49
  - file d'attente de transmission 31
  - gestionnaire de files d'attente 29
    - commandes traitées 40
    - modification 33, 42
    - modification par erreur 33
    - suppression accidentelle 251
  - identificateurs de droits (droits d'accès) 84
  - objets 16
  - objets système 319
  - variable d'environnement
    - MQSPREFIX 176
    - ORACLE\_HOME (Oracle) 125
    - ORACLE\_SID, Oracle 125
  - variable d'environnement
    - MQSPREFIX 176
  - vérification des commandes MQSC 45
  - visualisation des files d'attente 51

---

## Envoi de vos remarques à IBM

Si vous avez une ou plusieurs remarque(s) à formuler sur le présent manuel, utilisez l'une des méthodes présentées ci-dessous pour les transmettre à votre centre d'assistance IBM.

Vous pouvez formuler des remarques concernant des erreurs spécifiques ou des omissions, ou bien sur la précision des informations, l'organisation du manuel, son contenu, sa présentation générale.

Limitez vos remarques aux informations contenues dans ce manuel et à la façon dont elles sont traitées.

**Par ailleurs, nous vous rappelons que pour tout commentaire relatif à des produits ou des systèmes IBM, vous devez vous adresser à votre distributeur agréé, votre agent IBM ou votre ingénieur commercial IBM.**

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie de ces informations que, de votre côté, vous pourrez évidemment continuer à exploiter.

Vous pouvez transmettre vos remarques à IBM de l'une des façons suivantes :

- Par courrier, à l'adresse suivante :  
User Technologies Department (MP095)  
IBM United Kingdom Laboratories  
Hursley Park  
WINCHESTER,  
Hampshire  
SO21 2JN  
Royaume-Uni
- Par télécopie :
  - En dehors du Royaume-Uni, après le code international, composez le 44-1962-870229
  - A partir du Royaume-Uni, composez le 01962-870229
- Par courrier électronique, utilisez l'ID réseau approprié :
  - IBM Mail Exchange : GBIBM2Q9 at IBMMAIL
  - IBMLink : HURSLEY(IDRCF)
  - Internet : idrcf@hursley.ibm.com

Quelle que soit la méthode choisie, assurez-vous que vous avez précisé les éléments suivants :

- Le numéro et le titre de la publication
- Les numéros de page ou les rubriques concernés
- Vos nom et adresse/numéro de téléphone ou de télécopie/votre ID utilisateur.







SC11-1798-00

