

MQSeries® for Compaq OpenVMS Alpha®



システム管理ガイド

バージョン 5 リリース 1

MQSeries® for Compaq OpenVMS Alpha®



システム管理ガイド

バージョン 5 リリース 1

ご注意!

本書、および本書がサポートする製品をご使用になる前に、481ページの『付録L. 特記事項』にある一般的な情報を必ずお読みください。

本書の内容は、新版で特に指定がない限り、MQSeries for Compaq OpenVMS Alpha、バージョン 5.1 以降のすべてのリリースおよびモディフィケーションに適用されます。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

原典：	SC34-5884-00 MQSeries® for Compaq OpenVMS Alpha® System Administration Guide Version 5 Release 1
発行：	日本アイ・ビー・エム株式会社
担当：	ナショナル・ランゲージ・サポート

第1刷 2001.5

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1994, 2001. All rights reserved.

Translation: © Copyright IBM Japan 2001

目次

図	xi
表	xiii
本書について	xv
本書の対象読者	xv
本書を理解する上での前提事項	xv
本書の使い方	xvi
付録の使い方	xvi
インターネット上の MQSeries についての情報	xvi
MQSeries for Compaq OpenVMS Alpha V5.1 の新機能	xvii

第1部 手引き 1

第1章 MQSeries の概要	7
MQSeries とメッセージ・キューイング	7
時間に依存しないアプリケーション	7
メッセージ・ドリブン処理	8
メッセージとキュー	8
メッセージとは	8
キューとは	9
オブジェクト	10
オブジェクト名	11
オブジェクトの管理	11
MQSeries キュー・マネージャー	12
MQSeries キュー	13
プロセス定義	18
チャンネル	18
クラスター	18
名前リスト	19
システム・デフォルト・オブジェクト	19
ローカル管理およびリモート管理	20
クライアントおよびサーバー	20
クライアント / サーバー環境での MQSeries アプリケーション	21
キュー・マネージャー機能の拡張	21
ユーザー出口	22
インストール可能なサービス	22

セキュリティ	23
オブジェクト権限マネージャー (OAM) 機能	23
DCE セキュリティー	23
トランザクション・サポート	24

第2章 MQSeries 管理の紹介 25

ローカル管理およびリモート管理	25
制御コマンドによる管理タスクの実行	26
MQSC コマンドによる管理タスクの実行	26
PCF コマンドによる管理タスクの実行	27
MQSC および PCF での属性	28
エスケープ PCF	28
MQSeries のファイル名についての理解	28
キュー・マネージャー名の変換	29
オブジェクト名の変換	29
大文字小文字の区別についての理解	30
制御コマンドでの大文字小文字の区別	30
MQSC コマンドでの大文字小文字の区別	31

第3章 制御コマンドによるキュー・マネージャーの管理 33

制御コマンドの使用	33
制御コマンドの使用	34
キュー・マネージャーを作成する	35
キュー・マネージャーの作成のためのガイドライン	35
デフォルト・キュー・マネージャーを作成する	40
キュー・マネージャーを開始する	41
既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする	41
キュー・マネージャーを停止する	42
静止シャットダウン	42
即時シャットダウン	42
優先シャットダウン	43
キュー・マネージャーのシャットダウンに問題がある場合	43
キュー・マネージャーを再始動する	44
キュー・マネージャーを削除する	44

第4章 ローカル MQSeries オブジェクトの管理	45	第5章 管理用タスクの自動化	77
MQI を使用したアプリケーション・プログラムのサポート	45	PCF コマンド	77
MQSC コマンドによるローカル管理タスクの実行	47	MQSC および PCF での属性	78
開始する前に	48	エスケープ PCF	78
MQSC 機能を対話式で使用する	49	MQAI を使用して PCF の使い方を単純化する	79
MQSC コマンドからのフィードバック	50	リモート管理でコマンド・サーバーを管理する	80
対話式での MQSC への入力を終了する	50	コマンド・サーバーを開始する	80
キュー・マネージャーの属性を表示する	51	コマンド・サーバーの状況を表示する	80
デフォルトではないキュー・マネージャーを使用する	52	コマンド・サーバーを停止する	81
キュー・マネージャーの属性を更新する	52	第6章 リモート MQSeries オブジェクトの管理	83
テキスト・ファイルからの MQSC コマンドの実行	52	チャンネル、クラスターおよびリモート・キューイング	83
MQSC コマンド・ファイル	53	クラスターを使用するリモート管理	85
MQSC レポート	54	MQSC コマンドを使用するローカル・キュー・マネージャーからのリモート管理	85
システムに提供された MQSC コマンド・ファイルを実行する	55	リモート管理のためのキュー・マネージャーを作成する	86
runmqsc を使用してコマンドを確認する	55	リモート管理のためにチャンネルおよび伝送キューを作成する	87
MQSC で起こった問題の解決	56	チャンネルと伝送キューを定義する	88
ローカル・キューの取り扱い	58	チャンネルを開始する	89
ローカル・キューを定義する	59	チャンネルの自動定義	90
送達不能キューを定義する	60	リモートから MQSC コマンドを発行する	91
デフォルト・オブジェクトの属性を表示する	60	MVS/ESA でのキュー・マネージャーの取り扱い	92
ローカル・キュー定義をコピーする	61	リモートからの MQSC の使用に問題がある場合	93
ローカル・キューの属性を変更する	62	リモート・キューのローカル定義の作成	93
ローカル・キューをクリアする	63	リモート・キューのローカル定義の働きについて理解する	94
ローカル・キューを削除する	63	リモート・キューにメッセージを書き込む代替方法	96
キューのブラウズ	63	リモート・キューに関してその他のコマンドを使用する	96
別名キューの取り扱い	69	伝送キューを作成する	97
別名キューを定義する	69	リモート・キュー定義を別名として使用する	97
キュー別名に関してその他のコマンドを使用する	71	キュー・マネージャー別名	98
モデル・キューの取り扱い	71	応答先キュー別名付け	98
モデル・キューを定義する	72	データ変換	99
モデル・キューに関してその他のコマンドを使用する	72	キュー・マネージャー CCSID の変更	101
トリガー操作のためのオブジェクトの管理	73	第7章 MQSeries オブジェクトの保護	103
トリガー操作のためにアプリケーション・キューを定義する	73		
開始キューを定義する	75		
プロセス定義を作成する	75		
プロセス定義を表示する	76		

MQSeries リソースを保護する理由	103	ルール (パターンと処理)	135
開始する前に	104	ルール・テーブルのルール	139
リソース ID MQM を持つ MQSeries for Compaq OpenVMS のユーザー ID	104	ルール・テーブルの処理方法	141
詳細情報	105	すべての DLQ メッセージを確実に処理す る	142
オブジェクト権限マネージャーについての理 解	105	DLQ ハンドラーのルール・テーブルの例	144
OAM の機能	106	第9章 観測イベント	147
権限 ID を介してアクセスを管理する	106	観測イベントとは	147
デフォルト権限 ID	107	イベントを使用するのはなぜか	149
OAM で保護できるリソース	107	イベントのタイプ	149
許可に権限 ID を使用する	108	イベント・キューによるイベント通知	150
オブジェクト権限マネージャーを使用不可 にする	108	イベントの有効化および無効化	151
オブジェクト権限マネージャー・コマンドの 使用	109	イベント・メッセージ	152
OAM コマンドの使用時に指定するもの	109	第10章 トランザクション・サポート	153
setmqaut コマンドを使用する	110	データベースの調整	154
アクセス許可	111	制約事項	156
権限の表示コマンド	111	データベースの接続	156
オブジェクト権限マネージャーのガイドライ ン	112	データベース・マネージャーの構成	156
ユーザー ID	112	Oracle の構成	159
キュー・マネージャー・ディレクトリー	112	Oracle の最小サポート・レベル	159
キュー	113	環境変数の設定の確認	159
代替ユーザー権限	113	Oracle XA サポートの使用可能化	160
コンテキスト権限	114	Oracle スイッチ・ロード・ファイルの作 成	160
リモート・セキュリティーに関する考慮事 項	115	Oracle 用の XAResourceManager 構成情報 の追加	161
チャンネル・コマンド・セキュリティー	115	Oracle 構成パラメーターの変更	163
許可指定テーブルについての理解	117	管理タスク	164
MQI 許可	117	未確定の作業単位	165
管理許可	122	dspmqrn コマンドを使用する	165
エスケープ PCF の中の MQSC コマンド に関する許可	122	rsvmqtrn コマンドを使用する	167
許可ファイルの理解	125	エラーが混在する実行結果	168
許可ファイルのパス	126	構成情報の変更	169
許可ファイルの内容	127	第11章 リカバリーおよび再始動	173
許可ファイルの管理	129	メッセージの消失を確実に回避する (ロギン グ)	174
第8章 MQSeries 送達不能キュー・ハンドラ ー	131	ログの概要	174
DLQ ハンドラーの呼び出し	131	ログのタイプ	175
サンプル DLQ ハンドラー amqsdlq	132	チェックポイント機能 (完全なりカバリーの 保証)	177
DLQ ハンドラーのルール・テーブル	133	ログのサイズの計算	180
制御データ	133	ログの管理	182

ディスクが満杯になったときに何が起きるか	183	Channels スタンザ	238
ログ・ファイルを管理する	184	LU62 および TCP スタンザ	240
ログをリカバリーに使用する	185	ExitPath スタンザ	242
問題からリカバリーする	185	mqs.ini および qm.ini ファイルの例	242
メディア・リカバリー	186		
始動時に損傷オブジェクトをリカバリーする	188		
その他の場合の損傷オブジェクトのリカバリー	189		
MQSeries ログ・ファイルの保護	189		
バックアップと復元	189		
MQSeries のバックアップ	190		
MQSeries の復元	190		
リカバリーの手順	191		
ディスク・ドライブの障害	191		
キュー・マネージャーのオブジェクトの損傷	193		
単独オブジェクトの損傷	193		
自動メディア・リカバリーの障害	193		
dmpmqlog コマンドを使用したログの内容のダンプ	194		
第12章 名前サービスの使用	219		
さまざまなキュー・マネージャー上のキューを共用するために DCE を使用する	219		
共用キューのための構成タスク	219		
DCE 構成	220		
第13章 MQSeries の構成	221		
MQSeries 構成ファイル	221		
構成ファイルを編集する	222		
MQSeries 構成ファイル、mqs.ini	223		
キュー・マネージャー構成ファイル qm.ini	223		
MQSeries 構成情報を変更するための属性	224		
AllQueueManagers スタンザ	224		
ClientExitPath スタンザ	226		
DefaultQueueManager スタンザ	226		
ExitProperties スタンザ	226		
LogDefaults スタンザ	227		
QueueManager スタンザ	230		
キュー・マネージャー構成情報の変更	231		
Service スタンザ	231		
ServiceComponent スタンザ	232		
Log スタンザ	233		
XAResourceManager スタンザ	236		
		予備的な検査	247
		以前に MQSeries は正常に実行されたか	247
		エラー・メッセージが出されたか	248
		問題について説明した戻りコードが戻されていないか	248
		問題を再現できるか	248
		最後に正常に実行された後、変更が行われたか	248
		アプリケーションは以前に正常に実行されたか	249
		問題がネットワークの特定部分に影響するか	250
		1 日の特定の時刻に問題が起こるか	251
		問題は断続的に起こるか	251
		サービス更新を適用したか	251
		構成を適用する必要があるか	251
		共通プログラミング・エラー	252
		次に何を行うか	252
		不正な出力を得たか	253
		PCF コマンドからの応答を受け取れなかったか	253
		キューの一部が障害を起こしているか	254
		問題はリモート・キューにのみ影響するか	255
		アプリケーション設計の考慮事項	256
		メッセージ長の影響	256
		メッセージ持続性の影響	256
		特定のメッセージの検索	257
		長さが異なるメッセージを含んでいるキュー	257
		同期点の頻度	257
		MQPUT1 呼び出しの使用	258
		不正出力	258
		キューにメッセージが現れない場合	258
		メッセージが予期しない情報または破壊された情報を含んでいる場合	260
		分散キューを使用している場合の不正出力の問題	260
		エラー・ログ	262
		ログ・ファイル	262
		早期エラー	263

オペレーター・メッセージ	263
エラー・ログの例	264
送達不能キュー	267
構成ファイルと問題判別	267
MQSeries トレースの使用	268
トレース・ファイル名	268
トレース・データのサンプル	268
第 1 障害サポート・テクノロジー (FFST)	269
FFST の読み方	269
クライアント問題判別	275
クライアントの終了	275
クライアントによるエラー・メッセージ	276

第15章 パフォーマンスのチューニング	277
プロセス固有パラメーターの値の設定	279

第16章 MQSeries for OpenVMS およびク ラスタ化	281
OpenVMS クラスタへの MQSeries のイン ストール	282
OpenVMS クラスタ・フェールオーバー・ セット	283
OpenVMS クラスタ・フェールオーバ ー・セットの概説	283
OpenVMS クラスタ・フェールオーバ ー・セットの概念	284
OpenVMS クラスタ・フェールオーバ ー・セットの構成の準備	286
OpenVMS クラスタ・フェールオーバ ー・セットの構成	286
OpenVMS クラスタ・フェールオーバ ー・セットの構成後のタスク	287
FAILOVER.INI 構成ファイルの編集	288
フェールオーバー・セットにより使用され るコマンド・プロシーチャー	289
フェールオーバー・セットの管理	291
フェールオーバー・モニターの始動	292
フェールオーバー・セット内のキュー・マ ネージャーの始動	292
フェールオーバー・セット内のキュー・マ ネージャーの終了	293
フェールオーバー・セット内のキュー・マ ネージャーの移動	293
フェールオーバー・セットの状態の表示	294
フェールオーバー・セットの状態への DCL シンボルの設定	296

フェールオーバー・モニター・プロセスの 停止	297
更新の進行中のコマンドの実行	298
フェールオーバー・セットの状態の変更	298
ICC アソシエーションのセキュリティー の設定	299
フェールオーバー・セットのトラブルシュ ーティング	300
フェールオーバー・セットと MultiNet for OpenVMS の使用	301
フェールオーバー・セットの使用例	301

第2部 リファレンス 311

第17章 MQSeries 制御コマンド	313
MQSeries オブジェクトを命名する際の規則	313
オブジェクト・ファイルの表示	314
構文図の見方	314
構文ヘルプ	316
例	316
MQSeries 戻りコード	317
crtmqcvx (データ変換)	318
crtmqm (キュー・マネージャーの作成)	320
dlmqm (キュー・マネージャーの削除)	326
dmpmqlog (ダンプ・ログ)	328
dspmqaut (許可の表示)	330
dspmqcsv (コマンド・サーバーの表示)	335
dspmqfls (MQSeries ファイルの表示)	337
dspmqtrc (MQSeries 定様式トレース出力の表 示)	340
dspmqtrn (MQSeries トランザクションの表 示)	341
endmqcsv (コマンド・サーバーの終了)	343
endmqlsr (リスナーの終了)	345
endmqm (キュー・マネージャーの終了)	346
endmqtrc (MQSeries トレースの終了)	349
フェールオーバー (フェールオーバー・セッ トの管理)	351
rcdmqimg (メディア・イメージの記録)	355
rcrmqobj (オブジェクトの再作成)	358
rsvmqtrn (MQSeries トランザクションの解決)	361
runmqchi (チャンネル開始プログラムの実行)	363
runmqchl (チャンネルの実行)	364
runmqdlq (送達不能キュー・ハンドラーの実 行)	365

runmqfm (フェールオーバー・モニターの開始)	367
runmqslr (リスナーの実行)	369
runmqsc (MQSeries コマンドの実行)	371
runmqtmc (クライアント・トリガー・モニターの開始)	375
runmqtrm (トリガー・モニターの開始)	376
setmqaut (許可の設定 / リセット)	377
strmqcsv (コマンド・サーバーの開始)	384
strmqm (キュー・マネージャーの開始)	385
strmqtrc (MQSeries トレースの開始)	387

第3部 付録 391

付録A. MQSeries for Compaq OpenVMS の概要 393

プログラムと部品番号	393
ハードウェア要件	393
ソフトウェア要件	393
接続性	393
セキュリティ	394
メンテナンス機能	394
互換性	394
サポートされるコンパイラ	394
言語の選択	395
国際化対応	395

付録B. システム・デフォルト 397

付録C. ディレクトリー構造 401

MQS_ROOT:[MQM] 中のディレクトリーとファイル	403
MQS_ROOT:[MQM.QMGRS.QMNAME] サブディレクトリーの中のディレクトリーとファイル	404

付録D. コマンド・セットの比較 409

キュー・マネージャーの管理用のコマンド	409
コマンド・サーバーの管理用のコマンド	410
キューの管理用のコマンド	410
プロセスの管理用のコマンド	411
チャンネルの管理用のコマンド	411
その他の制御コマンド	412

付録E. MQI プログラムおよび MQSC ファイルのサンプル 413

MQSC コマンド・ファイルのサンプル	413
-------------------------------	-----

C および COBOL プログラム・サンプル	413
各種ツール	414

付録F. OpenVMS クラスタ・フェールオーバー・セット・テンプレート 417

テンプレート構成ファイル	
FAILOVER.TEMPLATE	417
テンプレート StartCommand プロシージャ	
START_QM.TEMPLATE	419
テンプレート EndCommand プロシージャ	
END_QM.TEMPLATE	421
テンプレート TidyCommand プロシージャ	
TIDY_QM.TEMPLATE	425

付録G. MQSeries for Compaq OpenVMS でのコード・セットのサポート 427

付録H. MONMQ 診断ユーティリティー . . . 431

概説	431
MONMQ 内の変数	432
デフォルト値の割り当て	434
トレース・セクションおよび関連するメールボックスのオープンまたは作成	435
論理装置定義の表示	435
LU のクローズおよび削除	436
チャンネル詳細の表示	436
チャンネルの現行のトレース・マスクを表示する	437
ターゲット・スレッド・スタックの内容を表示する	437
アクティブな MQSeries 関連プロセスおよびメモリー使用法の表示	438
チャンネルに保留されるすべてのメッセージの表示	438
現行ノード上のすべての MQSeries 関連グローバル・セクションを表示する	439
ターゲット・スレッドに対し、クライアント・トレース・プロセスに mutex テーブルを送信するようシグナルを送る	440
ターゲット・スレッドに対し、クライアント・トレース・プロセスに内部イベント・テーブルを送信するようシグナルを送る	441
ターゲット・スレッドに対し、クライアント・トレース・プロセスに内部マップ共用メモリー・テーブルを送信するようシグナルを送る	442

名前および 16 進 ID によるアクティブ	
MQSeries コンポーネントの表示	443
指定されたコンポーネント内の関数を表示する	444
プロセスの開始点からトレースをアクティブにする	445
MQSeries プロセスの始動直後のトレースを回避する	445
ターゲット・スレッドを指定されたチャンネルに接続する	446
ターゲット・スレッドを指定されたチャンネルから切断する	446
LU トレース・メールボックスに書き込まれたリアルタイム・トレース・メッセージを表示する	446
現行のクライアント・プロセスの切り離しと終了	446
トレース・データの指定	447
トレース・フィルター・テーブルから単一記入項目を除去する	447
クライアント・プロセスがトレース・メッセージをバイナリー・ファイルに書き込む	448
バイナリー・トレース・メッセージ・ファイルのクローズ	448
クライアント・プロセスがトレース・メッセージをテキスト・ファイルに書き込む	448
テキスト・トレース・メッセージ・ファイルのクローズ	449
タイム・スタンプ・メッセージ	449
メッセージのタイム・スタンプの停止	449
トレースを使用可能にする	449
トレースを使用不可にする	450
メッセージ・履歴の保管	450
メッセージ・履歴を使用不可にする	450
メッセージ・履歴の削除	450
履歴の深さの設定	450
チャンネルのスタックおよび履歴・データをリセットする	451
マスク・ビットを使用可能または使用不可にする	451
チャンネルにカラーを設定する	453
出力をファイルに宛先変更する	453
トレース・バイナリー・ファイルの分析	454
MQSeries スレッドの現行状態を表示する	456
トレースのクローズと MONMQ の終了	457
トレースをクローズしないで MONMQ を終了する	457
共用メモリーを MONMQ で管理する	458
MONMQ 中のスクリプトとマクロ	459
トレース・セッションのサンプル	460
付録I. ユーザー出口	473
チャンネルおよびワークロード出口	473
MQSeries クラスタ・ワークロード出口	473
付録J. トラストッド・アプリケーション 475	475
ユーザー・アプリケーション	475
トラストッド・アプリケーションのセットアップ	476
トラストッド・アプリケーションとしてチャネルとリスナーを実行する	476
高速、非持続性のメッセージ	477
付録K. 補足情報	479
アプリケーション・プログラミング・ガイド	479
アプリケーション・トリガー操作	479
付録L. 特記事項	481
商標	483
参考文献	485
MQSeries 複数のプラットフォームに共通する資料	485
MQSeries プラットフォーム固有の資料	486
ソフトコピー資料	487
HTML 形式	487
PDF	487
インターネットで利用できる MQSeries 情報	488
参考資料	488
用語集および略語集	491
索引	511



1. キュー、メッセージ、およびアプリケーション	46	17. 持続期間の長いトランザクションでのチェックポイント機能	179
2. DISPLAY QMGR コマンドからの一般的な出力	51	18. dmpmqlog 出力の例	200
3. MQSC コマンド・ファイル myprog.in からの抽出	54	19. MQSeries for Compaq OpenVMS システム用の MQSeries 構成ファイルの例	243
4. MQSC レポート・ファイル myprog.out からの抽出	55	20. キュー・マネージャー構成ファイル (MQSeries for Compaq OpenVMS) の例	244
5. キュー・ブラウザーからの一般的な結果	65	21. MQSeries for Compaq OpenVMS トレーズのサンプル	269
6. リモート管理	87	22. ICC\$SYSTARTUP.COM で必要なエンタリーのサンプル	300
7. リモート管理のためのチャンネルとキューのセットアップ	88	23. FAILOVER.INI 構成ファイルを作成するための failover.template	302
8. DLQ ハンドラーのルール・テーブルのルールの一例	135	24. start_failover_set コマンド・プロシージャ	305
9. 観測イベントの理解	148	25. end_failover_set コマンド・プロシージャ	308
10. 異なるプラットフォームにあるキュー・マネージャーを、単一のノードでモニターする	149	26. キュー・マネージャーが開始された後のデフォルト・ディレクトリ構造	402
11. Oracle スイッチ・ロード・ファイル oraswit.c のソース・コード	160	27. テンプレート構成ファイル: failover.template	418
12. Oracle 用サンプル XAResourceManager エントリー	163	28. テンプレート StartCommand プロシージャ: Start_QM.template	420
13. dspmqtrn による出力の例	167	29. テンプレート EndCommand プロシージャ: END_QM.template	424
14. dspmqtrnによるエラーのトランザクションの出力例	169	30. テンプレート TidyCommand プロシージャ: TIDY_QM.template	426
15. XAResourceManager スタンザのコメント化	171		
16. チェックポイント機能	178		

表

1. 制御コマンドのカテゴリ	33	16. dspmqaut コマンドからのセキュリテ	
2. MQI 呼び出しに必要なセキュリテ		ー許可	331
許可	118	17. 種々のオブジェクト・タイプについ	
3. MQSC コマンドと必要なセキュリ		ての許可の指定	380
テー許可	122	18. キューのシステムおよびデフォルト	
4. PCF コマンドと必要なセキュリ		オブジェクト	397
テー許可	123	19. チャンルのシステムおよびデフォルト	
5. XA 準拠リレーショナル・データ		オブジェクト	398
ベース 155		20. 名前リストのシステムおよびデフォ	
6. ログのオーバーヘッド・サイズ (す		ルト・オブジェクト	398
べての値は概算値です).	180	21. プロセスのシステムおよびデフォ	
7. 使用できる ISO CCSID のリスト	225	ルト・オブジェクト	399
8. デフォルトの未解決接続要求数	241	22. キュー・マネージャーの管理用の	
(TCP)		コマンド	409
9. FAILOVER.INI ファイル内のフィ		23. コマンド・サーバーの管理用の	
ールドの説明	288	コマンド	410
10. コマンド・プロシージャーに渡		24. キューの管理用のコマンド	410
されるパラメーター	289	25. プロセスの管理用のコマンド	411
11. フェールオーバー・セット・		26. チャンルの管理用のコマンド	411
キュー・マネージャーの状態	294	27. その他の制御コマンド	412
12. フェールオーバー・セット・		28. MQSC コマンド・ファイル	413
ノード・キュー・マネージャーの		29. サンプル・プログラム - ソース・	
状態	295	ファイル	414
13. フェールオーバー・セット・		30. 各種ファイル	414
ノード・モニター状態	295	31. ロケールと CCSID	427
14. DCL シンボルおよび説明	296		
15. 構文図の見方	314		

本書について

MQSeries for Compaq OpenVMS Alpha V5.1 (本書では文脈に応じて MQSeries for Compaq OpenVMS または MQSeries と呼ばれることもある) は、MQSeries プロダクト・ファミリーの一部です。これらのプロダクトは、アプリケーション・プログラムがメッセージ・キューを使用して相互に通信できるようにするアプリケーション・プログラミング・サービスを提供します。この形式の通信は、コマーシャル・メッセージングと呼ばれています。関係するアプリケーションは、広範囲のタイプのマシンおよびオペレーティング・システムの種々のノードに存在できます。アプリケーションは、メッセージ・キューイング・インターフェースまたは MQI と呼ばれる共通アプリケーション・プログラミング・インターフェースを使用するので、あるプラットフォームで開発されたプログラムは容易に別のプラットフォームに移すことができます。

本書は、MQSeries for Compaq OpenVMS Alpha V5.1、および OpenVMS 環境でのコマーシャル・メッセージングをサポートするために MQSeries for Compaq OpenVMS Alpha V5.1 が提供するサービスについて、システム管理の面から説明します。その中には、アプリケーションがメッセージを受け取るために使用するキューの管理や、必要とするキューにアプリケーションが確実にアクセスできるようにすることなども含まれています。

本書の対象読者

本書の主な対象読者は、システム管理者、および MQSeries の構成タスクと管理タスクを管理するシステム・プログラマーです。本書はまた、MQSeries の管理作業について理解しておく必要のあるアプリケーション・プログラマーにとっても役立つものです。

本書を理解する上での前提事項

本書を使用するためには、本書に記載されている OpenVMS オペレーティング・システム、および関連したユーティリティについてよく理解しておく必要があります。メッセージ・キューイング・プロダクトを使用した経験はなくても構いませんが、メッセージ・キューイングの基本概念は理解しておく必要があります。

本書の使い方

本書の主部は、次のことを扱っています。

- MQSeries の紹介
- OpenVMS システムの日常の管理の説明。ローカルおよびリモート MQSeries オブジェクトの管理、セキュリティー、トランザクション・サポート、および問題判別について検討します。

付録の使い方

付録には、参照資料があります。一部の情報は、別の機会に他の MQSeries の資料に組み込まれることになっています。

インターネット上の MQSeries についての情報

MQSeries の URL

MQSeries プロダクト・ファミリーの URL は、次のとおりです。

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

MQSeries for Compaq OpenVMS Alpha V5.1 の新機能

MQSeries for Compaq OpenVMS Alpha V5.1 システム管理ガイド のこの版では、以下の新機能について説明します。

MQSeries キュー・マネージャー・クラスター

複数の MQSeries キュー・マネージャーを結合することによって、キュー・マネージャーのクラスターをつくることができます。同じクラスター内では、どのキュー・マネージャーも、別のキュー・マネージャーに属するキューを使用することができます。キュー・マネージャーが同じクラスター内の他のキュー・マネージャーにメッセージを送る場合、特別なチャンネル定義やリモート・キュー定義、各宛先用の伝送キューの定義はいずれも必要ありません。MQSeries クラスターの主な利点を次に挙げます。

- システム管理タスクの減少
- 可用性の向上
- 作業負荷の平衡化

詳細については、18ページの『クラスター』を参照してください。

注: MQSeries クラスターは、OpenVMS クラスターとは異なります。クラスター という語が用いられる場合、それは MQSeries キュー・マネージャー・クラスターを指します。OpenVMS クラスターは、常に *OpenVMS* クラスター と呼ばれます。OpenVMS クラスターの詳細については、281ページの『第16章 MQSeries for OpenVMS およびクラスター化』を参照してください。

MQSeries アプリケーション・インターフェース (MQAI)

MQSeries for Compaq OpenVMS は、MQSeries アプリケーション・インターフェース (MQAI) をサポートするようになりました。これは、MQSeries を構成するための PCF メッセージの使用を単純化するプログラム・インターフェースです。コマンドの完全な説明を含めた MQAI の詳細については、*MQSeries* 管理インターフェースのプログラミングの手引きおよび解説書 を参照してください。

メッセージ・キューの容量

メッセージ・キューの最大容量は、2 GB です。

キュー・マネージャーの制御待機シャットダウン

endmqm コマンドに新しいオプションが追加されました。このオプションを指定すると、キュー・マネージャーを同期制御シャットダウンすることができます。

Java® のサポート

MQSeries for Compaq OpenVMS は、Java コンパイラーに対応するようになりました。

Web 管理

Windows NT® 上の HTML ブラウザー (Netscape Navigator や Microsoft® Internet Explorer など) を使用して、以下の MQSeries for Compaq OpenVMS 管理タスクをリモート側で実行できるようになりました。

- MQSeries 管理者としてログオンする。
- キュー・マネージャーを選択し、キュー・マネージャーに対して MQSC コマンドを発行する。
- MQSC スクリプトを作成、編集および削除する。

第1部 手引き

第1章 MQSeries の概要	7	ローカル管理およびリモート管理	25
MQSeries とメッセージ・キューイング	7	制御コマンドによる管理タスクの実行	26
時間に依存しないアプリケーション	7	MQSC コマンドによる管理タスクの実行	26
メッセージ・ドリブン処理	8	PCF コマンドによる管理タスクの実行	27
メッセージとキュー	8	MQSC および PCF での属性	28
メッセージとは	8	エスケープ PCF.	28
メッセージ長	9	MQSeries のファイル名についての理解	28
キューとは	9	キュー・マネージャー名の変換	29
アプリケーションによるキューの送信 /		オブジェクト名の変換	29
受信方法	9	大文字小文字の区別についての理解	30
事前定義キューと動的キュー	10	制御コマンドでの大文字小文字の区別	30
キューからメッセージを取り出す	10	MQSC コマンドでの大文字小文字の区別	31
オブジェクト	10	第3章 制御コマンドによるキュー・マネー	
オブジェクト名	11	ジャーの管理	33
オブジェクトの管理	11	制御コマンドの使用	33
オブジェクトの属性	12	制御コマンドの使用	34
MQSeries キュー・マネージャー	12	キュー・マネージャーを作成する	35
MQI 呼び出し	13	キュー・マネージャーの作成のためのガイ	
MQSeries キュー	13	ドライン	35
キュー・オブジェクトの使用	14	固有のキュー・マネージャー名の指定	36
MQSeries で使用される特別のローカ		キュー・マネージャーの数の制限	36
ル・キュー	15	デフォルト・キュー・マネージャーの指	
プロセス定義	18	定	37
チャンネル	18	送達不能キューの指定	37
クラスター	18	デフォルト伝送キューの指定	38
名前リスト	19	必須ロギング・パラメーターの指定	39
システム・デフォルト・オブジェクト	19	キュー・マネージャーの作成後の構成フ	
ローカル管理およびリモート管理	20	ァイルのバックアップ	39
クライアントおよびサーバー	20	デフォルト・キュー・マネージャーを作成す	
クライアント / サーバー環境での		る	40
MQSeries アプリケーション	21	キュー・マネージャーを開始する	41
キュー・マネージャー機能の拡張	21	既存のキュー・マネージャーをデフォルト・	
ユーザー出口	22	キュー・マネージャーにする	41
インストール可能なサービス	22	キュー・マネージャーを停止する	42
セキュリティ	23	静止シャットダウン	42
オブジェクト権限マネージャー (OAM) 機		即時シャットダウン	42
能	23	優先シャットダウン	43
DCE セキュリティ	23	キュー・マネージャーのシャットダウンに	
トランザクション・サポート	24	問題がある場合	43
第2章 MQSeries 管理の紹介	25	キュー・マネージャーを再始動する	44

キュー・マネージャーを削除する	44	トリガー操作のためにアプリケーション・ キューを定義する	73
第4章 ローカル MQSeries オブジェクトの管 理	45	開始キューを定義する	75
MQI を使用したアプリケーション・プログラ ムのサポート	45	プロセス定義を作成する	75
MQSC コマンドによるローカル管理タスクの 実行	47	プロセス定義を表示する	76
開始する前に	48	第5章 管理用タスクの自動化	77
MQSeries オブジェクト名	48	PCF コマンド	77
入出力のリダイレクト	48	MQSC および PCF での属性	78
MQSC 機能を対話式で使用する	49	エスケープ PCF	78
MQSC コマンドからのフィードバック	50	MQAI を使用して PCF の使い方を単純化 する	79
対話式での MQSC への入力を終了する	50	リモート管理でコマンド・サーバーを管理す る	80
キュー・マネージャーの属性を表示する	51	コマンド・サーバーを開始する	80
デフォルトではないキュー・マネージャー を使用する	52	コマンド・サーバーの状況を表示する	80
キュー・マネージャーの属性を更新する	52	コマンド・サーバーを停止する	81
テキスト・ファイルからの MQSC コマンドの 実行	52	第6章 リモート MQSeries オブジェクトの管 理	83
MQSC コマンド・ファイル	53	チャンネル、クラスターおよびリモート・キュ ーイング	83
MQSC レポート	54	クラスターを使用するリモート管理	85
システムに提供された MQSC コマンド・ ファイルを実行する	55	MQSC コマンドを使用するローカル・キュ ー・マネージャーからのリモート管理	85
runmqsc を使用してコマンドを確認する	55	リモート管理のためのキュー・マネージャ ーを作成する	86
MQSC で起こった問題の解決	56	リモート管理のためにチャンネルおよび伝送 キューを作成する	87
ローカル・キューの取り扱い	58	チャンネルと伝送キューを定義する	88
ローカル・キューを定義する	59	チャンネルを開始する	89
送達不能キューを定義する	60	チャンネルの自動定義	90
デフォルト・オブジェクトの属性を表示す る	60	リモートから MQSC コマンドを発行する	91
ローカル・キュー定義をコピーする	61	MVS/ESA でのキュー・マネージャーの取 り扱い	92
ローカル・キューの属性を変更する	62	リモート・キューイングについての勧告	92
ローカル・キューをクリアする	63	リモートからの MQSC の使用に問題があ る場合	93
ローカル・キューを削除する	63	リモート・キューのローカル定義の作成	93
キューのブラウズ	63	リモート・キューのローカル定義の働きに ついて理解する	94
別名キューの取り扱い	69	例	94
別名キューを定義する	69	リモート・キューにメッセージを書き込む 代替方法	96
キュー別名に関してその他のコマンドを使 用する	71	リモート・キューに関してその他のコマン ドを使用する	96
モデル・キューの取り扱い	71		
モデル・キューを定義する	72		
モデル・キューに関してその他のコマンド を使用する	72		
トリガー操作のためのオブジェクトの管理	73		

伝送キューを作成する	97	コンテキスト権限	114
デフォルト伝送キュー	97	リモート・セキュリティーに関する考慮事項	115
リモート・キュー定義を別名として使用する	97	チャンネル・コマンド・セキュリティー	115
キュー・マネージャー別名	98	PCF コマンド	115
応答先キュー別名付け	98	MQSC チャンネル・コマンド	116
データ変換	99	チャンネル用の制御コマンド	116
キュー・マネージャーがメッセージを組		許可指定テーブルについての理解	117
み込み形式に変換できない場合	99	MQI 許可	117
ファイル ccsid.tbl	99	管理許可	122
ユーザー定義形式でのメッセージの変換	100	エスケープ PCF 中の MQSC コマンドに関する許可	122
キュー・マネージャー CCSID の変更	101	PCF コマンドについての許可	123
第7章 MQSeries オブジェクトの保護	103	許可ファイルの理解	125
MQSeries リソースを保護する理由	103	許可ファイルのパス	126
開始する前に	104	許可ディレクトリー	126
リソース ID MQM を持つ MQSeries for		許可ファイルの内容	127
Compaq OpenVMS のユーザー ID	104	クラス許可ファイル	129
詳細情報	105	全クラス許可ファイル	129
オブジェクト権限マネージャーについての理解	105	許可ファイルの管理	129
OAM の機能	106	許可ファイルについての許可	130
権限 ID を介してアクセスを管理する	106	第8章 MQSeries 送達不能キュー・ハンドラ	
権限 ID および 1 次権限 ID	106	ー	131
プリンシパルが複数の権限 ID を保持している場合	106	DLQ ハンドラーの呼び出し	131
デフォルト権限 ID	107	サンプル DLQ ハンドラー amqsdlq	132
OAM で保護できるリソース	107	DLQ ハンドラーのルール・テーブル	133
許可に権限 ID を使用する	108	制御データ	133
オブジェクト権限マネージャーを使用不可にする	108	ルール (パターンと処理)	135
オブジェクト権限マネージャー・コマンドの使用	109	パターン・マッチング・キーワード	136
OAM コマンドの使用時に指定するもの	109	アクション・キーワード	137
許可リスト	109	ルール・テーブルのルール	139
setmqaut コマンドを使用する	110	ルール・テーブルの処理方法	141
権限コマンドとインストール可能なサービス	111	すべての DLQ メッセージを確実に処理する	142
アクセス許可	111	DLQ ハンドラーのルール・テーブルの例	144
権限の表示コマンド	111	第9章 観測イベント	147
オブジェクト権限マネージャーのガイドライン	112	観測イベントとは	147
ユーザー ID	112	イベントを使用するのはなぜか	149
キュー・マネージャー・ディレクトリー	112	イベントのタイプ	149
キュー	113	イベント・キューによるイベント通知	150
代替ユーザー権限	113	トリガー・イベント・キューを使用する	151
		イベントの有効化および無効化	151
		イベント・メッセージ	152

第10章 トランザクション・サポート	153	リカバリー用のメディア・イメージ	187
データベースの調整	154	始動時に損傷オブジェクトをリカバリーす	188
制約事項	156	る	188
データベースの接続	156	その他の場合の損傷オブジェクトのリカバ	
データベース・マネージャーの構成	156	リー	189
スイッチ・ロード・ファイルの作成	157	MQSeries ログ・ファイルの保護	189
データベース・マネージャーの定義	157	バックアップと復元	189
Oracle の構成	159	MQSeries のバックアップ	190
Oracle の最小サポート・レベル	159	MQSeries の復元	190
環境変数の設定の確認	159	リカバリーの手順	191
Oracle XA サポートの使用可能化	160	ディスク・ドライブの障害	191
Oracle スイッチ・ロード・ファイルの作		キュー・マネージャーのオブジェクトの損	
成	160	傷	193
OpenVMS システム上での Oracle スイ		単独オブジェクトの損傷	193
ッチ・ロード・ファイルの作成	160	自動メディア・リカバリーの障害	193
Oracle 用の XAResourceManager 構成情報		dmpmqlog コマンドを使用したログの内容の	
の追加	161	ダンプ	194
Oracle 構成パラメーターの変更	163	第12章 名前サービスの使用	219
管理タスク	164	さまざまなキュー・マネージャー上のキュー	
未確定の作業単位	165	を共用するために DCE を使用する	219
dspmqtrn コマンドを使用する	165	共用キューのための構成タスク	219
rsvmqtrn コマンドを使用する	167	DCE 構成	220
エラーが混在する実行結果	168	第13章 MQSeries の構成	221
構成情報の変更	169	MQSeries 構成ファイル	221
データベース・マネージャーのインス		構成ファイルを編集する	222
タンスの削除	170	どのようなときに構成ファイルの編集	
第11章 リカバリーおよび再始動	173	が必要か	222
メッセージの消失を確実に回避する (ロギン		構成ファイルの優先順位	222
グ)	174	構成ファイルへの変更のインプリメン	
ログの概要	174	ト	223
ログ制御ファイル	174	MQSeries 構成ファイル、mqseries.ini	223
ログのタイプ	175	キュー・マネージャー構成ファイル qmseries.ini	223
循環ロギング	175	MQSeries 構成情報を変更するための属性	224
リニア・ロギング	175	AllQueueManagers スタンザ	224
チェックポイント機能 (完全なリカバリーの		ClientExitPath スタンザ	226
保証)	177	DefaultQueueManager スタンザ	226
ログのサイズの計算	180	ExitProperties スタンザ	226
ログの管理	182	LogDefaults スタンザ	227
ディスクが満杯になったときに何が起き		QueueManager スタンザ	230
か	183	キュー・マネージャー構成情報の変更	231
ログ・ファイルを管理する	184	Service スタンザ	231
ログ・ファイルの位置	185	ServiceComponent スタンザ	232
ログをリカバリーに使用する	185	Log スタンザ	233
問題からリカバリーする	185	XAResourceManager スタンザ	236
メディア・リカバリー	186		

Channels スタンザ	238	ログ・ファイル	262
LU62 および TCP スタンザ	240	早期エラー	263
ExitPath スタンザ	242	オペレーター・メッセージ	263
mqs.ini および qm.ini ファイルの例	242	エラー・ログの例	264
第14章 問題判別	247	送達不能キュー	267
予備的な検査	247	構成ファイルと問題判別	267
以前に MQSeries は正常に実行されたか	247	MQSeries トレースの使用	268
エラー・メッセージが出されたか	248	トレース・ファイル名	268
問題について説明した戻りコードが戻され		トレース・データのサンプル	268
ていないか	248	第 1 障害サポート・テクノロジー (FFST)	269
問題を再現できるか	248	FFST の読み方	269
最後に正常に実行された後、変更が行われ		クライアント問題判別	275
たか	248	クライアントの終了	275
アプリケーションは以前に正常に実行され		クライアントによるエラー・メッセージ	276
たか	249	OS/2、UNIX、および OpenVMS シス	
アプリケーションが以前に正常に実行		テム・クライアント	276
されなかった場合	250	DOS および Windows® クライアント	276
問題がネットワークの特定部分に影響する		第15章 パフォーマンスのチューニング	277
か	250	プロセス固有パラメーターの値の設定	279
1 日の特定の時刻に問題が起こるか	251	第16章 MQSeries for OpenVMS およびク	
問題は断続的に起こるか	251	ラスター化	281
サービス更新を適用したか	251	OpenVMS クラスタへの MQSeries のイン	
構成を適用する必要があるか	251	ストール	282
共通プログラミング・エラー	252	OpenVMS クラスタ・フェールオーバー・	
次に何を行うか	252	セット	283
不正な出力を得たか	253	OpenVMS クラスタ・フェールオーバ	
PCF コマンドからの応答を受け取れなか		ー・セットの概説	283
ったか	253	OpenVMS クラスタ・フェールオーバ	
キューの一部が障害を起こしているか	254	ー・セットの概念	284
問題はリモート・キューにのみ影響するか		OpenVMS クラスタ・フェールオーバ	
アプリケーション設計の考慮事項	256	ー・セットの構成の準備	286
メッセージ長の影響	256	OpenVMS クラスタ・フェールオーバ	
メッセージ持続性の影響	256	ー・セットの構成	286
特定のメッセージの検索	257	OpenVMS クラスタ・フェールオーバ	
長さが異なるメッセージを含んでいるキュー		ー・セットの構成後のタスク	287
.	257	FAILOVER.INI 構成ファイルの編集	288
同期点の頻度	257	フェールオーバー・セットにより使用され	
MQPUT1 呼び出しの使用	258	るコマンド・プロシージャ	289
不正出力	258	フェールオーバー・セットの管理	291
キューにメッセージが現れない場合	258	フェールオーバー・モニターの始動	292
メッセージが予期しない情報または破壊さ		フェールオーバー・セット内のキュー・マ	
れた情報を含んでいる場合	260	ネージャーの始動	292
分散キューを使用している場合の不正出力		フェールオーバー・セット内のキュー・マ	
の問題	260	ネージャーの終了	293
エラー・ログ	262		

フェールオーバー・セット内のキュー・マネージャーの移動	293
フェールオーバー・セットの状態の表示	294
フェールオーバー・セットの状態への DCL シンボルの設定	296
フェールオーバー・モニター・プロセスの停止	297
更新の進行中のコマンドの実行	298
フェールオーバー・セットの状態の変更	298
ICC アソシエーションのセキュリティーの設定	299
フェールオーバー・セットのトラブルシューティング	300
フェールオーバー・セットと MultiNet for OpenVMS の使用	301
フェールオーバー・セットの使用例	301
failover.template のカスタマイズ	302
フェールオーバー・セット・コマンド・プロシージャの変更	303
フェールオーバー・セット始動コマンド・プロシージャの例、start_failover_set.com	303
フェールオーバー・セット終了コマンド・プロシージャの例、end_failover_set.com	306

第1章 MQSeries の概要

この章では、管理者の観点から MQSeries for Compaq OpenVMS を紹介し、MQSeries およびメッセージングの基本概念について説明します。この章には、次の節があります。

- 『MQSeries とメッセージ・キューイング』
- 8ページの『メッセージとキュー』
- 10ページの『オブジェクト』
- 19ページの『システム・デフォルト・オブジェクト』
- 20ページの『ローカル管理およびリモート管理』
- 20ページの『クライアントおよびサーバー』
- 21ページの『キュー・マネージャー機能の拡張』
- 23ページの『セキュリティ』

MQSeries とメッセージ・キューイング

MQSeries により、アプリケーション・プログラムは、メッセージ・キューイングを使用して、メッセージ・ドリブン処理を行うことができます。適切なメッセージ・キューイング・ソフトウェア・プロダクトを使用すれば、プラットフォームが異なっても、アプリケーション・プログラム相互間で通信することができます。たとえば、OpenVMS アプリケーションと MVS/ESA[®] アプリケーションは、それぞれ MQSeries for Compaq OpenVMS と MQSeries for OS/390[®] を介して通信を行うことができます。アプリケーションは、基礎となる通信機構からは保護されています。

MQSeries プロダクトでは、アプリケーションがどのプラットフォームで実行される場合でも、共通アプリケーション・プログラミング・インターフェース (メッセージ・キュー・インターフェース: MQI) が実現されます。このため、アプリケーションを、あるプラットフォームから別のプラットフォームに移植するのが容易になります。

MQI については、MQSeries アプリケーション・プログラミング・リファレンス で詳しく説明されています。

時間に依存しないアプリケーション

メッセージ・キューイングを使用すれば、送信側プログラムと受信側プログラム間のメッセージ交換は、時間に依存しなくなります。つまり、送信側と受信

MQSeries とメッセージ・キューイング

側のアプリケーションの相互の結び付きがなくなり、送信側は、受信側からの受信の通知を待たずに処理を続行することができます。そればかりか、ターゲット・アプリケーションが、メッセージの送信時に、実行中である必要もありません。ターゲット・アプリケーションは、始動後にメッセージを取り出すことができます。

メッセージ・ドリブン処理

トリガー操作と呼ばれるメカニズムを使用して、メッセージがキューに到着したときに自動的にアプリケーションを開始できます。必要に応じて、メッセージの処理が完了したときに、アプリケーションを停止させることもできます。

メッセージとキュー

メッセージとキューは、メッセージ・キューイング・システムの基本コンポーネントです。

メッセージとは

メッセージとは、そのメッセージを使用するアプリケーションにとってなんらかの意味を持つ、何バイトかの長さのストリングです。メッセージは、ある 1 つのアプリケーションから別のアプリケーションに（または、同じアプリケーションの異なる部分に）情報を転送するために使用されます。アプリケーションが実行されるのは、同じプラットフォーム上でも、異なるプラットフォーム上でも構いません。

MQSeries のメッセージは次の 2 つの部分で構成されます。

- アプリケーション・データ

アプリケーション・データの内容と構造は、それらのデータを使用するアプリケーション・プログラムによって定義されます。

- メッセージ記述子

メッセージ記述子はそのメッセージを識別するものですが、この中には、そのメッセージのタイプや送信側のアプリケーションによって与えられた優先順位などの他の制御情報も格納されています。

メッセージ記述子の形式は、MQSeries によって定義されます。メッセージ記述子の詳細については、*MQSeries アプリケーション・プログラミング・リファレンス* を参照してください。

メッセージ長

MQSeries では、最大メッセージ長は 100 MB です (1 MB は 1 048 576 バイトとします)。ただし、実際には、次のものによってメッセージ長に制限がつけます。

- 受信側のキュー用に定義した最大メッセージ長
- キュー・マネージャー用に定義した最大メッセージ長
- 送信側のアプリケーションまたは受信側のアプリケーションで定義した最大メッセージ長
- メッセージ用として使用可能なストレージ量

1 つのアプリケーションが必要とするすべての情報を送るには、複数のメッセージが必要な場合があります。

キューとは

キューとは、メッセージを保管するためのデータ構造体です。メッセージは、アプリケーション・プログラムまたはキュー・マネージャーによって、通常の操作の一部としてキューに書き込まれます。

キューはすべて、キュー・マネージャーに属しています。キュー・マネージャーは、所有するキューを管理し、受信したすべてのメッセージを適切なキューに格納します。

キューの最大サイズは、2 GB です。キューに必要な記憶容量の計画についての情報は、*MQSeries Planning Guide* を参照してください。また、次の Web サイトには、プラットフォーム別のパフォーマンス・レポートがあります。

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

アプリケーションによるキューの送信 / 受信方法

アプリケーションでは、*MQI* 呼び出しを使用してメッセージを送受信します。たとえば、キューにメッセージを書き込むために、アプリケーションでは以下の処理が行われます。

1. MQI MQOPEN 呼び出しを発行して必要なキューをオープンする。
2. MQI MQPUT 呼び出しを発行してキューにメッセージを書き込む。
3. 別のアプリケーションがそのキューからメッセージを取り出すときには、MQI MQGET 呼び出しを発行する。

MQI 呼び出しの詳細については、*MQSeries アプリケーション・プログラミング・リファレンス* を参照してください。

メッセージとキュー

事前定義キューと動的キュー

キューは、その作成方法によって特性をもたせることができます。

- 事前定義キューは、管理者が該当のコマンド・セットを使用して作成するものです。たとえば、MQSC コマンド `DEFINE QLOCAL` は、事前定義ローカル・キューを作成します。事前定義キューは、永続キューであって、それらを使用するアプリケーションとは無関係に存在し、MQSeries が再始動しても存続します。
- 動的キューは、アプリケーションがモデル・キューの名前を指定してオープン要求を出したときに作成されるものです。作成されたキューは、モデル・キューであるテンプレート・キュー定義に基づいています。モデル・キューは、MQSC `DEFINE QMODEL` を使用して作成できます。モデル・キューの属性 (たとえば、キューに保管できるメッセージの最大数) は、そのモデル・キューから作成される動的キューが継承します。

モデル・キューは、作成される動的キューが永続キューになるか一時キューになるかを指定する属性を持っています。永続キューは、アプリケーションやキュー・マネージャーが再始動しても存続しますが、一時キューは、再始動すると失われます。

キューからメッセージを取り出す

MQSeries では、アプリケーションに適切な許可が与えられている場合、次の取り出しアルゴリズムに従ってキューからメッセージを取り出すことができます。

- 先入れ先出し法 (FIFO)
- メッセージ記述子に定義されたメッセージ優先順位。同じ優先順位を持つメッセージは、FIFO 順に取り出されます。
- 特定のメッセージについてのプログラム要求。

アプリケーションからの `MQGET` 要求によって、使用される方式が決まります。

オブジェクト

本書で説明しているタスクの多くには、MQSeries オブジェクト の操作が含まれています。MQSeriesバージョン 5.1 の場合、オブジェクトのタイプには、キュー・マネージャー、キュー、プロセス定義、チャンネル、クラスター、名前リストがあります。

オブジェクトの操作または管理では、以下のことが行われます。

- キュー・マネージャーの開始および停止。

- アプリケーション用のオブジェクト、特にキューの作成。
- チャネルを使用して、他の (リモート) システムにあるキュー・マネージャーへの通信パスを作成。これについては、*MQSeries 相互通信* で詳しく説明されています。
- キュー・マネージャーのクラスターを作成することによって、管理プロセス全体を簡易化する、または作業負荷のバランスをとる。

本書では、以下の各章で管理についての詳細情報を掲載しています。

- 25ページの『第2章 MQSeries 管理の紹介』
- 33ページの『第3章 制御コマンドによるキュー・マネージャーの管理』
- 45ページの『第4章 ローカル MQSeries オブジェクトの管理』
- 83ページの『第6章 リモート MQSeries オブジェクトの管理』

オブジェクト名

MQSeries オブジェクトに関する命名規則は、各オブジェクトによって異なります。

キュー・マネージャーの各インスタンスは、その名前で見分けられます。この名前は、相互接続されたキュー・マネージャーのネットワーク内で固有である必要があります。固有になっていると、あるキュー・マネージャーは、所定のメッセージを送るべきターゲットのキュー・マネージャーを明確に識別することができます。

残りの他のタイプのオブジェクトの場合、各オブジェクトにはそれぞれに関連付けられた名前があり、各オブジェクトはその名前で参照できます。これらの名前は、1 つのキュー・マネージャーおよびオブジェクト・タイプ内において固有のものである必要があります。たとえば、同じ名前のついたキューとプロセスを持つことはできますが、同じ名前の 2 つのキューを持つことはできません。

MQSeries では、名前は最大 48 文字までです。ただし、チャネル は例外で、最大 20 文字までです。名前についての詳細は、313ページの『MQSeries オブジェクトを命名する際の規則』を参照してください。

オブジェクトの管理

オブジェクトの作成、変更、表示、削除を実行するには、以下のものを使用します。

- 制御コマンド。キーボードから入力します。

オブジェクト

- MQSeriesコマンド (MQSC)。これは、キーボードからの入力とファイルからの入力の両方が可能です。
- プログラム式コマンド形式 (PCF) のコマンド。自動プログラムで使用することができます。
- プログラムからの MQSeries 管理インターフェース (MQAI) 呼び出し。

詳細については、25ページの『第2章 MQSeries 管理の紹介』を参照してください。

オブジェクトの属性

オブジェクトの特性は、そのオブジェクトの属性によって定義されます。ユーザー側で指定できる属性もあれば、表示しかできない属性もあります。たとえば、キューが収容できる最大メッセージ長は、*MaxMsgLength* 属性によって定義されます。この属性は、キューの作成時に指定できます。*DefinitionType* 属性は、キューが作成された方法を指定します。この属性は表示のみできません。

MQSeriesでは、属性を参照する方法には次の2通りがあります。

- 属性の PCF 名 (たとえば、*MaxMsgLength*) を使用する方法
- 属性の MQSC 名 (たとえば、MAXMSGL) を使用する方法

属性の正式の名前は、その属性の PCF 名です。MQSC 機能の使用は、本書の重要な部分を占めているので、本書の例では、与えられている属性について PCF 名よりも MQSC 名の方が多く使用されています。

MQSeries キュー・マネージャー

キュー・マネージャーは、キューイング・サービスをアプリケーションに提供し、キュー・マネージャーに属しているキューを管理します。キュー・マネージャーによって、次のことが確実に行われます。

- オブジェクトの属性は、受け取ったコマンドに応じて変更されます。
- 該当の条件が満たされたときに、トリガー・イベントや観測イベントなどの特殊イベントが生成されます。
- メッセージが、MQPUT 呼び出しを行ったアプリケーションの要求により、正しいキューに書き込まれます。正しいキューに入れられなかった場合には、アプリケーションに通知され、該当の理由コードが戻されます。

それぞれのキューは、1つのキュー・マネージャーに属しており、そのキュー・マネージャーに対してローカル・キュー であるといえます。アプリケーションが接続されているキュー・マネージャーは、そのアプリケーションに対し

てローカル・キュー・マネージャーであるといいます。アプリケーションのローカル・キュー・マネージャーに属しているキューは、そのアプリケーションのためのローカル・キューです。

リモート・キューとは、単に別の管理プログラムに属しているキューのことです。

リモート・キュー・マネージャーとは、ローカル・キュー・マネージャー以外の任意のキュー・マネージャーのことです。リモート・キュー・マネージャーは、ネットワーク内のリモート・マシン上にある場合と、ローカル・キュー・マネージャーと同じマシン上にある場合があります。

MQSeries では、1 つのマシン上で複数のキュー・マネージャーを使用することができます。

MQI 呼び出し

キュー・マネージャー・オブジェクトは、一部の MQI 呼び出しで使用される場合があります。たとえば、キュー・マネージャー・オブジェクトの属性について、MQI 呼び出しの MQINQ を使用して問い合わせることができます。

注: メッセージをキュー・マネージャー・オブジェクトに書き込むことはできません。メッセージは、必ずキュー・オブジェクトに書き込まれ、キュー・マネージャー・オブジェクトには書き込まれません。

MQSeries キュー

MQSeries に対してキューを定義するには、次のコマンドを使用します。

- 該当の MQSC DEFINE コマンド
- PCF キュー作成コマンド

これらのコマンドは、キューのタイプおよびキューの属性を指定します。たとえば、ローカル・キュー・オブジェクトは、アプリケーションがそのキューを MQI 呼び出しで参照したときに何が起こるかを指定する属性を持っています。属性には次のものがあります。

- アプリケーションがメッセージをキューから取り出せるかどうか (読み取り (GET) 可能)
- アプリケーションがメッセージをキューに書き込めるかどうか (書き込み (PUT) 可能)
- キューへのアクセスが、1 つのアプリケーション専用になるか、または複数のアプリケーションで共用されるか
- 同時にキューに保管できるメッセージの最大数 (キューの最大サイズ)

オブジェクト

- キューに書き込むことのできる最大メッセージ長

キュー・オブジェクトの定義についての詳細は、 *MQSeries* コマンド・リファレンス または *MQSeries* プログラム式システム管理 を参照してください。

キュー・オブジェクトの使用

MQSeries には、さまざまなタイプのキュー・オブジェクトがあります。オブジェクトのそれぞれのタイプは、プロダクト・コマンドで操作することができ、それぞれ異なる方法で実際のキューと関連付けられます。

• ローカル・キュー・オブジェクト

ローカル・キュー・オブジェクトは、アプリケーションが接続されているキュー・マネージャーに属するローカル・キューを識別します。それぞれのキューが 1 つのキュー・マネージャーに属しているという意味では、すべてのキューがローカル・キューであり、そのキュー・マネージャーについては、それに属しているキューがローカル・キューです。

• リモート・キュー・オブジェクト

リモート・キュー・オブジェクトは、他のキュー・マネージャーに属するキューを識別します。このキューは、そのキュー・マネージャーのローカル・キューとして定義する必要があります。リモート・キュー・オブジェクトを定義したときに指定した情報により、ローカル・キュー・マネージャーはリモート・キュー・マネージャーを見つけることができ、それにより、そのリモート・キュー宛てのすべてのメッセージは、正しいキュー・マネージャーに渡されるようになります。

別のキュー・マネージャー上のキューにメッセージを送信するには、あらかじめ伝送キューと宛先のキュー・マネージャーとの間のチャンネルを定義しておく必要があります。ただし、複数のキュー・マネージャーをクラスターにグループ化してある場合は、その必要はありません。クラスターについての詳細は、85ページの『クラスターを使用するリモート管理』を参照してください。

• 別名キュー・オブジェクト

別名キュー・オブジェクトにより、アプリケーションは、MQI 呼び出しで間接的にキューを参照することにより、そのキューをアクセスすることができます。別名キュー名が MQI 呼び出しで使用されると、その名前は、実行時に、ローカル・キューまたはリモート・キューの名前に解決されます。これにより、アプリケーションを変更しなくても、単に別名キュー定義を変更して、別名によって決定される新しいキューの名前になるようにするだけで、アプリケーションが使用するキューを変更することができます。

別名キューは、キューではありませんが、他のキューへのアクセスに使用できるオブジェクトです。

• モデル・キュー・オブジェクト

モデル・キュー・オブジェクトは、動的キュー作成のテンプレートとして使用するキュー属性のセットを定義します。動的キューは、モデル・キューの名前であるキュー名を指定した MQOPEN 要求をアプリケーションが出す時に、キュー・マネージャーによって作成されます。このようにして作成された動的キューはローカル・キューであり、その属性は、モデル・キュー定義から継承されます。動的キュー名は、アプリケーションで指定するか、あるいは、キュー・マネージャーがその名前を生成して、アプリケーションにそれを戻すことができます。

このようにして定義された動的キューは、プログラムの再始動後に残存しない一時キューである場合もあれば、再始動後も残存する永続キューである場合もあります。

MQSeries で使用される特別のローカル・キュー

MQSeries は、一部のローカル・キューを、その操作に関連する特定の目的のために使用します。これらのキューは、MQSeries が使用する前に**必ず**定義する必要があります。

アプリケーション・キュー: アプリケーションが (MQI を介して) 使用するキューのことを アプリケーション・キュー と言います。このキューは、アプリケーションが接続されているキュー・マネージャー上のローカル・キューの場合もあれば、別のキュー・マネージャーが所有するリモート・キューの場合もあります。

アプリケーションは、メッセージを、ローカル・キューやリモート・キューに入れることができます。ただし、メッセージを読み取ることができるのは、ローカル・キューからのみです。

開始キュー: 開始キュー は、トリガー操作に使用するキューです。キュー・マネージャーは、トリガー・イベントが発生すると、開始キューにトリガー・メッセージを書き込みます。トリガー・イベントとは、キュー・マネージャーによって検出される条件の論理的組み合わせのことです。たとえば、キュー上のメッセージの数が、あらかじめ定義されたキューのサイズに達したときにトリガー・イベントが生成される場合があります。このイベントが発生すると、キュー・マネージャーは指定の開始キューにトリガー・メッセージを入れることとなります。このトリガー・メッセージは、開始キューをモニターする特殊

オブジェクト

アプリケーションであるトリガー・モニター によって取り出されます。次に、トリガー・モニターは、トリガー・メッセージに指定されているアプリケーション・プログラムを始動します。

キュー・マネージャーでこのトリガー操作を使用する場合は、少なくとも 1 つの開始キューを、そのキュー・マネージャー用に定義する必要があります。

73ページの『トリガー操作のためのオブジェクトの管理』 および 376ページの『runmqtrm (トリガー・モニターの開始)』 を参照してください。トリガー操作の詳細については、 *MQSeries* アプリケーション・プログラミング・ガイド を参照してください。

伝送キュー: 伝送キュー は、リモートのキュー・マネージャー宛てのメッセージを一時的に保管するキューです。ローカル・キュー・マネージャーがメッセージを直接送信する各リモート・キュー・マネージャーごとに、少なくとも 1 つの伝送キューを定義する必要があります。これらのキューは、リモート管理にも使用されます (85ページの『MQSC コマンドを使用するローカル・キュー・マネージャーからのリモート管理』を参照)。分散キューイングでの伝送キューの使用法の詳細は、 *MQSeries* 相互通信 を参照してください。

クラスター伝送キュー: クラスター内のそれぞれのキュー・マネージャーには、 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` というクラスター伝送キューがあります。 *MQSeries* バージョン 5.1 では、このキューはすべてのキュー・マネージャーに対してデフォルトで定義されます。

クラスター内のキュー・マネージャーは、同じクラスター内のすべてのキュー・マネージャーに、伝送キュー上のメッセージを送ることができます。

クラスター内のキュー・マネージャーは、同一クラスターには属さないキュー・マネージャーと通信することができます。そのためには、従来の分散キューイング環境での場合と同様に、クラスター内のキュー・マネージャーから相手先のキュー・マネージャーに対して、チャンネルと伝送キューを定義する必要があります。

ネーム・レゾリューション時には、クラスター伝送キューがデフォルト伝送キューより優先されます。クラスター外のキュー・マネージャーがリモート・キューにメッセージを書き込む場合、宛先のキュー・マネージャーと同名の伝送キューがなければ、デフォルトでは、デフォルト伝送キューが使用されます。

クラスター内のキュー・マネージャーの場合は、宛先のキュー・マネージャーが同じクラスターに属さないかぎり、デフォルトでは `SYSTEM.CLUSTER.TRANSMIT.QUEUE` が使用されます。

送達不能キュー: 送達不能キューは、正しい宛先に渡すことができないメッセージを保管します。このようなことが起きるのは、たとえば、宛先キューが満杯である場合です。システムに提供された送達不能キューは、SYSTEM.DEAD.LETTER.QUEUE と呼ばれます。このキューは、他のプラットフォームでは、未配布メッセージ・キューとも呼ばれています。

分散キューイングでは、関係する各キュー・マネージャーごとに 1 つの送達不能キューを定義する必要があります。

コマンド・キュー: SYSTEM.ADMIN.COMMAND.QUEUE というコマンド・キューは、許可を与えられたアプリケーションが処理のために MQSeries コマンドを送ることができる送り先のローカル・キューです。次に、これらのコマンドは、コマンド・サーバーと呼ばれる MQSeries コンポーネントによって取り出されます。コマンド・サーバーは、それらのコマンドを検査し、正しいものをキュー・マネージャーの処理用に渡し、該当の応答先キューに応答を戻します。

キュー・マネージャーを作成すると、各キュー・マネージャーごとにコマンド・キューが自動的に作成されます。

応答先キュー: あるアプリケーションが要求メッセージを送信した場合、そのメッセージを受信するアプリケーションは、送信側のアプリケーションに応答メッセージを戻すことができます。この応答メッセージは、応答先キューと呼ばれるキューに書き込まれます。このキューは、通常は送信側のアプリケーションのローカル・キューです。応答先キューの名前は、送信側のアプリケーションによってメッセージ記述子の一部として指定されます。

イベント・キュー: MQSeries バージョン 5.1 は、MQI アプリケーションとは無関係にキュー・マネージャーをモニターするのに使用できる観測イベントをサポートします。観測イベントが生成される条件には、次のようなものがあります。

- 使用不能または存在しないキューに、アプリケーションがメッセージを書き込もうとした。
- キューが満杯になった。
- チャンネルが開始された。

観測イベントが発生すると、キュー・マネージャーはイベント・キューにイベント・メッセージを書き込みます。書き込まれたこのメッセージは、イベントが問題を提示すると、管理者に通知したり何らかの矯正処置を開始するモニター・アプリケーションにより、読み取られます。

オブジェクト

注: トリガー・イベントは、同じ条件では発生しない、イベント・メッセージが生成されない、という 2 点で、観測イベントとはまったく異なります。

観測イベントの詳細については、*MQSeries* プログラム式システム管理 を参照してください。

プロセス定義

プロセス定義オブジェクト は、*MQSeries* キュー・マネージャーでのトリガー・イベントに応答して開始されるアプリケーションを定義します。詳細については、15ページの『開始キュー』 を参照してください。

プロセス定義の属性には、アプリケーション ID 、アプリケーション・タイプ、およびアプリケーション特有のデータがあります。

プロセス定義を作成するには、*MQSC* コマンド `DEFINE PROCESS` または `PCF` コマンド `Create Process` を使用します。

チャネル

チャネル とは、あるキュー・マネージャーと別のキュー・マネージャーを結ぶ通信パスを提供するオブジェクトのことです。チャネルは、あるキュー・マネージャーから別のキュー・マネージャーにメッセージを移すために、分散メッセージ・キューイングで使用されます。チャネルは、基礎をなす通信プロトコルからアプリケーションを保護します。キュー・マネージャーは、同一のプラットフォーム上に存在する場合もあれば、異なるプラットフォーム上に存在する場合もあります。キュー・マネージャーが相互に通信できるようにするには、メッセージを送信するキュー・マネージャーに 1 つのチャネル・オブジェクトを定義し、メッセージを受信するキュー・マネージャーに別の補完的なチャネル・オブジェクトを定義する必要があります。

チャネルおよびその使用方法については、*MQSeries* 相互通信 および 87ページの『リモート管理のためにチャネルおよび伝送キューを作成する』を参照してください。

クラスター

分散キューイングを使用する従来の *MQSeries* ネットワークでは、すべてのキュー・マネージャーは独立しています。1つのキュー・マネージャーが別のキュー・マネージャーにメッセージを送信する必要がある場合、伝送キュー、リモート・キュー・マネージャーへのチャネル、メッセージの送信先であるすべてのキューのリモート・キュー定義を定義しておく必要があります。

クラスターは、複雑な伝送キュー、チャネル、およびキューを定義しなくても、複数のキュー・マネージャーが単一のネットワーク上で互いに直接通信できる方法でセットアップされたキュー・マネージャーのグループです。

注: MQSeries クラスターは、OpenVMS クラスターとは異なります。クラスター という語が用いられる場合、それは MQSeries キュー・マネージャー・クラスターを指します。OpenVMS クラスターは、常に *OpenVMS* クラスターと呼ばれます。OpenVMS クラスターの詳細については、281ページの『第16章 MQSeries for OpenVMS およびクラスター化』を参照してください。

クラスターについては、83ページの『第6章 リモート MQSeries オブジェクトの管理』および *MQSeries* キュー・マネージャー・クラスター を参照してください。

名前リスト

名前リストは、MQSeries オブジェクトの 1 つで、他の MQSeries オブジェクトのリストが格納されています。通常、名前リストはトリガー・モニターなどのアプリケーションによって、キュー・グループの識別に使用されます。名前リストを使用した場合の利点は、アプリケーションとは別個に管理できるということです。したがって、更新時に、その名前リストを使用しているアプリケーションを停止する必要はありません。また、あるアプリケーションで障害が起こった場合でも、名前リストには影響はなく、他のアプリケーションは引き続きその名前リストを使用できます。

名前リストは、キュー・マネージャーのクラスター用にも使用できます。これによって、複数の MQSeries オブジェクトが参照するクラスターのリストを管理することができます。

システム・デフォルト・オブジェクト

システム・デフォルト・オブジェクト とは、キュー・マネージャーの作成時に各キュー・マネージャーごとに自動的に作成される 1 組のオブジェクト定義のことです。インストール・システムのアプリケーションで使用するために、これらのオブジェクト定義はすべてコピーしたり、修正することができます。デフォルト・オブジェクト名には、語幹 `SYSTEM.DEF` が付いています。たとえば、デフォルト・ローカル・キューは `SYSTEM.DEFAULT.LOCAL.QUEUE` であり、デフォルト受信チャネルは `SYSTEM.DEF.RECEIVER` です。これらのオブジェクトの名前を変更することはできません。これらの名前を持つデフォルト・オブジェクトは必須です。

システム・デフォルト・オブジェクト

オブジェクトを定義する際に、明示的に指定されなかった属性は該当するデフォルト・オブジェクトからコピーされます。たとえば、ローカル・キューを定義する場合、指定しなかった属性は、デフォルト・キュー `SYSTEM.DEFAULT.LOCAL.QUEUE` から取られます。

システム・デフォルトについての詳細は、397ページの『付録B. システム・デフォルト』を参照してください。

ローカル管理およびリモート管理

ローカル管理とは、ローカル・システムに定義したキュー・マネージャーで管理タスクを実行することです。たとえば、TCP/IP の端末エミュレーション・プログラム **telnet** を介して、他のシステムにアクセスし、そこで管理作業を行うことができます。MQSeries では、これをローカル管理と考えることができます。なぜなら、チャンネルは無関係であり、通信はオペレーティング・システムによって管理されるからです。

MQSeries は、リモート管理 という管理方法による、1つの地点からの管理をサポートします。これにより、別のシステムで処理されるコマンドを、ユーザーのローカル・システムから出すことができます。この場合、別のシステムにログオンする必要はありませんが、適切なチャンネルを定義しておく必要があります。ターゲット・システム上のキュー・マネージャーおよびコマンド・サーバーは、実行中である必要があります。たとえば、リモート・コマンドを出して、リモート・キュー・マネージャー上のキュー定義を変更することができます。

一部のコマンドは、このような方法では発行することができません。特に、キュー・マネージャーの作成や開始、およびコマンド・サーバーの開始などの際には、このような方法では発行できません。この種のタスクを実行するためには、リモート・システムにログオンしてそこからコマンドを発行するか、あるいはユーザーの代わりにコマンドを発行するプロセスを作成する必要があります。

クライアントおよびサーバー

MQSeries は、MQSeries アプリケーション用のクライアント / サーバー構成をサポートします。

MQSeries クライアント は、マシン上にインストールされる MQSeries プロダクトの一部であり、アプリケーションからの MQI 呼び出しを受け入れて、それを MQI サーバー・マシンに渡します。MQI 呼び出しは、MQI サーバー・

マシン上でキュー・マネージャーによって処理されます。クライアントとサーバーは、通常は異なるマシン上にありますが、同一マシン上に存在する場合があります。

MQI サーバー とは、キューイング・サービスを 1 つまたは複数のクライアントに提供するキュー・マネージャーのことです。たとえばキューなどの *MQSeries* オブジェクトのすべては、キュー・マネージャー・マシン上、つまり、*MQI* サーバー・マシン上にのみ存在します。サーバーは、通常のローカル *MQSeries* アプリケーションもサポートできます。

MQI サーバーと通常のキュー・マネージャーとの相違点は、サーバーには、各クライアントとの専用通信リンクが備わっているという点です。クライアントおよびサーバー用のチャンネル作成方法についての詳細は、*MQSeries* 相互通信を参照してください。

一般的なクライアントのサポートについては、*MQSeries* クライアント を参照してください。

クライアント / サーバー環境での *MQSeries* アプリケーション

サーバーにリンクされていれば、クライアント *MQSeries* アプリケーションは、ローカル・アプリケーションの場合と同様にして *MQI* 呼び出しを発行することができます。クライアント・アプリケーションは、*MQCONN* 呼び出しを発行して、指定のキュー・マネージャーに接続します。接続要求から戻された接続ハンドルを指定した追加の *MQI* 呼び出しは、このキュー・マネージャーによって処理されます。

ユーザーのアプリケーションを、該当のクライアント・ライブラリーにリンクする必要があります。詳細については、*MQSeries* クライアント を参照してください。

キュー・マネージャー機能の拡張

キュー・マネージャーが提供する機能は、次のものによって拡張できます。

- ユーザー出口
- インストール可能なサービス

ユーザー出口

ユーザー出口は、独自のコードをキュー・マネージャー機能に挿入するメカニズムを提供します。サポートされているユーザー出口は次のとおりです。

- **チャンネル出口**

この出口では、チャンネルの動作方法を変更できます。チャンネル出口については、*MQSeries 相互通信* で説明されています。

- **データ変換出口**

この出口では、アプリケーション・プログラムに書き込んでデータの形式を変換するための部分ソース・コードを作成できます。データ変換出口については、*MQSeries アプリケーション・プログラミング・ガイド* で説明されています。

- **クラスター・ワークロード出口**

この出口により実行できる機能は、出口のプロバイダーにより定義されます。呼び出し定義情報については、*MQSeries キュー・マネージャー・クラスター* で説明されています。

すべての出口タイプは、分散キューイングに関連付けられています。これらの出口およびその使用方法については、*MQSeries 相互通信* を参照してください。

インストール可能なサービス

インストール可能なサービスは、複数のエントリー・ポイントを持つ形式化されたインターフェース (API) があるという点で、出口よりもさらに利用範囲の広いものです。

インストール可能なサービスは、サービス・コンポーネント と呼ばれるもので実現されます。プロダクトで提供されているコンポーネントを使用することもできますし、必要な機能を実行するコンポーネントを独自に作成することもできます。

現在、次のようなインストール可能なサービスが用意されています。

- **許可サービス。**

これにより、独自のセキュリティー機能を構築することができます。

このサービスを実現するデフォルト・サービス・コンポーネントは、オブジェクト権限マネージャー (OAM) です。これはプロダクトと共に提供されています。デフォルトでは、OAM はアクティブになっています。つまり、OAM を構成するための作業は一切必要ありません。許可サービス・インターフェースを使用して別のコンポーネントを作成し、OAM を置き換えたり

OAM を補強したりすることができます。OAM の詳細については、103ページの『第7章 MQSeries オブジェクトの保護』を参照してください。

- **名前サービス。**

これにより、アプリケーションはリモート・キューをローカル・キューであるかのように認識することができ、キューを共用できます。このサービスを実現するデフォルト・サービス・コンポーネントは、MQSeries バージョン 5.1 で提供されています。この MQSeries V5 プロダクトでは、オープン・ソフトウェア・ファウンデーション (OSF) 分散コンピューター環境 (DCE) を使用しています。たとえば、DCE をインストールしていない場合は、独自の名前サービス・コンポーネントを作成することもできます。デフォルトでは、名前サービスは使用できない状態になっています。

219ページの『第12章 名前サービスの使用』 および *MQSeries* プログラム式システム管理 を参照してください。

セキュリティ

MQSeries バージョン 5 プロダクトでは、次の 2 つの方法によりセキュリティを提供しています。

- オブジェクト権限マネージャー (Object Authority Manager: OAM) 機能
- DCE セキュリティー

オブジェクト権限マネージャー (OAM) 機能

MQI 呼び出しの使用、コマンドの発行、およびオブジェクトへのアクセスのための許可は、オブジェクト権限マネージャー (OAM) によって提供されます。OAM はデフォルトでは使用可能になっています。MQSeries エンティティーへのアクセスは、MQSeries ユーザー・グループと OAM によって制御されます。管理者が必要に応じて許可を与えたり取り消したりできるように、コマンド行インターフェースが提供されています。

許可サービス・コンポーネントについては、*MQSeries* プログラム式システム管理 を参照してください。

DCE セキュリティー

DCE Generic Security Service (GSS) を使用するチャネル出口が MQSeries によって提供されています。詳細については、*MQSeries* 相互通信 を参照してください。

トランザクション・サポート

アプリケーション・プログラムは、一連の更新内容を 1 つの作業単位としてグループ化します。作業単位としてグループ化される更新は相互に論理的に関連しているため、データの一貫性を維持できるように正常に処理されなければなりません。ある更新処理が正常に行われ、一方で他の更新処理が失敗すると、データの一貫性が失われます。

処理が正常に完了すると、作業単位はコミットされます。この時点で作業単位内のすべての更新内容が永久的なものとなり、これ以降は取り消し不可能になります。作業単位が失敗した場合は、すべての更新がバックアウトされます。同期点調整とは、データの一貫性を維持しながら作業単位をコミットまたはバックアウトするプロセスです。

ローカル 作業単位では、更新されるリソースのみが MQSeries キュー・マネージャーのリソースとなります。この場合、キュー・マネージャー自体が単一フェーズ・コミットによって同期点を調整します。

グローバル 作業単位では、XA 準拠データベースなどの他のリソース・マネージャーにより管理されているリソースも更新されます。この場合、2 フェーズ・コミット手順を実行する必要があり、作業単位はキュー・マネージャー自体によって調整されます。

詳細については、153ページの『第10章 トランザクション・サポート』を参照してください。

第2章 MQSeries 管理の紹介

この章では、MQSeries での管理の問題について説明します。

管理タスクには、MQSeries オブジェクト (キュー・マネージャー、キュー、プロセス、名前リスト、クラスター、およびチャネル) の作成、起動、変更、表示、停止、および削除があります。

この章には次の節があります。

- 『ローカル管理およびリモート管理』
- 26ページの『制御コマンドによる管理タスクの実行』
- 26ページの『MQSC コマンドによる管理タスクの実行』
- 27ページの『PCF コマンドによる管理タスクの実行』
- 28ページの『MQSeries のファイル名についての理解』
- 30ページの『大文字小文字の区別についての理解』

ローカル管理およびリモート管理

MQSeries オブジェクトに対しては、ローカル管理またはリモート管理ができます。

ローカル管理とは、ローカル・システムに定義したキュー・マネージャーで管理タスクを実行することです。たとえば、TCP/IP の端末エミュレーション・プログラム telnet を介して、他のシステムにアクセスし、そこで管理作業を行うことができます。MQSeries では、これをローカル管理と考えることができます。なぜなら、チャネルは無関係であり、通信はオペレーティング・システムによって管理されるからです。

MQSeries は、リモート管理という管理方法により、1 つの地点からの管理をサポートします。これにより、別のシステムで処理されるコマンドを、ユーザーのローカル・システムから出すことができます。この場合、別のシステムにログオンする必要はありませんが、適切なチャネルを定義しておく必要があります。ターゲット・システム上のキュー・マネージャーおよびコマンド・サーバーは、実行中である必要があります。たとえば、リモート・コマンドを出して、リモート・キュー・マネージャー上のキュー定義を変更することができます。

ローカル管理およびリモート管理

一部のコマンドは、このような方法では発行することができません。特に、キュー・マネージャーの作成や開始、およびコマンド・サーバーの開始などの際には、このような方法では発行できません。この種のタスクを実行するためには、リモート・システムにログオンしてそこからコマンドを発行するか、あるいはユーザーの代わりにコマンドを発行するプロセスを作成する必要があります。

リモート管理の詳細は、83ページの『第6章 リモート MQSeries オブジェクトの管理』に記載されています。

制御コマンドによる管理タスクの実行

制御コマンドを使用すると、キュー・マネージャーそのものに対して管理タスクを実行することができます。

制御コマンドについての詳細は、33ページの『第3章 制御コマンドによるキュー・マネージャーの管理』を参照してください。

MQSC コマンドによる管理タスクの実行

キュー・マネージャー自体、チャンネル、キュー、およびプロセス定義などのキュー・マネージャー・オブジェクトを管理するためには、MQSeries コマンド (MQSC) を使用します。たとえば、指定したキューを定義、更新、表示、および削除するためのコマンドがあります。

MQSC コマンドを実行するには、コマンド行から制御コマンド **runmqsc** を起動します。MQSC コマンドは、次のように実行できます。

- キーボードからタイプして対話式に実行する。49ページの『MQSC 機能を対話式で使用する』を参照してください。
- ASCII テキスト・ファイルからコマンド・シーケンスとして実行する。52ページの『テキスト・ファイルからの MQSCコマンドの実行』を参照してください。

コマンドに設定したフラグによって、**runmqsc** コマンドを次の3通りのモードで実行できます。

- 検証モード。このモードでは、MQSC コマンドはローカル・キュー・マネージャー上で検査され、実際には実行されません。
- 直接モード。このモードでは、MQSC コマンドはローカル・キュー・マネージャー上で実行されます。

- 間接モード。このモードでは、MQSC コマンドはリモート・キュー・マネージャー上で実行されます。

MQSC 機能およびテキスト・ファイルの使い方の詳細については、52ページの『テキスト・ファイルからの MQSCコマンドの実行』を参照してください。
runmqsc コマンドの詳細については、371ページの『runmqsc (MQSeries コマンドの実行)』を参照してください。

MQSC に指定するオブジェクト属性は、本書では大文字 (たとえば、RQMNAME) で示されます。ただし、大文字小文字の区別はありません。MQSC 属性名は 8 文字までに制限されています。

MQSC コマンドは、AS/400[®] および OS/390 などの他のプラットフォームでも使用できます。

MQSC コマンドについては、409ページの『付録D. コマンド・セットの比較』にその概要があります。

各 MQSC コマンドとその構文については、MQSeries コマンド・リファレンスを参照してください。

MQSC コマンドを使用したローカル管理についての詳細は、47ページの『MQSC コマンドによるローカル管理タスクの実行』を参照してください。

PCF コマンドによる管理タスクの実行

MQSeries プログラム式コマンド形式 (PCF) コマンドの目的は、管理タスクを管理プログラムに組み込めるようにすることです。このようにして、プログラムから、キューやプロセス定義を作成したり、キュー・マネージャーを変更したりすることができます。

PCF コマンドは、MQSC 機能が提供する機能と同様の処理を行います。

詳細については、77ページの『PCF コマンド』を参照してください。

PCF データ構造およびそれをインプリメントする方法の詳しい説明については、MQSeries プログラム式システム管理を参照してください。

MQSeries 管理インターフェース (MQAI) を使用すると、PCF メッセージへのプログラミング上のアクセスが容易になります。詳細については、79ページの『MQAI を使用して PCF の使い方を単純化する』を参照してください。

MQSC および PCF での属性

MQSC に指定するオブジェクト属性は、本書では大文字 (たとえば、RQMNAME) で示されます。ただし、大文字小文字の区別はありません。これらの属性名は 8 文字までに制限されているので、意味を理解するのが難しいものもあります (たとえば、QDPHIEV)。PCF のオブジェクト属性はイタリック体で表示され、8 文字までに制限されていないので、読みやすくなっています。RQMNAME と同等の PCF は *RemoteQMgrName* で、QDPHIEV と同等の PCF は *QDepthHighEvent* です。

エスケープ PCF

エスケープ PCF は、メッセージ・テキスト内に MQSC コマンドを含んでいる PCF コマンドです。PCF を使用して、リモート・キュー・マネージャーにコマンドを送信することができます。エスケープ PCF の使い方の詳細については、*MQSeries プログラム式システム管理* を参照してください。

MQSeries のファイル名についての理解

MQSeries のキュー、キュー・マネージャー、名前リスト、およびプロセスの各オブジェクトは、それぞれファイルによって表されます。これらのオブジェクト名は必ずしも有効なファイル名ではないので、キュー・マネージャーは、必要に応じてそのオブジェクト名を有効なファイル名に変換します。

キュー・マネージャー・ディレクトリーへのパスは、次のものから作られます。

- 接頭部 - 名前の最初の部分

`MQS_ROOT:[MQM]`

接頭部は、キュー・マネージャー構成ファイルに定義されています。

- リテラル

`QMGRS`

- コード化されたキュー・マネージャー名。これは、有効なディレクトリー名に変換されたキュー・マネージャー名です。たとえば、

`QUEUE.MANAGER`

というキュー・マネージャーは、次のように表されます。

`QUEUE$MANAGER`

この処理のことを、**名前変換** と呼びます。

キュー・マネージャー名の変換

MQSeries では、キュー・マネージャーに 48 文字までの名前を付けることができます。

たとえば、キュー・マネージャーに次の名前を付けることができます。

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

しかし、各キュー・マネージャーはファイルで表されるため、ファイル名の最大長および名前に使用できる文字に制限があります。そのため、オブジェクトを表すファイルの名前は、ファイル・システムの要件に合うように自動的に変換されます。

名前が `QUEUE.MANAGER` であるキュー・マネージャーの例を使用した、キュー・マネージャー名の変換の規則を以下に示します。

1. 個々の文字が次のように変換されます。
 - ・ は \$ に変換
 - / は _ に変換
 - % は _ に変換
2. それでも名前が正しくない場合
 - a. 8 文字に切り捨てられます。
 - b. 3 文字の数値接尾部が付加されます。

たとえば、デフォルト接頭部であるとする、キュー・マネージャー名は次のようになります。

```
MQS_ROOT:[MQM.QMGRS.QUEUE$MANAGER]
```

変換アルゴリズムでは、大文字小文字の区別のないファイル・システム上で、大文字小文字の違いしかない名前を区別することもできます。

オブジェクト名の変換

オブジェクト名は、必ずしも有効なファイル・システム名になっていません。そのため、オブジェクト名を変換しなければならない場合があります。使用される変換方法は、キュー・マネージャー名の場合とは異なります。つまり、1 つのマシンにつきキュー・マネージャー名はわずかしかありませんが、各キュー・マネージャーごとに相当数の他のオブジェクトが存在する可能性があるためです。ファイル・システムには、プロセス定義、キュー、および名前リストのみが提示されます。チャンネルは関係しません。

MQSeries の名前についての理解

変換処理で新しい名前が生成されるときに、元のオブジェクト名との関係は簡単ではありません。 **dspmqls** コマンドを使用して、本当のオブジェクト名と変換されたオブジェクト名の間で変換を行うことができます。

キュー・ファイル名は、文字 “Q” で始まります。

オブジェクトの命名の詳細については、313ページの『MQSeries オブジェクトを命名する際の規則』を参照してください。

大文字小文字の区別についての理解

制御コマンドでの大文字小文字の区別

通常、OpenVMS は、大文字小文字を区別しないオペレーティング・システムであると説明されます。つまり、一般に、次の 3 つのコマンドはすべて “QUEUEMANAGER” というキュー・マネージャーを作成するということです。

```
$ crtmqm QueueManager
$ crtmqm queuemanager
$ crtmqm QUEUEMANAGER
```

MQSeries for Compaq OpenVMS では、キュー・マネージャーの名前 (または類似のパラメーター) を二重引用符で囲むことによって、大文字小文字の別を保つことができます。二重引用符を使用すると、次の 3 つのコマンドは、3 つの異なるキュー・マネージャーを作成するようになります。

```
$ crtmqm "QueueManager"   QueueManager というキュー・マネージャーを作成します
$ crtmqm "queuemanager"   queuemanager というキュー・マネージャーを作成します
$ crtmqm "QUEUEMANAGER"  QUEUEMANAGER というキュー・マネージャーを作成します
```

OpenVMS バージョン 7.2 には、次の新しいコマンドが導入されています。

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

このコマンドは、OpenVMS が大文字小文字を処理する方法を変更します。

set process /parse_style コマンドを使用しない場合、または **traditional** オプションを指定してこのコマンドを使用する場合は、OpenVMS は、大文字小文字の区別に関してこれまで動作してきたのと同じように動作します。

extended オプションを指定してこのコマンドを使用すると、LIB\$GET_FOREIGN ランタイム・ライブラリー・ルーチンの動作は、検索するテキストの大文字小文字を保持するよう変更されます。MQSeries は、このルーチンを使用してコマンド行パラメーターを入手するので、パラメーターを二重引用符で囲まなくても、パラメーターの大文字小文字は保持されます。

たとえば、次のコマンド・シーケンスは、3 つの異なるキュー・マネージャーを作成します。パラメーターが二重引用符で囲まれていないことに注意してください。

```
$ set process /parse_style = extended
$ crtmqm QueueManager QueueManager というキュー・マネージャーを作成します
$ crtmqm queuemanager queuemanager というキュー・マネージャーを作成します
$ crtmqm QUEUEMANAGER QUEUEMANAGER というキュー・マネージャーを作成します
```

OpenVMS の **set process /parse_style** コマンドは、大文字小文字の区別の他に多くのことを変更します。OpenVMS DCL 辞書で提供されている情報からこのコマンドについての理解を深めた後で、このコマンドをシステムに適用することができます。

MQSC コマンドでの大文字小文字の区別

MQSeries 制御コマンド (たとえば、MQSC 機能呼び出す **runmqsc**) は、大文字小文字を区別しません。

MQSC コマンドは、属性も含めて、大文字で書いても小文字で書いても構いません。MQSC コマンドの中のオブジェクト名は、単一引用符で名前を囲まない限り、自動的に大文字に変換されます。単一引用符を使用しないと、オブジェクトは大文字の名前で処理されます。詳細については、MQSeries コマンド・リファレンス を参照してください。

第3章 制御コマンドによるキュー・マネージャーの管理

この章では、キュー・マネージャーとコマンド・サーバーに対する操作の実行方法について説明します。この章には、次の節があります。

- 『制御コマンドの使用』
- 35ページの『キュー・マネージャーの作成のためのガイドライン』
- 40ページの『デフォルト・キュー・マネージャーを作成する』
- 41ページの『キュー・マネージャーを開始する』
- 41ページの『既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする』
- 42ページの『キュー・マネージャーを停止する』
- 44ページの『キュー・マネージャーを再始動する』
- 44ページの『キュー・マネージャーを削除する』
- 314ページの『オブジェクト・ファイルの表示』

制御コマンドの使用

制御コマンドを使用して、キュー・マネージャー、コマンド・サーバーおよびチャンネルに対する操作を実行できます。表1に示すように、制御コマンドは3つのカテゴリに分類できます。

表1. 制御コマンドのカテゴリ

カテゴリ	説明
キュー・マネージャー・コマンド	キュー・マネージャーの制御コマンドには、キュー・マネージャーおよびコマンド・サーバーを作成、開始、終了、および削除するためのコマンドが含まれます。
チャンネル・コマンド	チャンネル・コマンドには、チャンネルやチャンネル開始プログラムを開始したり、停止するコマンドが含まれます。

制御コマンドの使用

表 1. 制御コマンドのカテゴリ (続き)

カテゴリ	説明
ユーティリティー・コマンド	ユーティリティー・コマンド以下のものと関連したコマンドが含まれます。 <ul style="list-style-type: none">• MQSC コマンドの実行• 変換出口• 許可管理• キュー・マネージャー・リソースのメディア・イメージの記録および回復• トランザクションの表示および解決• トリガー・モニター• MQSeries オブジェクトのファイル名の表示

チャンネルの管理タスクに関する詳細については、*MQSeries 相互通信* を参照してください。

制御コマンドの使用

MQSeries for Compaq OpenVMS では、制御コマンドは DCL プロンプトで入力します。コマンド名およびフラグ自体は大文字小文字の区別はありませんが、パラメーターが大文字に変換される場合もあれば、されない場合もあります。これは、OpenVMS プロセス・オプションや、大文字小文字を保護するためにパラメーターを二重引用符で囲むかどうかによって異なります。OpenVMS コマンドと二重引用符が大文字小文字に影響する方法の詳細については、30ページの『大文字小文字の区別についての理解』を参照してください。

一般に、この例の場合は、次のようになります。

```
crtmqm -u system.dead.letter.queue "jupiter.queue.manager"
```

- 送達不能キューは、小文字で入力しても SYSTEM.DEAD.LETTER.QUEUE になることがあります。文字が自動的に大文字に変換されるかどうかは、OpenVMS コマンド **set process/parse_style** の設定値によって決まります。30ページの『大文字小文字の区別についての理解』を参照してください。

- キュー・マネージャー名は、二重引用符で囲んでいるので
"jupiter.queue.manager" ("JUPITER.queue.manager" とは異なる) として指定されます。

したがって、コマンドは例に示されているとおりに入力してください。

キュー・マネージャーを作成する

キュー・マネージャーは、関連したリソース、特にキュー・マネージャー自体が所有しているキューを管理します。キュー・マネージャーは、メッセージ・キューイング・インターフェース (MQI) 呼び出しおよびコマンドが MQSeries オブジェクトを作成、修正、表示、および削除できるように、アプリケーションに対してキューイング・サービスを提供します。

メッセージおよびキューに関して何かを行うためには、その前に少なくとも 1 つのキュー・マネージャーとそれに関連するオブジェクトを作成しておく必要があります。キュー・マネージャーを作成するには、MQSeries 制御コマンド **crtmqm** を使用します。**crtmqm** コマンドは、必要なデフォルト・オブジェクトおよびシステム・オブジェクトを自動的に作成します。デフォルト・オブジェクトは、作成されるすべてのオブジェクト定義の基礎となるものです。システム・オブジェクトはキュー・マネージャーの操作に必要なものです。キュー・マネージャーとそのオブジェクトが作成されている場合は、**strmqm** コマンドを使用してそのキュー・マネージャーを開始します。

キュー・マネージャーの作成のためのガイドライン

キュー・マネージャーを作成する前に、(特に、実稼働環境では) いくつかの点を考慮する必要があります。次のチェックリストを調べてください。

- 固有のキュー・マネージャー名の指定。
- キュー・マネージャーの数の制限。
- デフォルト・キュー・マネージャーの指定。
- 送達不能キューの指定。
- デフォルト・伝送キューの指定。
- 必須ロギング・パラメーターの指定。
- キュー・マネージャーの作成後の構成ファイルのバックアップ。

このリストの作業について、これ以降の節で説明します。

キュー・マネージャーの作成

固有のキュー・マネージャー名の指定

キュー・マネージャーを作成するときには、ネットワーク内のどこにも同じ名前のキュー・マネージャーがないことを確かめる必要があります。キュー・マネージャーの名前は、作成時には検査されません。固有の名前でないと、分散キューイング用のチャンネルを使用することができなくなります。

確実に固有の名前になるようにする方法の1つは、各キュー・マネージャー名に、接頭部として固有のノード名を付けることです。たとえば、ノードの名前が `accounts` の場合は、キュー・マネージャーの名前を `accounts.saturn.queue.manager` にすることもできます。ここで、`saturn` は特定のキュー・マネージャーを識別し、`queue.manager` はすべてのキュー・マネージャーに与えることができる拡張子です。この拡張子は省略することもできますが、その場合、`accounts.saturn` と `accounts.saturn.queue.manager` とは異なる キュー・マネージャー名になることに注意してください。

他の企業との通信のために `MQSeries` を使用する場合には、自分の企業を接頭部として組み込むこともできます。この方法の具体例は示しません。この方法を一層理解しにくくしてしまうためです。

注: 制御コマンドでのキュー・マネージャー名は、大文字に変換される場合もあれば、されない場合もあります。これは、`OpenVMS` プロセス・オプションや、大文字小文字を保護するために名前を二重引用符で囲むかどうかによって異なります。つまり、`jupiter.queue.manager` という名前と `JUPITER.queue.manager` という名前の2つのキュー・マネージャーを作成できるということです。 `OpenVMS` 処理オプションと二重引用符が大文字小文字に影響する方法の詳細については、30ページの『大文字小文字の区別についての理解』を参照してください。

キュー・マネージャーの数の制限

リソースに余裕がある限り、いくつでもキュー・マネージャーを作成できます。ただし、各キュー・マネージャーは、独自のリソースを必要とするため、それぞれが10個のキューを持つ10個のキュー・マネージャーを作成するよりも、ノードに100個のキューを持つ1個のキュー・マネージャーを作成する方が一般的には有効です。

実動システムでは、多数のノードが1つのキュー・マネージャーで実行されますが、大規模なサーバー・マシンは複数のキュー・マネージャーで実行される場合があります。

デフォルト・キュー・マネージャーの指定

デフォルト・キュー・マネージャーを持たない MQSeries をノード上に構成することは可能ですが、各ノードにデフォルト・キュー・マネージャーを 1 つずつ持つのが望ましいかたちです。

キュー・マネージャーを作成するには、**crtmqm** コマンドを使用します。このコマンドおよびそのパラメーターの詳細については、320ページの『**crtmqm** (キュー・マネージャーの作成)』を参照してください。

デフォルト・キュー・マネージャーとは

デフォルト・キュー・マネージャーとは、アプリケーションが MQCONN 呼び出しでキュー・マネージャー名を指定しなかった場合に、そのアプリケーションが接続されるキュー・マネージャーのことです。また、これは、キュー・マネージャー名を指定せずに **runmqsc** コマンドを呼び出したときに MQSC コマンドを処理するキュー・マネージャーのことでもあります。

デフォルト・キュー・マネージャーの指定方法

-q フラグを **crtmqm** コマンドに組み込み、作成中のキュー・マネージャーをデフォルト・キュー・マネージャーとして指定します。作成中のキュー・マネージャーをデフォルト・キュー・マネージャーにしない場合は、このフラグを省略してください。

あるキュー・マネージャーをデフォルト・キュー・マネージャーとして指定すると、ノードについての既存のデフォルト・キュー・マネージャーの指定は置き換えられます。

デフォルト・キュー・マネージャーを変更する場合の影響

デフォルト・キュー・マネージャーを変更すると決める場合、他のユーザーまたはアプリケーションに影響を与える可能性があることに注意してください。変更しても、現在接続されているアプリケーションに影響はありません。それらのアプリケーションは、以後の MQI 呼び出しで元の接続呼び出しからのハンドルを使用できるためです。このハンドルにより、呼び出しは確実に同じキュー・マネージャーに対して出されます。変更後に接続されるアプリケーションは、新しいデフォルト・キュー・マネージャーに接続されます。

こうしたことを行う場合には、デフォルト・キュー・マネージャーを変更する場合はその前に、上記の事柄を考慮する必要があります。

送達不能キューの指定

送達不能キューとは、正しい宛先に送ることができなかったメッセージを入れるローカル・キューのことです。

重要:

ネットワークの各キュー・マネージャーが送達不能キューを持っていることが非常に重要です。キュー・マネージャーがこれを持っていないと、アプリケーション・プログラムのエラーによってチャネルがクローズされたり、管理者コマンドに対する応答が受信されなかったりする場合があります。

たとえば、アプリケーションが別のキュー・マネージャーのキューにメッセージを入れようとしたときに、誤ったキュー名が与えられると、チャネルが停止し、メッセージは伝送キューに残ったままになります。すると、他のアプリケーションは、このチャネルを使用して自分のメッセージを送ることができなくなります。

キュー・マネージャーに送達不能キューがある場合には、チャネルは影響を受けません。未送達メッセージは受信側の送達不能キューに入れられるだけで、チャネルとその伝送キューはそのまま使用できます。

したがって、キュー・マネージャーを作成するときには、`-u` フラグを使用して送達不能キューの名前を指定する必要があります。MQSC コマンドを使用してキュー・マネージャーの属性を更新し、使用する送達不能キューを指定することもできます。MQSC ALTER コマンドの例については、52ページの『キュー・マネージャーの属性を更新する』を参照してください。

送達不能キュー上にメッセージが見つかった場合、MQSeries と共に提供されている送達不能キュー・ハンドラーを使用して、それらのメッセージを処理することができます。送達不能キュー・ハンドラーの詳細と、送達不能キューを指定していれば送達不能キューに入れられるようなメッセージの数を少なくする方法については、131ページの『第8章 MQSeries 送達不能キュー・ハンドラー』を参照してください。

デフォルト伝送キューの指定

伝送キューとは、リモート・キュー・マネージャーへ送られる途中のメッセージが、伝送されるまでの間キューイングされるローカル・キューのことです。デフォルト伝送キューとは、伝送キューが明示的に定義されなかった場合に使用されるキューのことです。各キュー・マネージャーには、1つのデフォルト伝送キューを割り当てることができます。

キュー・マネージャーを作成するときには、`-d` フラグを使用してデフォルト伝送キューの名前を指定する必要があります。これによって実際にキューが作成されるわけではありません。後で明示的にキューを作成する必要があります。詳細については、58ページの『ローカル・キューの取り扱い』を参照してください。

必須ロギング・パラメーターの指定

crtmqm コマンドには、ロギングのタイプ、およびログ・ファイルのパスとサイズなどのログ・パラメーターを指定できます。開発環境では、デフォルト・ロギング・パラメーターで十分です。しかし、たとえば、次のような場合にはデフォルトを変更できます。

- 大きなログをサポートできないローエンドのシステム構成である場合
- 同時に多数の長いメッセージがキューに留まることが予想される場合

ロギング・パラメーターの指定の詳細に関しては、以下を参照してください。

- **crtmqm** コマンドについては、320ページの『**crtmqm** (キュー・マネージャーの作成)』を参照してください。
- 構成ファイルの使用については、233ページの『Log スタンザ』を参照してください。

キュー・マネージャーの作成後の構成ファイルのバックアップ

考慮すべき構成ファイルには、次の2つがあります。

1. プロダクトをインストールするときに、MQSeries 構成ファイル (`mqs.ini`) が作成されます。これには、キュー・マネージャーのリストが含まれます。このファイルは、キュー・マネージャーを作成したり削除したりするたびに更新されます。1 ノードに1つの `mqs.ini` ファイルがあります。
2. 新しいキュー・マネージャーを作成するときに、新しいキュー・マネージャー構成ファイル (`qm.ini`) が自動的に作成されます。これには、そのキュー・マネージャーの構成パラメーターが含まれます。

これらのファイルのバックアップを作成する必要があります。問題の原因となる別のキュー・マネージャーを後で作成した場合、問題の原因を取り除いたときにバックアップを回復させることができます。一般に、新しいキュー・マネージャーを作成するたびに、構成ファイルのバックアップを取る必要があります。

構成ファイルの詳細については、221ページの『第13章 MQSeries の構成』を参照してください。

デフォルト・キュー・マネージャーを作成する

crtmqm コマンドを使用して、デフォルト・キュー・マネージャーを作成します。 **q** フラグを使用して指定した **crtmqm** コマンドは、以下を実行します。

- **saturn.queue.manager** というデフォルト・キュー・マネージャーを作成する。
- デフォルトおよびシステム・オブジェクトを作成する。
- デフォルト伝送キューおよび送達不能キュー両方の名前を指定する。

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE "saturn.queue.manager"
```

説明

-q このキュー・マネージャーがデフォルト・キュー・マネージャーであることを示します。

-d MY.DEFAULT.XMIT.QUEUE
デフォルト伝送キューの名前です。

-u SYSTEM.DEAD.LETTER.QUEUE
送達不能キューの名前です。

“saturn.queue.manager”
このキュー・マネージャーの名前です。 **crtmqm** の場合、これはコマンドの最後のパラメーターでなければなりません。

デフォルト・キュー・マネージャーを作成すると、キュー・マネージャー名を指定しなくても、そのキュー・マネージャーに対していくつかのコマンド (**strmqm** や **runmqsc** など) を発行することができます。他のコマンド (**endmqm** や **dltmqm** など) については、キュー・マネージャー名を指定する必要があります。

この例のキュー・マネージャー名は小文字であり、小文字は二重引用符によって保護されていることに注意してください。パラメーターについて大文字小文字の区別が扱われる方法の詳細については、30ページの『大文字小文字の区別についての理解』を参照してください。

キュー・マネージャーを開始する

キュー・マネージャーを作成しても、キュー・マネージャーが開始されるまでコマンドまたは MQI 呼び出しを処理できません。次のコマンドをタイプして、キュー・マネージャーを開始します。

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

strmqm コマンドは、キュー・マネージャーが開始して、接続要求を受け入れる用意ができるまで、制御を戻しません。

既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする

デフォルト・キュー・マネージャーを作成するときに、デフォルト・キュー・マネージャーの名前が MQSeries 構成ファイル (mq5.ini) の中の *DefaultQueueManager* スタンザのに挿入されます。スタンザとその内容は、スタンザが存在していない場合には自動的に作成されます。

次の場合、このスタンザを編集しなければならないことがあります。

- **既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする場合。** このためには、このスタンザのキュー・マネージャー名を、新しいデフォルト・キュー・マネージャーの名前に変更する必要があります。これは、テキスト編集プログラムを使用して手動で行います。
- **ノード上にデフォルト・キュー・マネージャーがなく、既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする場合。** このためには、必要な名前を挿入した *DefaultQueueManager* スタンザをユーザー自身で作成する必要があります。
- **別のキュー・マネージャーを誤ってデフォルト・キュー・マネージャーにしてしまったため、元のデフォルト・キュー・マネージャーに戻したい場合。** このためには、MQSeries 構成ファイル内の *DefaultQueueManager* スタンザを編集し、誤ってデフォルト・キュー・マネージャーにしたキュー・マネージャーの名前を、希望するデフォルト・キュー・マネージャーの名前に置き換えます。

構成ファイルについては、221ページの『第13章 MQSeries の構成』を参照してください。

スタンザに必須情報を入れたら、キュー・マネージャーを停止して再始動します。

キュー・マネージャーの作成

キュー・マネージャーを停止する

endmqm コマンドを使用して、キュー・マネージャーを停止します。たとえば、`saturn.queue.manager` という名前のキュー・マネージャーを停止するには、次のように入力します。

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

静止シャットダウン

デフォルトでは、**endmqm** コマンドは、指定されたキュー・マネージャーの制御された または静止 シャットダウンを実行します。これは完了するまでに多少の時間がかかります。制御シャットダウンは、接続されたアプリケーションすべてが切断されるまで待機するためです。

このタイプのシャットダウンを使用して、アプリケーションに対して停止するよう通知します。次のように入力する場合、

```
endmqm -c "saturn.queue.manager"
```

すべてのアプリケーションが停止されたのがどの時点かは通知されません。(endmqm -c “saturn.queue.manager” コマンドは、endmqm “saturn.queue.manager” コマンドと同等です。)

即時シャットダウン

即時シャットダウンの場合、現在の MQI 呼び出しは完了することができますが、新しい呼び出しは失敗します。このタイプのシャットダウンは、アプリケーションがキュー・マネージャーに接続中でも実行されます。

静止時間の後に任意でキュー・マネージャーを停止するには、通常の方法としてこのシャットダウンを使用します。即時シャットダウンの場合、次のように入力します。

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

優先シャットダウン

重要: この方法は、 **endmqm** コマンドを使ってもキュー・マネージャーを停止できなかったときにのみ使用してください。この方法は、接続されているアプリケーションに予測不能な結果を及ぼす可能性があります。

即時シャットダウンが機能しない場合は、 **-p** フラグを指定した優先 シャットダウンを用いる必要があります。たとえば、次のとおりです。

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

これによって、キュー・マネージャーのコードが即時に停止されます。

注: 強制シャットダウンまたは優先シャットダウンの後、またはキュー・マネージャーが失敗した場合は、キュー・マネージャーは、所有する共用メモリーをクリーンアップせずに終了した可能性があります。この場合には、問題が再び生じるようになることがあります。 **MONMQ** ユーティリティーを使用して、この種の異常終了をクリーンアップする方法についての詳細は、458ページの『共用メモリーを **MONMQ** で管理する』を参照してください。

キュー・マネージャーのシャットダウンに問題がある場合

キュー・マネージャーのシャットダウンにおける問題は、アプリケーションによって頻繁に引き起こされます。たとえば、次のような場合です。

- アプリケーションが **MQI** 戻りコードを正しく検査しない場合。
- アプリケーションが静止の通知を要求しない場合。
- アプリケーションが (**MQDISC** 呼び出しを出して)、キュー・マネージャーからの切断を行わずに終了する場合。

キュー・マネージャーのシャットダウンに非常に時間がかかる場合、またはキュー・マネージャーが停止しないと確信する場合は、 **Ctrl-Y** を使用して **endmqm** コマンドを中断することができます。その後、別の **endmqm** コマンドを発行できますが、この場合は、即時シャットダウンまたは優先シャットダウンを指定するフラグを付加します。

endmqm コマンドおよびそのオプションの詳細については、346ページの『**endmqm** (キュー・マネージャーの終了)』を参照してください。

キュー・マネージャーの作成

キュー・マネージャーを再始動する

キュー・マネージャーを再始動するには、次のコマンドを使用します。

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

キュー・マネージャーを削除する

キュー・マネージャーを削除するには、まずキュー・マネージャーを停止してから、その後次のコマンドを使用します。

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

重要: キュー・マネージャーの削除は、キュー・マネージャーに関連するすべてのリソースも削除されるため、大きな影響を与えるステップです。これには、すべてのキューとそのメッセージだけでなく、すべてのオブジェクト定義も含まれます。

dltmqm コマンドとそのオプションの詳細については、326ページの『dltmqm (キュー・マネージャーの削除)』を参照してください。信頼できる管理者のみにこのコマンドの使用許可を与えるようにします。

第4章 ローカル MQSeries オブジェクトの管理

この章では、メッセージ・キュー・インターフェース (MQI) を使用したアプリケーション・プログラムをサポートするための、ローカル MQSeries オブジェクトの管理方法を説明します。ここでは、ローカル管理とは、MQSeries オブジェクトを作成、表示、変更、コピー、および削除することを意味しています。

この章には、次の節があります。

- 『MQI を使用したアプリケーション・プログラムのサポート』
- 47ページの『MQSC コマンドによるローカル管理タスクの実行』
- 52ページの『テキスト・ファイルからの MQSCコマンドの実行』
- 56ページの『MQSC で起こった問題の解決』
- 58ページの『ローカル・キューの取り扱い』
- 69ページの『別名キューの取り扱い』
- 71ページの『モデル・キューの取り扱い』
- 73ページの『トリガー操作のためのオブジェクトの管理』

MQI を使用したアプリケーション・プログラムのサポート

MQSeries のアプリケーション・プログラムを正常に実行するためには、一定のオブジェクトが必要です。たとえば、46ページの図1 は、キューからメッセージを取り出し、そのメッセージを処理し、その後同じキュー・マネージャー上の別のキューにいくつかの結果を送信するアプリケーションを示しています。

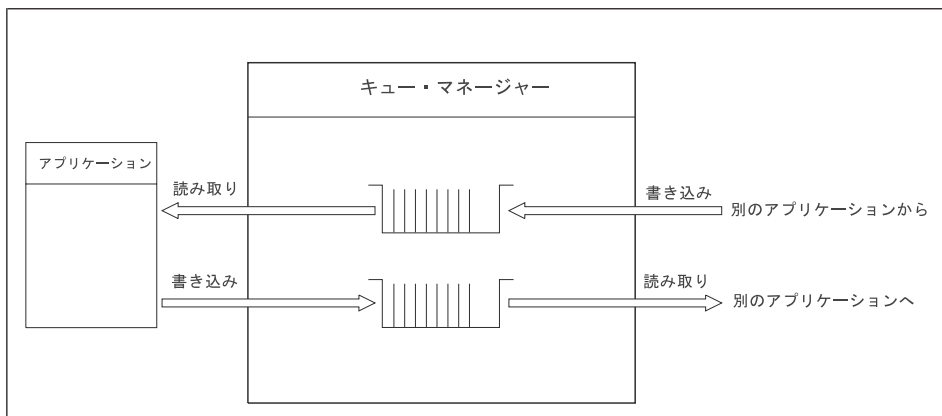


図1. キュー、メッセージ、およびアプリケーション

アプリケーションは、ローカル・キューまたはリモート・キューにメッセージを (MQPUT を使用して) 書き込むことができますが、ローカル・キューからしかメッセージを (MQGET を使用して) 直接読み取れません。

このアプリケーションが実行されるには、次の条件が満たされている必要があります。

- キュー・マネージャーが存在しており、実行されている。
- 最初のアプリケーション・キュー (ここからメッセージが取り出される) が定義されている。
- 2 番目のキュー (アプリケーションはここにメッセージを書き込む) も定義されている。
- アプリケーションが、キュー・マネージャーに接続可能である。このためには、アプリケーションはプロダクト・コードにリンクされている必要があります。詳細については、*MQSeries アプリケーション・プログラミング・ガイド* を参照してください。
- 最初のキューにメッセージを書き込むアプリケーションも、キュー・マネージャーに接続する。また、それらのアプリケーションがリモートである場合も、伝送キューとチャネルは共にセットアップされている。図1 には、システムのこの部分は示されていません。

MQSC コマンドによるローカル管理タスクの実行

この節では、**runmqsc** コマンドを使用してコマンドを発行することを想定しています。このことは、対話式で行うことができますし（キーボードからコマンドを入力）、**SYSS\$INPUT** をリダイレクトして ASCII テキスト・ファイルから一連のコマンドを実行することもできます。どちらの場合も、コマンドの形式は同じです。

MQSeries コマンド・リファレンス には、各 MQSC コマンドとその構文が説明されています。

MQSeries スクリプト・コマンド (MQSC) を使用して、キュー・マネージャー自体、クラスター、チャンネル、キュー、およびプロセス定義などのキュー・マネージャー・オブジェクトを管理することができます。この節では、キュー・マネージャー、キューおよびプロセス定義を扱います。チャンネル・オブジェクトの管理に関する詳細については、*MQSeries* 相互通信 の『DQM のインプリメンテーション』を参照してください。

runmqsc コマンドを使用して、キュー・マネージャーに対して、MQSC コマンドを発行します。このことは、対話式で行うことができますし（キーボードからコマンドを発行）、標準入力をリダイレクトして ASCII テキスト・ファイルから一連のコマンドを実行することもできます。どちらの場合も、コマンドの形式は同じです。

コマンドに設定したフラグによって、**runmqsc** コマンドを次の 3 通りのモードで実行できます。

- 検証モード。このモードでは、MQSC コマンドはローカル・キュー・マネージャー上で検査され、実際には実行されません。
- 直接モード。このモードでは、MQSC コマンドはローカル・キュー・マネージャー上で実行されます。
- 間接モード。このモードでは、MQSC コマンドはリモート・キュー・マネージャー上で実行されます。

MQSC に指定するオブジェクト属性は、本書では大文字（たとえば、**RQMNAME**）で示されます。ただし、大文字小文字の区別はありません。（大文字小文字の区別の詳細については、31ページの『MQSC コマンドでの大文字小文字の区別』を参照してください。）MQSC 属性名は 8 文字までに制限されています。

MQSC コマンドの発行

開始する前に

MQSC コマンドを実行するには、事前にそのコマンドを実行するキュー・マネージャーを作成して開始しておく必要があります。40ページの『デフォルト・キュー・マネージャーを作成する』を参照してください。

MQSeries オブジェクト名

例では、いくつかの長い名前をオブジェクトに使用しています。これは、取り扱うオブジェクトのタイプを識別するのに役立ちます。

MQSC コマンドを出す場合、必要なことはキューのローカル名を指定することのみです。例では、次のようなキュー名を使用しています。

```
ORANGE.LOCAL.QUEUE
```

この名前の LOCAL.QUEUE という部分は、このキューがローカル・キューであることを示すためにのみ使用しています。一般に、ローカル・キューの名前に、この部分は必要ではありません。

キュー・マネージャー名として saturn.queue.manager という名前も使用しています。

この名前の queue.manager の部分は、このオブジェクトがキュー・マネージャーであることを示すために使用しているだけです。一般に、キュー・マネージャーの名前に、この部分は必要ではありません。

このような名前を使用する必要はありませんが、使用しない場合には、これらの名前を指定している例の中のコマンドを修正する必要があります。

入出力のリダイレクト

他のオペレーティング・システムから OpenVMS へのマイグレーションをより簡単にするために、MQSeries では、次のような sys\$input、sys\$output、および sys\$error 用の UNIX[®] スタイルのリダイレクト標識をサポートしています。

- < は、SYSSINPUT のソースを指定します
- > は、SYS\$OUTPUT のソースを指定します
- 2> は、SYS\$ERROR のソースを指定します

この機能は、サンプル・プログラムの実行可能バージョンにも組み込まれています。ただし、サンプルのソースには組み込まれていません。したがって、ソース・コードからサンプルを再作成する場合は、この機能は使用できません。

MQSC 機能対話式で使用する

コマンドを対話式に入力するには、DCL で次のように入力します。

```
runmqsc
```

このコマンドでは、キュー・マネージャー名が指定されていません。したがって、MQSC コマンドは、デフォルト・キュー・マネージャーによって処理されます。この時点で、必要に応じて任意の MQSC コマンドを入力できます。たとえば、次のコマンドを試してください。

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

コマンドが次の行に続くことを示すためには連結文字を使用する必要があります。

- 負符号 (-) は、コマンドが次の行の先頭に続くことを示します。
- 正符号 (+) は、コマンドが次の行の最初の空白でない文字に続くことを示します。

コマンドの入力は、連結文字でなく空白でもない行の最後の文字で終了します。セミコロン (;) を入力して、明示的にコマンド入力を終了することもできます。(これは、コマンド入力の最終行の終わりで誤って連結文字を入力してしまった場合などに特に役立ちます。)

MQSC コマンドの発行

MQSC コマンドからのフィードバック

MQSC 機能からコマンドを発行すると、キュー・マネージャーは、そのアクションを確認するオペレーター・メッセージ、または操作エラーがあったことを示すオペレーター・メッセージを戻します。たとえば、次のとおりです。

```
AMQ8006: MQSeries queue created
.
.
.
AMQ8405: Syntax error detected at or near end of command segment below:-
Z
AMQ8426: Valid MQSC commands are:
ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND
```

最初のメッセージは、キューが作成されたことを確認するものですが、2 番目のメッセージは、構文エラーがあったことを示すものです。これらのメッセージは、標準出力装置に送られます。コマンドを正しく入力していなかった場合は、*MQSeries* コマンド・リファレンス を参照して正しい構文を確認してください。

対話式での MQSC への入力を終了する

MQSC コマンドの対話式入力を終了するには、MQSC END コマンドを入力します。

```
END
```

あるいは、EOF 文字 <CTRL Z> を入力して終了することもできます。

テキスト・ファイルなどの他の送信元から入力をリダイレクトしている場合は、このような作業を行う必要はありません。

キュー・マネージャーの属性を表示する

runmqsc コマンドに指定されたキュー・マネージャーの属性を表示するには、次の MQSC コマンドを使用します。

```
DISPLAY QMGR ALL
```

一般的な出力を 図2 に示してあります。

```
1 : display qmgr all
AMQ8408: Display Queue Manager details.
DESCR( )                DEADQ( )
DEFXMITQ( )             CHADEXIT( )
CLWLXIT( )              CLWLDATA( )
REPOS( )                REPOSNL( )
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) QMNAME(saturn.queue.manager)
CRDATE(2001-01-16)      CRTIME(11.13.56)
ALTDATE(2001-01-16)     ALTTIME(11.13.56)
QMID(saturn.queue.manager_2001-01-16_11.13.56)
TRIGINT(999999999)     MAXHANDS(256)
MAXUMSGS(10000)        AUTHOREV(DISABLED)
INHIBTEV(DISABLED)    LOCALEV(DISABLED)
REMOTEEV(DISABLED)    PERFMEEV(DISABLED)
STRSTPEV(ENABLED)     CHAD(DISABLED)
CHADEV(DISABLED)      CLWLEN(100)
MAXMSGL(4194304)      CCSID(819)
MAXPRTY(9)             CMDLEVEL(510)
PLATFORM(OpenVMS)     SYNCPT
DISTL(YES)
```

図2. DISPLAY QMGR コマンドからの一般的な出力

DISPLAY QMGR コマンドに ALL パラメーターを指定すると、キュー・マネージャーのすべての属性が表示されます。特に、コマンドの実行時にキュー・マネージャーの名前を指定していなかったため、出力では、デフォルト・キュー・マネージャー名 (saturn.queue.manager)、送達不能キューの名前 (SYSTEM.DEAD.LETTER.QUEUE) およびコマンド・キューの名前 (SYSTEM.ADMIN.COMMAND.QUEUE) が示されます。

次に進む前に、次のコマンドを入力して、これらのキューが作成されていることを確認してください。

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

これにより、語幹 'SYSTEM.*' に一致したキューのリストが表示されます。括弧は必ず付けてください。

MQSC コマンドの発行

デフォルトではないキュー・マネージャーを使用する

runmqsc コマンドでキュー・マネージャーの名前を指定して、デフォルトとは別のローカル・キュー・マネージャーに対して MQSC コマンドを実行することができます。たとえば、キュー・マネージャー `jupiter.queue.manager` に対して MQSC コマンドを実行するには、次のコマンドを使用します。

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

この後、入力するすべての MQSC コマンドは、このキュー・マネージャーによって処理されます。ただし、このキュー・マネージャーが同じノード上にあって、すでに実行されている場合に限りです。

リモート・キュー・マネージャーに対しても MQSC コマンドを実行できます。91ページの『リモートから MQSC コマンドを発行する』を参照してください。

キュー・マネージャーの属性を更新する

runmqsc コマンドに指定されたキュー・マネージャーの属性を更新するには、変更したい属性および値を指定した MQSC コマンド `ALTER QMGR` を使用します。たとえば、`jupiter.queue.manager` の属性を更新するには、次のコマンドを使用します。

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

`ALTER QMGR` コマンドにより、使用されている送達不能キューが変更され、禁止イベントが使用可能になります。

テキスト・ファイルからの MQSCコマンドの実行

MQSC コマンドを対話式で実行する方法は、迅速にテストを行うのに適していますが、非常に長いコマンドや、繰り返し実行したい一連のコマンドの場合には、テキスト・ファイルから入力するようにしてください。(リダイレクト標識の詳細については、48ページの『入出力のリダイレクト』を参照してください。) これを行うためには、まず、MQSC コマンドを含んでいるテキスト・ファイルを、使い慣れたテキスト・エディターを使用して作成します。たとえ

ば、次のコマンドは、テキスト・ファイル `myprog.in` に含まれている一連のコマンドを実行します。

```
runmqsc < myprog.in
```

同様に、出力をファイルにリダイレクトすることもできます。入力用の MQSC コマンドが格納されているファイルを *MQSC コマンド・ファイル* と呼びます。キュー・マネージャーからの応答が格納されている出力ファイルを *レポート・ファイル* と呼びます。

runmqsc コマンドで `SYS$INPUT` と `SYS$OUTPUT` の両方をリダイレクトするためには、次の形式のコマンドを使用します。

```
runmqsc < myprog.in > myprog.out
```

このコマンドにより、MQSC コマンド・ファイル `myprog.in` に含まれている MQSC コマンドが呼び出されます。キュー・マネージャー名を指定していないので、MQSC コマンドはデフォルト・キュー・マネージャーに対して実行されます。出力は、レポート・ファイル `myprog.out` に送られます。54ページの図3は、MQSC コマンド・ファイル `myprog.in` からの抜粋を示しており、55ページの図4は、レポート・ファイル `myprog.out` の中の対応する部分を示しています。

デフォルトのものではないキュー・マネージャー (`saturn.queue.manager`) について、**runmqsc** コマンドで `SYS$INPUT` および `SYS$OUTPUT` をリダイレクトするために、次の形式のコマンドを使用します。

```
runmqsc "saturn.queue.manager" < myprog.in > myprog.out
```

MQSC コマンド・ファイル

MQSC コマンドは、人が理解できる形式、つまり ASCII テキストで書きます。

54ページの図3は、MQSC コマンド・ファイルからの抜粋で、属性が指定された MQSC コマンド (`DEFINE QLOCAL`) を示しています。MQSeries コマンド・リファレンスには、各 MQSC コマンドとその構文が説明されています。

```
.  
. .  
. .  
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +  
  DESCR(' ') +  
  PUT(ENABLED) +  
  DEFPRTY(0) +  
  DEFPSIST(NO) +  
  GET(ENABLED) +  
  MAXDEPTH(5000) +  
  MAXMSGL(1024) +  
  DEFSOPT(SHARED) +  
  NOHARDENBO +  
  USAGE(NORMAL) +  
  NOTRIGGER  
. . .
```

図3. MQSC コマンド・ファイル *myprog.in* からの抽出

MQSeries 環境間での移植性を考えた場合、MQSC コマンド・ファイルでの行の長さは、最大 72 文字に制限することをお勧めします。正符号 (+) は、コマンドが次の行に続くことを示します。

MQSeries for Compaq OpenVMS の場合は、各行を継続文字を含めて最大で 80 文字に制限する必要があります。正符号 (+) は、コマンドが次の行に続くことを示します。

MQSC レポート

runmqsc コマンドはレポート を戻し、これは SYS\$OUTPUT に送られます。レポートには、次のものが含まれています。

- MQSC をレポートの出所として識別するヘッダー。
Starting MQSeries Commands.
- 発行された MQSC コマンドの番号付きリスト (オプション)。デフォルトでは、入力のテキストが出力にエコーされます。この出力の中では、55ページの図4 に示されているように、各コマンドには順序番号が先頭に付けられています。ただし、**runmqsc** コマンドに **-e** フラグを指定すると、この出力を抑制することができます。
- エラーのあったコマンドについての構文エラー・メッセージ。
- 各コマンドの実行結果を示すオペレーター・メッセージ。たとえば、**DEFINE QLOCAL** コマンドの正常終了を示すオペレーター・メッセージは、次のとおりです。

```
AMQ8006: MQSeries queue created.
```

- スクリプト・ファイルの実行時の一般エラーのために出されるその他のメッセージ。
- 読み取られたコマンドの数、構文エラーのあったコマンドの数、処理できなかったコマンドの数を示す、レポートの簡単な統計の要約。

注: キュー・マネージャーが処理を試みるのは、構文エラーのなかったコマンドのみです。

```
Starting MQSeries Commands.

.
.
12:   DEFINE QLOCAL('RED.LOCAL.QUEUE') REPLACE +
:     DESCR(' ') +
:     PUT(ENABLED) +
:     DEFPRTY(0) +
:     DEFPSIST(NO) +
:     GET(ENABLED) +
:     MAXDEPTH(5000) +
:     MAXMSGL(1024) +
:     DEFSOPT(SHARED) +
:     USAGE(NORMAL) +
:     NOTRIGGER
AMQ8006: MQSeries queue created.
:
.
.
15 MQSC commands read.
0 commands have a syntax error.
0 commands cannot be processed.
```

図4. MQSC レポート・ファイル *myprog.out* からの抽出

システムに提供された MQSC コマンド・ファイルを実行する

MQSeries for Compaq OpenVMS をインストールするときは、次の MQSC コマンド・ファイルが提供されます。

amqscos0.tst

サンプル・プログラムが使用するオブジェクトの定義

このファイルは、ディレクトリー `MQS_EXAMPLES:` にあります。

runmqsc を使用してコマンドを確認する

runmqsc コマンドを使用すると、MQSC コマンドを実際に行なわずに、ローカル・キュー・マネージャーでそれらのコマンドを確認することができま

MQSC コマンドの実行

す。これを行うには、たとえば次のように、`-v` フラグを `runmqsc` コマンドに設定します。

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

MQSC コマンド・ファイルに対して `runmqsc` を呼び出すと、キュー・マネージャーは、実際に MQSC コマンドを実行することなく、各コマンドを確認してレポートを戻します。これによって、コマンド・ファイル内のすべてのコマンドの構文を検査できます。これは、コマンド・ファイルから多数のコマンドを実行する場合は特に重要です。

このレポートは、55ページの図4 に示されているものと同様です。

リモートから MQSC コマンドを確認するには、この方法を用いることはできません。たとえば、次のコマンドを試行するとします。

```
runmqsc -w 30 -v "jupiter.queue.manager" < myprog.in > myprog.out
```

キュー・マネージャーがリモートであることを示すために使用する `-w` フラグは無視され、コマンドは検証モードでローカルで実行されます。

MQSC で起こった問題の解決

MQSC コマンドを実行できない場合は、次のチェックリストに記載された共通問題に該当するものがないか確認してください。生成されたエラーを読んでも、問題は必ずしも明らかにはなりません。

`runmqsc` コマンドを使用するとき、次の点に注意してください。

- ファイルから入力のリダイレクトする際に間接演算子「`<`」を使用します。この間接演算子を省略すると、キュー・マネージャーはそのファイル名をキュー・マネージャー名として解釈し、次のエラー・メッセージを発行しません。

```
AMQ8118: MQSeries queue manager does not exist.
```

- 出力をファイルにリダイレクトする場合には、「`>`」間接演算子を使用します。デフォルトでは、`runmqsc` コマンドを実行したディレクトリーに出力されます。出力を特定のファイルおよびディレクトリーに送るには、完全修飾ファイル名を指定してください。

- コマンドを実行するキュー・マネージャーが作成されているかどうか確認します。

これを行うためには、構成ファイル `mqs.ini` を調べます。このファイルは、デフォルトでは `MQS_ROOT:[MQM]` ディレクトリーにあります。このファイルには、キュー・マネージャーの名前と、すでに作成されている場合はデフォルト・キュー・マネージャーの名前が入っています。

- キュー・マネージャーはすでに開始されている必要があります。開始されていない場合は、開始します。41ページの『キュー・マネージャーを開始する』を参照してください。すでに開始されている場合に開始しようとすると、エラー・メッセージが出されます。
- デフォルト・キュー・マネージャーが定義されていない場合、**runmqsc** コマンドにキュー・マネージャー名を指定します。これを定義しなければ、次のようなエラー・メッセージが出されます。

```
AMQ8146: MQSeries queue manager not available.
```

この種の問題を訂正するには、41ページの『既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする』を参照してください。

- **MQSC** コマンドを **runmqsc** パラメーターとして指定することはできません。たとえば、次のものは無効です。

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

- **runmqsc** コマンドを発行する前に、DCL から **MQSC** コマンドを入力することはできません。たとえば、次のとおりです。

```
DEFINE QLOCAL(Queue1)
```

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

MQSC で起こった問題

- **runmqsc** から制御コマンドを実行することはできません。たとえば、MQSC を対話式で実行すると、キュー・マネージャーを開始できません。

```
$ runmqsc
0790997, 5724-A38 (C) Copyright IBM Corp. 1996, 2001 ALL RIGHTS RESERVED.
Starting MQSeries Commands.

strmqm saturn.queue.manager
  1 : strmqm saturn.queue.manager
AMQ8405: Syntax error detected at or near end of command segment below:-
s
AMQ8426: Valid MQSC commands are:
  ALTER
  CLEAR
  DEFINE
  DELETE
  DISPLAY
  END
  PING
  REFRESH
  RESET
  RESOLVE
  RESUME
  START
  STOP
  SUSPEND

*CANCEL*
One MQSC command read.
One command has a syntax error.
All valid MQSC commands were processed.
$
```

93ページの『リモートからの MQSC の使用に問題がある場合』も参照してください。

ローカル・キューの取り扱い

この節では、使用できる MQSC コマンドの例をいくつか示します。これらのコマンドの詳細については、*MQSeries* コマンド・リファレンスを参照してください。

ローカル・キューを定義する

アプリケーションにとって、ローカル・キュー・マネージャーとは、アプリケーションが接続されているキュー・マネージャーです。ローカル・キュー・マネージャーによって管理されるキューは、そのキュー・マネージャーに対してローカルであるといえます。

ローカル・キューの定義を作成するため、またキューと呼ばれるデータ構造を作成するためには、MQSC コマンド `DEFINE QLOCAL` を使用します。デフォルト・ローカル・キューの特性からのキュー特性を修正することもできます。

この例では、定義するキュー `ORANGE.LOCAL.QUEUE` は、次のような特性を持つものとして指定します。

- 読み取りは可能、書き込みは不可、先入れ先出し法 (FIFO) で操作が行われる。
- 「通常の」キュー。つまり、開始キューや伝送キューではなく、トリガー・メッセージを生成しない。
- キューの最大サイズは、1000 個のメッセージで、最大メッセージ長は、2000 バイトである。

次の MQSC コマンドによって、このようなキューが作成されます。

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL)
```

注:

1. これらの属性のほとんどは、プロダクトで提供されているデフォルト値です。ただし、ここでは、これらは説明の目的で示してあります。デフォルト値が自分の希望するものであるか、デフォルト値が変更されていないことが確実ならば、これらは省略することができます。60ページの『デフォルト・オブジェクトの属性を表示する』も参照してください。
2. `USAGE (NORMAL)` は、このキューが伝送キューではないことを示します。

ローカル・キューの取り扱い

- 名前が `ORANGE.LOCAL.QUEUE` である同じキュー・マネージャーにすでにローカル・キューがあると、このコマンドは失敗します。既存のキューの定義を上書きする場合には、`REPLACE` 属性を使用してください。ただし、62ページの『ローカル・キューの属性を変更する』も参照してください。

送達不能キューを定義する

正しい宛先に送達できないメッセージを後で取り出すために保管することができるよう、各キュー・マネージャーは、送達不能キューとして使用されるローカル・キューを持っている必要があります。送達不能キューについては、キュー・マネージャーに明示的に通知する必要があります。このことは、送達不能キューを `crtmqm` コマンドに指定することにより行えます。あるいは `ALTER QMGR` コマンドを使用して後でそれを指定することができます。送達不能キューを使用するためには、その前にそれを定義しておくことも必要です。

`SYSTEM.DEAD.LETTER.QUEUE` という名前のサンプル送達不能キューが、プロダクトと共に提供されています。このキューは、サンプルを実行すると、自動的に作成されます。必要ならば、この定義を修正できます。その名前を変更する必要はありません。

送達不能キューには、以下に示すものを除いて、特別な要件はありません。

- ローカル・キューでなければならない。
- その `MAXMSGL` (最大メッセージ長) 属性は、キュー・マネージャーが取り扱う最大メッセージおよび送達不能ヘッダー (`MQDLH`) をキューに収容できるようにしておく必要があります。

MQSeries には送達不能キュー・ハンドラーがあり、これによって、送達不能キュー上で見つかったメッセージの処理方法または除去方法を指定できます。詳細については、131ページの『第8章 MQSeries 送達不能キュー・ハンドラー』を参照してください。

デフォルト・オブジェクトの属性を表示する

MQSeries オブジェクトを定義した場合、指定していない属性はデフォルト・オブジェクトから取られます。たとえば、ローカル・キューを定義すると、このキューは、定義の中で省略された属性を、`SYSTEM.DEFAULT.LOCAL.QUEUE` と呼ばれるデフォルト・ローカル・キューから継承します。これらの属性を正確に知りたい場合には、次のコマンドを使用します。

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE) ALL
```


注: このコマンドの構文は、対応する DEFINE コマンドの構文とは異なっています。

属性を個別に指定すると、属性を選択的に表示できます。たとえば、次のとおりです。

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +
    MAXDEPTH +
    MAXMSGL +
    CURDEPTH
```

このコマンドにより、次のような 3 つの指定の属性が表示されます。

```
AMQ8409: Display Queue details.
    QUEUE (ORANGE.LOCAL.QUEUE)
    MAXDEPTH(1000)
    MAXMSGL(2000)
    CURDEPTH(0)
```

CURDEPTH は、現行キュー項目数、つまりキュー上のメッセージ数です。これは、表示すると便利な属性です。キュー項目数を監視することによって、キューが満杯にならないようにすることができます。

ローカル・キュー定義をコピーする

DEFINE コマンドに LIKE 属性を指定すると、キュー定義をコピーできます。たとえば、次のとおりです。

```
DEFINE QLOCAL (MAGENTA.QUEUE) +
    LIKE (ORANGE.LOCAL.QUEUE)
```

このコマンドにより、システム・デフォルト・ローカル・キューの属性ではなく、コピー元のキュー ORANGE.LOCAL.QUEUE と同じ属性を持つキューが作成されます。

この同じ形式の DEFINE コマンドを使用して、キュー定義をコピーし、元のキューの属性をいくつか変更することもできます。たとえば、次のとおりです。

ローカル・キューの取り扱い

```
DEFINE QLOCAL (THIRD.QUEUE) +  
  LIKE (ORANGE.LOCAL.QUEUE) +  
  MAXMSGL(1024)
```

このコマンドにより、キュー ORANGE.LOCAL.QUEUE の属性がキュー THIRD.QUEUE にコピーされ、新しいキューの最大メッセージ長は、2000 バイトではなく、1024 バイトになるように指定されます。

注:

1. DEFINE コマンドの LIKE 属性を使用した場合、キューの属性だけをコピーします。キュー上のメッセージはコピーしません。
2. LIKE を指定せずにローカル・キューを定義する場合、それは DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE) と同じになります。

ローカル・キューの属性を変更する

キューの属性は 2 通りの方法で変更できます。つまり、ALTER QLOCAL コマンドを使用するか、あるいは REPLACE 属性を指定した DEFINE QLOCAL コマンドを使用する方法です。59ページの『ローカル・キューを定義する』では、キュー ORANGE.LOCAL.QUEUE を定義しました。ここで、たとえば、このキューの最大メッセージ長を 10 000 バイトに増やしたいとします。

- ALTER コマンドを使用

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

このコマンドにより、1 つの属性、つまり最大メッセージ長の属性は変更されますが、他の属性はすべて変更されません。

- REPLACE オプションを指定した DEFINE コマンドを使用。たとえば、次のように指定します。

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

このコマンドにより、最大メッセージ長だけでなく、他のすべての属性も変更されます。他のすべての属性にはデフォルト値が与えられます。このキューは、以前は書き込み禁止でしたが、これで書き込み可能になります。キュー SYSTEM.DEFAULT.LOCAL.QUEUE で指定されているとおり、変更されていない限り、書き込み可能はデフォルト値です。

既存のキューの最大メッセージ長を短くしても、既存のメッセージは影響を受けません。ただし、新しいメッセージはこの新しい基準に適合する必要があります。

ローカル・キューをクリアする

MAGENTA.QUEUE という名前のローカル・キューからすべてのメッセージを削除するためには、次のコマンドを使用します。

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

次の場合には、キューの内容をクリアすることができません。

- 同期点でコミットされていないメッセージで、そのキューに書き込まれているものがある場合
- アプリケーションがそのキューを現在オープンしている場合

ローカル・キューを削除する

ローカル・キューを削除するには、MQSC コマンド DELETE QLOCAL を使用します。キュー上にコミットされていないメッセージがある場合、そのキューは削除できません。ただし、キューがコミットされたメッセージを 1 つまたは複数持っているが、コミットされていないメッセージがない場合は、PURGE オプションを指定した場合にのみ削除することができます。たとえば、次のとおりです。

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

PURGE の代わりに NOPURGE を指定すると、コミットされたメッセージがキューに含まれている場合、そのキューが削除されることはありません。

キューのブラウズ

キュー上のメッセージの内容を調べる必要がある場合、MQSeries for OpenVMS では、この目的のためのサンプル・キュー・ブラウザーが用意されています。ブラウザーは、ソースとしても実行可能なモジュールとしても提供されています。デフォルトでは、ファイル名とパスは次のとおりです。

ソース MQS_EXAMPLES:AMQSBCG0.C

実行可能モジュール

[.BIN]AMQSBCG.EXE (次のディレクトリーの下にある)

MQS_EXAMPLES:

ローカル・キューの取り扱い

サンプルは、次の 2 つのパラメーターをとります。

- キュー名。たとえば、SYSTEM.ADMIN.RESPQ.TEST。
- キュー・マネージャー名。たとえば、JJJH。

たとえば、次のコマンドのようになります。

```
amqsbcg "SYSTEM.ADMIN.RESPQ.TEST" "JJJH"
```

デフォルトはありません。パラメーターは両方とも必須です。このコマンドの一般的な結果を 65 ページの図5 に示します。


```

**** Message ****

length - 81 bytes

00000000: 4A4F 484E 534F 4E2C 4441 5649 4420 4D52 'JOHNSON,DAVID MR'
00000010: 2020 2020 3239 2D4A 414E 2D32 3030 3120 ' 29-JAN-2001 '
00000020: 3133 3A34 3220 3431 3233 3030 3831 2031 '13:42 41230081 1'
00000030: 3238 332E 3334 2020 3030 3235 2E32 3220 '283.34 0025.22 '
00000040: 2030 3030 302E 3030 2020 3739 3235 2E36 ' 0000.00 7925.6 '
00000050: 35 '5 '

No more messages
MQCLOSE
MQDISC
$

```

図5. キュー・ブラウザーからの一般的な結果 (5/5)

別名キューの取り扱い

別名キュー (キュー別名としても知られています) は、MQI 呼び出しのリダイレクトの方法を提供します。別名キューは、実際のキューではなく、実際のキューに解決される定義です。別名キュー定義は、TARGQ 属性 (PCF では *BaseQName*) で指定される宛先キュー名を含んでいます。アプリケーションが MQI 呼び出しの中で別名キューを指定すると、キュー・マネージャーは実行時に実際のキュー名に解決します。

たとえば、MY.ALIAS.QUEUE という名前のキューにメッセージを入れるようなアプリケーションが開発されたとします。このアプリケーションは、MQOPEN 要求を出すときにこのキューの名前を指定し、メッセージをこのキューに書き込む場合には、間接的にこのキューの名前を指定します。アプリケーションは、キューが別名キューであることに気が付きません。この別名を使用した各 MQI 呼び出しについて、キュー・マネージャーは実際のキュー名に解決します。このキュー名はローカル・キューか、このキュー・マネージャーに定義されたりリモート・キューのいずれかです。

TARGQ 属性の値を変更することにより、MQI 呼び出しを別のキュー (おそらく別のキュー・マネージャー上の別のキュー) にリダイレクトできます。これは、保守、マイグレーション、および負荷平衡に役立ちます。

別名キューを定義する

次のコマンドにより、別名キューが作成されます。

別名キューの取り扱い

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (YELLOW.QUEUE)
```

このコマンドは、MQI 呼び出し (MY.ALIAS.QUEUE を指定している) をキュー YELLOW.QUEUE にリダイレクトします。このコマンドは、ターゲット・キューを作成しないので、キュー YELLOW.QUEUE が実行時に存在しなければ、MQI 呼び出しは失敗します。

別名定義を変更すると、MQI 呼び出しを別のキューにリダイレクトできます。たとえば、次のとおりです。

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (MAGENTA.QUEUE) REPLACE
```

このコマンドは、MQI 呼び出しを別のキュー MAGENTA.QUEUE にリダイレクトします。

別名キューを使用すると、単一のキュー (ターゲット・キュー) が、異なるアプリケーションについては異なる属性を持っているように見えるようにすることもできます。これは、アプリケーションごとに 1 つの別名、つまり合計 2 つの別名を定義すると行えます。2 つのアプリケーションがあるとします。

- アプリケーション ALPHA は、メッセージを YELLOW.QUEUE に書き込むことができますが、そこからメッセージを読み取ることはできません。
- アプリケーション BETA は、YELLOW.QUEUE からメッセージを読み取ることはできますが、そこにメッセージを書き込むことはできません。

これは、次のコマンドを使用して行います。

```
* This alias is put enabled and get disabled for application ALPHA
```

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
    TARGQ (YELLOW.QUEUE) +  
    PUT (ENABLED) +  
    GET (DISABLED)
```

```
* This alias is put disabled and get enabled for application BETA
```

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
    TARGQ (YELLOW.QUEUE) +  
    PUT (DISABLED) +  
    GET (ENABLED)
```

ALPHA は、MQI 呼び出しの中でキュー名 ALPHAS.ALIAS.QUEUE を使用しますが、BETAは、キュー名 BETAS.ALIAS.QUEUE を使用します。これらはいずれも同じキューをアクセスしますが、その方法は異なっています。

キュー別名を定義する際には、ローカル・キューの場合と同様にして、LIKE 属性および REPLACE 属性を使用することができます。

キュー別名に関してその他のコマンドを使用する

該当の MQSC コマンドを使用すると、キュー別名の属性を表示または更新したり、キュー別名オブジェクトを削除したりできます。たとえば、次のとおりです。

```
* Display the queue alias' attributes
* ALL = Display all attributes

DISPLAY QUEUE (ALPHAS.ALIAS.QUEUE) ALL

* ALTER the base queue name, to which the alias resolves.
* FORCE = Force the change even if the queue is open.

ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGQ(ORANGE.LOCAL.QUEUE) FORCE

* Delete this queue alias, if you can.

DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

キュー別名は、たとえば、アプリケーションがそのキューを現在オープンしていたり、このキューに解決されるキューをオープンしている場合には、削除することができません。キュー別名コマンドの詳細については、*MQSeries* コマンド・リファレンスを参照してください。

モデル・キューの取り扱い

キュー・マネージャーは、モデル・キューとして定義されているキュー名を指定した MQI 呼び出しをアプリケーションから受け取ると、動的キューを作成します。新しい動的キューの名前は、そのキューの作成時にキュー・マネージャーによって生成されます。モデル・キューとは、動的キューの属性を指定しているテンプレートのことで、動的キューはこのモデル・キューから作成されます。

モデル・キューは、アプリケーションがキューを必要とするときにそのキューを作成するための便利な方法を提供します。

モデル・キューの取り扱い

モデル・キューを定義する

ローカル・キューを定義するのと同じ方法で、1組の属性を持つモデル・キューを定義します。モデル・キューには、作成される動的キューが一時キューとなるか永続キューとなるかを指定できること以外は、モデル・キューとローカル・キューは同じ1組の属性を持っています。(永続キューはキュー・マネージャーが再始動しても維持されますが、一時キューは維持されません。)たとえば、次のとおりです。

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
  DESCR('Queue for messages from application X') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL) +
  DEFTYPE (PERDYN)
```

このコマンドにより、モデル・キュー定義が作成されます。DEFTYPE 属性により、このテンプレートから作成される実際のキューは、永続動的キューになります。

注: 指定されていない属性は、SYSYTEM.DEFAULT.MODEL.QUEUE デフォルト・キューから自動的にコピーされます。

モデル・キューを定義する際には、ローカル・キューの場合と同様にして、LIKE 属性および REPLACE 属性を使用することができます。

モデル・キューに関してその他のコマンドを使用する

該当の MQSC コマンドを使用すると、モデル・キューの属性を表示または更新したり、モデル・キュー・オブジェクトを削除したりできます。たとえば、次のとおりです。

```

* Display the model queue's attributes
* ALL = Display all attributes

DISPLAY QUEUE (GREEN.MODEL.QUEUE) ALL

* ALTER the model to enable puts on any
* dynamic queue created from this model.

ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)

* Delete this model queue:

DELETE QMODEL (RED.MODEL.QUEUE)

```

トリガー操作のためのオブジェクトの管理

MQSeries には、キューで特定の条件が満たされると自動的にアプリケーションを開始するための機能があります。その条件の一例として、キュー上のメッセージ数が指定の数に達した場合があります。この機能は、トリガー操作と呼ばれていて、詳細は、*MQSeries アプリケーション・プログラミング・ガイド* に記載されています。この節では、MQSeries for Compaq OpenVMS でのトリガー操作をサポートするために必要なオブジェクトのセットアップ方法を説明します。

トリガー操作のためにアプリケーション・キューを定義する

アプリケーション・キューとは、アプリケーションが MQI を介してメッセージング用に使用するローカル・キューのことです。トリガー操作では、いくつかのキュー属性をアプリケーション・キューに定義する必要があります。トリガー操作自体は、*Trigger* 属性 (MQSC の中の TRIGGER) によって使用可能になります。

以下に示す例では、トリガー・イベントは、MOTOR.INS.QUEUE というローカル・キューに優先順位 5 以上のメッセージが 100 個入れられたときに生成されます。

トリガー操作のためのオブジェクトの管理

```
DEFINE QLOCAL (MOTOR.INS.QUEUE) +  
  PROCESS (MOTOR.INS.PROC) +  
  MAXMSGL (2000) +  
  DEFPSIST (YES) +  
  INITQ (MOTOR.INS.INT.Q) +  
  TRIGGER +  
  TRIGTYPE (DEPTH) +  
  TRIGDPTH (100)+  
  TRIGMPRI (5)
```

説明

QLOCAL (MOTOR.INS.QUEUE)

定義するアプリケーション・キューの名前を指定します。

PROCESS (MOTOR.INS.PROC)

トリガー・モニター・プログラムによって開始するアプリケーションの名前を指定します。

MAXMSGL (2000)

キューに入れるメッセージの最大長を指定します。

DEFPSIST (YES)

メッセージがこのキューでは永続であることを指定します。

INITQ (MOTOR.INS.INT.Q)

キュー・マネージャーがトリガー・メッセージを入れる開始キューの名前です。

TRIGGER

トリガー属性値です。

TRIGTYPE (DEPTH)

要求した優先順位 (TRIMPRI) を持つメッセージの数が TRIGDPTH で指定した数に達したときにトリガー・イベントを生成するように指定します。

TRIGDPTH (100)

トリガー・イベントを生成するのに必要なメッセージ数を指定します。

TRIGMPRI (5)

トリガー・イベントを生成するかどうかを決める際に、キュー・マネージャーが考慮に入れるメッセージの優先順位です。優先順位 5 以上のメッセージだけが考慮に入れます。

開始キューを定義する

トリガー・イベントが発生すると、キュー・マネージャーは、アプリケーション・キュー定義に指定された開始キューにトリガー・メッセージを入れます。開始キューには特別の設定値はありませんが、参考として以下に示す `MOTOR.INS.INT.Q` というローカル・キューの定義を使用することができます。

```
DEFINE QLOCAL(MOTOR.INS.INT.Q) +
  GET (ENABLED) +
  NOSHARE +
  NOTRIGGER +
  MAXMSGL (2000) +
  MAXDEPTH (10)
```

プロセス定義を作成する

プロセス定義を作成するには、`DEFINE PROCESS` コマンドを使用します。プロセス定義は、アプリケーション・キューを、キューからのメッセージを処理するアプリケーションと関連付けます。この関連付けは、アプリケーション・キュー `MOTOR.INS.QUEUE` の `PROCESS` 属性によって行われます。次の `MQSC` コマンドは、この例で識別されている必須プロセス `MOTOR.INS.PROC` を定義します。

```
DEFINE PROCESS (MOTOR.INS.PROC) +
  DESCRC ('Insurance request message processing') +
  APPLTYPE (OPENVMS) +
  APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE') +
  USERDATA ('open, close, 235')
```

説明

MOTOR.INS.PROC

プロセス定義の名前です (15 文字までに制限されています)。

DESCRC ('Insurance request message processing')

キーワードに続く、定義が関連付けられているアプリケーション・プログラムの説明テキストです。このテキストは、`DISPLAY PROCESS` コマンドを使用すると表示されます。これは、プロセスが行う事柄を識別するのに役立ちます。ストリングの中でスペースを使用する場合は、ストリングを単一引用符で囲む必要があります。

APPLTYPE (OPENVMS)

OPENVMS 上で実行されるアプリケーションのタイプです。

トリガー操作のためのオブジェクトの管理

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

アプリケーション実行可能プログラムの名前です。

USERDATA ('open, close, 235')

アプリケーションで使用できるユーザー定義のデータです。

プロセス定義を表示する

定義の結果を調べるには、ALL キーワードを指定して DISPLAY PROCESS コマンドを使用します。たとえば、次のとおりです。

```
DISPLAY PROCESS (MOTOR.INS.PROC) ALL

      24 : DISPLAY PROCESS (MOTOR.INS.PROC) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing') +
APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE') +
USERDATA (open, close, 235) +
PROCESS (MOTOR.INS.PROC) +
APPLTYPE (OPENVMS)
```

MQSC コマンド ALTER PROCESS を使用して既存のプロセス定義を更新したり、DELETE PROCESS を使用してプロセス定義を削除することもできます。

第5章 管理用タスクの自動化

この章の内容を理解するためには、MQSeries オブジェクト管理の経験が必要です。

タスクのモニターおよび一部の管理を自動化することがインストールに役立つと判断する場合があります。プログラム式コマンド形式 (PCF) コマンドを使用して、ローカル・キュー・マネージャーおよびリモート・キュー・マネージャー両方の管理タスクを自動化することができます。

この章は次のことについて説明します。

- プログラム式コマンド形式を使用して、『PCF コマンド』の管理用タスクを自動化する方法。
- 80ページの『リモート管理でコマンド・サーバーを管理する』におけるコマンド・サーバーの使用法

PCF コマンド

MQSeries プログラム式コマンド形式 (PCF) コマンドの目的は、管理タスクを管理プログラムに組み込めるようにすることです。これにより、プログラムから、キュー、プロセス定義、チャンネル、および名前リストを作成したり、キュー・マネージャーを変更したりすることができます。

PCF コマンドは、MQSC 機能が提供する機能と同様の処理を行います。

したがって、単一のノードからネットワーク内の任意のキュー・マネージャーへ PCF コマンドを発行するプログラムを作成することができます。これにより、管理タスクを中央集中方式にすると同時に自動化することができます。

各 PCF コマンドは、MQSeries メッセージのアプリケーション・データ部分に組み込まれたデータ構造です。各コマンドは、他のメッセージの場合と同様に、MQI 機能 MQPUT を使用して宛先キュー・マネージャーに送られます。メッセージを受信するキュー・マネージャー上のコマンド・サーバーは、そのコマンドをコマンド・メッセージとして解釈し、実行します。応答を入手するには、アプリケーションが MQGET 呼び出しを発行します。すると、応答データが別のデータ構造に戻されます。次に、アプリケーションはその応答を処理し、その応答に応じてアクションを実行します。

PCF コマンド

注: MQSC コマンドとは異なり、PCF コマンドおよびそれらの応答は、テキスト形式では読み取ることはできません。

次に、PCF コマンド・メッセージを作成するためにアプリケーション・プログラマーが指定する必要がある事項のいくつか簡単に示します。

メッセージ記述子

これは、標準 MQSeries メッセージ記述子です。この中で、次のように指定します。

メッセージ・タイプ (*MsgType*) には、MQMT_REQUEST を指定します。

メッセージ形式 (*Format*) には、MQFMT_ADMIN を指定します。

アプリケーション・データ

これには、PCF ヘッダーを含む PCF メッセージが入ります。この中で、

PCF メッセージ・タイプ (*Type*) には MQCFT_COMMAND を指定します。

コマンド ID には、*Change Queue* (MQCMD_CHANGE_Q) などのコマンドを指定します。

PCF データ構造およびそれをインプリメントする方法の詳しい説明については、*MQSeries プログラム式システム管理* を参照してください。

MQSC および PCF での属性

MQSC に指定するオブジェクト属性は、本書では大文字 (たとえば、RQMNAME) で示されます。ただし、大文字小文字の区別はありません。MQSC 属性名は 8 文字までに制限されています。

PCF でのオブジェクト属性はイタリックで示され、8 文字までに制限されていません。たとえば、PCF では、RQMNAME に相当するものは *RemoteQMgrName* です。

エスケープ PCF

エスケープ PCF は、メッセージ・テキスト内に MQSC コマンドを含んでいる PCF コマンドです。PCF を使用して、リモート・キュー・マネージャーにコマンドを送信することができます。エスケープ PCF の使い方の詳細については、*MQSeries プログラム式システム管理* を参照してください。

MQAI を使用して PCF の使い方を単純化する

MQAI は、OpenVMS 上で使用できる MQSeries に対する管理インターフェースです。

MQAI は、データ・バッグを使用することにより、キュー・マネージャー上の管理タスクを実行します。データ・バッグを使用すると、PCF を使用するよりも簡単な方法で、オブジェクトのプロパティ（またはパラメーター）を処理することができます。

MQAI は、次のことを実行するために使用できます。

- **PCF メッセージの使い方を単純化する**

MQAI は、MQSeries を管理する簡単な方法です。独自の PCF メッセージを書き込む必要がないので、複雑なデータ構造に関連する問題を回避することができます。

MQI 呼び出しを使用して書き込まれたプログラムにおいてパラメーターを渡すには、PCF メッセージにコマンドおよびストリングまたは整数データの詳細を入れる必要があります。これを行うには、すべての構造についてプログラムに複数のステートメントを指定する必要があり、メモリー・スペースを割り振る必要があります。このタスクは、時間がかかる大変な作業です。

一方、MQAI を使用して書き込まれたプログラムはパラメーターを適切なデータ・バッグに渡し、構造ごとに1つのステートメントしか必要ありません。MQAI データ・バッグを使用すると、配列を処理したり、ストレージを割り振る必要がなく、PCF の詳細を入れる必要がある程度なくなります。

- **自己管理アプリケーションおよびアプリケーション・ツールを実装する**

たとえば、MQSeries for Windows NT and Windows 2000 バージョン 5.2 に装備されているアクティブ・ディレクトリー・サービスは、MQAI を使用します。（現在、OpenVMS プラットフォーム上でこれを使用している例はありません。）

- **エラー条件をより簡単に処理する**

MQSC コマンドから戻りコードを戻すのは難しいですが、MQAI によりエラー条件の処理が簡単になります。

データ・バッグを作成し入力した後、mqExecute 呼び出しを使用して、管理コマンド・メッセージをキュー・マネージャーのコマンド・サーバーに送信することができます。mqExecute 呼び出しはどの応答メッセージも待ちます。mqExecute 呼び出しは、コマンド・サーバーとの交換を処理し、応答を応答バッグに返します。

PCF コマンド

MQAI の使い方の詳細については、*MQSeries* 管理インターフェースのプログラミング・ガイドおよびリファレンス を参照してください。

一般的な PCF の詳細については、*MQSeries* プログラム式システム管理 を参照してください。

リモート管理でコマンド・サーバーを管理する

各キュー・マネージャーには、それぞれに関連付けられた 1 つのコマンド・サーバーがあります。コマンド・サーバーは、リモート・キュー・マネージャーからの着信コマンド、またはアプリケーションからの PCF コマンドを処理します。コマンド・サーバーは処理のために、そのコマンドをキュー・マネージャーに渡し、コマンドの発信元に応じて、完了コードやオペレーター・メッセージを戻します。

コマンド・サーバーは、PCF、MQAI に関するすべての管理およびリモート管理にも必須です。

注: リモート管理では、宛先キュー・マネージャーを確実に実行しているようにする必要があります。実行していないと、コマンドを含んだメッセージは、メッセージの発信元のキュー・マネージャーから出ていくことができません。代わりに、それらのメッセージは、リモート・キュー・マネージャーが使用しているローカル伝送キューに保持されます。こうした状況は、できるだけ回避する必要があります。

コマンド・サーバーを開始する

コマンド・サーバーを開始するには、次のコマンドを使用します。

```
strmqcsv "saturn.queue.manager"
```

ここで、`saturn.queue.manager` は、開始されるコマンド・サーバーに関連したキュー・マネージャーです。

コマンド・サーバーの状況を表示する

リモート管理では、宛先キュー・マネージャー上でコマンド・サーバーが実行中であることを確認します。これが実行されていないと、リモート・コマンドを処理できません。コマンドを含んだメッセージは、宛先キュー・マネージャーのコマンド・キューに入れられます。

キュー・マネージャー (ここでは `saturn.queue.manager`) のコマンド・サーバーの状況を表示するには、次のコマンドを出します。

```
dspmqcsv "saturn.queue.manager"
```

このコマンドは、ターゲット・マシンに対して出す必要があります。コマンド・サーバーが実行されていると、次のメッセージが戻されます。

```
AMQ8027    MQSeries Command Server Status ...: Running
```

コマンド・サーバーを停止する

コマンド・サーバーを停止するには、前の例を使用して、次のコマンドを出します。

```
endmqcsv "saturn.queue.manager"
```

コマンド・サーバーを停止するには、次の 2 つの方法があります。

- 制御された停止の場合、`-c` フラグを指定した **endmqcsv** コマンドを使用します。これはデフォルトです。
- 即時停止の場合、`-i` フラグを指定した **endmqcsv** コマンドを使用します。

注: キュー・マネージャーを停止すると、そのキュー・マネージャーと関連付けられているコマンド・サーバーも (開始されている場合) 終了します。

第6章 リモート MQSeries オブジェクトの管理

この章では、別のキュー・マネージャー上の MQSeries オブジェクトを管理する方法について説明します。また、リモート・キュー・オブジェクトを使用して、メッセージおよび応答メッセージの宛先を制御する方法についても説明します。

この章には、次の節があります。

- 『チャンネル、クラスターおよびリモート・キューイング』
- 85ページの『MQSC コマンドを使用するローカル・キュー・マネージャーからのリモート管理』
- 93ページの『リモート・キューのローカル定義の作成』
- 97ページの『リモート・キュー定義を別名として使用する』

チャンネル、その属性およびセットアップ方法の詳細については、*MQSeries 相互通信* を参照してください。

チャンネル、クラスターおよびリモート・キューイング

キュー・マネージャーはメッセージを送信し、必要な場合には、戻される応答を受信して他のキュー・マネージャーと通信します。受信側キュー・マネージャーは、次のようになります。

- 同じマシン上
- 同じ場所または別の場所にある別のマシン上
- ローカル・キュー・マネージャーと同じプラットフォームで実行
- MQSeries がサポートする別のプラットフォームで実行

メッセージの発信元としては次のようなものがあります。

- あるノードから別のノードにデータを転送するユーザー作成アプリケーション・プログラム
- PCF を使用するユーザー作成管理アプリケーション。
- キュー・マネージャーは、次のものを送信します。
 - 別のキュー・マネージャーに観測イベント・メッセージを送信する。
 - **runmqsc** コマンドから発行される MQSC コマンドを間接モードで送信する (この場合、コマンドは別のキュー・マネージャーで実行されます)。

リモート・オブジェクトの管理

メッセージをリモート・キュー・マネージャーに送信する前に、ローカル・キュー・マネージャーには、メッセージの到着を検出したり、メッセージを転送するメカニズムが必要です。そのメカニズムは次のもので構成されます。

- 少なくとも1つのチャンネル
- 伝送キュー
- メッセージ・チャンネル・エージェント (MCA)
- チャンネル・リスナー
- チャンネル開始プログラム

チャンネルは、2つのキュー・マネージャー間の単方向通信リンクであり、リモート・キュー・マネージャーの多数のキューにメッセージを伝送することができます。

チャンネルの各端は、個別に定義されます。たとえば、一方の端が送信側、つまりサーバーであれば、他方の端は受信側、つまり要求側にする必要があります。簡単なチャンネルは、ローカル・キュー・マネージャーの端の送信側チャンネル定義 およびリモート・キュー・マネージャーの端の受信側チャンネル定義 で構成されます。2つの定義の名前は同じにして、共に単一のチャンネルを構成する必要があります。

リモート・キュー・マネージャーを、ローカル・キュー・マネージャーが送信したメッセージに対して応答するよう設定する場合、2番目のチャンネルは、ローカル・キュー・マネージャーに応答を戻すようにセットアップする必要があります。

チャンネルは `MQSC DEFINE CHANNEL` コマンドを使用して定義されます。この章では、特に断りがない限り、チャンネルに関連した例はデフォルト・チャンネル属性を使用しています。

チャンネルの各端にはメッセージ・チャンネル・エージェント (MCA) があり、メッセージの送受信を制御します。伝送キューからメッセージを受け取り、キュー・マネージャー間の通信リンクにメッセージを書き込むジョブは、MCA が担当します。

伝送キューは、専用ローカル・キューであり、メッセージを MCA が受信し、リモート・キュー・マネージャーに送信する前に、一時的にメッセージを保管します。リモート・キュー定義 上の伝送キューの名前を指定します。

87ページの『リモート管理のためにチャンネルおよび伝送キューを作成する』は、リモート管理をセットアップするためにこれらの定義をどのように使用するかを示しています。

分散キューイングのセットアップに関する詳細については、*MQSeries 相互通信* を参照してください。

クラスターを使用するリモート管理

分散キューイングを使用する従来の *MQSeries* ネットワークでは、すべてのキュー・マネージャーは独立しています。1つのキュー・マネージャーが別のキュー・マネージャーにメッセージを送信する必要がある場合、伝送キュー、リモート・キュー・マネージャーへのチャンネル、メッセージの送信先であるすべてのキューのリモート・キュー定義を定義しておく必要があります。

クラスターは、複雑な伝送キュー、チャンネル、およびキューを定義しなくても、複数のキュー・マネージャーが単一のネットワーク上で互いに直接通信できる方法でセットアップされたキュー・マネージャーのグループです。クラスターは、簡単にセットアップでき、通常は、一部論理的に関連付けられるキュー・マネージャーを含み、データまたはアプリケーションを共有する必要があります。

クラスターを作成すると、クラスター内のキュー・マネージャーは、複雑なチャンネルまたはリモート・キューを定義しなくても互いに通信することができます。最小クラスターでも、システム管理のオーバーヘッドが短縮します。

クラスター内のキュー・マネージャーのネットワークを確立する場合、従来の分散キューイング環境を確立するのに比べて、定義が少なくてすみます。作成する定義が少ないので、ネットワークを素早く、簡単にセットアップまたは変更することができます。定義にエラーが発生するリスクが軽減されます。

クラスターをセットアップするには、通常、キュー・マネージャーにつき1つのクラスター送信側 (CLUSSDR) 定義と1つのクラスター受信側 (CLUSRCVR) 定義が必要です。伝送キュー定義またはリモート・キュー定義は必要ありません。リモート管理の基本は、クラスター内で使用されるときは同じですが、定義自体は大幅に単純化されます。

クラスター、その属性およびセットアップ方法の詳細については、*MQSeries キュー・マネージャー・クラスター* を参照してください。

MQSC コマンドを使用するローカル・キュー・マネージャーからのリモート管理

この節では、ローカル・キュー・マネージャーからリモート・キュー・マネージャーを管理する方法について説明します。次のものを使用して、ローカル・ノードからリモート管理を実現できます。

- MQSC コマンド

リモート管理

- PCF コマンド

キューおよびチャネルの作成方法は、基本的にはどちらの方法についても同じです。本書では、理解しやすいように各例で MQSC コマンドを示しています。ただし、必要に応じて、それらの例を PCF に変換することができます。PCF を使用した管理プログラムの作成方法に関する詳細については、*MQSeries プログラム式システム管理* を参照してください。

リモート管理では、対話式で、あるいはコマンドを含んでいるテキスト・ファイルから、MQSC コマンドをリモート・キュー・マネージャーに送ります。リモート・キュー・マネージャーは、同じマシン上に存在する場合がありますが、通常は、異なったマシン上に存在します。キュー・マネージャーは、AIX[®]、AS/400、MVS/ESA、および OS/2[®] など、さまざまな MQSeries 環境でリモートから管理することができます。

リモート管理を実施するには、特定のオブジェクトを作成する必要があります。特殊な要件がない限り、デフォルト値 (メッセージ長など) で十分です。

リモート管理のためのキュー・マネージャーを作成する

87ページの図6 は、`runmqsc` コマンドを使用するリモート管理に必要なキュー・マネージャーおよびチャネルの構成を示しています。

`source.queue.manager` は、MQSC コマンドを発行できる ソース キュー・マネージャーであり、それらのコマンド (オペレーター・メッセージ) の結果も、可能ならここに戻されます。`target.queue.manager` は、コマンドを処理し、オペレーター・メッセージを生成する宛先キュー・マネージャーです。

注: `source.queue.manager` は、デフォルト・キュー・マネージャーでなければなりません。キュー・マネージャーの作成についての詳細は、320ページの『`crtmqm` (キュー・マネージャーの作成)』を参照してください。

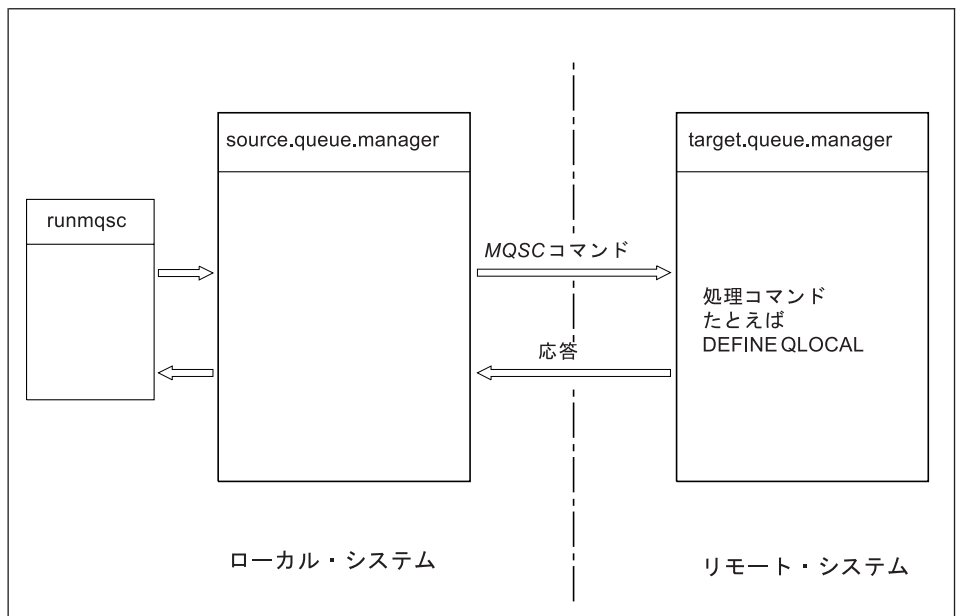


図6. リモート管理

両方のシステムで、次の事柄を行う必要があります (まだ行っていない場合)。

- **crtmqm** コマンドを使用して、キュー・マネージャーを作成します。
- **strmqm** コマンドを使用して、キュー・マネージャーを開始します。

これらのコマンドは、ローカルで実行するか、Telnet などのネットワーク機能を介して実行する必要があります。

宛先キュー・マネージャーに対して、次のようにします。

- コマンド・キュー `SYSTEM.ADMIN.COMMAND.QUEUE` が存在している。
このキューは、キュー・マネージャーが作成されるときにデフォルトとして作成されます。
- **strmqcsv** コマンドを使用してコマンド・サーバーを開始する。

リモート管理のためにチャンネルおよび伝送キューを作成する

リモートから MQSC コマンドを実行するには、2 つのチャンネル (各方向ごとに 1 つ) およびそれらに関連した伝送キューをセットアップする必要があります。この例では、TCP/IP がトランスポート・タイプとして使用されていること、および関係している TCP/IP アドレスをユーザーが把握していることが前提条件です。

チャンネル `source.to.target` は、MQSC コマンドをソース・キュー・マネージャーから宛先に送るためのものです。送信側は `source.queue.manager` であ

リモート管理

り、受信側はキュー・マネージャー `target.queue.manager` です。チャンネル `target.to.source` は、コマンドの出力および生成されたオペレーター・メッセージをソース・キュー・マネージャーに戻すためのものです。各送信側ごとに伝送キューを定義する必要もあります。このキューは、受信側のキュー・マネージャーの名前が付けられたローカル・キューです。キュー・マネージャー別名を使用していない限り、リモート管理を行うために、`XMITQ` 名がリモート・キュー・マネージャー名と一致している必要があります。図7にこの構成が示されています。

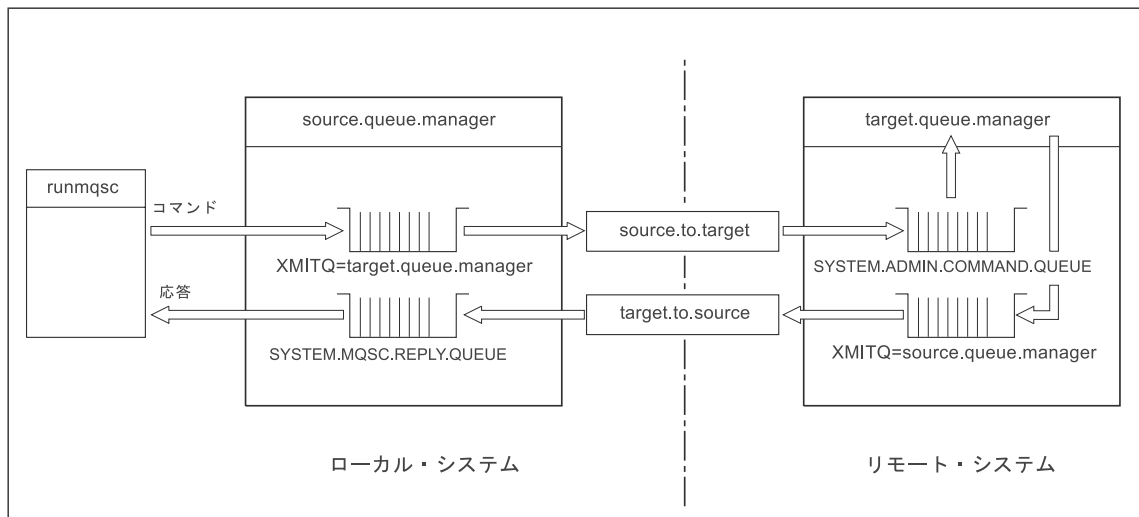


図7. リモート管理のためのチャンネルとキューのセットアップ

リモート・チャンネルのセットアップに関する詳細については、*MQSeries 相互通信* を参照してください。

チャンネルと伝送キューを定義する

送信元キュー・マネージャーに対して次の MQSC コマンドを出して、チャンネルと伝送キューを定義します。

```

* Define the sender channel at the source queue manager
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME (RHX5498) +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)

* Define the receiver channel at the source queue manager

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Define the transmission queue on the source

DEFINE QLOCAL ('target.queue.manager') +
  USAGE (XMITQ)

```

これらのコマンドを宛先キュー・マネージャー (target.queue.manager) に対して発行し、そこにチャンネルと伝送キューを作成します。

```

* Define the sender channel on the destination queue manager

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(SDR) +
  CONNAME (RHX7721) +
  XMITQ ('source.queue.manager') +
  TRPTYPE(TCP)

* Define the receiver channel on the destination queue manager

DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Define the transmission queue on the destination queue manager

DEFINE QLOCAL ('source.queue.manager') +
  USAGE (XMITQ)

```

注: 送信側チャンネル定義の CONNAME 属性に指定された TCP/IP 接続名は、説明のためだけに示したものです。これは、接続の他方側のマシンのネットワーク名です。各自ネットワークに合った値を使用してください。

チャンネルを開始する

次の説明は、両側のチャンネルが MQSeries for Compaq OpenVMS 上で実行されていることを想定しています。このようになっていない場合は、OpenVMS ではない側のチャンネルの該当する資料を参照してください。

リモート管理

2 つのチャンネルを開始するには、両方のノードで TCP/IP サービスが MQSeries 用に構成されていて、接続の両側で実行されていることをまず確認してください。

各チャンネルの受信側でリスナーを開始します。

- 送信元キュー・マネージャーに対して、次のように入力します。

```
runmqtsr -m "source.queue.manager" -t tcp
```

- 宛先キュー・マネージャーに対して、次のように入力します。

```
runmqtsr -m "target.queue.manager" -t tcp
```

ソース・キュー・マネージャーはデフォルト・キュー・マネージャーでなければならないので、ソース・キュー・マネージャーの **runmqchl** コマンドでは、キュー・マネージャー名を指定する必要はありません。宛先キュー・マネージャーがノード上でデフォルト・キュー・マネージャーになっていない場合は、宛先キュー・マネージャーの **runmchl** コマンドでは、キュー・マネージャー名は必須です。

その後、両方のチャンネルを開始します。

- 送信元キュー・マネージャーに対して、次のように入力します。

```
runmqchl -m "source.queue.manager" -c "source.to.target"
```

- 宛先キュー・マネージャーに対して、次のように入力します。

```
runmqchl -m "target.queue.manager" -c "target.to.source"
```

runmqtsr コマンドおよび **runmqchl** コマンドも MQSeries 制御コマンドです。これらも **runmqsc** を使用して発行することはできません。リスナーとチャンネルは、**runmqsc** コマンドまたはスクリプト (start channel と start listener) を使用して開始することができます。

チャンネルの自動定義

チャンネルの自動定義は、MQSeries バージョン 5.1 以降のプロダクトにおいて宛先キュー・マネージャーが実行している場合にだけ適用します。インバウンド接続要求を受信して、チャンネル定義ファイル (CDF) に適切な受信側またはサ

サーバー接続定義がない場合、MQSeries は自動的に定義を作成し、その定義を CDF に追加します。自動定義は MQSeries で提供される 2 つのデフォルト定義 SYSTEM.AUTO.RECEIVER および SYSTEM.AUTO.SVRCONN に基づいています。

MQSC コマンド、ALTER QMGR (またはキュー・マネージャーの変更 PCF コマンド) を使用してキュー・マネージャー・オブジェクトを更新することによって、受信側の自動定義およびサーバー接続定義を使用可能にします。

チャンネル定義の自動作成に関する詳細については、MQSeries 相互通信 を参照してください。

クラスターのためのチャンネルの自動定義に関する詳細については、MQSeries キュー・マネージャー・クラスター を参照してください。

リモートから MQSC コマンドを発行する

MQSC コマンドをリモートから処理する場合には、コマンド・サーバーが宛先キュー・マネージャー上で実行されている**必要があります**。(ソース・キュー・マネージャーで実行されている必要はありません)。

- 宛先キュー・マネージャーに対して、次のように入力します。

```
strmqcsv "target.queue.manager"
```

- 次に、送信元キュー・マネージャーに対して次のように入力することにより、キュー・モードで対話的に MQSC を実行できます。

```
runmqsc -w 30 "target.queue.manager"
```

この形式の **runmqsc** コマンド (-w フラグ付き) は、キュー・モードで MQSC コマンドを実行します。このモードでは、コマンドがコマンド・サーバーの入力キューに (修正された形式で) 書き込まれ、順次に行われます。

MQSC コマンドを入力すると、このコマンドはリモート・キュー・マネージャー (ここでは、target.queue.manager) にリダイレクトされます。タイムアウトは 30 秒に設定されます。したがって、30 秒以内に応答がなければ、ローカル (送信元) キュー・マネージャーで次のメッセージが生成されます。

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

リモート管理

MQSC セッションの終了時に、ローカル・キュー・マネージャーは、到着したタイムアウト応答を表示します。MQSC セッションが終了すると、それ以降の応答は破棄されます。

間接モードでは、MQSC コマンドもリモート・キュー・マネージャー上で実行することができます。たとえば、次のとおりです。

```
runmqsc -w 60 "target.queue.manager" < mycomds.in > report.out
```

ここで、mycomds.in は MQSC コマンドを含んでいるファイルであり、report.out はレポート・ファイルです。

MVS/ESA でのキュー・マネージャーの取り扱い

MQSeries for Compaq OpenVMS キュー・マネージャーから、MVS/ESA キュー・マネージャーに対して MQSC コマンドを発行することができます。ただし、これを行うには、送信側の **runmqsc** コマンドとチャネル定義を修正する必要があります。

特に、OpenVMS ノード側で **runmqsc** コマンドに **-x** フラグを追加します。

```
runmqsc -w 30 -x "target.queue.manager"
```

送信側チャネルで、**CONVERT** 属性を **YES** に設定します。これで、システム間で必要なデータ変換が OpenVMS 側で実行されることが指定されます。チャネル定義コマンドは、次のようになります。

```
* Define the sender channel at the source queue manager on OpenVMS  
  
DEFINE CHANNEL ('source.to.target') +  
  CHLTYPE(SDR) +  
  CONNAME (RHX5498) +  
  XMITQ ('target.queue.manager') +  
  TRPTYPE(TCP) +  
  CONVERT (YES)
```

前と同じように、送信元キュー・マネージャーに受信側チャネルと伝送キューを定義する必要もあります。この例でも、TCP/IP が伝送プロトコルとして使用されていることを想定しています。

リモート・キューイングについての勧告

リモート・キューイングを実施する際に、次のようにしてください。

1. リモート・システムで実行する MQSC コマンドは、コマンド・ファイルに入れます。
2. **runmqsc** コマンドに **-v** フラグを指定することにより、MQSC コマンドをローカルで確認します。
runmqsc を使用して別のキュー・マネージャー上の MQSC コマンドを確認することはできません。
3. コマンド・ファイルがローカルでエラーなしで実行されることを可能な限り確認します。
4. 最後に、そのコマンド・ファイルをリモート・システムに対して実行します。

リモートからの MQSC の使用に問題がある場合

リモートから MQSC コマンドを実行する際に問題が生じた場合、次のリストに従ってチェックしてください。

- 宛先キュー・マネージャーのコマンド・サーバーを開始しましたか。
- 有効な伝送キューを定義しましたか。
- 次の両方について、メッセージ・チャネルの両端を定義しましたか。
 - コマンドが送信されるチャネル
 - 応答が戻されるチャネル
- チャネル定義に正しい結合名 (CONNAME) を指定しましたか。
- メッセージ・チャネルを開始する前に、リスナーを開始しましたか。
- 切断時間の間隔が期限切れでないかどうか (チャネルが開始したが、しばらくしてシャットダウンされた場合など) について検査しましたか。これは、チャネルを手動操作で開始した場合に特に重要です。
- 意味をなさない要求 (新しいパラメーターを含んだ要求など) を発信元キュー・マネージャーから宛先キュー・マネージャーに送信しないようにしてください。

56ページの『MQSC で起こった問題の解決』も参照してください。

リモート・キューのローカル定義の作成

リモート・キュー定義は、リモート・キューのローカル定義として使用できません。別のキュー・マネージャー上のローカル・キューを識別するために、自分のローカル・キュー・マネージャーにリモート・キュー・オブジェクトを作成します。

リモート・キューのローカル定義

リモート・キューのローカル定義の働きについて理解する

アプリケーションは、ローカル・キュー・マネージャーに接続し、その後 MQOPEN 呼び出しを出します。オープン呼び出しで指定されるキュー名は、ローカル・キュー・マネージャー上のリモート・キュー定義のキュー名です。リモート・キュー定義は、宛先キューの名前、宛先キュー・マネージャーの名前、および伝送キューの名前（オプション）を提供します。リモート・キューにメッセージを書き込むためには、アプリケーションは、MQOPEN 呼び出しから戻されたハンドルを指定して、MQPUT 呼び出しを出します。キュー・マネージャーは、リモート・キュー名およびリモート・キュー・マネージャー名を、メッセージの伝送ヘッダーに追加します。この情報は、ネットワーク内の正しい宛先にメッセージを転送するために使用されます。

管理者は、リモート・キュー定義を更新することにより、メッセージの宛先を制御できます。

例

目的: アプリケーションは、リモート・キュー・マネージャーが所有しているキューにメッセージを書き込む必要があります。

処理内容: アプリケーションはキュー・マネージャー (saturn.queue.manager など) に接続します。宛先キューは、別のキュー・マネージャーが所有しています。

MQOPEN 呼び出しで、アプリケーションは次のフィールドを指定します。

フィールド値	説明
<i>ObjectName</i> CYAN.REMOTE.QUEUE	リモート・キュー・オブジェクトのローカル名を指定します。これは、宛先キューと宛先キュー・マネージャーを定義します。
<i>ObjectType</i> (キュー)	このオブジェクトをキューとして識別します。
<i>ObjectQmgrName</i> ブランク または saturn.queue.manager	このフィールドの指定はオプションです。 ブランクの場合、ローカル・キュー・マネージャーの名前と見なされます。(これは、リモート・キュー定義が作成され、アプリケーションが接続されているキュー・マネージャーです。) ブランクでない場合、ローカル・キュー・マネージャーの名前を指定する必要があります。

この後、アプリケーションはこのキューにメッセージを書き込むために、MQPUT 呼び出しを出します。

ローカル・キュー・マネージャーでは、次の MQSC コマンドを使用してリモート・キューのローカル定義を作成することができます。

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
  DESCR ('Queue for auto insurance requests from the branches') +
  RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
  RQMNAME ('jupiter.queue.manager') +
  XMITQ (INQUOTE.XMIT.QUEUE)
```

説明

QREMOTE (CYAN.REMOTE.QUEUE)

リモート・キュー・オブジェクトのローカル名を指定します。これは、このキュー・マネージャーに接続されたアプリケーションが、リモート・キュー・マネージャー `jupiter.queue.manager` 上のキュー `AUTOMOBILE.INSURANCE.QUOTE.QUEUE` をオープンするために、MQOPEN 呼び出しに指定する必要がある名前です。

DESCR ('Queue for auto insurance requests from the branches')

キューの用途を説明する追加テキストです。

RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)

リモート・キュー・マネージャーの宛先キューの名前を指定します。これは、アプリケーションが送信する、キュー名 `CYAN.REMOTE.QUEUE` を指定したメッセージの実際の宛先キューです。キュー `AUTOMOBILE.INSURANCE.QUOTE.QUEUE` を、ローカル・キューとしてリモート・キュー・マネージャーに定義する必要があります。

RQMNAME ('jupiter.queue.manager')

宛先キュー `AUTOMOBILE.INSURANCE.QUOTE.QUEUE` を所有するリモート・キュー・マネージャーの名前を指定します。この名前は、単一引用符で囲む必要があります。

XMITQ (INQUOTE.XMIT.QUEUE)

伝送キューの名前を指定します。この指定はオプションです。これを指定しないと、リモート・キュー・マネージャーと同じ名前のキューが使用されます。

リモート・キューのローカル定義

いずれの場合でも、伝送キューであることを指定する *Usage* 属性 (MQSC に USAGE(XMIT) を指定) を持つローカル・キューとして、該当の伝送キューを定義する必要があります。

リモート・キューにメッセージを書き込む代替方法

リモート・キューのローカル定義を使用する方法以外にも、リモート・キューにメッセージを書き込む方法があります。アプリケーションは、リモート・キュー・マネージャー名を含んでいる完全なキュー名を、MQOPEN 呼び出しの一部として指定することができます。この場合、リモート・キューのローカル定義は不要です。ただし、この代替方法の場合、アプリケーションがリモート・キュー・マネージャーの名前を認識しているか、実行時にリモート・キュー・マネージャーの名前にアクセスできなければなりません。

リモート・キューに関してその他のコマンドを使用する

該当の MQSC コマンドを使用すると、リモート・キュー・オブジェクトの属性を表示または変更したり、リモート・キュー・オブジェクトを削除したりすることができます。たとえば、次のとおりです。

```
* Display the remote queue's attributes.  
* ALL = Display all attributes  
  
DISPLAY QUEUE (CYAN.REMOTE.QUEUE) ALL  
  
* ALTER the remote queue to enable puts.  
* This does not affect the destination queue,  
* only applications that specify this remote queue.  
  
ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)  
  
* Delete this remote queue  
* This does not affect the destination queue  
* only its local definition  
  
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

注: リモート・キューを削除する場合、削除するのはリモート・キューのローカル表示のみです。リモート・キューやリモート・キュー上のメッセージは削除されません。

伝送キューを作成する

伝送キューとは、キュー・マネージャーがメッセージ・チャンネルを介してメッセージをリモート・キュー・マネージャーに転送する際に使用されるローカル・キューのことです。チャンネルは、リモート・キュー・マネージャーへの片方向リンクを提供します。メッセージは、チャンネルがメッセージを受け入れることができるまで、伝送キューにキューイングされます。チャンネルを定義する際には、メッセージ・チャンネルの送信側に伝送キュー名を指定してください。

Usage 属性 (MQSC 内の USAGE) は、キューが伝送キューであるか、通常キューであるかを定義します。

デフォルト伝送キュー

*XmitQName*属性 (MQSC 内の XMITQ) を任意で使用して、伝送キューをリモート・キュー・オブジェクトに指定できます。伝送キューを定義しなければ、デフォルト値が使用されます。アプリケーションがメッセージをリモート・キューに書き込む際に、宛先キュー・マネージャーと同じ名前を持つ伝送キューが存在している場合、そのキューが使用されます。このキューが存在していなければ、ローカル・キュー・マネージャーの *DefaultXmitQ* 属性 (MQSC 内の DEFXMITQ) によって指定されたキューが使用されます。

たとえば、次の MQSC コマンドでは、`target.queue.manager` に送られるメッセージ用として、デフォルト伝送キューが `source.queue.manager` に作成されます。

```
DEFINE QLOCAL ('target.queue.manager') +
  DESCR ('Default transmission queue for target qm') +
  USAGE (XMITQ)
```

アプリケーションは、適切なヘッダーを指定して伝送キューに直接メッセージを書き込んだり、間接的に (たとえば、リモート・キュー定義を介して) 書き込んだりすることができます。93ページの『リモート・キューのローカル定義の作成』も参照してください。

リモート・キュー定義を別名として使用する

キューを別のキュー・マネージャーに置くだけでなく、リモート・キューのローカル定義を次の 2 つの作業に使用することもできます。

- キュー・マネージャー別名付け
- 応答先キュー別名付け

別名

いずれのタイプの別名も、リモート・キューのローカル定義を使用して解決されます。

リモート・キューイングで通常行われているように、メッセージがその宛先に到着するためには、該当のチャンネルをセットアップする必要があります。

キュー・マネージャー別名

別名付けとは、宛先キュー・マネージャーの名前（メッセージ内に指定されている）をメッセージ経路上のキュー・マネージャーによって変更するためのプロセスです。キュー・マネージャー別名は重要です。キュー・マネージャーのネットワーク内でメッセージの宛先を制御するのに、この別名を使用できるためです。

これを行うには、制御点でキュー・マネージャーのリモート・キュー定義を変更します。送信アプリケーションは、指定されたキュー・マネージャー名が別名であることを認識しません。

キュー・マネージャー別名に関する詳細については、*MQSeries 相互通信* を参照してください。

応答先キュー別名付け

オプションとして、アプリケーションは、要求メッセージをキューに入れる際に、応答先キューの名前を指定することができます。メッセージを処理するアプリケーションは、その応答先キューの名前を取り出すときに、必要に応じて応答メッセージの送り先を確認します。

応答先キュー別名付けとは、応答先キューの名前（要求メッセージ内で指定されている）をメッセージ経路上のキュー・マネージャーによって変更するためのプロセスです。送信アプリケーションは、指定された応答先キュー名が別名であることを認識しません。

応答先キュー別名を使用すると、応答先キューの名前を更新でき、オプションでそのキュー・マネージャーを更新することもできます。これによって、応答メッセージに使用される経路を制御することができます。

要求メッセージ、応答メッセージおよび応答先キューの詳細については、*MQSeries アプリケーション・プログラミング・リファレンス* を参照してください。応答先キュー別名に関する詳細については、*MQSeries 相互通信* を参照してください。

データ変換

MQSeries で定義された形式 (組み込み形式とも呼ばれる) のメッセージ・データは、キュー・マネージャーによって 1 つのコード化文字セットからもう 1 つのコード化文字セットに変換することができます。ただし、2 つのコード化文字セットが、1 つの言語または類似する言語グループに関連付けられていることが必要です。

たとえば、ID (CCSID) がそれぞれに 850 と 500 であるコード化文字セット間の変換は、両方とも西欧の言語に該当するため、サポートされます。

ASCIIへの EBCDIC 改行 (NL) 文字変換については、224ページの『AllQueueManagers スタンザ』を参照してください。

サポートされている変換は、*MQSeries* アプリケーション・プログラミング・リファレンス のコード・ページ変換表で定義されています。

キュー・マネージャーがメッセージを組み込み形式に変換できない場合

CCSID が別の各国語グループを表している場合には、キュー・マネージャーはメッセージを組み込み形式に自動的に変換することはできません。たとえば、CCSID 850 と CCSID 1025 (キリル文字スクリプトを使用する言語用の EBCDIC コード化文字セット) 間の変換はサポートされていません。これは、一方のコード化文字セットの文字の多くが、もう一方のコード化文字セットで表現できないためです。さまざまな各国語で稼働しているキュー・マネージャーのネットワークがあり、一部のコード化文字セット間でのデータ変換がサポートされていない場合に、デフォルト変換を使用することができます。デフォルトのデータ変換については、100ページの『デフォルトのデータ変換』で説明しています。

ファイル ccsid.tbl

ファイル ccsid.tbl は、次のものを指定します。

- すべての追加のコード・セット。追加のコード・セットを指定するには、ccsid.tbl を編集する必要があります (これを行う方法はファイルで指示されています)。
- すべてのデフォルトのデータ変換。

ccsid.tbl に記録されている情報を更新することができます。たとえば、使用しているオペレーティング・システムの将来のリリースで追加のコード化文字セットがサポートされる場合に、更新が必要となる場合があります。

MQSeries for Compaq OpenVMS では、サンプルの ccsid.tbl ファイルが次のファイルとして提供されています。

MQS_EXAMPLES:CCSID.TBL

アクティブ `ccsid.tbl` ファイルは、次のディレクトリーにあります。

MQS_ROOT:[MQM.CONV.TABLE]

デフォルトのデータ変換: デフォルトのデータ変換を実施するには、`ccsid.tbl` を編集して、デフォルト EBCDIC CCSID とデフォルト ASCII CCSID を指定し、さらにデフォルト CCSID も指定します。この方法に関する指示は、`ccsid.tbl` に入っています。

`ccsid.tbl` を更新してデフォルト・データ変換を実施する場合は、変更内容が有効になる前にキュー・マネージャーを再始動する必要があります。

デフォルトのデータ変換プロセスは、次のようになります。

- ソース CCSID とターゲット CCSID の間の変換がサポートされていなくても、CCSID のソース環境およびターゲット環境の両方が EBCDIC または ASCII のいずれかである場合は、文字データは変換されずにターゲット・アプリケーションに渡されます。
- 一方の CCSID が ASCII コード化文字セットを表し、もう一方の CCSID が EBCDIC コード化文字セットを表す場合、MQSeries は `ccsid.tbl` で定義されているデフォルトのデータ変換機構 CCSID を使用してデータを変換します。

注: メッセージ用として指定されたコード化文字セットとデフォルトのコード化文字セット中で同じコード値を持つ文字に、変換対象文字を制限してください。MQSeries オブジェクト名にとって有効な文字のセットのみを使用する場合には、通常、この要件を満たします。日本で使用されている EBCDIC CCSID 290、930、1279、および 5026 では例外が発生します。この場合、小文字は他の EBCDIC CCSID で使用されるものとは異なるコードを持ちます。

ユーザー定義形式でのメッセージの変換

ユーザー定義形式のメッセージを、キュー・マネージャーによって 1 つのコード化文字セットから別のコード化文字セットに変換することはできません。ユーザー定義形式のデータが変換を必要とする場合は、それぞれの形式ごとにデータ変換出口が必要となります。ユーザー定義形式の文字データの变換のために、デフォルト CCSID を使用することはできますが、お勧めできません。ユーザー定義形式のデータ変換およびデータ変換出口の作成に関する詳細については、*MQSeries アプリケーション・プログラミング・ガイド* を参照してください。

キュー・マネージャー CCSID の変更

キュー・マネージャーの CCSID の変更時には、ALTER QMGR コマンドの CCSID 属性を使用して、キュー・マネージャーを停止して再始動することをお勧めします。

これによって、コマンド・サーバーとチャンネル・プログラムを含むすべての実行中のアプリケーションが、停止して再始動するようになります。

キュー・マネージャー CCSID の変更時に実行中であったアプリケーションは、既存の CCSID を使用し続けるので、このことが必要です。

第7章 MQSeries オブジェクトの保護

この章では、MQSeries for Compaq OpenVMS でのセキュリティー管理について、またセキュリティー管理をインプリメントする方法について説明します。

この章には、次の節があります。

- 『MQSeries リソースを保護する理由』
- 104ページの『開始する前に』
- 105ページの『オブジェクト権限マネージャーについての理解』
- 109ページの『オブジェクト権限マネージャー・コマンドの使用』
- 112ページの『オブジェクト権限マネージャーのガイドライン』
- 117ページの『許可指定テーブルについての理解』
- 125ページの『許可ファイルの理解』

MQSeries リソースを保護する理由

MQSeries キュー・マネージャーは潜在的価値のある情報の転送を取り扱うため、許可システムによる安全対策が必要になります。これによって、キュー・マネージャーが所有して取り扱うリソースが無許可アクセスから保護されることになり、データの消失または開示のおそれなくなります。システム・セキュリティーでは、無許可ユーザーまたは無許可アプリケーションが次のものをアクセスしたり、変更できないようにすることが重要です。

- キュー・マネージャーとの接続。
- キュー、クラスター、チャンネルおよびプロセスなどの MQSeries オブジェクトへのアクセス。
- MQSC コマンドおよび PCF コマンドを含むキュー・マネージャー管理用のコマンド。
- MQSeries メッセージへのアクセス。
- メッセージに関連したコンテキスト情報。

どのユーザーにどのリソースをアクセスさせるかについて、独自の方針を確立してください。

開始する前に

すべてのキュー・マネージャー・リソースは、次の VMS 権限 ID を使用して実行されます。

MQM

この権限 ID は、MQSeries のインストール中に作成され、MQSeries リソースを制御する必要があるすべてのユーザーにこのリソース属性を許可する**必要があります**。

リソース ID MQM を持つ MQSeries for Compaq OpenVMS のユーザー ID

ユーザー ID が MQM OpenVMS 権限 ID を保持している場合は、すべての MQSeries リソースに対するすべての許可を持っています。 **crtmqcvx** を除くすべての MQSeries for Compaq OpenVMS 制御コマンドを使用するためには、ユーザー ID が OpenVMS MQM 権限を保持している**必要があります**。特に、次の許可が必要です。

- **runmqsc** コマンドを使用して MQSC コマンドを実行する。
- **setmqaut** コマンドを使用して、MQSeries for Compaq OpenVMS に関する許可を管理する。

リモート・システムのキュー・マネージャーにチャンネル・コマンドを送信する場合は、ユーザー ID はターゲット・システムの OpenVMS 権限 ID MQM を保持している**必要があります**。 PCF および MQSC チャンネル・コマンドについては、115ページの『チャンネル・コマンド・セキュリティ』のリストを参照してください。

さらに、MQSeries をインストールすると、ID MQS_SERVER が作成されます。これには、VMS が MQSeries 用のロック情報を保持するリソース・ドメインの所有権が付与されます。デフォルトでは、この ID へのアクセス権限は次のユーザーに認可されます。

- ユーザー MQM と同じユーザー・グループに属するユーザー。または、
- システム・ユーザー。または、
- SYSPRV、SYSLOCK、または BYPASS 特権セットを持つユーザー。

他のユーザーが MQ リソースにアクセスできるようにするには、次のコマンドを実行して、MQS_SERVER ID が適切な WORLD 特権を持っていることを確かめる**必要があります**。

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

注: 次のものを発行する場合、ユーザー ID が権限 ID MQM を保持していなくても構いません。

- 管理プログラムから PCF コマンド (エスケープ PCF を含む) を発行する場合
- アプリケーション・プログラムから MQI 呼び出しを発行する場合。

詳細情報

次のものの詳細については、示されている資料を参照してください。

- MQSeries for Compaq OpenVMS コマンド・セットについては、25ページの『第2章 MQSeries 管理の紹介』を参照してください。
- MQSeries for Compaq OpenVMS 制御コマンドについては、313ページの『第17章 MQSeries 制御コマンド』を参照してください。
- PCF コマンドおよびエスケープ PCF については、MQSeries プログラム式システム管理 を参照してください。
- MQI 呼び出しについては、MQSeries アプリケーション・プログラミング・ガイド および MQSeries アプリケーション・プログラミング・リファレンスを参照してください。

オブジェクト権限マネージャーについての理解

デフォルトでは、キュー・マネージャー・リソースへのアクセスは、許可サービスのインストール可能なコンポーネントを介して管理されます。このコンポーネントは、正式には MQSeries for Compaq OpenVMS 用オブジェクト権限マネージャー (OAM) と呼ばれます。これは MQSeries for Compaq OpenVMS と共に提供されており、特別の指定をしない限り、自動的にインストールされ、作成されるキュー・マネージャーごとに使用可能になります。この章で使用される OAM とは、このプロダクトと共に提供されるオブジェクト権限マネージャーのことです。

OAM は、許可サービスのインストール可能なコンポーネント です。OAM がインストール可能なサービスとして提供されるため、次のような柔軟な対応が可能です。

- 提供されているインターフェースを使用して、この OAM を独自の許可サービス・コンポーネントで置き換える。
- 提供されているインターフェースを再度使用して、独自の許可サービス・コンポーネントにより、この OAM の機能を強化する。
- OAM を除去するか、使用不可にして、許可サービスなしで実行する。

オブジェクト権限マネージャー

インストール可能サービスの詳細については、 *MQSeries* プログラム式システム管理 を参照してください。

OAM は、キュー、プロセス定義、およびチャネルなどの MQSeries オブジェクトを操作するためのユーザーの許可を管理します。また、OAM は、特定のオブジェクトへのアクセス許可を特定のグループのユーザーに与えたり、取り消したりするためのコマンド・インターフェースを提供します。あるリソースへのアクセスを認める決定は OAM が行い、キュー・マネージャーはその決定に従います。OAM が決定できない場合は、キュー・マネージャーは該当のリソースへのアクセスを妨げます。

OAM の機能

OAM は、基礎となる OpenVMS オペレーティング・システムのセキュリティ機能を利用して働きます。特に、OAM は OpenVMS のユーザー、グループ ID、および権限 ID を使用します。ユーザーは、必要な許可を持っている場合にだけ、キュー・マネージャー・オブジェクトにアクセスできます。

権限 ID を介してアクセスを管理する

コマンド・インターフェースでは、ユーザー ID ではなく、プリンシパルという用語を使用します。その理由は、ユーザー ID に与えられる許可は、別のエンティティ、たとえば、MQI 呼び出しを発行するアプリケーション・プログラムや PCF コマンドを発行する管理プログラムにも与えられるためです。これらの場合、プログラムに関連付けられたプリンシパルは、プログラムの開始時に使用されたユーザー ID とは限りません。ただし、この章の説明では、プリンシパルおよびユーザー ID は常に OpenVMS ユーザー ID です。

権限 ID および 1 次権限 ID

MQSeries リソースへのアクセス許可の管理は、OpenVMS 権限 ID、すなわちプリンシパルが保持している ID に基づいています。1 つのプリンシパルは、1 つまたは複数の OpenVMS 権限 ID を保持できます。グループは、特定の権限 ID を許可されたすべてのプリンシパルのセットとして定義されます。

OAM は、個々のプリンシパルではなく権限 ID のレベルで許可を保守します。プリンシパルの ID 名へのマッピングは、OAM 内で実行され、オペレーションは権限 ID レベルで実行されます。ただし、個々のプリンシパルの許可を表示することはできません。

プリンシパルが複数の権限 ID を保持している場合

プリンシパルの持つ許可は、プリンシパルが保持しているすべての権限 ID の許可のユニオン、つまりプリンシパルのプロセス権限です。プリンシパルがリ

ソースへのアクセスを要求すると、OAM はこのユニオンを計算し、これに対する許可を検査します。制御コマンド **setmqaut** を使用すると、特定のプリンシパルまたは ID の許可を設定できます。

注: setmqaut コマンドを使用して行われたすべての変更の影響は、オブジェクトが使用中でない限り即時に現れます。オブジェクトが使用中の場合は、オブジェクトが次にオープンされるときに変更が有効になります。ただし、プリンシパルの権限 ID リストに対する変更は、キュー・マネージャーがリセットされない限り、つまり停止されて再始動されない限り有効になりません。

プリンシパルに関連する許可は、OAM によって計算される時にキャッシュに入れられます。キャッシュに入れられた後で ID の許可に対して行われる変更は、キュー・マネージャーが再始動されるまで認識されません。キュー・マネージャーの実行中は、どの許可も変更しないでください。

デフォルト権限 ID

OAM は、すべてのユーザーが名目上割り当てられるデフォルトを認識します。このグループは、'NOBODY' という疑似権限 ID によって定義されます。'NOBODY' は、MQSeries コマンドを使用して許可を割り当てるための有効な権限 ID であるかのように使用できます。デフォルトでは、この ID には許可は与えられていません。特定の許可を持たないユーザーは、この権限 ID を使用して MQSeries リソースへのアクセスが許可されます。

OAM で保護できるリソース

OAM を使用することによって次の事柄が制御できます。

- MQI を介する MQSeries オブジェクトへのアクセス。アプリケーション・プログラムがオブジェクトにアクセスしようとする時、OAM は、要求された操作に関する (保持されている ID を介した) 許可を要求元のユーザー ID が持っているかどうかを調べます。
特に、これはキューおよびキュー上のメッセージを無許可アクセスから保護することを意味します。
- MQSC コマンドを使用する許可。権限 ID MQM を保持しているプリンシパルだけが、たとえば、キューを作成するために、キュー・マネージャー管理コマンドを実行できます。
- 制御コマンドを使用する許可。権限 ID MQM を保持しているプリンシパルだけが、たとえば、キュー・マネージャーの作成、コマンド・サーバーの開始、または **runmqsc** の使用などの、制御コマンドを実行できます。
- PCF コマンドを使用する許可。

オブジェクト権限マネージャー

同じオブジェクトに対して、ユーザーごとに異なる種類のアクセス許可を与えることができます。たとえば、特定のキューに対して、ある ID を保持しているユーザーには書き込み操作と読み取り操作の両方を許可し、別の ID を保持しているユーザーにはキューのブラウズ (ブラウズ・オプションによる MQGET) のみを許可することができます。また、複数の ID を持つユーザーには、あるキューの読み取りおよび書き込みは許可するが、そのキューの変更または削除は認めないということもできます。

許可に権限 ID を使用する

個別のプリンシパルでなく、ID を使用することにより、必要な管理タスクの量を減らすことができます。特定の種類のアクセスを複数のプリンシパルが必要とするのが普通であるためです。たとえば、特定のアプリケーションの実行を希望するエンド・ユーザーからなる ID を定義できます。新しいユーザーの場合は、適切な ID をそのユーザーの OpenVMS ユーザー ID に与えるだけで、アクセスを許可できます。

ID の数はできるだけ少なくします。たとえば、最初は、プリンシパルを 1 グループのアプリケーション・ユーザーと 1 グループの管理者とに分けることをお勧めします。

オブジェクト権限マネージャーを使用不可にする

デフォルトでは、OAM は使用可能になっています。キュー・マネージャーを作成する前に、次のように論理名 MQSNOAUT を設定すると、OAM を使用不可にすることができます。

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

ただし、一般にこの方法では、あとで OAM を再始動できません。OAM を使用可能にして、すべてのユーザーおよびアプリケーションが適切なユーザー ID を介してアクセスできるようにしておく方が賢明です。

また、キュー・マネージャー構成ファイル (qm.ini) の許可サービス・スタンザを除去することにより、テスト目的だけに OAM を使用不可にすることができます。

オブジェクト権限マネージャー・コマンドの使用

OAM は、許可を与えたり、取り消したりするためのコマンド・インターフェースを提供しています。これらのコマンドを使用するには、そのための許可が必要です。つまり、ユーザー ID は OpenVMS 権限 ID MQM を保持している必要があります。この ID は、プロダクトのインストール時にセットアップされているはずです。

ユーザー ID が ID MQM を保持している場合は、キュー・マネージャーに対する管理権限があります。これは、ユーザー ID から、すべての MQI 要求またはコマンドを発行する許可が与えられているということです。

OAM は、ユーザーの許可を管理するために OpenVMS DCL から起動できる 2 つのコマンドを提供しています。そのコマンドは次のとおりです。

- **setmqaut** (権限の設定またはリセット)
- **dspmqaut** (権限の表示)

権限検査は、MQCONN、MQOPEN、MQPUT1、および MQCLOSE の呼び出しの際に発生します。

権限検査は、これらすべての呼び出しの最初のインスタンスだけで実行され、リセットする (つまり、クローズして再オープンする) まで権限は修正されません。

したがって、**setmqaut** を使用してオブジェクトの許可に加えられた変更は、いずれもオブジェクトをリセットするまでは有効になりません。

OAM コマンドの使用時に指定するもの

権限コマンドは指定したキュー・マネージャーに適用されます。キュー・マネージャーを指定しないと、デフォルト・キュー・マネージャーが使用されます。これらのコマンドには、オブジェクトを固有に指定する必要があります。つまり、オブジェクト名とそのタイプを指定する必要があります。また、権限を適用するプリンシパルまたは ID 名も指定する必要があります。

許可リスト

setmqaut コマンドには許可のリストを指定します。この方法は、許可の付与または取り消しを指定したり、さらに許可が適用されるリソースを指定する場合に役立ちます。リストの各許可は、小文字のキーワードの前に + または - 符号を付けて指定します。指定した許可を追加するときには + 符号を使用し、

OAM コマンドの使用

許可を取り消すときには - 符号を使用します。単一のコマンドで、許可をいくつでも指定できます。たとえば、次のとおりです。

```
+browse -get +put
```

setmqaut コマンドを使用する

必要な許可を持っている場合は、**setmqaut** コマンドを使用すると、特定のオブジェクトをアクセスする許可をプリンシパルまたは権限 ID に与えたり、取り消したりできます。次の例は、**setmqaut** コマンドの使い方を示しています。

```
setmqaut -m "saturn.queue.manager" -t queue -n RED.LOCAL.QUEUE -g GROUPA +browse -get +put
```

この例のそれぞれの指定の意味は次のとおりです。

語句	指定内容
saturn.queue.manager	キュー・マネージャー名。
queue	オブジェクト・タイプ。
RED.LOCAL.QUEUE	オブジェクト名。
GROUPA	許可を与えるグループの ID。
+browse -get +put	指定したキューの許可リスト。「+」または「-」符号とキーワードの間にはスペースを入れません。

許可リストは、付与される許可を指定します。次のとおりです。

語句	結果
+browse	キュー上のメッセージをブラウズする (ブラウズ・オプションを指定した MQGET) 許可を追加します。
-get	キューからメッセージを読み取る (MQGET) 許可を取り消します。
+put	キューにメッセージを書き込む (MQPUT) 許可を追加します。

これは、OpenVMS ID GROUPA を保持しているユーザー ID で開始したアプリケーションには、これらの許可があるということです。

1 つまたは複数のプリンシパルを指定することができ、また、同時に 1 つまたは複数の ID を指定することができます。たとえば、次のコマンドはキュー MyQueue に対する書き込み権限を、プリンシパル FVUSER と、ID GROUPA および GROUPB に対して取り消します。

```
setmqaut -m "saturn.queue.manager" -t queue -n "MyQueue" -p FVUSER -g GROUPA -g GROUPB -put
```

注: このコマンドは、FVUSER によって保持されているすべての権限 ID (つまり、FVUSER が属するすべてのグループ) の書き込み権限も取り消します。

このコマンドの形式的定義と構文については、377ページの『setmqaut (許可の設定 / リセット)』を参照してください。

権限コマンドとインストール可能なサービス

setmqaut コマンドは、更新が適用されるインストール可能なサービス・コンポーネントの名前を指定する追加パラメーターを取ります。同時に実行される複数のインストール可能なコンポーネントがある場合は、このパラメーターを指定する必要があります。デフォルトでは、このパラメーターは使用されません。このパラメーターを省略した場合は、最初の該当タイプのインストール可能なコンポーネント (存在する場合) が更新されます。デフォルトでは、これはシステムに提供された OAM です。

アクセス許可

setmqaut コマンドに関連した許可リストで定義される許可は、次のように類別できます。

- MQI 呼び出しに関する許可
- 管理コマンドに関する許可
- コンテキスト許可
- 一般許可、すなわち、MQI 呼び出しまたはコマンドに関するもの、あるいはその両方に関するもの

各許可は、**setmqaut** および **dspmqaut** コマンドで使用するキーワードで指定します。これらのコマンドについては、377ページの『setmqaut (許可の設定 / リセット)』で説明しています。

権限の表示コマンド

コマンド **dspmqaut** を使用して、特定のオブジェクトに関して特定のプリンシパルまたは ID が持つ許可を表示することができます。フラグの意味は、**setmqaut** コマンドの場合と同じです。一度に 1 つの ID またはプリンシパル

OAM コマンドの使用

の許可のみが表示されます。このコマンドの形式的な指定については、330ページの『dspmqaut (許可の表示)』を参照してください。

たとえば、次のコマンドは、キュー・マネージャー QueueMan1 にある Annuities という名前のプロセス定義について GpAdmin が持っている許可を表示しています。

```
dspmqaut -m "QueueMan1" -t process -n "Annuities" -g "GpAdmin"
```

このコマンドの結果として表示されるキーワードは、アクティブな許可を示しています。

オブジェクト権限マネージャーのガイドライン

一部の操作は特に重要度が高いので、その実行は特権ユーザーに限る必要があります。たとえば、次のような操作があります。

- キュー・マネージャーの開始および停止
- 伝送キューまたはコマンド・キュー SYSTEM.ADMIN.COMMAND.QUEUE などの特殊キューへのアクセス
- 完全な MQI コンテキスト・オプションを使用するプログラム
- 一般に、アプリケーション・キューの作成およびコピー

ユーザー ID

プロダクトのインストール中に作成した特殊なユーザー ID MQM は、プロダクトによる使用のみを目的としています。特権のないユーザーは使用できません。

MQ プロセスと関連した許可検査に使用されるユーザー ID は、OpenVMS ユーザー ID です。

キュー・マネージャー・ディレクトリー

キューおよびその他のキュー・マネージャー・データを入れるディレクトリーは、プロダクト専用です。このディレクトリーの中のオブジェクトには、OAM 許可に関連する OpenVMS ユーザー許可があります。ただし、OpenVMS の標準コマンドを使用して MQI リソースへの許可を与えたり、取り消したりしないでください。理由は次のとおりです。

- MQSeries オブジェクトは、対応するシステム・オブジェクトと必ずしも同じではないからです。この点に関する詳細については、28ページの『MQSeries のファイル名についての理解』を参照してください。

- すべてのオブジェクトはリソース ID MQM によって所有されています。

キュー

動的キューに対する権限は、それが派生したモデル・キューに対する権限に基づきます (ただし、必ずしも同じではありません)。詳細については、121 ページの1 を参照してください。

別名キューまたはリモート・キューの場合、許可はオブジェクト自体に関するものであり、別名キューまたはリモート・キューが解決されるキューの許可ではありません。したがって、ユーザー ID に別名キューへのアクセスを許可し、その解決先のローカル・キューへのアクセスは認めないという場合もあります。

キューを作成する権限は、特権ユーザーに限定してください。限定しないと、一部のユーザーが単に別名を作成して通常のアクセス管理を逃れる事態になりかねません。

代替ユーザー権限

代替ユーザー権限は、1 つのユーザー ID が MQSeries オブジェクトにアクセスしている間に、別のユーザー ID の権限を使用できるかどうかを制御するものです。これは、サーバーがプログラムから要求を受け取り、その要求に必要な許可をプログラムに与えようとする場合に重要です。サーバーは、要求に必要な許可があっても、要求したアクションに関する許可がプログラムにあるかどうかを確認する必要があります。

たとえば、次のとおりです。

- ユーザー ID PAYSERV のもとで実行中のサーバーが、キューから要求メッセージを取り出したとします。この要求メッセージは、ユーザー ID USER1 によってキューに置かれます。
- サーバー・プログラムは、要求メッセージを読み取ると、要求を処理し、要求メッセージで指定されている応答先キューに応答を書き戻します。
- サーバーは、サーバーのユーザー ID (PAYSERV) を使用して応答先キューのオープンを許可する代わりに、別のユーザー ID (この場合は USER1) を指定することができます。この例では、PAYSERV が応答先キューをオープンするときに代替ユーザー ID として USER1 を指定できるかどうかを制御するために、代替ユーザー権限を使用することができます。

代替ユーザー ID は、オブジェクト記述子の *AlternateUserId* フィールドに指定します。

OAM のガイドライン

注: 代替ユーザー ID は、どの MQSeries オブジェクトでも使用できます。代替ユーザー ID を使用しても、別のリソース・マネージャーが使用するユーザー ID には影響しません。

コンテキスト権限

コンテキストは、特定のメッセージに適用される情報であって、メッセージの一部であるメッセージ記述子 MQMD に含まれています。コンテキスト情報は 2 つのセクションからなります。

ID セクション

この部分は、メッセージの発信者を示します。次のフィールドからなります。

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

起点セクション

このセクションは、メッセージの発信元のアプリケーションと、キューに書き込まれた日時を示します。次のフィールドからなります。

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

アプリケーションは、MQOPEN 呼び出しまたは MQPUT 呼び出しのいずれかを出すときにコンテキスト・データを指定することができます。このデータは、アプリケーションによって生成されたり、別のメッセージから渡されたり、デフォルトでキュー・マネージャーによって生成されたりします。たとえば、コンテキスト・データはサーバー・プログラムによって、要求側の ID の検査、メッセージの発信元が許可ユーザー ID のもとで実行中のアプリケーションであるかどうかのテストに使用されることがあります。

サーバー・プログラムは、*UserIdentifier* を使用して、代替ユーザーのユーザー ID を判別することができます。

コンテキスト許可は、ユーザーが MQOPEN 呼び出しまたは MQPUT1 呼び出しにコンテキスト・オプションを使用できるかどうかを制御するのに使用できます。コンテキスト・オプションの詳細については、*MQSeries アプリケーション・プログラミング・ガイド* を参照してください。コンテキストに関連する

メッセージ記述子フィールドの説明については、*MQSeries* アプリケーション・プログラミング・リファレンス を参照してください。

リモート・セキュリティーに関する考慮事項

リモート・セキュリティーについては、次のことを考慮する必要があります。

書き込み権限

複数のキュー・マネージャーにまたがるセキュリティーについては、チャンネルが別のキュー・マネージャーから送られたメッセージを受け取ったときに使用する書き込み権限を指定することができます。

チャンネル属性 `PUTAUT` は次のように指定します。

DEF デフォルト・ユーザー ID。これはメッセージ・チャンネル・エージェントが実行されるユーザー ID です。

CTX メッセージ・コンテキスト内のユーザー ID。

伝送キュー

キュー・マネージャーは、伝送キューにリモート・メッセージを自動的に書き込みます。これには特殊権限は必要ありません。しかし、メッセージを伝送キューに直接書き込むには、特殊な許可が必要です。118ページの表2 を参照してください。

チャンネル出口

チャンネル出口は、追加されたセキュリティーに使用されます。

詳細については、*MQSeries* 相互通信 を参照してください。

チャンネル・コマンド・セキュリティー

チャンネル・コマンドは、MQAI、MQSC コマンドおよび制御コマンドにより、PCF コマンドとして発行することができます。

PCF コマンド

リモート OpenVMS システムの `SYSTEM.ADMIN.COMMAND.QUEUE` に PCF メッセージを送ることによって、PCF チャンネル・コマンドを発行できます。PCF メッセージのメッセージ記述子に指定するユーザー ID は、ターゲット・システムの権限 ID `MQM` を保持している必要があります。これらのコマンドは、次のとおりです。

- *ChangeChannel*
- *CopyChannel*
- *CreateChannel*
- *DeleteChannel*
- *PingChannel*

OAM のガイドライン

- *ResetChannel*
- *StartChannel*
- *StartChannelInitiator*
- *StartChannelListener*
- *StopChannel*
- *ResolveChannel*

PCF セキュリティー要件の詳細については、*MQSeries* プログラム式システム管理 を参照してください。

MQSC チャネル・コマンド

MQSC チャネル・コマンドをリモート OpenVMS システムに出すには、PCF エスケープ・メッセージでコマンドを直接送信するか、**runmqsc**を使用して間接モードでコマンドを発行するかの 2 つの方法があります。関連した PCF メッセージのメッセージ記述子に指定するユーザー ID は、ターゲット・システムの権限 ID MQM を保持している必要があります。(PCF コマンドは、間接モードで **runmqsc**から出された MQSC コマンドに暗黙に含まれています。) これらのコマンドは、次のとおりです。

- ALTER CHANNEL
- DEFINE CHANNEL
- DELETE CHANNEL
- PING CHANNEL
- RESET CHANNEL
- START CHANNEL
- START CHINIT
- START LISTENER
- STOP CHANNEL
- RESOLVE CHANNEL

runmqsc コマンドから発行された MQSC コマンドの場合、PCF メッセージ内のユーザー ID は通常、現行ユーザーの ID です。

チャネル用の制御コマンド

チャネル用の制御コマンドの場合、それらが発行するユーザー ID は権限 ID MQM を保持している必要があります。これらのコマンドは、次のとおりです。

- **runmqchi** (チャネル開始プログラムの実行)
- **runmqchl** (チャネルの実行)

許可指定テーブルについての理解

118 ページ以降の許可指定テーブルには、許可の機能と、適用される制限が正確に定義されています。これらの表は、次のような状態に適用されます。

- MQI 呼び出しを発行するアプリケーション。
- MQSC コマンドをエスケープ PCF として発行する管理プログラム。
- PCF コマンドを発行する管理プログラム。

この節では、次のものを指定する 1 組のテーブルとして情報を示します。

実行するアクション

MQI オプション、MQSC コマンド、または PCF コマンド

アクセス制御オブジェクト

キュー、プロセス、またはキュー・マネージャー

必要な許可

「MQZAO_」定数で表す

テーブルの中で、接頭部が MQZAO_ の定数は、特定のエンティティーに関する **setmqaut** コマンドの許可リストのキーワードに対応します。たとえば、MQZAO_BROWSE はキーワード **+browse** に対応します。また、キーワード MQZAO_SET_ALL_CONTEXT はキーワード **+setall** に対応します。これらの定数は、プロダクトと共に提供されるヘッダー・ファイル `cmqzc.h` に定義されています。詳細については、127ページの『許可ファイルの内容』を参照してください。

MQI 許可

MQI 呼び出しおよびオプションのいくつかは、アプリケーションを実行するユーザー ID (またはアプリケーションが許可を想定できるユーザー ID) が適切な許可を与えられている場合にのみ、アプリケーションから発行できます。

許可検査を必要とする MQI 呼び出しは、MQCONN、MQOPEN、MQPUT1、MQCLOSE の 4 つです。

MQOPEN および MQPUT1 の場合、許可検査は、名前が解決された結果の 1 つまたは複数の名前についてではなく、オープンされるオブジェクトの名前について行われます。たとえば、アプリケーションが別名キューをオープンする許可を与えられていても、別名が解決される基本キューをオープンする許可は与えられていない場合があります。検査の規則は次のとおりです。名前解決の過程で最初に検出された定義について検査が行われます。この定義は、キュー・マネージャー別名定義が直接オープンされる場合以外はキュー・マネージャー別名ではない定義です。つまり、オブジェクト記述子の *ObjectName* フィ

許可指定テーブル

ールドに現れた名前について検査が行われます。特定のオブジェクトをオープンするには必ず許可が必要です。さらに、キューに依存しない追加の許可 (キュー・マネージャー・オブジェクトに関する許可を介して取得する) が必要なこともあります。

表2 は、各呼び出しに必要な許可を要約したものです。

表2. MQI 呼び出しに必要なセキュリティ許可

必要な許可	キュー・オブジェクト (1)	プロセス・オブジェクト	キュー・マネージャー・オブジェクト	名前リスト
MQCONN オプション	適用しない	適用しない	MQZAO_CONNECT	適用しない
MQOPEN オプション				
MQOO_INQUIRE	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)
MQOO_BROWSE	MQZAO_BROWSE	適用しない	検査しない	適用しない
MQOO_INPUT_*	MQZAO_INPUT	適用しない	検査しない	適用しない
MQOO_SAVE_ALL_CONTEXT (3)	MQZAO_INPUT	適用しない	検査しない	適用しない
MQOO_OUTPUT (通常キュー) (4)	MQZAO_OUTPUT	適用しない	検査しない	適用しない
MQOO_PASS_IDENTITY_CONTEXT (5)	MQZAO_PASS_IDENTITY_CONTEXT	適用しない	検査しない	適用しない
MQOO_PASS_ALL_CONTEXT (5, 6)	MQZAO_PASS_ALL_CONTEXT	適用しない	検査しない	適用しない
MQOO_SET_IDENTITY_CONTEXT (5, 6)	MQZAO_SET_IDENTITY_CONTEXT	適用しない	MQZAO_SET_IDENTITY_CONTEXT (7)	適用しない
MQOO_SET_ALL_CONTEXT (5, 8)	MQZAO_SET_ALL_CONTEXT	適用しない	MQZAO_SET_ALL_CONTEXT (7)	適用しない
MQOO_OUTPUT (伝送キュー) (9)	MQZAO_SET_ALL_CONTEXT	適用しない	MQZAO_SET_ALL_CONTEXT (7)	適用しない
MQOO_SET	MQZAO_SET	適用しない	検査しない	適用しない

表 2. MQI 呼び出しに必要なセキュリティ許可 (続き)

必要な許可	キュー・ オブジェクト (1)	プロセス・ オブジェクト	キュー・ マネージャー・ オブジェクト	名前リスト
MQOO_ALTERNATE_ USER_AUTHORITY	(10)	(10)	MQZAO_ ALTERNATE_ USER_ AUTHORITY (10、11)	(10)
MQPUT1 オプション				
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_ CONTEXT (12)	適用しない	検査しない	適用しない
MQPMO_PASS_ ALL_CONTEXT	MQZAO_PASS_ ALL_CONTEXT (12)	適用しない	検査しない	適用しない
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_ CONTEXT (12)	適用しない	MQZAO_SET_ IDENTITY_ CONTEXT (7)	適用しない
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT (12)	適用しない	MQZAO_SET_ ALL_CONTEXT (7)	適用しない
(伝送キュー) (9)	MQZAO_SET_ ALL_CONTEXT	適用しない	MQZAO_SET_ ALL_CONTEXT (7)	適用しない
MQPMO_ALTERNATE_ USER_AUTHORITY	(13)	適用しない	MQZAO_ _ALTERNATE_ USER_ AUTHORITY (11)	適用しない
MQCLOSE オプション				
MQCO_DELETE	MQZAO_DELETE (14)	適用しない	適用しない	適用しない
MQCO_DELETE_PURGE	MQZAO_DELETE (14)	適用しない	適用しない	適用しない

特別な注:

1. モデル・キューがオープンされる場合は、次のようになります。

許可指定テーブル

- モデル・キューの場合、指定したオープン・オプションに必要な他のすべての権限に加えて (モデル・キューのための権限も)、MQZAO_DISPLAY 権限が必要です。
 - 動的キューを作成する場合、MQZAO_CREATE 権限は必要ありません。
 - モデル・キューのオープンに使用したユーザー ID には、作成された動的キューに関するキュー特有のあらゆる権限が自動的に与えられます (MQZAO_ALL と同等)。
2. オープンされるオブジェクトのタイプに応じて、キュー、プロセス、名前リスト、またはキュー・マネージャー・オブジェクトのいずれかが検査されます。
 3. MQOO_INPUT_* も指定する必要があります。これは、ローカル・キュー、モデル・キュー、または別名キューの場合に有効です。
 4. この検査は、注 9 に示した場合以外は、すべての場合の出力において実行されます。
 5. MQOO_OUTPUT も指定する必要があります。
 6. このオプションは、MQOO_PASS_IDENTITY_CONTEXT も暗黙的に指定されます。
 7. この許可は、キュー・マネージャー・オブジェクトと個々のキューの両方に対して必要です。
 8. MQOO_PASS_IDENTITY_CONTEXT、MQOO_PASS_ALL_CONTEXT、および MQOO_SET_IDENTITY_CONTEXT も、このオプションによって暗黙的に指定されます。
 9. この検査は、Usage キュー属性として MQUS_TRANSMISSION を持ち、出力のために直接オープンされているローカル・キューまたはモデル・キューについて実行されます。リモート・キューがオープンされる場合 (リモート・キュー・マネージャーとリモート・キューの名前を指定するか、リモート・キューのローカル定義の名前を指定して) は、この検査は適用されません。
 10. MQOO_INQUIRE (あらゆるオブジェクト・タイプの場合)、または (キューの場合) MQOO_BROWSE、MQOO_INPUT_*、MQOO_OUTPUT、または MQOO_SET の中から、少なくとも 1 つを指定する必要があります。検査は他の指定されたオプションの場合と同じで、提供されている代替ユーザー ID を使用

し、特有の名前のあるオブジェクト権限と、
MQZAO_ALTERNATE_USER_IDENTIFIER 検査の現行アプリケーション権限を調べます。

11. この許可では、任意の *AlternateUserId* を指定できます。
12. MQUS_TRANSMISSION の *Usage* キュー属性がないキューの場合は、MQZAO_OUTPUT 検査も行われます。
13. 検査は他の指定されたオプションの場合と同じで、提供されている代替ユーザー ID を使用し、特有の名前のあるキュー権限と、MQZAO_ALTERNATE_USER_IDENTIFIER 検査の現行アプリケーション権限を調べます。
14. この検査は、次の 2 つの条件が両方とも満たされた場合にのみ行われます。
 - 永続動的キューがクローズされて削除中である。
 - 使用中のオブジェクト・ハンドルを戻した MQOPEN が作成したキューではない。

上記以外の場合は、検査は行われません。

一般的な注:

1. 特殊許可 MQZAO_ALL_MQI には、オブジェクト・タイプに関係する次の許可がすべて含まれます。
 - 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 (注 14 を参照) および MQZAO_DISPLAY は、管理許可として分類されます。したがって、MQZAO_ALL_MQI には含まれません。
3. 「検査しない」は、許可検査が行われないことを意味します。
4. 「適用しない」は、その検査がこの操作には該当しないことを意味します。たとえば、プロセス・オブジェクトに MQPUT 呼び出しを発行できません。

許可指定テーブル

管理許可

これらの許可を持つユーザーは、管理コマンドを発行できます。これは、エスケープ PCF メッセージとしての MQSC コマンドの場合と、PCF コマンドそれ自体の場合があります。こうした方法を使用して、プログラムは、管理コマンドを管理ユーザーに代わって実行させるためにメッセージとしてキュー・マネージャーに送ることができます。

エスケープ PCF の中の MQSC コマンドに関する許可

表3 は、エスケープ PCF に含まれる各 MQSC コマンドに必要な許可を要約したものです。

表3. MQSC コマンドと必要なセキュリティ許可

(2) 必要な許可	キュー・オブジェクト	プロセス・オブジェクト	キュー・マネージャー・オブジェクト	名前リスト
MQSC コマンド				
ALTER オブジェクト	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
CLEAR QLOCAL	MQZAO_CLEAR	適用しない	適用しない	適用しない
DEFINE オブジェクト NOREPLACE (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	適用しない	MQZAO_CREATE (4)
DEFINE オブジェクト REPLACE (3, 5)	MQZAO_CHANGE	MQZAO_CHANGE	適用しない	MQZAO_CHANGE
DELETE オブジェクト	MQZAO_DELETE	MQZAO_DELETE	適用しない	MQZAO_DELETE
DISPLAY オブジェクト	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY

特別な注:

1. コマンドを実行依頼するプログラム (たとえば、**runmqsc**) を実行させるユーザー ID は、キュー・マネージャーに関する MQZAO_CONNECT 権限も持っている必要があります。
2. オブジェクトのタイプに応じて、キュー、プロセス、名前リスト、またはキュー・マネージャー・オブジェクトのいずれかが検査されます。
3. DEFINE コマンドでは、LIKE オブジェクトが指定されている場合は LIKE オブジェクトに関する、また LIKE が省略されて

いる場合は適切な SYSTEM.DEFAULT.xxx オブジェクトに関する、MQZAO_DISPLAY 許可も必要です。

4. MQZAO_CREATE 権限は、特定のオブジェクトまたはオブジェクト・タイプに特有のものではありません。SETMQAUT コマンドで QMGR のオブジェクト・タイプを指定すれば、指定したキュー・マネージャーのすべてのオブジェクトについて作成権限が与えられます。
5. これは、置き換えられるオブジェクトが実際に存在している場合に適用されます。存在していない場合は、DEFINE オブジェクト NOREPLACE の検査になります。

一般的な注:

1. PCF コマンドを実行するためには、キュー・マネージャーに DISPLAY 権限を持っている必要があります。
2. エスケープ PCF を実行する権限は、エスケープ PCF メッセージのテキスト内の MQSC コマンドに応じて決まります。
3. 「適用しない」は、その検査がこの操作には該当しないことを意味します。たとえば、キュー・マネージャー・オブジェクトに CLEAR QLOCAL を発行することはできません。

PCF コマンドについての許可

表4 は、各 PCF コマンドに必要な許可を要約したものです。

表4. PCF コマンドと必要なセキュリティー許可

(2) 必要な許可	キュー・オブジェクト	プロセス・オブジェクト	キュー・マネージャー・オブジェクト	名前リスト
PCF コマンド				
オブジェクトの変更	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
キューのクリア	MQZAO_CLEAR	適用しない	適用しない	適用しない
オブジェクトのコピー (置き換えなし) (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	適用しない	MQZAO_CREATE (4)
オブジェクトのコピー (置き換えあり) (3、6)	MQZAO_CHANGE	MQZAO_CHANGE	適用しない	MQZAO_CHANGE

許可指定テーブル

表 4. PCF コマンドと必要なセキュリティ許可 (続き)

(2) 必要な許可	キュー・オブジェクト	プロセス・オブジェクト	キュー・マネージャー・オブジェクト	名前リスト
オブジェクトの作成 (置き換えなし) (5)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	適用しない	MQZAO_CREATE (4)
オブジェクトの作成 (置き換えあり) (5、6)	MQZAO_CHANGE	MQZAO_CHANGE	適用しない	MQZAO_CHANGE
オブジェクトの削除	MQZAO_DELETE	MQZAO_DELETE	適用しない	MQZAO_DELETE
オブジェクトの照会	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY
オブジェクト名の照会	検査しない	検査しない	検査しない	検査しない
キュー統計のリセット	MQZAO_DISPLAY および MQZAO_CHANGE	適用しない	適用しない	適用しない

特別な注:

1. コマンドを実行依頼するプログラムを実行させるユーザー ID は、そのローカル・キュー・マネージャーに接続する許可と、出力のためにコマンド管理キューをオープンする許可も持っている必要があります。
2. オブジェクトのタイプに応じて、キュー、プロセス、名前リスト、またはキュー・マネージャー・オブジェクトのいずれかが検査されます。
3. Copy コマンドでは、From オブジェクトに関する MQZAO_DISPLAY 許可も必要です。
4. MQZAO_CREATE 権限は、特定のオブジェクトまたはオブジェクト・タイプに特有のものではありません。SETMQAUT コマンドで QMGR のオブジェクト・タイプを指定すれば、指定したキュー・マネージャーのすべてのオブジェクトについて作成権限が与えられます。
5. Create コマンドでは、適切な SYSTEM.DEFAULT.* オブジェクトに関する MQZAO_DISPLAY 権限も必要です。

6. これは、置き換えようとするオブジェクトがすでに存在している場合に適用されます。存在しない場合は、Copy または Create (置き換えなし) と同じ検査になります。

一般的な注:

1. PCF コマンドを実行するためには、キュー・マネージャーに DISPLAY 権限を持っている必要があります。
2. 特殊許可 MQZAO_ALL_ADMIN には、オブジェクト・タイプに関係する次の許可がすべて含まれます。
 - MQZAO_CHANGE
 - MQZAO_CLEAR
 - MQZAO_DELETE
 - MQZAO_DISPLAY

MQZAO_CREATE は、個々のオブジェクトまたはオブジェクト・タイプ固有の許可ではないため、これには含まれません。

3. 「検査しない」は、許可検査が行われないことを意味します。
4. 「適用しない」は、その検査がこの操作には該当しないことを意味します。たとえば、プロセス・オブジェクトに **Clear Queue** コマンドは使用できません。

許可ファイルの理解

注: この節の情報は問題判別用です。通常的环境下では、許可情報の表示と変更には許可コマンドを使用してください。

MQSeries for Compaq OpenVMS は、特定のファイル構造体を使用してセキュリティを実現します。すべての許可ファイル自体が安全であることを確認する以外、これらのファイルには何もする必要はありません。

セキュリティは許可ファイルによって実現されます。この観点からすると、許可には次の 3 つのタイプがあります。

- 単一のオブジェクトに適用される許可 (キューにメッセージを書き込む権限など)。
- オブジェクトのクラスに適用される許可 (キューを作成する権限など)。
- すべてのクラスのオブジェクトに適用される許可 (別のユーザーの代わりに操作を行う権限など)

許可ファイル

許可ファイルのパス

許可ファイルへのパスは、そのタイプによって異なります。たとえば、1 つのオブジェクトに関する許可を指定すると、キュー・マネージャーが該当する許可ファイルを作成します。これらのファイルは 1 つのサブディレクトリーに置かれ、そのパスは、キュー・マネージャー名、許可のタイプ、および適切な場合はオブジェクト名によって定義されます。

すべての許可がオブジェクトのインスタンスに直接適用されるわけではありません。たとえば、オブジェクトを作成する許可は、個別のインスタンスではなくオブジェクトのクラスに適用されます。また、キュー・マネージャー全体に適用される許可もあります。たとえば、代替ユーザー権限は、ユーザーが別のユーザーに関連した権限を仮に受け取ることを意味します。

許可ディレクトリー

デフォルトでは、saturn と呼ばれるキュー・マネージャーのデフォルトの許可ディレクトリーは次のとおりです。

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.QUEUES]

キューの許可ファイル

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.PROCDEF]

プロセス定義の許可ファイル

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.QMANAGER]

キュー・マネージャーの許可ファイル

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH]\$ACCLASS

すべてのクラスに適用する許可

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.NAMELIST]

すべての名前リストに適用する許可

ディレクトリー・ディレクトリーの中で、\$ACCLASS ファイルはクラス全体に関連した許可を保持します。

注: \$ACCLASS (特定のクラスに関する許可を指定する許可ファイル) と \$SACCLASS (すべてのクラスに関する許可を指定する許可ファイル) は異なります。

オブジェクト許可ファイルのパスは、オブジェクト自体のパスが基礎となりますが、オブジェクト・タイプ・ディレクトリーの前に AUTH が挿入されます。**dspmqls** コマンドを使用すると、指定したオブジェクトへのパスを表示できます。

たとえば、SYSTEM.DEFAULT.LOCAL.QUEUE の名前とパスが次のとおりとします。

```
MQS_ROOT:[MQM.QMGRS.SATURN.QUEUES.SYSTEM$DEFAULT$LOCAL$QUEUE]
```

これに対応する許可ファイルの名前とパスは次のようになります。

```
MQS_ROOT:[MQM.QMGRS.SATURN.AUTH.QUEUES.SYSTEM$DEFAULT$LOCAL$QUEUE]
```

注: この場合、キューに関連付けられたファイルの実際の名前は、キュー自体の名前と同じではありません。詳細については、28ページの『MQSeries のファイル名についての理解』を参照してください。

許可ファイルの内容

特定の ID の許可は、許可ファイルの中で 1 組のスタンザによって定義されます。詳細については、125ページの『許可ファイルの理解』を参照してください。許可は、このファイルに関連付けられたオブジェクトに適用されます。たとえば、次のとおりです。

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

このスタンザは、ID GROUPB の権限を定義しています。許可指定は、次に示す割り当てに基づく個別のビット・パターンユニオンです。

許可ファイル

Authorization keyword	Formal name	Hexadecimal Value
connect	MQZAO_CONNECT	0x00000001
browse	MQZAO_BROWSE	0x00000002
get	MQZAO_INPUT	0x00000004
put	MQZAO_OUTPUT	0x00000008
inq	MQZAO_INQUIRE	0x00000010
set	MQZAO_SET	0x00000020
passid	MQZAO_PASS_IDENTITY_CONTEXT	0x00000040
passall	MQZAO_PASS_ALL_CONTEXT	0x00000080
setid	MQZAO_SET_IDENTITY_CONTEXT	0x00000100
setall	MQZAO_SET_ALL_CONTEXT	0x00000200
altusr	MQZAO_ALTERNATE_USER_AUTHORITY	0x00000400
allmqi	MQZAO_ALL_MQI	0x000007FF
crt	MQZAO_CREATE	0x00010000
dlt	MQZAO_DELETE	0x00020000
dsp	MQZAO_DISPLAY	0x00040000
chg	MQZAO_CHANGE	0x00080000
clr	MQZAO_CLEAR	0x00100000
chgaut	MQZAO_AUTHORIZE	0x00800000
alladm	MQZAO_ALL_ADMIN	0x009E0000
none	MQZAO_NONE	0x00000000
all	MQZAO_ALL	0x009E07FF

これらの定義は、ヘッダー・ファイル `cmqzc.h` 内で行われます。次の例では、16 進数 `0x40007` に基づいて、GROUPB に許可が与えられています。これは次に対応します。

MQZAO_CONNECT	0x00000001
MQZAO_BROWSE	0x00000002
MQZAO_INPUT	0x00000004
MQZAO_DISPLAY	0x00040000

Authority is:	0x00040007

これらのアクセス権限は、GROUPB のユーザーであれば、次の MQI 呼び出しを発行できることを意味します。

```
MQCONN
MQGET (ブラウズ付き)
```

また、この許可ファイルに関連付けられたオブジェクトに関する DISPLAY 権限を持っていることを意味します。

クラス許可ファイル

クラス許可ファイルは、クラス全体に関連した許可を保持します。これらのファイルは、「\$CLASS」と呼ばれ、特定オブジェクトのファイルと同じディレクトリーに存在します。\$CLASS ファイルの項目 MQZAO_CRT は、このクラスにオブジェクトを作成する許可を与えます。これは唯一のクラス権限です。

全クラス許可ファイル

全クラス許可ファイルは、キュー・マネージャー全体に適用される許可を保持します。このファイルは、「\$ACCLASS」と呼ばれ、キュー・マネージャーの auth サブディレクトリーに存在します。

次の許可は、キュー・マネージャー全体に適用され、全クラス許可ファイルに保持されます。

項目 与える許可**MQZAO_ALTERNATE_USER_AUTHORITY**

MQSeries オブジェクトと対話するときに別のユーザーと一致すると想定します。

MQZAO_SET_ALL_CONTEXT

MQPUT を発行するときにメッセージのコンテキストを設定します。

MQZAO_SET_IDENTITY_CONTEXT

MQPUT を発行するときにメッセージの一致コンテキストを設定します。

許可ファイルの管理

許可ファイルを管理するときに考慮する必要のある事柄を以下に示します。

1. 許可ファイルが安全に保護され、信頼性が確認されていない一般ユーザーによる書き込みアクセスができないようにしておく必要があります。130ページの『許可ファイルについての許可』を参照してください。
2. ファイル許可の複製を作成できるように、次の2つの作業のうち少なくとも1つを実行します。
 - 意味のある更新を行うたびに、AUTH サブディレクトリーのバックアップを取る。
 - 使用したコマンドが入っている DCL コマンド・ファイルを保存する。
3. 許可ファイルは、コピーして編集することができます。ただし、通常は、許可ファイルを手動で作成したり、修理したりする必要はありません。万一緊

許可ファイル

急の事態が発生した場合は、ここで示した説明を使用して消失または損傷した許可ファイルを回復することができます。

許可ファイルについての許可

許可ファイルをすべてのプリンシパルが読み取れるようにしておく必要があります。ただし、システム管理者と MQM ID を持つユーザーだけにはこれらのファイルの更新が許可されています。

OAM によって作成された許可ファイルに関する許可は次のとおりです。

S:RWD, O:RWD, G:RWD, W:R (ID=MQM, ACCESS=R+W+E+D+C)

セキュリティを損なうことがないかどうかを入念に検討せずにこれらの許可を変更しないでください。

MQSeries for Compaq OpenVMS で提供されているコマンドを使用して許可を変更するためには、プロセスには MQM 権限 ID が必要です。

第8章 MQSeries 送達不能キュー・ハンドラー

未配布メッセージ・キュー とも呼ばれる送達不能キュー (DLQ) とは、宛先キューに送達できないメッセージ用の保留キューです。ネットワーク内のすべてのキュー・マネージャーが、関連した DLQ を持つ必要があります。

キュー・マネージャー、メッセージ・チャンネル・エージェント (MCA)、およびアプリケーションは、メッセージを DLQ に書き込むことができます。DLQ 上のすべてのメッセージの先頭に、送達不能ヘッダー 構造体 MQDLH を付ける必要があります。キュー・マネージャーまたはメッセージ・チャンネル・エージェントが DLQ に書き込むメッセージには、常に MQDLH があります。メッセージを DLQ に書き込むアプリケーションが MQDLH を提供することを推奨します。MQDLH 構造体の *Reason* フィールドは、メッセージが DLQ 上にある理由を識別する理由コードを含みます。

DLQ 上のメッセージを処理するために定期的に行われるルーチンが、すべての MQSeries 環境になければなりません。MQSeries は、送達不能キュー・ハンドラー (DLQ ハンドラー) と呼ばれるデフォルト・ルーチンを提供しています。DLQ ハンドラーは、**runmqdlq** コマンドを使用して呼び出します。

DLQ 上のメッセージを処理する命令は、ユーザー作成ルール・テーブル を介して DLQ ハンドラーに提供されます。つまり、DLQ ハンドラーは DLQ 上のメッセージとルール・テーブルの項目の突き合わせを行います。DLQ メッセージがルール・テーブルの項目と一致すると、DLQ ハンドラーはその項目に関連付けられたアクションを実行します。

この章には次の節があります。

- 『DLQ ハンドラーの呼び出し』
- 133ページの『DLQ ハンドラーのルール・テーブル』
- 141ページの『ルール・テーブルの処理方法』
- 144ページの『DLQ ハンドラーのルール・テーブルの例』

DLQ ハンドラーの呼び出し

DLQ ハンドラーは、**runmqdlq** コマンドを使用して呼び出します。処理したい DLQ の名前および使用したいキュー・マネージャーの名前を指定するには、次の 2 つの方法があります。

DLQ ハンドラー

- **runmqdlq** へのパラメーターとしてコマンド・プロンプトから指定する。たとえば、次のとおりです。

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER < qrule.ru1
```

- ルール・テーブルの中で指定する。たとえば、次のとおりです。

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

上記の例は、キュー・マネージャー **ABC1.QUEUE.MANAGER** が所有する **ABC1.DEAD.LETTER.QUEUE** という DLQ に適用されます。

DLQ またはキュー・マネージャーを上記の例のように指定しなかった場合は、インストール先のデフォルト・キュー・マネージャーと共にそのキュー・マネージャーが所有する DLQ が使用されます。

runmqdlq コマンドは、**SYS\$INPUT** から入力を取ります。ユーザーはルール・テーブルから **SYS\$INPUT** をリダイレクトして、ルール・テーブルを **runmqdlq** に関連付けます。

重要: **SYS\$INPUT** をルール・ファイルにリダイレクトしないで DLQ ハンドラーを実行すると、DLQ ハンドラーがループしてしまいます。

DLQ ハンドラーを実行するためには、DLQ 自体、および DLQ 上のメッセージの転送先となるあらゆるメッセージ・キューの両方へのアクセスが許可されていることが必要です。さらに、DLQ ハンドラーがメッセージ・コンテキスト中のユーザー ID の権限を使用してキューにメッセージを書き込むことが可能になっている場合には、DLQ ハンドラーを実行するユーザーは他のユーザーの ID を借用する許可を持っている必要があります。

runmqdlq コマンドの詳細については、365ページの『**runmqdlq** (送達不能キュー・ハンドラーの実行)』を参照してください。

サンプル DLQ ハンドラー **amqsdq**

runmqdlq コマンドを使用して呼び出される DLQ ハンドラーの他に、MQSeries はサンプル DLQ ハンドラー **amqsdq** のソース・コードを提供します。サンプル DLQ ハンドラーには、**runmqdlq** を介して呼び出されるものと同様な機能があります。ソースはテンプレートとしてのみ提供されているので、特定のローカル要件に適合した DLQ ハンドラーを与えるには、これをカスタマイズする必要があります。たとえば、送達不能ヘッダーのないメッセー

ジを処理できる DLQ ハンドラーが必要となる場合があります。(デフォルト DLQ ハンドラーとサンプルの `amqsdlq` は、両方共、DLQ 上のメッセージのうち送達不能ヘッダー `MQDLH` で始まるもののみを処理するようになっています。 `MQDLH` で始まらないメッセージはエラーとして識別され、DLQ 上にいつまでも残ることになります。)

`amqsdlq` のソースは、次のディレクトリーに提供されています。

`[.DLQ]`, under `MQS_EXAMPLES`

また、コンパイル済みバージョンは次のディレクトリー内にあります。

`[.BIN]`, under `MQS_EXAMPLES`

DLQ ハンドラーのルール・テーブル

DLQ ハンドラーのルール・テーブルは、DLQ に到着したメッセージを DLQ ハンドラーがどのように処理するかを定義するものです。ルール・テーブルには、次の 2 種類の項目があります。

- テーブルの最初の項目は **制御データ** で、この項目はオプションです。
- テーブルの中の他のすべての項目は、DLQ ハンドラーが従うルールです。各ルールは、パターン (1 組のメッセージ特性) とアクション からなります。メッセージはパターンと突き合わされ、指定のパターンと DLQ 上の一致したメッセージにはアクションが実行されます。ルール・テーブルには、ルールが少なくとも 1 つ必要です。

ルール・テーブルの各項目は、1 つまたは複数のキーワードからなります。

制御データ

この節では、DLQ ハンドラーのルール・テーブルの制御データ項目に入れることができるキーワードについて説明します。次の点に注意してください。

- キーワードのデフォルト値 (ある場合) には、下線が引いてあります。
- 縦線 (|) によって、代替の値を区切っています。値のうちの 1 つのみを指定できます。
- キーワードはすべてオプションです。

INPUTQ (*QueueName*['_'])

処理したい DLQ の名前を指定します。

1. INPUTQ 値を **runmqdlq** コマンドのパラメーターとして指定した場合は、ルール・テーブル内の INPUTQ 値がそのパラメーターによって指定変更されます。
2. **runmqdlq** コマンドのパラメーターとして INPUTQ 値を指定しないで、ルール・テーブルの中に値を指定すると、ルール・テーブルの INPUTQ 値が使用されます。
3. DLQ を指定しなかった場合、または ルール・テーブルの中で INPUTQ(' ') を指定した場合は、**runmqdlq** コマンドのパラメーターとして名前を指定したキュー・マネージャーが所有する DLQ の名前が使用されます。
4. INPUTQ 値を **runmqdlq** コマンドのパラメーターとしても、ルール・テーブルの中の値としても指定しなかった場合は、ルール・テーブル内の INPUTQM キーワードで指定したキュー・マネージャーが所有する DLQ が使用されます。

INPUTQM (*QueueManagerName*['_'])

INPUTQ キーワードで指定した DLQ を所有するキュー・マネージャーの名前を指定します。

1. INPUTQM 値を **runmqdlq** コマンドのパラメーターとして指定した場合は、ルール・テーブル内の INPUTQM 値がそのパラメーターによって指定変更されます。
2. INPUTQM 値を **runmqdlq** コマンドのパラメーターとして指定しなかった場合は、ルール・テーブル内の INPUTQM 値が使用されます。
3. キュー・マネージャーを指定しない場合、または INPUTQM(' ') をルール・テーブルの中に指定した場合は、インストール・システムのデフォルト・キュー・マネージャーが使用されます。

RETRYINT (*Interval***|60**)

最初の試みで処理できなかった DLQ 上のメッセージについて、試みの反復が要求されている場合に DLQ ハンドラーが再処理を試みる間隔 (秒数) です。デフォルトでは、再試行間隔は 60 秒です。

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

DLQ ハンドラーが処理できるメッセージがこれ以上ないことを DLQ ハンドラーが検出したとき、DLQ にメッセージが新たに到着するのを DLQ ハンドラーが待機するかどうかを指示します。

YES DLQ ハンドラーは無限に待機します。

- NO** DLQ ハンドラーは、DLQ が空になるか、あるいは処理できるメッセージがなくなったことを検出すると終了します。
- nnn* キューが空であるか、または DLQ ハンドラーが処理できるメッセージがキューにないことを DLQ ハンドラーが検出したあと、メッセージが新たに到着するのを DLQ ハンドラーが *nnn* 秒間待機するようにします。

使用頻度の高い DLQ については WAIT (YES)、使用頻度の低い DLQ については WAIT (NO) または WAIT (*nnn*) を指定することをお勧めします。DLQ ハンドラーが終了するようにした場合は、トリガー操作によって DLQ ハンドラーを再び呼び出すことをお勧めします。

ルール・テーブルに制御データを組み込む代わりに、`runmqdlq` コマンドの入力パラメーターとして DLQ とそのキュー・マネージャーの名前を指定することができます。ルール・テーブルに値を指定した場合や `runmqdlq` コマンドの入力で値を指定した場合は、`runmqdlq` コマンドで指定した値が優先されます。

注: 制御データ項目をルール・テーブルに組み込む場合、その項目は、ルール・テーブルの最初の項目でなければなりません。

ルール (パターンと処理)

図8 は、DLQ ハンドラーのルール・テーブルのルールの一例です。

```
PERSIST(MQPER PERSISTENT) REASON(MQRC_PUT_INHIBITED) +
ACTION(RETRY) RETRY(3)
```

図8. DLQ ハンドラーのルール・テーブルのルールの一例。このルールは、MQPUT および MQPUTI の使用が禁止されたため DLQ に書き込まれた持続メッセージをその宛先キューに送達する試みを 3 回行うように、DLQ ハンドラーに指示しています。

ルールに使用できるすべてのキーワードについて、この節で後程説明します。次の点に注意してください。

- キーワードのデフォルト値 (ある場合) には、下線が引いてあります。ほとんどのキーワードの場合、デフォルト値は * (アスタリスク) であり、これはどの値でも構いません。
- 縦線 (|) によって、代替の値を区切っています。値のうちの 1 つのみを指定できます。
- ACTION 以外のキーワードはすべてオプションです。

DLQ ハンドラー

この節では、まず、パターン・マッチング・キーワード (DLQ 上のメッセージを突き合わせるキーワード) について説明し、次に、処理キーワード (一致するメッセージを DLQ ハンドラーが処理する方法を決定するキーワード) について説明します。

パターン・マッチング・キーワード

DLQ 上のメッセージを突き合わせる値を指定するために使用するパターン・マッチング・キーワードについて、これから説明します。パターン・マッチング・キーワードのすべてがオプションです。

APPLIDAT (*ApplIdentityData**)

DLQ 上のメッセージのメッセージ記述子 MQMD に指定した *ApplIdentityData* 値。

APPLNAME (*PutApplName**)

DLQ 上のメッセージのメッセージ記述子 MQMD の *PutApplName* フィールドに指定した、MQPUT 呼び出しまたは MQPUT1 呼び出しを出したアプリケーションの名前。

APPLTYPE (*PutApplType**)

DLQ 上のメッセージのメッセージ記述子 MQMD に指定した *PutApplType* 値。

DESTQ (*QueueName**)

メッセージの宛先であるメッセージ・キューの名前。

DESTQM (*QueueManagerName**)

メッセージの宛先であるメッセージ・キューのキュー・マネージャーの名前。

FEEDBACK (*Feedback**)

MsgType 値が MQFB_REPORT の場合、*Feedback* はレポートの性質を記述します。

記号名を使用できます。たとえば、記号名 MQFB_COA を使用して、DLQ 上のメッセージのうち、宛先キューへの到着の確認が必要なものを識別することができます。

FORMAT (*Format**)

メッセージ・データの形式を記述するためにメッセージの送信側が使用する名前。

MSGTYPE (*MsgType**)

DLQ 上のメッセージのメッセージ・タイプ。

記号名を使用できます。たとえば、記号名 MQMT_REQUEST を使用して、DLQ 上のメッセージのうち応答が必要なものを識別することができます。

PERSIST (Persistence|*)

メッセージの持続性の値。(メッセージの持続性によって、キュー・マネージャーの再始動後もメッセージが存続するかどうかが決まります。)

記号名を使用できます。たとえば、記号名 MQPER_PERSISTENT を使用して、DLQ 上のメッセージのうち持続するものを識別することができます。

REASON (ReasonCode|*)

メッセージが DLQ に書き込まれた理由を記述する理由コード。

記号名を使用できます。たとえば、記号名 MQRC_Q_FULL を使用して、宛先キューが満杯だったために DLQ に置かれたメッセージを識別することができます。

REPLYQ (QueueName|*)

DLQ 上のメッセージのメッセージ記述子 MQMD に指定した応答先キューの名前。

REPLYQM (QueueManagerName|*)

DLQ 上のメッセージのメッセージ記述子 MQMD に指定した応答先キューのキュー・マネージャーの名前。

USERID (UserIdentifier|*)

メッセージ記述子 MQMD に指定した DLQ 上のメッセージの発信元のユーザーのユーザー ID。

アクション・キーワード

一致するメッセージの処理方法の記述に使用するアクション・キーワードについて、次に説明します。

ACTION (DISCARD|IGNORE|RETRY|FWD)

このルールに定義したパターンに一致する DLQ 上のメッセージに関して実行されるアクション。

DISCARD

メッセージは、DLQ から削除されます。

IGNORE

メッセージは DLQ 上に残されます。

RETRY

DLQ ハンドラーは、メッセージを宛先キューに入れることをもう一度試みます。

FWD FWDQ キーワードに指定したキューへのメッセージの転送を、DLQ ハンドラーに実行させます。

ACTION キーワードは必須指定です。アクションを実行するための試行の回数は、RETRY キーワードで制御します。試行相互間の間隔は、制御データの RETRYINT キーワードで制御します。

FWDQ (*QueueName*|&DESTQ|&REPLYQ)

ACTION (FWD) を要求したときのメッセージの転送先のメッセージ・キューの名前。

QueueName

メッセージ・キューの名前です。FWDQ(' ')は無効です。

&DESTQ

MQDLH 構造体の *DestQName* フィールドからキュー名が入手されるようにします。

&REPLYQ

メッセージ記述子 MQMD の *ReplyToQ* フィールドからキュー名が入手されるようにします。

FWDQ(&REPLYQ) を指定したルールが、ブランクの *ReplyToQ* フィールドを持つメッセージと一致したときにエラー・メッセージを避けるために、メッセージ・パターンで REPLYQ(?) を指定することができます。

FWDQM (*QueueManagerName*|&DESTQM|&REPLYQM|' _')

メッセージの転送先のキューのキュー・マネージャーを識別します。

QueueManagerName

ACTION (FWD) を要求したときの、メッセージの転送先のキューのキュー・マネージャーの名前。

&DESTQM

MQDLH 構造体の *DestQMgrName* フィールドからキュー・マネージャー名が入手されるようにします。

&REPLYQM

メッセージ記述子 MQMD の *ReplyToQMgr* フィールドからキュー・マネージャー名が入手されるようにします。

- ・ FWDQM(' ') がデフォルト値です。この値は、ローカル・キュー・マネージャーを識別します。

HEADER (YES|NO)

ACTION (FWD) が要求されたメッセージに MQDLH を残すかどうかを指定します。デフォルトでは、MQDLH はメッセージに残ります。HEADER キーワードは、FWD 以外の処理に関しては無効です。

PUTAUT (DEF|CTX)

DLQ ハンドラーがメッセージを書き込む際の権限を定義します。

DEF メッセージは DLQ ハンドラー自体の権限で書き込まれます。

CTX メッセージはメッセージ・コンテキストの中のユーザー ID の権限で書き込まれます。PUTAUT (CTX) を指定する場合、他のユーザーの ID を使用することが許可されている必要があります。

RETRY (RetryCount|1)

1 から 999,999,999 の範囲の数で、(制御データの RETRYINT キーワードに指定されている間隔で) アクションを試みる回数です。

注: DLQ ハンドラーが特定のルールを実行するために行う試みのカウントは、DLQ ハンドラーの現行インスタンスに特有のものであり、再始動後には持ちこされません。DLQ ハンドラーが再始動すると、あるルールに適用された試行のカウントはゼロにリセットされます。

ルール・テーブルのルール

ルール・テーブルは、その構文、構造、内容に関して次のルールに従う必要があります。

- ・ ルール・テーブルには、ルールが少なくとも 1 つはなければなりません。
- ・ キーワードは、任意の順序で指定して構いません。
- ・ キーワードは、どのルールにも 1 回のみ指定できます。
- ・ キーワードには大文字小文字の区別はありません。
- ・ キーワードとそのパラメーター値は、1 つ以上の空白またはコンマによって他のキーワードから分離する必要があります。
- ・ ルールの始めまたは終わり、およびキーワード、句読点、値の間には、空白をいくつ入れても構いません。
- ・ 各ルールは、新しい行で始まらなければなりません。
- ・ 移植性のために、行の有効長は 72 文字を超えないようにする必要があります。

DLQ ハンドラー

- 次行の最初の非空白文字にルールが継続するよう指示するには、行の最後の非空白文字として正符号 (+) を使用します。次行の先頭にルールが継続するよう指示するには、行の最後の非空白文字として負符号 (-) を使用します。連結文字がキーワードおよびパラメーターの内部に現れても構いません。
- 注釈行は、アスタリスク (*) で始まり、ルール・テーブルのどこにでも指定することができます。
- 空白行は無視されます。
- DLQ ハンドラーのルール・テーブルの各項目は、1 つまたは複数のキーワードと、それらに関連付けられたパラメーターからなります。そのパラメーターは、次の構文ルールに従う必要があります。
 - 各パラメーター値には、有効な文字を少なくとも 1 個は含める必要があります。引用符で囲んだ値の区切り用の引用符は無効と見なされます。たとえば、次のパラメーターは有効です。

FORMAT('ABC') 有効な文字は 3 個

FORMAT(ABC) 有効な文字は 3 個

FORMAT('A') 有効な文字は 1 個

FORMAT(A) 有効な文字は 1 個

FORMAT(' ') 有効な文字は 1 個

次のパラメーターは、有効な文字が 1 個も含まれていないため無効です。

FORMAT('')

FORMAT()

FORMAT()

FORMAT

- ワイルドカード文字がサポートされます。後書き空白以外の 1 文字の代わりに疑問符 (?) を使用することができます。0 個以上の隣接した文字の代わりにアスタリスク (*) を使用することができます。アスタリスク (*) および疑問符 (?) は、パラメーター値の中では常に ワイルドカード文字と解釈されます。
- 次のキーワードのパラメーターの中には、ワイルドカード文字を含めることはできません。ACTION、HEADER、RETRY、FWDQ、FWDQM、および PUTAUT。
- パラメーター値の中の後書き空白、および DLQ 上のメッセージ内のそれに対応するフィールドの中の後書き空白は、ワイルドカード突き

合わせの実行時には無効です。しかし、引用符で囲んだストリングの中の先行空白と組み込み空白は、ワイルドカード突き合わせでも有効です。

- 数値パラメーターには、疑問符 (?) のワイルドカード文字を含むことはできません。1 個の数値パラメーター全体の代わりにアスタリスク (*) を使用できますが、数値パラメーターの一部として含めることはできません。たとえば、次の数値パラメーターは有効です。

MSGTYPE(2) 応答メッセージのみが適格
MSGTYPE(*) どのメッセージ・タイプも適格
MSGTYPE('*') どのメッセージ・タイプも適格

しかし、MSGTYPE('2*') は無効です。数値パラメーターの一部としてアスタリスク (*) が含まれているためです。

- 数値パラメーターの有効範囲は、0 から 999,999,999 です。パラメーター値がこの範囲内であるなら、キーワードが関連するフィールドで現在無効であっても、パラメーター値は受け入れられます。数値パラメーターには、記号名を使用できます。
- キーワードが関連する MQDLH または MQMD 内のフィールドよりもストリング値が短い場合、そのストリング値は、フィールドの長さになるまで空白が埋め込まれます。ストリング値 (アスタリスクを除外して) がフィールドより長い場合は、エラーの診断が下されます。たとえば、次のストリング値は、8 文字のフィールドに関してすべて有効です。

'ABCDEFGH' 8 文字
'A*C*E*G*I' アスタリスクを除外して 5 文字
'*A*C*E*G*I*K*M*O*' アスタリスクを除外して 8 文字

- 空白、小文字、もしくはピリオド (.), 斜線 (/), 下線 (_), およびパーセント記号 (%) を除く特殊文字が含まれているストリングは、単一引用符で囲む必要があります。引用符で囲んでいない小文字は、大文字に変換されます。ストリングに引用符を含む場合は、その引用符の始めと終わりの両方を示すために、2 個の単一引用符を使用する必要があります。ストリングの長さが計算されるとき、二重引用符の各オカレンスは、1 個の文字としてカウントされます。

ルール・テーブルの処理方法

DLQ ハンドラーはルール・テーブルを検索し、DLQ 上のメッセージとそのパターンが一致するルールを求めます。検索は、ルール・テーブルの最初のルールから始まって、表中を順番に進んでゆきます。一致するパターンを持つルールが見つかったら、そのルールの処理が試行されます。DLQ ハンドラーは、

DLQ ハンドラー

そのルール適用を試行するたびに、ルールの再試行カウントを 1 つずつ増分します。最初の試行が失敗すると、なされた試行のカウントが、RETRY キーワードに指定した数と一致するまで、試行は繰り返されます。試行がすべて失敗すると、DLQ ハンドラーは、ルール・テーブルの中の次に一致するルールを検索します。

このプロセスは、アクションが正常に実行されるまで、一致するルールについて順番に繰り返されます。一致するルールがそれぞれ RETRY キーワードで指定されている回数試みられ、その試みがすべて失敗した場合は、ACTION (IGNORE) であると見なされます。一致するルールがまったく見つからないときにも、ACTION (IGNORE) が想定されます。

注:

1. 一致するルールのパターンは、接頭部が MQDLH の DLQ 上のメッセージについてのみ検索されます。接頭部が MQDLH 以外のメッセージは、エラーとして定期的に報告され、DLQ 上にいつまでも残ります。
2. すべてのパターン・キーワードを、デフォルトにして、ルールがアクションのみで構成されるようにすることができます。ただし、そのキューにおいて、テーブル内のその他のルールに従って処理されておらず、MQDLH があるキューの全メッセージにアクションのみのルールが適用されることに注意してください。
3. ルール・テーブルは、DLQ ハンドラーの開始時に検査され、その時点でエラーのフラグが付けられます。ルール・テーブルにはいつでも変更を加えることができますが、DLQ ハンドラーが再始動されないと、その変更は有効になりません。
4. DLQ ハンドラーは、メッセージ、MQDLH、メッセージ記述子のいずれの内容も更新しません。DLQ ハンドラーは、常にメッセージ・オプション MQPMO_PASS_ALL_CONTEXT でメッセージを他のキューに書き込みます。
5. DLQ ハンドラーは、オプション MQOO_INPUT_AS_Q_DEF を持つ DLQ をオープンします。
6. 同じルール・テーブルを使用して同じキューに DLQ ハンドラーの複数インスタンスが並行して実行される場合もあります。しかし、DLQ と DLQ ハンドラーの間には、1 対 1 の関係があるのが普通です。

すべての DLQ メッセージを確実に処理する

DLQ ハンドラーは、すでに参照されたが除去されていない DLQ 上のすべてのメッセージを記録しておきます。DLQ からメッセージの小さいサブセットを抽出するためのフィルターとして DLQ ハンドラーを使用する場合にも、

DLQ ハンドラーは、DLQ 上にある未処理のメッセージの記録を保持し続けます。また、DLQ が先入れ先出し (FIFO) として定義されても、DLQ に到着する新規メッセージが参照されることを DLQ ハンドラーは保証できません。したがって、キューが空でない限り、すべてのメッセージを検査するために DLQ が定期的に再走査されます。以上の点から、必ず DLQ にはできるだけ少数のメッセージを入れるようにしてください。廃棄したり他のキューに転送したりできない (その理由が何であっても) メッセージをキュー上に累積させると、DLQ ハンドラーの作業負荷が増大し、DLQ 自体が満杯になる恐れがあります。

特定の対策を取って、DLQ ハンドラーに DLQ を空にさせることができます。たとえば、ACTION (IGNORE) は、DLQ 上のメッセージを単に放置するだけなので、使用しないようにしてください。(テーブルの中の他のルールによって明示的に処理されないメッセージには、ACTION (IGNORE) が適用されることに注意してください。) その代わりに、無視するメッセージに関して、別のキューにそのメッセージを移動するアクションを実行してください。たとえば、次のとおりです。

ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)

また、テーブルの最後のルールは、テーブルの中のそれまでのルールから漏れたメッセージをまとめて扱えるものにする必要があります。たとえば、テーブルの中の最後のルールは、次のようなものにすることができます。

ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
--

テーブルの中の最後のルールが適用されることになったメッセージは、このアクションによってキュー REALLY.DEAD.QUEUE に転送されます。このキューで、そのメッセージを手動によって処理できます。このようなルールがない場合は、メッセージが DLQ に無期限に残ることになります。

DLQ ハンドラーのルール・テーブルの例

1 つの制御データ項目といくつかのルールが含まれているルール・テーブルの例を示します。

```
*****
*           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.

```
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)
```

DLQ ハンドラー

第9章 観測イベント

キュー・マネージャーの操作をモニターするために、MQSeries の観測イベントを使用することができます。この章では、観測イベントについて簡単に紹介します。詳細については、*MQSeries プログラム式システム管理* の観測イベントについて節を参照してください。

観測イベントとは

事前定義された 1 組の条件をキュー・マネージャーが検出するたびに、観測イベントによって、イベント・メッセージと呼ばれる特殊なメッセージが生成されます。たとえば、次のような条件によってキュー満杯 イベントが発生することが考えられます。

- キュー満杯イベントは、指定されたキューについて有効です。
- アプリケーションは、そのキューにメッセージを書き込むために MQPUT 呼び出しを発行しますが、キューが満杯であるため、その呼び出しは失敗します。

観測イベントを引き起こす可能性のあるその他の条件には、次のようなものがあります。

- キューのメッセージ数に関するしきい値の限度に達した。
- 指定した時間枠内にキューの運用が実行されない。
- チャンネル・インスタンスが開始または停止した。
- MQSeries for Compaq OpenVMS システムでは、アプリケーションが、許可されていないユーザー ID を指定してキューをオープンしようとした。

チャンネル・イベントの例外として、すべての観測イベントは、生成される前に有効な状態になっている必要があります。

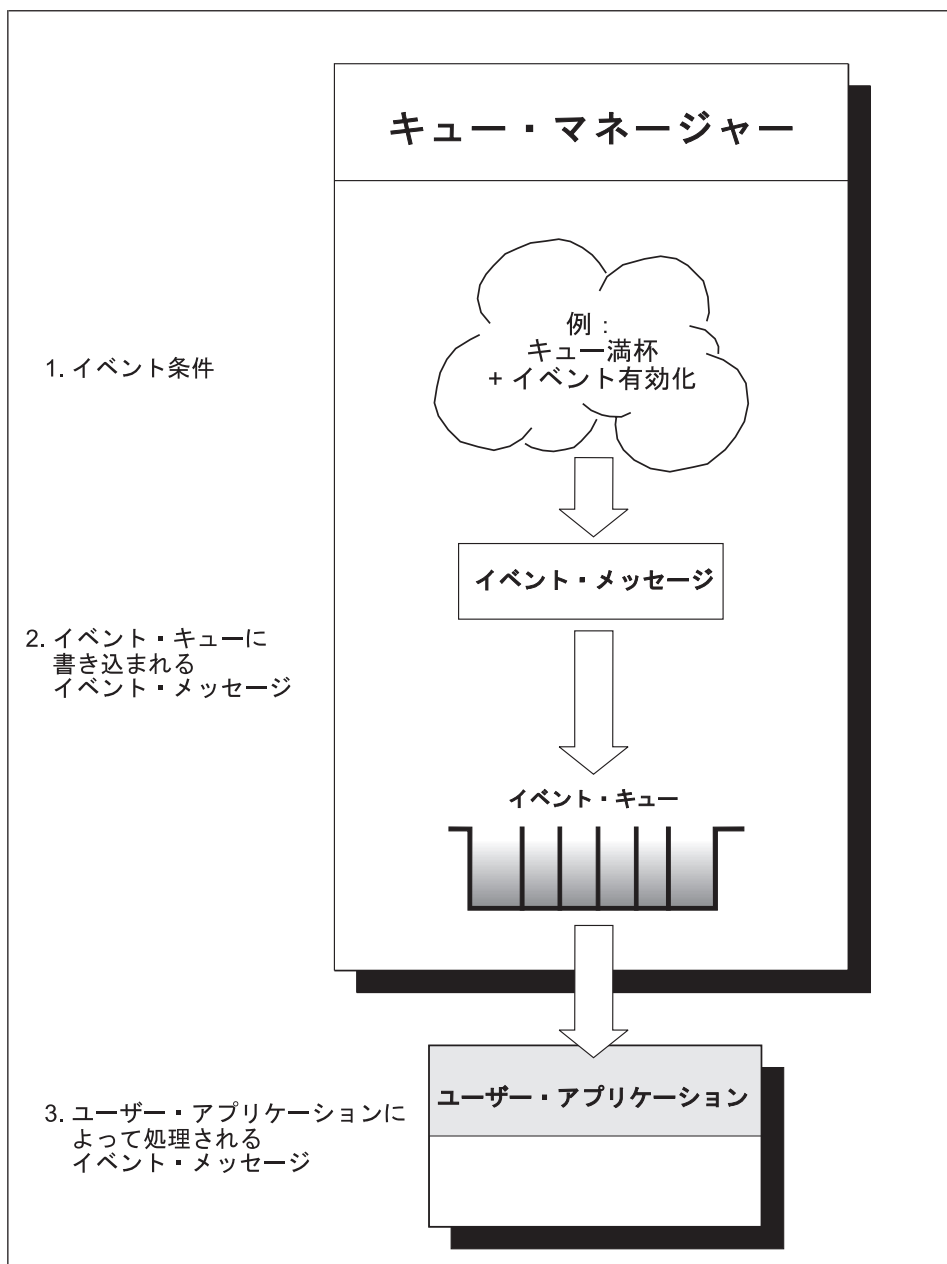


図9 観測イベントの理解。観測イベントの条件が満たされたことを検出すると、キュー・マネージャーは、イベント・メッセージを該当のイベント・キューに書き込みます。

イベント・メッセージ (イベントを引き起こした条件についての情報が入っている) は、イベント・キューに書き込まれます。アプリケーションは、分析のためにこのキューからイベント・メッセージを取り出すことができます。

イベントを使用するのはなぜか

イベント・キューをリモート・キューとして指定する場合、すべてのイベント・キューを（観測イベントをサポートする複数のノード用として）1つのキュー・マネージャーに置くことができます。すると、生成されたイベントを使用して、単一のノードからキュー・マネージャーのネットワークをモニターできます。図10にこのことが示されています。

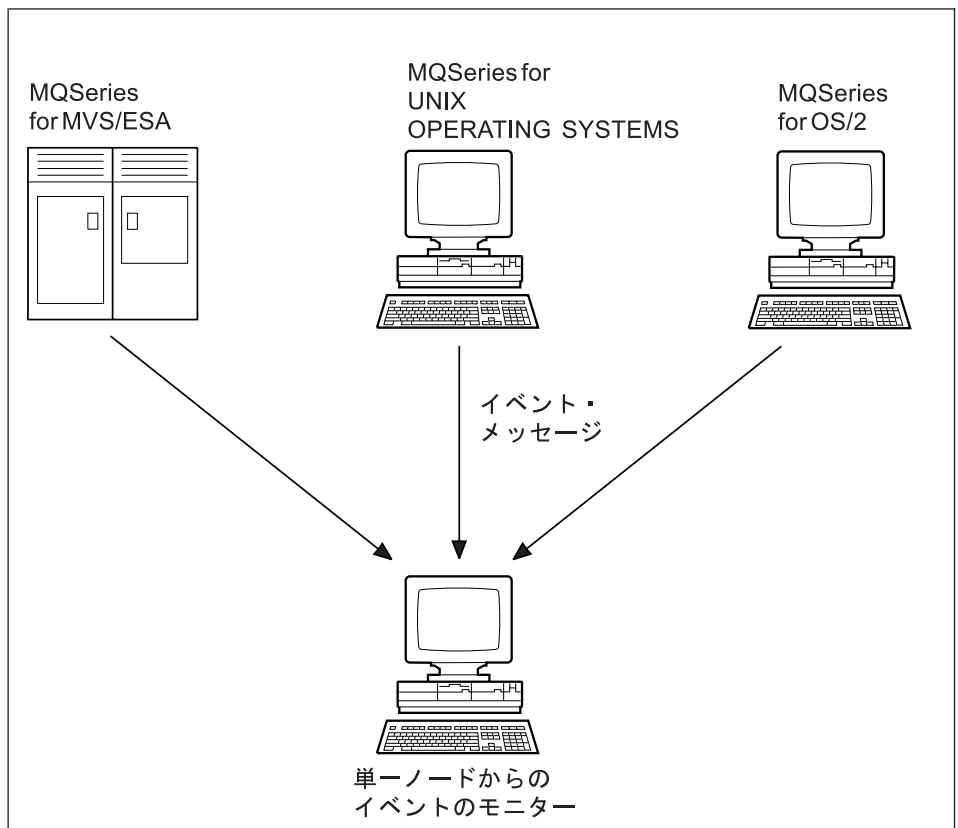


図10. 異なるプラットフォームにあるキュー・マネージャーを、単一のノードでモニターする

イベントのタイプ

MQSeries イベントは、次のように分類することができます。

キュー・マネージャー・イベント

このタイプのイベントは、キュー・マネージャー内のリソースの定義に関係したものです。たとえば、アプリケーションが、存在しないキューにメッセージを書き込もうとする場合です。

イベントの使用

パフォーマンス・イベント

このタイプのイベントは、リソースのしきい値条件に達したことを通知するものです。たとえば、キューの制限項目数に達した場合や、あるいは取得後に、事前定義された時間制限内にキューが運用されなかった場合です。

チャンネル・イベント

このタイプのイベントは、チャンネル操作時に検出された条件の結果として、チャンネルによって報告されます。たとえば、チャンネルが停止する場合です。

トリガー・イベント

本書および他の MQSeries の資料でトリガー操作について説明するときに、トリガー・イベントに言及することがあります。このイベントは、トリガー・イベントの条件が満たされたことをキュー・マネージャーが検出したときに起こります。たとえば、メッセージが到着するたびにトリガー・イベントを生成するようキューを構成できます。(トリガー・イベントと観測イベントの条件はまったく異なります。)

トリガー・イベントが起きると、トリガー・メッセージが開始キューに入れられ、オプションで、アプリケーション・プログラムが開始されます。

イベント・キューによるイベント通知

イベントが起こると、キュー・マネージャーはイベント・メッセージを該当のイベント・キュー(定義されている場合)に書き込みます。イベント・メッセージには、次のような適切な MQI アプリケーション・プログラムを作成すれば取り出すことができるイベントの情報が入っています。

- キューからメッセージを取得するアプリケーション。
- イベント・データを取り出すメッセージを処理するアプリケーション。イベント・メッセージの形式については、MQSeries プログラム式システム管理を参照してください。

イベントの各カテゴリーには、それぞれのイベント・キューがあります。そのカテゴリーのイベントが起こると、すべて同じキューにイベント・メッセージが入れられることとなります。

イベント・キュー

入れられるメッセージの発生元

SYSTEM.ADMIN.QMGR.EVENT

キュー・マネージャー・イベント

SYSTEM.ADMIN.PERFM.EVENT

パフォーマンス・イベント

SYSTEM.ADMIN.CHANNEL.EVENT

チャンネル・イベント

イベント・キューは、ローカル・キューとして、またはリモート・キューとして定義できます。イベント・キューすべてをリモート・キューとして同じキュー・マネージャーに定義すると、モニター活動を単一地点で統括的に行うことができます。

トリガー・イベント・キューを使用する

イベントが生成されると、イベント・メッセージがイベント・キューに書き込まれ、それによって (ユーザー作成) モニター・アプリケーションが開始されるように、トリガー付きのイベント・キューをセットアップできます。このアプリケーションはイベント・メッセージを処理し、適切なアクションを実行できます。たとえば、オペレーターに通知する必要があるイベントもあれば、一部の管理タスクを自動的に実行するアプリケーションを開始するイベントもあります。

イベントの有効化および無効化

イベントのタイプに応じて、キュー・マネージャーまたはキュー属性、あるいはその両方に該当の値を指定することにより、イベントを有効化または無効化します。これには、次のいずれかを使用します。

- MQSC コマンド。詳細については、*MQSeries* コマンド・リファレンス を参照してください。
- UNIX システム、OpenVMS システム、および OS/2 上の PCF コマンド。詳細については、*MQSeries* プログラム式システム管理 を参照してください。
- MQAI コマンド。詳細については、*MQSeries* 管理インターフェースのプログラミングの手引きおよび解説書 を参照してください。

イベントの有効化は、イベントのカテゴリによって異なります。

- キュー・マネージャー・イベントは、キュー・マネージャー上の属性を設定することにより、有効化されます。

イベントの使用

- パフォーマンス・イベントは全体として、キュー・マネージャー上で有効化する必要があります。そうしないと、パフォーマンス・イベントは起こりません。その後、該当のキュー属性を設定して、特定のパフォーマンス・イベントを有効化します。さらに、イベントを引き起こす条件（たとえば、キュー項目数の上限）も指定する必要があります。
- チャンネル・イベントは自動的に起こるので、有効化する必要はありません。チャンネル・イベントをモニターしない場合には、チャンネル・イベント・キューを書き込み禁止にすることができます。

イベント・メッセージ

イベント・メッセージには、イベントの発生元に関係した情報（イベントのタイプ、イベントを引き起こしたアプリケーションの名前など）、さらにパフォーマンス・イベントの場合にはキューの短い統計要約も含まれています。

イベント・メッセージの形式は、PCF 応答メッセージの形式と類似しています。メッセージ・データは、*MQSeries* プログラム式システム管理 に説明されているデータ構造を使用したユーザー作成の管理プログラムによって、イベント・メッセージから取り出すことができます。

第10章 トランザクション・サポート

MQSeries アプリケーション・プログラミング・ガイド に、この章のトピックに関する詳しい紹介があります。この章ではトランザクション・サポートについて簡単に解説します。

アプリケーション・プログラムは、一連の更新内容を 1 つの作業単位としてグループ化します。作業単位としてグループ化される更新は相互に論理的に関連しているため、データの一貫性を維持できるように正常に処理されなければなりません。ある更新処理が正常に行われ、一方で他の更新処理が失敗すると、データの一貫性が失われます。

処理が正常に完了すると、作業単位は**コミット**されます。この時点で作業単位内のすべての更新内容が永久的なものとなり、これ以降は取り消し不可能になります。作業単位が失敗した場合は、すべての更新が**バックアウト**されます。**同期点調整**とは、データの一貫性を維持しながら作業単位をコミットまたはバックアウトするプロセスです。

ローカル 作業単位では、更新されるリソースのみが **MQSeries キュー・マネージャー**のリソースとなります。この場合、キュー・マネージャー自体が単一フェーズ・コミットによって同期点を調整します。

グローバル 作業単位では、**XA 準拠データベース**などの他のリソース・マネージャーにより管理されているリソースも更新されます。この場合、2 フェーズ・コミット手順を実行する必要があり、作業単位はキュー・マネージャー自体によって調整されます。

つまり、キュー・マネージャーのリソースはローカルまたはグローバルの作業単位の一部として更新できます。

ローカル作業単位

MQSeries キュー・マネージャーのリソースのみを更新する場合にローカル作業単位を使用します。更新のコミットには **MQCMIT** を、更新のバックアウトには **MQBACK** をそれぞれ使用します。

グローバル作業単位

XA 準拠データベース・マネージャーのリソースの更新も必要な場合に、グローバル作業単位を使用します。調整処理はキュー・マネージャーの内部または外部で行われます。

キュー・マネージャー調整

MQBEGIN でグローバル作業単位を開始し、MQCMIT でコミットするか、または MQBACK でバックアウトします。2 フェーズ・コミット・プロセスは、Oracle® などの XA 準拠のリソース・マネージャーに、まず最初にコミットの準備をさせる必要がある場合に使用します。すべてのプログラムで正常にコミットの準備ができると、次にコミットするよう通知されます。コミットの準備ができないと通知してくるリソース・マネージャーには、バックアウトを実行するよう通知されます。

外部調整

ここでは、IBM CICS®、Transarc Encina、または BEA Tuxedo などの XA 準拠のトランザクション・マネージャーによって調整が行われます。作業単位の開始とコミットはこれらのトランザクション・マネージャーの管理下で行われます。

MQBEGIN、MQCMIT、MQBACK は実行できません。

この章では、グローバル作業単位のサポートを使用可能にする方法について説明します（ローカル作業単位のサポートは、特別に使用可能にする必要はありません）。

この章には、次の節があります。

- 『データベースの調整』
- 159ページの『Oracle の構成』
- 164ページの『管理タスク』

データベースの調整

キュー・マネージャーによりグローバル作業単位を管理している場合、データベースの更新を MQ の作業単位に統合できるようになります。つまり、MQI と SQL の両方を使用してアプリケーションを作成し、MQCMIT と MQBACK を使用してキューとデータベースへの変更を一斉にコミットまたはロールバックできます。

キュー・マネージャーでは、2 フェーズ・コミット・プロトコルによりこの処理を行います。作業単位をコミットする場合、キュー・マネージャーはまず作業単位にかかわっている各データベース・マネージャーに対して、更新する準備ができているかどうかを確認します。キュー・マネージャーをはじめ、作業単位にかかわっているすべてのプログラムおよびデータベースがコミットでき

る状態にある場合にのみ、キューおよびデータベースの更新がすべてコミットされます。更新できない状態にあるプログラムが 1 つでもあると、作業単位はバックアウトされます。

コミット・プロトコル中にキュー・マネージャーとデータベース・マネージャーとの間の連絡が断たれた場合には、完全リカバリー・サポートを使用できます。データベース・マネージャーが未確定で使用できない場合、つまりコミットできる状態になっているがコミットまたはバックアウトの通知を受信していない場合には、キュー・マネージャーは通知を正常に送信できるまで作業単位の実行結果を保管しておきます。同様に、コミット操作が完了していないままの状態にキュー・マネージャーが終了した場合には、キュー・マネージャーの再開時にはこの未完了の状態が保管されています。

データベースの更新が含まれる作業単位を示すには、必ず MQI 動詞の MQBEGIN を使用します。MQSeries アプリケーション・プログラミング・ガイドに、MQSeries とデータベースの更新を 1 つの作業単位にまとめるプログラム例が記載されています。

キュー・マネージャーは *X/Open Distributed Transaction Processing: The XA Specification* (ISBN 1 872630 24 3) に記載されている XA インターフェースを使ってデータベース・マネージャーと通信します。つまり、キュー・マネージャーが通信できるデータベース・マネージャーは、この規格に準拠していることとなります。このようなデータベース・マネージャーは、XA 準拠データベース・マネージャーと呼ばれています。

表5 に、MQSeries バージョン 5.1 がサポートしている XA 準拠のデータベース・マネージャーを示します。

表5. XA 準拠リレーショナル・データベース

MQSeries プロダクト	DB2 [®]	Oracle	Sybase [®]
MQSeries for AIX	はい	はい	はい
MQSeries for HP-UX	はい	はい	いいえ
MQSeries for Sun Solaris	はい	はい	はい
MQSeries for Compaq OpenVMS Alpha	いいえ	はい	いいえ
MQSeries for Windows NT and Windows 2000	はい	いいえ	はい

制約事項

制約事項

データベース調整サポートに適用される制約事項は次のとおりです。

- MQSeries の作業単位内でのデータベース更新の調整は、MQI クライアント・アプリケーションではサポートされていません。
- MQI 更新とデータベース更新は、同一のキュー・マネージャー・サーバー・マシンで実行してください。
- データベース・サーバーは、キュー・マネージャー・サーバーとは別のマシンにインストールされていることがあります。この場合、データベースにはそのデータベース・マネージャーの XA 準拠クライアント機能を使ってアクセスしなければなりません。
- キュー・マネージャー自体は XA に準拠していますが、その他のキュー・マネージャーをグローバル作業単位の一部として構成できない場合もあります。これは、一度に 1 つの接続のみがサポートされるためです。

データベースの接続

キュー・マネージャーへの標準接続を確立するアプリケーションは、個別のローカル・キュー管理エージェント・プロセスのスレッドと関連付けられます。アプリケーションによって MQBEGIN が出されると、次にこのアプリケーションとエージェント・プロセスはいずれも作業単位に属しているデータベースへ接続しなければなりません。アプリケーションがキュー・マネージャーに接続している限り、データベースへの接続は維持されます。このことは、データベースによってサポートされるユーザーおよび接続の数が限られている場合には重要な考慮事項となります。

接続数を減らす方法として、アプリケーションで MQCONNX 呼び出しを使用して高速パス結合を要求します。この場合、アプリケーションとローカル・キュー・マネージャー・エージェントは同一のプロセスとなり、1 つのデータベース接続を共有できるようになります。この操作を行うまえに、高速パス・アプリケーションに適用される制限事項を確認するため、MQSeries アプリケーション・プログラミング・ガイド を参照してください。

データベース・マネージャーの構成

キュー・マネージャーが調整するグローバル作業単位でデータベース・マネージャーを使用するには、前もって次の作業を行ってください。

1. データベース・マネージャー用の XA スイッチ・ロード・ファイル¹ を作成する。
2. キュー・マネージャーの構成ファイル `qm.ini` 内に、データベース・マネージャーを定義する。
スイッチ・ロード・ファイルの名前を始めとするいくつかの項目を、`qm.ini` ファイル内に、定義する必要があります。

スイッチ・ロード・ファイルの作成

サポートされているデータベース・マネージャーのスイッチ・ロード・ファイルを作成するための説明については、160ページの『Oracle スイッチ・ロード・ファイルの作成』を参照してください。

インストール手順についての詳細は、MQSeries のインストール・マニュアルを参照してください。

スイッチ・ロード・ファイルの作成に使用するすべてのサンプル・ソース・モジュールに、`MQStart` という関数が含まれています。スイッチ・ロード・ファイルがロードされると、キュー・マネージャーにより `MQStart` が呼び出され、この `MQStart` によって XA スイッチと呼ばれる構造体のアドレスが戻されます。スイッチ・ロード・ファイルはデータベース・マネージャーのライブラリーとリンクされます。これにより、MQSeries はこのデータベース・マネージャーを呼び出すことができるようになります。

Oracle 用のスイッチ・ロード・ファイルの作成で使用するサンプル・ソース・モジュールは、`oraswit.c` です。

データベース・マネージャーの定義

データベース・マネージャー用のスイッチ・ロード・ファイルを作成したあとで、その位置をキュー・マネージャーに指定する必要があります。これは、キュー・マネージャーの `qm.ini` ファイルの `XAResourceManager` スタンザに指定します。

キュー・マネージャーで調整する予定の個々のデータベース・マネージャーについて、`XAResourceManager` スタンザを 1 つずつ追加する必要があります。

`XAResourceManager` スタンザの属性は次のとおりです。

1. XA スイッチ・ロード・ファイルは動的にロードされるオブジェクトで、キュー・マネージャーとデータベース・マネージャーの間の通信を可能にします。

データベース・マネージャーの構成

Name=name

データベース・マネージャーのインスタンスを識別するユーザー指定のストリング。

キュー・マネージャーの名前は必須指定属性で、長さは 31 文字までです。これは固有の名前でなければなりません。単にデータベース・マネージャーの名前を指定することもできますが、複雑な構成で固有の名前を維持するには、更新するデータベースの名前などを含めておく方法などがあります。

dspmqtrn コマンドの実行時には、キュー・マネージャーはメッセージと出力の両方でこの名前を使用してデータベース・マネージャーのインスタンスを参照するため、指定する名前は分かりやすいものにしておくことをお勧めします。

一度指定した名前は**変更しないでください**。構成情報の変更については、169ページの『構成情報の変更』に説明があります。

SwitchFile=name

データベース・マネージャーの XA スイッチ・ロード・ファイルの完全修飾名です。この属性は必須指定属性です。

XAOpenString=string

これは、呼び出しに入れてデータベース・マネージャーの `xa_open` 入り口点へ渡されるデータ・ストリングです。このストリングの形式はデータベース・マネージャーによって異なりますが、通常は更新されるデータベースの名前を識別するものでなければなりません。

この属性はオプションです。省略すると空白のストリングが想定されます。

XACloseString=string

これは、呼び出しに入れてデータベース・マネージャーの `xa_close` 入り口点へ渡されるデータ・ストリングです。このストリングの形式はデータベース・マネージャーによって異なります。

この属性はオプションです。省略すると空白のストリングが想定されます。

ThreadOfControl=THREAD|PROCESS

`ThreadOfControl` 値は `THREAD` または `PROCESS` のいずれかです。キュー・マネージャーはこの値を使用して直列化を行います。

データベース・マネージャが「スレッド・セーフ」である場合は、*ThreadOfControl* の値として `THREAD` を指定することができ、キュー・マネージャは複数のスレッドから同時にデータベース・マネージャを呼び出すことができます。

データベース・マネージャがスレッド・セーフでない場合は、*ThreadOfControl* の値に `PROCESS` を指定する必要があります。キュー・マネージャは、データベース・マネージャへの呼び出しをすべて直列化し、特定のプロセス内からの呼び出しが一度に 1 回ずつ行われるようにします。

これらの属性の詳細については、236ページの『`XAResourceManager` スタンザ』を参照してください。

サポートされている各データベース・マネージャと共に `MQSeries` を構成する場合に必要な特定の作業については、『`Oracle` の構成』に詳しい説明があります。

Oracle の構成

次の作業を行う必要があります。

- Oracle レベルの確認およびパッチの適用 (まだ行っていない場合)
- 環境変数の設定の確認
- Oracle XA サポートの使用可能化
- Oracle スイッチ・ロード・ファイルの作成
- `XAResourceManager` 構成情報の `qm.ini` ファイルへの追加
- 必要な場合には、Oracle 構成パラメーターの変更

Oracle の最小サポート・レベル

`OpenVMS` 上の Oracle では 8.1.6 以上がサポートされています。

環境変数の設定の確認

Oracle 環境変数がアプリケーション・プロセスとキュー・マネージャ・プロセスに対応して設定されていることを確認する必要があります。特に、キュー・マネージャを開始する前に次の環境変数を設定しておかなければなりません。

ORACLE_HOME

Oracle ホーム・ディレクトリーです。

ORACLE_SID

使用中の Oracle SID です。

Oracle XA サポートの使用可能化

Oracle XA サポートが使用できる状態になっていることを確認してください。特に、Oracle 共用ライブラリーが作成されていることを確認します。このライブラリーは Oracle XA ライブラリーのインストール時に作成されます。

Oracle8 のインストール時には、このライブラリーは自動的に作成されます。ライブラリーを再作成する必要がある場合は、ご使用のプラットフォームに対応する *Oracle8 Administrator's Reference* を参照してください。

Oracle スイッチ・ロード・ファイルの作成

Oracle スイッチ・ロード・ファイルを作成する最も簡単な方法は、サンプル・ファイルを使用する方法です。Oracle スイッチ・ロード・ファイルを作成するために使用するソース・コードを図11 に示します。

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

extern struct xa_switch_t xaosw;

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

図11. Oracle スイッチ・ロード・ファイル *oraswit.c* のソース・コード

組み込まれている *xa.h* ヘッダー・ファイルは *MQSeries* に添付されており、*oraswit.c* と同じディレクトリーに入っています。

OpenVMS システム上での Oracle スイッチ・ロード・ファイルの作成

OpenVMS システムで Oracle スイッチ・ロード・ファイルを作成する場合、*oraswit.c* をコンパイルして、Oracle クライアント・ライブラリー "*oraclient_v816.exe*" にリンクする必要があります。

1. 作成する Oracle スイッチ・ロード・ファイル *oraswit* の保存先ディレクトリーを作成します。
2. 次のファイルを *mq5_examples:[xatm]* から、作成したディレクトリーにコピーします。
 - *xa.h*
 - *oraswit.c*
3. コピーしたソース・ファイル (*oraswit.c*) をコンパイルします。たとえば、次のとおりです。

```
$ cc oraswit0.c
```

4. 次のようにして、スイッチ・ロード・ファイルを生成します。

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

5. MQStart プロシージャは、生成されたイメージから実行時に動的にロードされます。したがって、".exe" 拡張子のない生成ファイルを指すようにするために、論理名がシステム論理名テーブルで定義されているか、ファイルを sys\$share にコピーする必要があります。たとえば、作成されたファイルの位置が、ディスク \$a_device:[a_directory]oraswit0.exe の場合は、次のいずれかを使用してください。

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

または、

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

Oracle 用の XResourceManager 構成情報の追加

次に、キュー・マネージャの qm.ini 構成ファイルを変更して、Oracle をグローバル作業単位に参加するプログラムとして定義します。

XResourceManager スタンザに次の属性を追加する必要があります。

Name=name

これは必須属性です。作業単位にかかわるプログラムに適切な名前を指定します。更新するデータベースの 名前を含めることもできます。

SwitchFile=name

これは必須属性です。Oracle スイッチ・ロード・ファイルの完全修飾名。

XAOpenString=string

Oracle 用 XA オープン・ストリングの形式は次のとおりです。

```
Oracle_XA+Acc=P//|P/userName/passWord
+SesTm=sessionTimeLimit
[+DB=dataBaseName]
[+GPwd=P/groupPassWord]
[+LogDir=logDir]
[+MaxCur=maximumOpenCursors]
[+SqlNet=connectString]
```

説明

Acc= 必須指定属性です。ユーザー・アクセス情報を指定します。 **P//** を指定すると、ユーザーまたはパスワード情報が明示的に指定されず、 **ops\$login** 形式が使用されます。 **P/userName/passWord** には、有効な ORACLE ユーザー ID と対応するパスワードを指定します。

SesTm=

必須指定属性です。トランザクションが自動的に削除されるまでに、そのトランザクションがアクティブでない状態にある最大時間を指定します。秒単位で指定します。

DB= データベース名を指定します。 **DataBaseName** には、Oracle プリコンパイラーがデータベースを識別する際に使用する名前を指定します。このフィールドを指定する必要があるのは、アプリケーションによって明示的にデータベース名が指定される場合 (SQL 文の AT 句を使用する場合) です。

GPwd=

サーバー・セキュリティー・パスワードを指定します。
P/groupPassWord に、サーバー・セキュリティー・グループ・パスワード名を指定します。サーバー・セキュリティー・グループにより、1 つの ORACLE インスタンスに対して稼動している複数のアプリケーションの保護が強化されます。デフォルトでは ORACLE 定義のサーバー・セキュリティー・グループが指定されます。

LogDir=

ローカル・マシンで、Oracle XA ライブラリー・エラーおよびトレース情報のログを記録するディレクトリーを指定します。値を指定しないと、現行ディレクトリーが想定されます。ユーザー **mqm** がこのディレクトリーへの書き込みアクセスを持っていることを確認してください。

MaxCur=

データベースのオープン時に割り振られるカーソルの数を指定します。このオプションを指定すると、プリコンパイラ・オプションの `maxopencursors` を指定したときと同じ効果が得られます。

SqlNet=

システムへのログオン時に使用される SQL*Net 接続ストリングを指定します。指定できる接続ストリングは SQL*Net V1 ストリング、SQL*Net V2ストリング、SQL*Net V2 別名のいずれかです。キュー・マネージャーが稼動しているマシンとは別のマシン上に Oracle をセットアップする場合は、このフィールドを必ず指定してください。

詳細については、*Oracle8 Server Application Developer's Guide* (資料番号 A54642-01) を参照してください。

XACloseString=string

Oracle では XA クローズ・ストリングは必要ありません。

ThreadOfControl=THREAD|PROCESS

MQSeries for Compaq OpenVMS プラットフォームではこのパラメーターを指定する必要はありません。

上記の各属性の詳細な説明については、236ページの『XAResourceManager スタンザ』を参照してください。

図12 では、更新されるデータベースは MQBankDB です。XA オープン・ストリングに `LogDir` を追加して、すべてのエラーおよびトレース情報のログが同一のディレクトリー内に記録されるようにしておくことをお勧めします。この例では、Oracle スイッチ・ロード・ファイルの作成後、このファイルが `sys$share` ディレクトリーにコピーされていることを想定しています。

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

図12. Oracle 用サンプル XAResourceManager エントリー

Oracle 構成パラメーターの変更

キュー・マネージャーとユーザー・アプリケーションでは、Oracle への接続時には XA オープン・ストリングに指定されているユーザー ID が使用されません。

Oracle の構成

- **データベース権限** オープン・ストリングに指定される Oracle ユーザー ID は、DBA_PENDING_TRANSACTIONS ビューへのアクセス権限が付与されているものでなければなりません。

この権限は次のコマンドによって付与できます。 userID には、アクセス権限が付与されるユーザー ID が入ります。

```
grant select on DBA_PENDING_TRANSACTIONS to userID;
```

セキュリティーについての詳細は、103ページの『第7章 MQSeries オブジェクトの保護』を参照してください。

管理タスク

通常の運用では、構成タスクの完了後には、最小限の管理タスクのみが必要とされます。キュー・マネージャーは、データベース・マネージャーが使用できない状況でもその影響を受けることが少ないため、管理ジョブは容易です。特に、次のような特長があります。

- キュー・マネージャーを開始する際に、前もって各データベース・マネージャーを開始する必要はありません。
- いずれかのデータベース・マネージャーが使用できなくなった場合に、キュー・マネージャーを停止して再始動する必要はありません。

つまり、データベース・マネージャーとは別にキュー・マネージャーを開始および停止できます。同様に、データベース・マネージャーがキュー・マネージャーをサポートしている場合には、キュー・マネージャーとは別にデータベース・マネージャーを開始および停止できます。

キュー・マネージャーとデータベース・マネージャーの連絡が失われた場合には、両方のプログラムが回復した後に再同期する必要があります。

再同期とは、該当するデータベースが関連している未確定の作業単位を完了するプロセスです。通常、再同期は自動的に実行されるため、ユーザーが介入する必要はありません。キュー・マネージャーはデータベース・マネージャーに対して、未確定の作業単位のリストを要求します。次に、キュー・マネージャーは未確定の作業単位をそれぞれコミットまたはロールバックするよう、データベース・マネージャーに通知します。

キュー・マネージャーが停止した場合、プログラムの再始動時に各データベース・マネージャー・インスタンスと再同期する必要があります。あるデータベース・マネージャーが使用できない状態になった場合、キュー・マネージャー

が再同期する必要があるのはこのプログラムのみです。キュー・マネージャーは次の再始動時にこのデータベース・マネージャーが使用可能な状態に戻ったと認識します。

新しいグローバル作業単位が開始されると、キュー・マネージャーは使用できない状態にあるデータベース・マネージャーとの連絡を自動的に回復しようとします。あるいは、**rsvmqtrn** コマンドを実行して、すべての未確定の作業単位を明示的に解決することもあります。

未確定の作業単位

データベース・マネージャーは、準備 (PREPARE) するよう通知を受けた後にキュー・マネージャーとの連絡を失った場合、このデータ・マネージャーがかかわっている作業単位が未確定になることがあります。データベース・マネージャーは、キュー・マネージャーから COMMIT または ROLLBACK の実行結果を受信するまでは、更新に関連するデータベース・ロックを維持する必要があります。

このロックにより、他のアプリケーションがデータベース・レコードの更新または読み取りを実行できなくなるため、できるだけ早く再同期を実行する必要があります。

何らかの理由により、キュー・マネージャーによるデータベースの自動再同期よりも早い時点で再同期を行わなければならない場合には、データベース・マネージャーの機能を使用してデータベースの更新を手動でコミットまたはロールバックすることもあります。これは発見的 (*heuristic*) 決定法と呼ばれる方法ですが、データの一貫性を失う可能性があるため、できるだけ使用しないようにしてください。あるデータベースの更新を手動でコミットしたところ、その他のすべての接続データベースがロールバックしたり、その逆の事態が発生することがあります。

このような場合、自動再同期を開始するには、キュー・マネージャーを再始動するか、データベースがすでに再始動している場合には **rsvmqtrn** コマンドを実行する方法のほうがはるかに安全です。

dspmqtrn コマンドを使用する

データベース・マネージャーが使用不能な状態になっている場合には、**dspmqtrn** コマンドを実行して、該当するデータベースが含まれる未解決の作業単位 (UOW) の状態を確認できます。

データベース・マネージャーが使用不能な状態になると、2 フェーズ・コミット・プロセスに入る前に、データベース・マネージャーが関与している未完了

管理タスク

の UOW はすべてロールバックされます。データベース・マネージャー自体は、次の再始動時に未完了の UOW をロールバックします。

dspmqtrn コマンドを実行すると、キュー・マネージャーからの COMMIT または ROLLBACK を待機しており、未確定の状態にあるプログラム (participant) が 1 つまたは複数ある作業単位のみを表示します。

作業単位ごとに、各接続データベース (participant) の状態が表示されます。特定のリソース・マネージャーのリソースを更新しなかった作業単位は表示されません。

未確定の作業単位の状態に応じて、リソース・マネージャーは、次のいずれかが済んだ状態と見なされます。

準備済み (Prepared)

リソース・マネージャーでは更新をコミットする準備ができています。

コミット済み (Committed)

リソース・マネージャーによる更新のコミットはすでに実行されています。

ロールバック済み (Rolled-back)

リソース・マネージャーによる更新のロールバックはすでに実行されています。

認識 (Participated)

リソース・マネージャーは作業単位にかかわっていますが、準備できておらず、またコミットもロールバックも済んでいません。

キュー・マネージャーでは、作業単位にかかわっているプログラムの個別の状態は再始動時には保管されません。キュー・マネージャーを再始動した時に連絡をとれないデータベース・マネージャーがある場合、再始動時にそのデータベース・マネージャーがかかわっている未確定の作業単位は解決されません。この場合、再同期が行われるまで、データベース・マネージャーは準備済み 状態であるとして報告されます。

dspmqtrn コマンドによって未確定の UOW を表示すると、まず最初に、関与している可能性のあるリソース・マネージャーがすべてリストされます。これらのプログラムには固有の ID *RMIId* が割り振られています。この ID は、未確定の UOW に基づいてリソース・マネージャーの状態が報告される場合に、リソース・マネージャーの *Name* の代わりに使用されます。

図13 は、次のコマンドを発行した結果の出力を示します。

```
dspmqrn -m MY_QMGR
```

```
AMQ7107: Resource manager 0 is MQSeries.
AMQ7107: Resource manager 1 is Oracle MQBankDB
AMQ7107: Resource manager 2 is Oracle MQFeedB

AMQ7056: Transaction number 0,1.
        XID: formatID 5067085, gtrid_length 12, bqual_length 4
            gtrid [3291A5060000201374657374]
            bqual [00000001]
AMQ7105: Resource manager 0 has committed.
AMQ7104: Resource manager 1 has prepared.
AMQ7104: Resource manager 2 has prepared.
```

図13. *dspmqrn* による出力の例

図13 の出力は、3つのリソース・マネージャーがキュー・マネージャーに関連付けられていることを示しています。1番目のリソース・マネージャー0は、キュー・マネージャー自体です。その他の2つのリソース・マネージャー・インスタンスはMQBankDB およびMQFeedB Oracle データベースです。

この例では、未確定の作業単位は1つのみです。3つのリソース・マネージャーすべてについて1つのメッセージが出されます。これは、キュー・マネージャーと2つのOracleデータベースに対して作業単位内で変更が行われたことを示します。

キュー・マネージャー、つまりリソース・マネージャー0に対して行われた更新は、コミット済み の状態です。Oracleデータベースに対する更新は準備済み 状態にあります。これは、データベースMQBankDB およびMQFeedB に対する更新をコミットするためにOracleが呼び出された時点よりも前に、Oracleが使用できない状態にあったことを示します。

未確定の作業単位には、拡張子XIDが付いています。これは、Oracleと更新を関連付ける拡張子です。

rsvmqtrn コマンドを使用する

図13 に示す出力には、両方のOracleデータベースに対してコミットの決定がまだ送達されていない未確定UOWが1つあることが示されています。

この作業単位を完了するには、Oracle が次に使用可能な状態になった時点でキュー・マネージャーと Oracle を再同期する必要があります。キュー・マネージャーでは、新しい作業単位の開始時に Oracle との連絡を再度獲得しようとし、あるいは、**rsvmqtrn** コマンドを使用して明示的に再同期を実行するよう、キュー・マネージャーに指定することもできます。未確定の作業単位と関連するデータベース・ロックをできるだけ早く解放するため、Oracle が再始動した直後にキュー・マネージャーに対してこのように指定してください。

このためには、キュー・マネージャーに対してすべての未確定の作業単位を解決するよう指示する **-a** オプションを使用します。次の例では、Oracle が再始動しているため、キュー・マネージャーが未確定の作業単位を解決できます。

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

未確定のトランザクションは、すべて解決されます。

エラーが混在する実行結果

キュー・マネージャーは 2 フェーズ・コミット・プロトコルを使用しますが、作業単位の実行結果とエラーが混在する可能性を完全に排除するわけではありません。作業単位にかかわっているプログラムのうち、一部が更新をコミットし、一部が更新をバックアウトした場合にこのような混在が発生します。

ある作業単位で実行結果にエラーが混在すると、共用リソースが整合性のない状態になってしまうため、深刻な影響が出る可能性があります。

実行結果にエラーが混在する事態は、キュー・マネージャーによって未確定の作業単位を解決せずに、発見的な方法で作業単位について決定を下した場合に主に発生します。

キュー・マネージャーは発見的方法による障害を検出すると、必ず **FFST**[®] 情報を作成し、エラー・ログにその障害を記録し、次のいずれかのメッセージを表示します。

- データベース・マネージャーがコミットの代わりにロールバックした場合には次のメッセージが表示されます。

```
AMQ7606 A transaction has been committed but one or more resource  
managers have rolled back.
```

- データベース・マネージャーがロールバックの代わりにコミットした場合には次のメッセージが表示されます。

```
AMQ7607 A transaction has been rolled back but one or more resource
managers have committed.
```

さらに、発見的方法によって障害が発生したデータベースを識別するメッセージが表示されます。この場合、障害のあるデータベースに対して適切なりカバリー処置を行い、整合性を復元してください。このりカバリー処置は複雑です。まず、誤ってコミットまたはロールバックした更新を隔離し、次にデータベースの変更を手動でやり直しまたは再実行しなければなりません。

ソフトウェアが原因で損害を受ける可能性はこれよりも低くなります。ソフトウェアが原因となって発生したエラーの影響を受ける作業単位には、メッセージ AMQ7112 によってそのトランザクション番号が報告されます。この場合、作業単位にかかわっているプログラムなどは整合性のない状態になっている可能性があります。

```
rsvmqtrn -m MY_QMGR

AMQ7107: Resource manager 0 is MQSeries.
AMQ7107: Resource manager 1 is Oracle MQBankDB
AMQ7107: Resource manager 2 is Oracle MQFeedB

AMQ7112: Transaction number 0,1 has encountered an error.
        XID: formatID 5067085, gtrid_length 12, bqual_length 4
        gtrid [3291A5060000201374657374]
        bqual [00000001]
AMQ7105: Resource manager 0 has committed.
AMQ7104: Resource manager 1 has prepared.
AMQ7104: Resource manager 2 has rolled back.
```

図 14. *dspmqrn*によるエラーのトランザクションの出力例

キュー・マネージャーは、次の再始動時までこのような障害のりカバリーを行いません。図14 では、作業単位を解決する **rsvmqtrn** が出されても、リソース・マネージャー **1**、つまりデータベース **MQBankDB** に対する更新は **準備済み (prepared)** 状態のままになります。

構成情報の変更

キュー・マネージャーが正常に開始され、グローバル作業単位を調整する準備ができた後で、`qm.ini` ファイル内の `XAResourceManager` スタンザに変更を加える際には、十分に注意する必要があります。

qm.ini ファイルは、変更する必要があるればいつでも変更できますが、変更内容が有効になるのはキュー・マネージャーの再始動後です。たとえば、データベース・マネージャーに渡される XA オープン・ストリングを変更しなければならない場合、このストリングの変更を有効にするにはキュー・マネージャーを再始動する必要があります。

XAResourceManager スタンザを削除すると、実際にはキュー・マネージャーがデータベース・マネージャーと連絡をとる機能が除去されるので注意してください。

XAResourceManager スタンザの *Name* 属性は、絶対に変更しないでください。この属性により、キュー・マネージャーに対してデータベース・マネージャーのインスタンスが識別されます。この固有の ID が変更されると、キュー・マネージャーはデータベース・マネージャーのインスタンスが削除されており、まったく新しいインスタンスが追加されているものと想定します。キュー・マネージャーは未解決の作業単位をその古い名前 (*Name*) に関連付けており、この時データベースは未確定の状態になっています。

データベース・マネージャーのインスタンスの削除

システム構成からデータベースまたはデータベース・マネージャーを削除する必要がある場合、まず初めにデータベースが未確定状態でないことを確認してください。また、この確認はキュー・マネージャーを再始動する前に行ってください。ほとんどのデータベース・マネージャーには、未確定のトランザクションのリストを表示するコマンドがあります。未確定トランザクションがある場合は、XAResourceManager スタンザを削除する前に、まずキュー・マネージャーとデータベース・マネージャーを再同期させます。

ユーザーがこのプロシージャーを監視できない場合、キュー・マネージャーが、該当するデータベースを対象とするすべての未確定作業単位を保管しています。キュー・マネージャーの再始動時に毎回警告メッセージ AMQ7623 が出されます。今後、このデータベースをキュー・マネージャーを使用して構成することがない場合、**rsvmqtrn** コマンドの **-r** オプションを使用して、このような未確定のトランザクション内のデータベースの関連するプログラムをすべて消去するようプログラムに指示します。

注: キュー・マネージャーは、すべての関連するプログラムについて同期点処理が完了した時にはじめて、完全にトランザクションを消去します。

XAResourceManager スタンザを一時的に削除する必要がある場合があります。このような場合には、今後スタンザを復元する際に簡単に復元できるように、スタンザをコメント化する方法が最も適切です。キュー・マネージャーが特定

のデータベースまたはデータベース・マネージャーと連絡をとる時に毎回エラーが発生する場合、この処置を行います。該当する `XAResourceManager` エントリーを一時的に削除すると、キュー・マネージャーはその作業単位にかかわるその他のプログラムなどがすべてが含まれているグローバル作業単位を開始します。 `XAResourceManager` スタanzasのコメント化の例を次に示します。

```
# This database has been temporarily removed
#XAResourceManager:
# Name=Oracle MQBankDB
# SwitchFile=sys$share:oraswit0
# XAOpenString=MQBankDB
```

図 15. `XAResourceManager` スタanzasのコメント化

第11章 リカバリーおよび再始動

メッセージ・システムでは、システムに入ってきたメッセージが確実にその宛先に配布されます。これは、システム内のメッセージを追跡し、何らかの理由でシステムに障害が起きた場合でもメッセージをリカバリーする手段が必要であることを意味します。

MQSeriesでは、メッセージの受信、伝送、および配布を取り扱うキュー・マネージャーの活動の記録 (ログ) を維持することにより、メッセージが失われないようにしています。MQSeriesは、次の3つのタイプのリカバリーに関してログを使用します。

1. 再始動リカバリー。計画どおりにMQSeriesを停止した場合
2. 破損リカバリー。予期しない障害によってMQSeriesが停止した場合
3. メディア・リカバリー。損傷したオブジェクトを復元する場合

いずれの場合も、リカバリーによってキュー・マネージャーは、停止した時の状態に復元されます。すべての未完了トランザクションはロールバックされ、キュー・マネージャーの停止時にコミットされていなかったメッセージはキューから除去されます。リカバリーによって、持続メッセージはすべて復元され、非持続メッセージはこのプロセスの間に失われることとなります。

この章の残りの部分では、リカバリーと再始動の概念をさらに詳しく紹介し、問題が起きたときにリカバリーする方法を説明します。この章では、次の事柄を説明します。

- 174ページの『メッセージの消失を確実に回避する (ロギング)』
- 177ページの『チェックポイント機能 (完全なりカバリーの保証)』
- 180ページの『ログのサイズの計算』
- 182ページの『ログの管理』
- 185ページの『ログをリカバリーに使用する』
- 189ページの『MQSeries ログ・ファイルの保護』
- 189ページの『バックアップと復元』
- 191ページの『リカバリーの手順』
- 194ページの『dmpmqlog コマンドを使用したログの内容のダンプ』

メッセージの消失を確実に回避する (ロギング)

MQSeries は、キュー・マネージャーが制御するデータに対する重要な変更をすべてログに記録します。ログには、オブジェクトの作成および削除、すべての持続メッセージの更新、トランザクション状態、オブジェクト属性の変更、およびチャネル活動が含まれます。したがって、ログは、メッセージ・キューのすべての更新をリカバリーするために必要な情報を、次の方法によって格納します。

- キュー・マネージャーの変更の記録を保持する
- 再始動プロセス用としてキューの更新の記録を保持する
- ハードウェア障害またはソフトウェア障害の後にデータを復元できるようにする

ログの概要

MQSeriesのログは、次の 2 つのコンポーネントからなっています。

1. 1 つまたは複数のログ・データ・ファイル
2. ログ制御ファイル

記録されているデータを含んでいるいくつかのログ・ファイルがあります。それらの数とサイズを (180ページの『ログのサイズの計算』で説明されているように) 定義することもできますし、あるいはシステム・デフォルト値の 3 ファイル (それぞれ 4MB のサイズ) を取ることもできます。

キュー・マネージャーを作成するときに、定義するログ・ファイルの数は、割り振られる 1 次 ログ・ファイルの数です。数を指定しないと、デフォルト値が使用されます。ログ・パスを変更していなければ、ログ・ファイルは次のディレクトリーに作成されます。

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

MQSeriesは、これらの 1 次ログ・ファイルと共に開始しますが、このログが満杯になり始めると、2 次 ログ・ファイルを割り振ります。MQSeries はこれを動的に行い、ログ・スペースの要求が少なくなったら、それを除去します。デフォルトでは、8MB の余分のディスク・スペースがあれば、最高 2 個の 2 次ログ・ファイルを割り振ることができます。デフォルトの数を変更することもできます。221ページの『第13章 MQSeries の構成』を参照してください。

ログ制御ファイル

ログ制御ファイルには、ログ・ファイルの使用状況をモニターするために必要な情報 (サイズと位置、次に使用可能なファイルの名前など) が入っています。

注: キュー・マネージャーを開始するときに作成されるログが、アプリケーションで取り扱うメッセージのサイズとボリュームに十分見合った大きさになるようにしてください。デフォルトのログの数とサイズは、要件に合わせて修正する必要があるでしょう。デフォルト値の変更方法については、180 ページで説明されています。

ログのタイプ

MQSeries では、ロギングに使用されるファイルの数は、ファイル・サイズ、受け取ったメッセージの数、およびメッセージ長によって異なります。キュー・マネージャー・アクティビティのレコードを維持する方法には、循環ロギングとリニア・ロギングの 2 つがあります。

循環ロギング

ログを使用してシステムの停止時に処理中であったトランザクションをロールバックするだけの再始動リカバリーでよい場合には、循環ロギングを使用してください。

循環ロギングでは、すべての再始動データをログ・ファイルのリングに保持します。ロギングでは、リングの最初のファイルに記入され、それから次のファイルに進むといった方法で、すべてのファイルに記入されます。その後、リングの最初のファイルに戻って、再び記入が開始されます。この操作は、プロダクトが使用されている間ずっと行われるので、ログ・ファイルが不足するという事は決してありません。

上記は循環ロギングの簡単な説明です。しかし、複雑な問題点もあります。キュー・マネージャーをデータの消失なしに再始動するために必要なログ項目は、キュー・マネージャーのデータ・リカバリーの必要性がなくなるまで保持されるという点です。再利用のためにログ・ファイルを解放するためのメカニズムについては、177ページの『チェックポイント機能 (完全なリカバリーの保証)』で説明されています。現時点では、MQSeries は、ログの容量を必要に応じて拡張するために、2 次ログ・ファイルを使用するという事を知っておいてください。

リニア・ロギング

再始動リカバリーおよびメディア・リカバリー (または順方向リカバリー) の両方を望む場合には、リニア・ロギングを使用します。メディア・リカバリーは、ログの内容を再生することにより、消失または損傷したデータを再作成するものです。

ロギング

リニア・ロギングでは、ログ・データを連続したファイルに保持します。この方法では、スペースは再利用されません。キュー・マネージャーが作成された時点以降のログ記録をいつでも取り出すことができます。

ディスク・スペースは限られているので、保存の形式について考慮を要する場合があります。管理者は、必要に応じて既存のスペースを再利用したり拡張したりして、ログのディスク・スペースを管理してください。

リニア・ロギングで使用されるログ・ファイルの数は、メッセージ・フローやキュー・マネージャーの持続期間によっては、非常に大きくなる可能性があります。ただし、アクティブと呼ばれているいくつかのファイルがあります。アクティブ・ファイルには、キュー・マネージャーを再始動するのに必要なログ項目が入ります。アクティブ・ログ・ファイルの数は、通常は、構成ファイルに定義された 1 次ログ・ファイルの数です。(数を定義する方法の詳細については、180ページの『ログのサイズの計算』を参照してください。)

ログ・ファイルがアクティブと呼ばれるかそうでないかを制御する重要なイベントは、チェックポイントです。MQSeries チェックポイントは、キュー・マネージャーの再始動を正常に行えるようにする情報が含まれているログ・レコードのグループです。以前に記録された情報は、キュー・マネージャーを再始動するには不要なので、非アクティブと呼ばれます。(チェックポイント機能の詳細については、177ページの『チェックポイント機能 (完全なりカバリーの保証)』を参照してください。)

非アクティブ・ログ・ファイルが必要でなくなる時期を判別する必要があります。それらのログ・ファイルを保存することもできますし、操作に必要でなくなった場合は削除することもできます。ログ・ファイルの後処理の詳細については、182ページの『ログの管理』を参照してください。

新しいチェックポイントが、2 番目以降の 1 次ログ・ファイルに記録された場合、最初のファイルは非アクティブになり、新しい 1 次ファイルがフォーマットされて 1 次プールの最後に追加されます。こうしてロギングに使用される 1 次ファイルの数が復元されます。このようにして、1 次ログ・ファイル・プールは、絶えず拡張されるログ・ファイル・リスト中にある現行のファイル・セットのように見えます。また、管理者は操作の要件に応じて非アクティブ・ファイルを管理してください。

2 次ログ・ファイルはリニア・ロギング用として定義されますが、このファイルは通常の操作では使用されません。持続期間の長いトランザクションのために、アクティブ・プールからファイルを解放できないような状態が生じた場合

には (そのファイルが再始動に必要とされる可能性がまだ存在するため)、 2 次ファイルがフォーマットされて、アクティブ・ログ・ファイル・プールに追加されます。

使用可能な 2 次ファイルの数が使い果たされたら、ログ活動を必要とするそれ以降のほとんどの操作の要求は拒否され、MQRC_RESOURCE_PROBLEM がアプリケーションに戻されます。

いずれのタイプのロギングも、予期しない停電に対処できます。ただし、ハードウェア障害が起こらないことを想定した場合です。

チェックポイント機能 (完全なリカバリーの保証)

メッセージ・キューに対する持続更新は、2 段階で行われます。まず、更新を表すレコードがログに書き込まれ、その後にキュー・ファイルが更新されます。つまり、ログ・ファイルの方が、キュー・ファイルよりも最新のものになります。再始動処理が確実に整合点から開始されるようにするために、MQSeries はチェックポイントを使用します。チェックポイントとは、ログに記述されているレコードとキューのレコードが一致する時点 (ポイント) です。チェックポイント自体は、キュー・マネージャーを再始動するのに必要な一連のログ・レコードです。たとえば、チェックポイントの時点でアクティブであったすべてのトランザクション (つまり作業単位) の状態です。

チェックポイントは、MQSeries によって自動的に生成されます。チェックポイントは、キュー・マネージャーの開始時、シャットダウン時、ログ・スペースが不足してきたとき、および 1000 回の操作がログに記録されるたびに取られます。キューが取り扱うメッセージが増えるにつれて、チェックポイント・レコードは、キューの現在の状態と一致しなくなります。

MQSeries は、再始動された時に、ログの中の最新のチェックポイント・レコードを見つけます。その情報は、すべてのチェックポイントの最後に更新されるチェックポイント・ファイルに保持されています。チェックポイント・レコードは、ログとデータの間最新の整合点を表すものです。このチェックポイントのデータは、チェックポイント時に存在していたとおりのキューを再作成するのに使用されます。キューが再作成されると、システムの障害時以前または停止時以前の状態にキューを戻すために、ログを作動させます。

MQSeries は、ログのヘッドとテールを指し示す内部ポインターを維持します。MQSeries は、ヘッドを指すポインターを、リカバリー・メッセージ・データとの整合性がある最新のチェックポイントに動かします。

チェックポイント機能

チェックポイントは、リカバリーを効率よく行ったり、1 次ログ・ファイルおよび 2 次ログ・ファイルの再利用を制御するために使用されます。

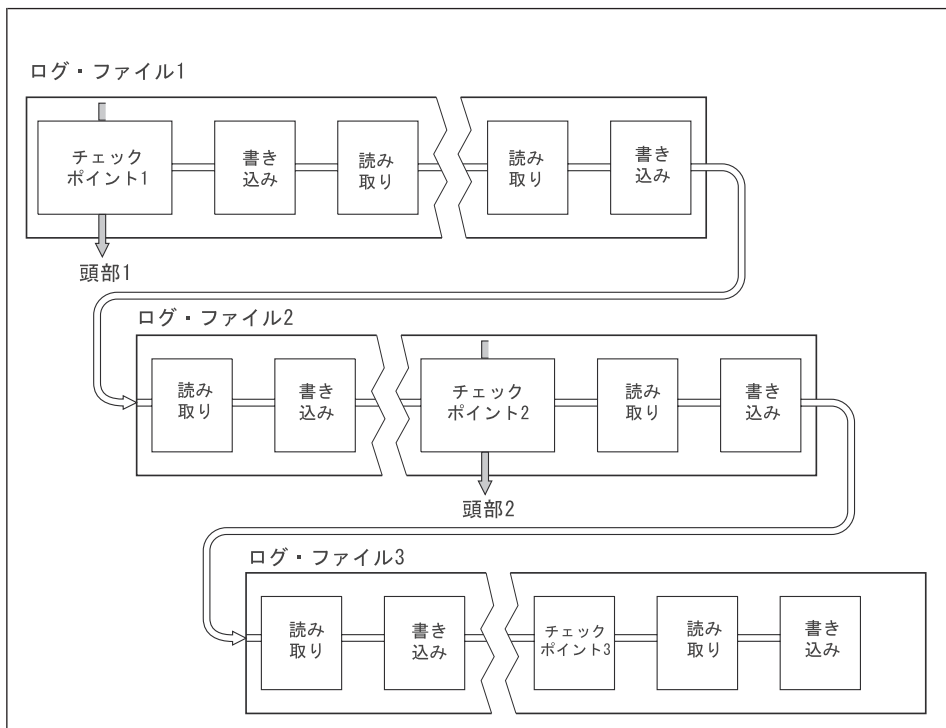


図 16. チェックポイント機能. 説明を簡単にするために、ログ・ファイルの両端のみを示してあります。

図16 では、最新のチェックポイント (チェックポイント 2) より前のすべてのレコードは、MQSeries にはもう必要ありません。このチェックポイント情報およびその後のログ項目から、キューをリカバリーすることができます。循環ロギングでは、チェックポイントより前の解放されたファイルは再利用できます。リニア・ロギングでは、解放されたログ・ファイルは、通常操作ではアクセスする必要がなくなり、非アクティブになります。例では、キューのヘッド・ポインターは、最新のチェックポイント (チェックポイント 2) の地点に移され、これが新しいキューのヘッド (ヘッド 2) になります。これでログ・ファイル 1 を再利用できるようになりました。

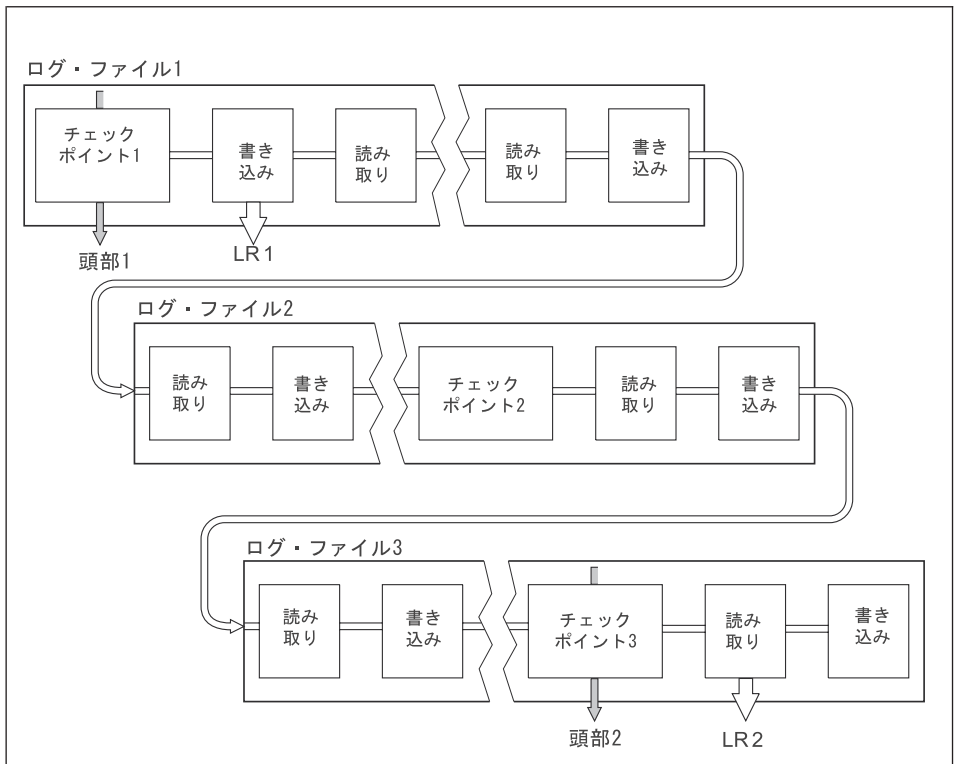


図 17. 持続期間の長いトランザクションでのチェックポイント機能。説明を簡単にするために、ログ・ファイルの両端のみを示してあります。

図17 は、持続期間の長いトランザクションが、ログ・ファイルの再利用にどんな影響を与えるかを示したものです。例では、持続期間の長いトランザクションによって、示されている最初のチェックポイントの後に LR 1 で示されているログへの入力が始まりました。3 番目のチェックポイントの後までトランザクションは完了しません (LR 2 で示されています)。LR 1 以降のすべてのログ情報は、このトランザクションのリカバリーを行うために、必要であればトランザクションが完了するまで保持されます。

持続期間の長いトランザクションが完了した後に (LR 2)、ログのヘッドがチェックポイント 3 (ログに記録された最も新しいチェックポイント) に移されます。チェックポイント 3 (ヘッド 2) より前のログ・レコードを含んでいるファイルは、不要になります。循環ロギングを使用している場合には、スペースは再利用できます。

持続期間の長いトランザクションが完了する前に 1 次ログ・ファイルがすべて記入された場合には、ログ満杯状況のリスクを避けるために、可能ならば 2 次ログ・ファイルが使用されます。

チェックポイント機能

循環ロギングを使用している場合に、ログのヘッドが移されたときは、1 次ログ・ファイルが再利用できます。ロガーは、カレント・ファイルが満杯になった後に、使用できる最初の 1 次ファイルを再利用します。リニア・ロギングを使用している場合は、ログのヘッドはアクティブ・プールのさらに後方に移されて、最初のファイルが非アクティブになります。新しい 1 次ファイルがフォーマットされて、将来のロギング活動に備えてプールの下部に追加されます。

ログのサイズの計算

キュー・マネージャーで循環ロギングとリニア・ロギングのどちらを使用するかを決めたら、次の作業は、キュー・マネージャーに必要なログのサイズを見積もることです。ログのサイズは、次のログ構成パラメーターによって決まります。

LogFilePages

4 KB のページを 1 単位として表した、個々の 1 次ログ・ファイルまたは 2 次ログ・ファイルのサイズ

LogPrimaryFiles

事前に割り振る 1 次ログ・ファイルの数

LogSecondaryFiles

1 次ログ・ファイルが満杯になったときに使用するために作成可能な 2 次ログ・ファイルの数

表6 は、各種の操作の場合にキュー・マネージャーがログに記録するデータの量を示します。キュー・マネージャーが実行するほとんどの操作では、最少量のログ・スペースしか必要ありません。しかし、持続メッセージをキューに入れる場合は、メッセージのリカバリーを可能にするために、すべてのメッセージ・データをログに書き込む必要があります。したがって、ログのサイズは、通常、キュー・マネージャーが取り扱う必要がある持続メッセージの数とサイズによって決まります。

表6. ログのオーバーヘッド・サイズ (すべての値は概算値です)

操作	サイズ
持続メッセージの書き込み	750 バイト + メッセージ長。メッセージが大きい場合、これは 15700 バイトのセグメントに分割され、それぞれのセグメントに 300 バイトのオーバーヘッドがかかります。
メッセージの読み取り	260 バイト
同期点、コミット	750 バイト

表 6. ログのオーバーヘッド・サイズ (すべての値は概算値です) (続き)

操作	サイズ
同期点、ロールバック	1000 バイト + 12 バイト (ロールバックされる読み取りまたは書き込みごとに)
オブジェクトの作成	1500 バイト
オブジェクトの削除	300 バイト
属性の更新	1024 バイト
メディア・イメージの記録	800 バイト + イメージ。イメージは 15700 バイトのセグメントに分割され、それぞれのセグメントに 300 バイトのオーバーヘッドがかかります。
チェックポイント	750 バイト + 200 バイト (各アクティブ作業単位ごとに)。パフォーマンス上の理由でバッファに入れられている、コミットされていない書き込みまたは読み取り用に、さらに他のデータがログに記録される可能性があります。

注:

- 1 次ログ・ファイルと 2 次ログ・ファイルの数は、キュー・マネージャーを始動するたびに変更することができます。
- ログ・ファイルのサイズは変更できません。また、このサイズは、キュー・マネージャーを作成する前に決定する必要があります。
- 1 次ログ・ファイルの数およびログ・ファイル・サイズによって、キュー・マネージャーの作成時に事前に割り振るログ・スペースの量が決まります。このスペースの編成は、小さいログ・ファイルが多数入った形より、大きいログ・ファイルが少数入った形にすることをお勧めします。
- 1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 63 以下でなければなりません。したがって、長時間実行型のトランザクションがある場合は、キュー・マネージャーが再始動リカバリーに使用できるログ・スペースの最大量が制限されることになります。メディア・リカバリーのためにキュー・マネージャーが使用するログ・スペースの量は、この制限には拘束されません。
- 循環 ロギングを使用している場合は、キュー・マネージャーは 1 次ログ・スペースを再利用します。これは、キュー・マネージャーがログに記録する必要がある量として見積もったデータ量より、キュー・マネージャーのログの方が小さくてもよいということです。ログ・ファイルが満杯になり、シー

チェックポイント機能

ケンス内の次の 1 次ログ・ファイルが使用できない場合には、キュー・マネージャーは 2 次ログ・ファイルを割り振り、限度に達するまでこれを続けます。

6. チェックポイントの時点で、1 次ログ・ファイルは再利用できるようになります。ログ・スペースの量は徐々に小さくなるので、キュー・マネージャーは、1 次ログ・スペースおよび 2 次ログ・スペースの両方を考慮に入れてチェックポイントをとります。

定義した 1 次ログ・ファイルの数が 2 次ログ・ファイルより少なかった場合は、チェックポイントがとられる前にキュー・マネージャーが 2 次ログ・ファイルを割り振る可能性があります。これで、1 次ログ・ファイルの再利用ができる状態になります。

ログの管理

時間の経過と共に、作成されたログ・レコードの一部はキュー・マネージャーの再始動には不要になり、キュー・マネージャーはログ・ファイルの空きスペースを再利用します。この活動はユーザーには気付かれなで行われます。割り振られるスペースは迅速に再利用されるため、使用されるディスク・スペースの量が減少するのは通常は分かりません。

ログ・レコードのうち、最後の完全なチェックポイントの開始以降に書き込まれたもの、およびアクティブ・トランザクションによって書き込まれたもののみが、キュー・マネージャーの再始動に必要です。チェックポイントが長い間取られなかったり、持続期間の長いトランザクションがずっと前にログ・レコードを作成した場合は、ログが満杯になる可能性があります。キュー・マネージャーは、最初の問題を回避するために、頻繁にチェックポイントを取ろうとします。

持続期間の長いトランザクションがログを満杯にしてしまうと、ログ・レコードを書き込む試みは失敗し、一部の MQI 呼び出しは MQRC_RESOURCE_PROBLEM を戻します。(すべての未完了トランザクションをコミットまたはロールバックするためにスペースが予約されているので、MQCMIT や MQBACK は失敗しません。)

キュー・マネージャーは、過度にログ・スペースを消費するトランザクションをロールバックします。トランザクションがこのようにしてロールバックされたアプリケーションは、それ以降の MQPUT や MQGET の操作を、同じトランザクションの下で同期点を指定して実行することができなくなります。この状態の同期点でメッセージの書き込みまたは読み取りを行おうとすると、MQRC_BACKED_OUT が戻されます。この場合、アプリケーションは、

MQCMIT を出す (MQRC_BACKED_OUT が戻される) か、MQBACK を出して、新しいトランザクションを開始することができます。過度にログ・スペースを消費するトランザクションがロールバックされると、そのログ・スペースが解放され、キュー・マネージャーは操作を正常に続行します。

ログが満杯になると、メッセージが出されます (AMQ7463)。さらに、持続期間の長いトランザクションがスペースの解放を妨げているためにログが満杯になった場合、メッセージ AMQ7465 が出されます。

最後に、非同期ハウスキーピング・プロセスが処理できる以上の速度でレコードがログに書き込まれていると、メッセージ AMQ7466 が出されます。このメッセージが出されたら、ログ・ファイルの数を増やすか、あるいはキュー・マネージャーが処理するデータの量を減らす必要があります。

ディスクが満杯になったときに何が起きるか

キュー・マネージャーのロギング・コンポーネントは、満杯のディスクおよび満杯のログ・ファイルに対処できます。ログ・ファイルを含んでいるディスクが満杯になると、キュー・マネージャーはメッセージ AMQ6708 を出し、エラー・レコードが取られます。

ログ・ファイルは、ログ・レコードが書き込まれるにつれて拡張されるのではなく、最大のサイズのもので作成されます。これは、MQSeriesが新しいファイルを作成しているときにのみ、ディスク・スペースが不足する可能性があることを意味します。したがって、レコードをログに書き込んでいるときにスペースが不足することはありません。MQSeries は、既存のログ・ファイルに使用可能なスペースがどれだけあるかを常に認識しており、ファイル内のスペースをそれに応じて管理します。

ログ・ファイルを含んでいるドライブが満杯になった場合には、一部のディスク・スペースを解放できることがあります。リニア・ログを使用している場合は、別のドライブまたは装置にコピーできる非活動ログ・ファイルがログ・ディレクトリーにいくつかある場合があります。それでもスペースが足りない場合には、キュー・マネージャーの構成ファイル内のログの構成が正しいかどうか検査してください。使用可能なスペースよりもログが大きくなるように、1 次または 2 次ログ・ファイルの数を減らすこともできます。既存のキュー・マネージャーでは、ログ・ファイルのサイズを変更することはできないので注意してください。キュー・マネージャーは、すべてのログ・ファイルが同じサイズであると想定しています。

ログ・ファイルを管理する

循環ロギングを使用している場合、ログ・ファイルを保持できるだけの十分なスペースを確保してください。これは、システムの構成時に行います(227ページの『LogDefaults スタンザ』および 233ページの『Log スタンザ』を参照してください)。ログが使用するディスク・スペースの量は、必要な時に作成される 2 次ファイルのスペースも含めて、ディスクの構成サイズによって制限されます。

リニア・ログを使用している場合は、データがログに記録されるにつれて、ログ・ファイルが連続的に追加され、使用されるディスク・スペースの量は時間と共に増大します。データのログの速度が速いと、ディスク・スペースは新しいログ・ファイルによって急速に使用されます。

時間の経過と共に、リニア・ログの古いログ・ファイルは、キュー・マネージャーの再始動または損傷したオブジェクトのメディア・リカバリーの実行には、必要なくなります。キュー・マネージャーは、周期的に次のような一対のメッセージを出して、どのログ・ファイルが必要かを示します。

- メッセージ AMQ7467 は、キュー・マネージャーを再始動するために必要な最も古いログ・ファイルの名前を示します。このログ・ファイルおよびそれより新しいすべてのログ・ファイルは、キュー・マネージャーの再始動時に使用できる必要があります。
- メッセージ AMQ7468 は、メディア・リカバリーを実行するために必要な最も古いログ・ファイルの名前を示します。

これらよりも古いログ・ファイルは、オンラインにする必要はありません。それらは、災害時リカバリー用にテープなどのアーカイブ・メディアにコピーして、アクティブ・ログ・ディレクトリーから除去することができます。メディア・リカバリーに必要であっても再始動には不要なログ・ファイルもすべてアーカイブ・メディアにコピーして、負荷を軽減させることができます。

必要なログ・ファイルが見つからないと、オペレーター・メッセージ AMQ6767 が出されます。そのログ・ファイル、およびそれ以降のすべてのログ・ファイルをキュー・マネージャーが使用できるようにして、操作を再試行してください。

注: メディア・リカバリーの実行中に、必要なログ・ファイルはすべてログ・ファイル・ディレクトリーの中で同時に使用可能である必要があります。必要なログ・ファイルをすべて保持するためのディスク・スペースが不足しないようにするために、リカバリーしたいオブジェクトのメディア・イメージを必ず定期的にとるようにしてください。

ログ・ファイルの位置

ログ・ファイルの位置を選択する際には、ディスク・スペースの不足のために MQSeries が新しいログをフォーマットできないと、操作に重大な影響が及ぶので注意してください。

循環ログを使用している場合は、少なくとも構成された 1 次ログ・ファイル用の十分なスペースをドライブ上に確保してください。また、ログが大きくなる場合に必要十分な少なくとも 1 つの 2 次ログ・ファイル用のスペースも残しておく必要があります。

リニア・ログを使用している場合は、十分にスペースを見込んでおく必要があります。というのは、ログが使用するスペースは、データがログに記録されるにつれて絶えず増大するからです。

ログ・ファイルは、キュー・マネージャーのデータとは別個のディスク・ドライブに置くのが理想的です。これは、パフォーマンスの面からの利点があります。また、ミラーリング配置の複数のディスク・ドライブにログ・ファイルを置くこともできます。この場合、ログを含んでいるドライブに障害が起きた場合に保護となります。ミラーリングがない場合にはおそらく、MQSeries システムの最新のバックアップを使用するしか方法がありません。

ログをリカバリーに使用する

データが損傷する原因は、いくつか考えられます。MQSeries for Compaq OpenVMS は次のような状況からリカバリーするのに役立ちます。

- データ・オブジェクトの損傷
- システムの停電
- 通信障害
- ログ・ボリュームの損傷

この節では、これらの問題からリカバリーするのにログがどのように使用されるかを調べます。

問題からリカバリーする

MQSeriesは、通信障害や停電の場合からリカバリーすることができます。さらに、場合によっては、不注意によるファイルの削除といった、その他のタイプの問題からもリカバリーすることができます。

通信障害の場合、メッセージは、受信側のアプリケーションがそれらを取り除くまでキューに残っています。メッセージが伝送中である場合は、正常に伝送

ログの使用

できるまで、メッセージは伝送キューにとどまっています。通信障害からリカバリーするためには、通常は、障害を起こしたリンクを使用しているチャンネルを再始動するだけで十分です。

停電の場合には、キュー・マネージャーが再始動されたときに、MQSeries が障害が起きたときの状態にキューを復元します。これにより、持続メッセージが失われることはなくなります。非持続メッセージは破棄されます。これらのメッセージは、MQSeries が停止した場合には残りません。

MQSeries オブジェクトが使用不能になる場合があります。たとえば、不注意で損傷した場合などです。そのようなときには、システム全体かまたはその一部をリカバリーする必要があります。どのような処置が必要かは、損傷が検出された時期、選択されているログ方法がメディア・リカバリーをサポートするかどうか、およびどのオブジェクトが損傷したかによって異なります。

メディア・リカバリー

メディア・リカバリーとは、リニア・ログだけに記録されている情報からオブジェクトを再作成することを意味します。循環ロギングでは、メディア・リカバリーはサポートされていません。たとえば、オブジェクト・ファイルが不注意で削除された場合、あるいはその他の理由で使用不能になった場合、それを再作成するのにメディア・リカバリーを使用することができます。オブジェクトのメディア・リカバリーに必要なログ内の情報のことを、メディア・イメージと呼びます。メディア・イメージは、**rcdmqimg** コマンドを使用して手動で記録するか、あるいは特定の状況では自動的に記録することができます。

メディア・イメージは、オブジェクトのイメージを含んでいる一連のログ・レコードであり、そこからオブジェクト自体を再作成できます。

オブジェクトを再作成するのに必要な最初のログ・レコードは、メディア・リカバリーとして知られています。これは、オブジェクトの最新のメディア・イメージの開始点です。各オブジェクトのメディア・リカバリー・レコードは、チェックポイント時に記録された情報の一部です。

メディア・イメージからオブジェクトを再作成する際には、最新のイメージが取られた時以降にオブジェクトに対して行われた更新を記述しているログ・ファイルを再生する必要もあります。

たとえば、持続メッセージがキューに書き込まれる前にとられたキュー・オブジェクトのイメージを持っているローカル・キューを考えてみます。そのオブ

ジェットの最新のイメージを再作成するには、イメージ自体を再生するだけでなく、キューにメッセージが書き込まれたことを記録しているログ項目を再生することも必要です。

オブジェクトが作成されるときに、作成されるログ・レコードには、そのオブジェクトを完全に再作成するのに十分な情報が入れられます。それらのレコードは、オブジェクトの最初のメディア・イメージを構成することになります。その後、メディア・イメージは、次の状態ではキュー・マネージャーによって自動的に記録されます。

- すべてのプロセス・オブジェクトおよび非ローカル・キューのイメージが、シャットダウンのたびに取られる
- キューが空になるとローカル・キュー・イメージが取られる

メディア・イメージは、355ページの『`rcdmqimg` (メディア・イメージの記録)』で説明されている `rcdmqimg` コマンドを使用して、手動で記録することもできます。

リカバリー用のメディア・イメージ

MQSeriesは、いくつかのオブジェクトが破壊されているかまたは損傷していることを検出すると、メディア・イメージからそれらを自動的にリカバリーします。特に、通常のキュー・マネージャーの始動時に損傷していることが検出されたオブジェクトに、このことが当てはまります。キュー・マネージャーの最後のシャットダウン時にいずれかのトランザクションが完了していなければ、影響を受けたキューも、始動操作を完了するために自動的にリカバリーされません。

`rcrmqobj` コマンドを使用して、他のオブジェクトを手動でリカバリーする必要があります。このコマンドは、MQSeries オブジェクトを再作成するために、ログ中のレコードを再生します。オブジェクトは、ログ中にある最新のイメージから再作成されます。また、その際に、イメージが保管されてから再作成コマンドが出されるまでの間のすべての適当なログ・イベントも再作成に使用されます。MQSeries オブジェクトが損傷した場合には、実行できる有効な処置は、オブジェクトを削除するか、あるいはこの方法でオブジェクトを再作成することです。ただし、非持続メッセージは、この方法ではリカバリーできないことに注意してください。

`rcrmqobj` コマンドの詳細については、358ページの『`rcrmqobj` (オブジェクトの再作成)』を参照してください。

オブジェクトのメディア・リカバリーを試みる際には、メディア・リカバリー・レコードを含んでいるログ・ファイル、およびそれ以降のすべてのログ・

ファイルがログ・ファイル・ディレクトリーの中であって、使用可能状態になっていなければならないということを忘れないでください。このことは重要です。必要なファイルが見つからない場合には、オペレーター・メッセージ AMQ6767 が出され、メディア・リカバリー操作は失敗します。再作成したいオブジェクトのメディア・イメージを定期的にとっていないと、オブジェクトを再作成するために必要なすべてのログ・ファイルを保持するためのディスク・スペースが不足するといった事態に陥る可能性があります。

始動時に損傷オブジェクトをリカバリーする

始動時に損傷オブジェクトを見つけた場合、キュー・マネージャーが行うアクションは、オブジェクトのタイプや、キュー・マネージャーがメディア・リカバリーをサポートするように構成されているかどうかによって異なります。

キュー・マネージャーのオブジェクトが損傷している場合、キュー・マネージャーはそのオブジェクトをリカバリーできなければ、開始できません。キュー・マネージャーがリニア・ログと共に構成されていて、メディア・リカバリーをサポートしている場合には、MQSeries は、自動的にメディア・イメージから MQSeries のオブジェクトを再作成しようとします。選択したログの方法がメディア・リカバリーをサポートしていない場合には、キュー・マネージャーのバックアップを復元するか、あるいはキュー・マネージャーを削除することができます。

キュー・マネージャーが停止したときにアクティブ状態のトランザクションがあると、それらのトランザクションの中で書き込まれたり読み取られた持続メッセージ（コミットされていないもの）のあるローカル・キューも、キュー・マネージャーを正常に開始させるためには必要です。キュー・マネージャーがメディア・リカバリーをサポートしている場合で、これらのキューのいずれかが損傷していることが分かった場合は、キュー・マネージャーは、メディア・イメージからそれらのキューを自動的に再作成しようとします。それらのキューのいずれかがリカバリーできないときは、MQSeries は開始できません。

メディア・リカバリーをサポートしないキュー・マネージャーの始動処理時に、コミットされていないメッセージを含んでいる損傷したローカル・キューが検出された場合、それらのキューには損傷オブジェクトのマークが付けられ、それに入っているコミットされていないメッセージは無視されます。その理由は、そのようなキュー・マネージャーでは、損傷オブジェクトのメディア・リカバリーを実行できず、それらを削除するという処置しか行えないためです。損傷を報告するメッセージ AMQ7472 が出されます。

その他の場合の損傷オブジェクトのリカバリー

オブジェクトのメディア・リカバリーが自動的に行われるのは、始動時のみです。その他の場合に、オブジェクトの損傷が検出されたときには、オペレーター・メッセージ AMQ7472 が出され、そのオブジェクトを使用するほとんどの操作は失敗します。キュー・マネージャーが開始された後の任意の時点でキュー・マネージャーのオブジェクトが損傷した場合には、キュー・マネージャーは優先シャットダウンを実行します。オブジェクトが損傷している場合には、それを削除することができます。あるいは、キュー・マネージャーがリニア・ログを使用している場合には、`rcrmqobj` コマンドを使用して、メディア・イメージからそのオブジェクトのリカバリーを試みてください (詳細については、358ページの『`rcrmqobj` (オブジェクトの再作成)』を参照してください)。

MQSeries ログ・ファイルの保護

MQSeries キュー・マネージャーの実行中は、ログ・ファイルを手動で削除しないようにすることが重要です。キュー・マネージャーの再始動に必要なログ・ファイルをユーザーが誤って (または何らかの悪意により) 削除した場合でも、MQSeries はエラー・メッセージを発行しないで、持続メッセージが入っているデータの処理を続けます。この場合、キュー・マネージャーは正常に終了しますが、再始動はできなくなります。そして、メッセージのメディア・リカバリーも不可能になります。

アクティブなキュー・マネージャーで使用中のログを削除する権限を持つユーザーは、キュー・マネージャーのその他の重要なリソース (許可ファイル、キュー・ファイル、オブジェクト・カタログ、MQSeries 実行ファイルなど) を削除する権限も持っています。したがって、そのようなユーザーは、おそらくは経験不足から、場合によっては意図的に、MQSeries が防ぎきれない方法で実行中または休止中のキュー・マネージャーに損傷を与える場合があります。

ユーザーに高い特権または MQM 権限 ID を与えるときには、注意してください。

バックアップと復元

定期的にキュー・マネージャーのデータのバックアップを取ることによって、ハードウェア障害によるデータの破壊からシステムを保護することができます。ただし、メッセージ・データは持続期間が短いことが多いため、バックアップを取らないようにすることもできます。

MQSeries のバックアップ

キュー・マネージャーのデータのバックアップを取るには、次の作業が必要です。

1. キュー・マネージャーが実行中でないことを確認します。

キュー・マネージャーが実行されている場合は、 **endmqm** コマンドでそれを停止してください。

注: 実行中のキュー・マネージャーのバックアップを取ろうとすると、ファイルがコピーされたときに更新が進行中であるため、バックアップが一貫していない可能性があります。

2. キュー・マネージャーのデータとログ・ファイルが置かれているディレクトリーを探し出します。

これらのディレクトリーを判別するために、構成ファイル中の情報を使用できます。この作業の詳細については、221ページの『第13章 MQSeries の構成』を参照してください。

注: ディレクトリーに現れている名前を理解するのは難しい場合があります。これは、使用しているMQSeriesのプラットフォームとの互換性を持たせるために、名前が変換されているためです。名前の変換の詳細については、28ページの『MQSeries のファイル名についての理解』を参照してください。

3. キュー・マネージャーのデータおよびログのすべてのディレクトリーを、すべてのサブディレクトリーも含めて、コピーしてください。

どのファイルもコピーし忘れないようにしてください。特に、ログ制御ファイルと構成ファイルのコピーは忘れないでください。一部のディレクトリーは空である場合がありますが、後日バックアップを復元する際にはそれらもすべて必要になるので、それらも保管するようお勧めします。

4. ファイルの所有権も存続させるようにしてください。 **BACKUP** コマンドと **/BY_OWNER** パラメーターを使用して、これを行うことができます。

MQSeries の復元

キュー・マネージャーのデータのバックアップを復元するには、次の作業が必要です。

1. キュー・マネージャーが実行中でないことを確認します。

2. キュー・マネージャーのデータとログ・ファイルが置かれているディレクトリーを探し出します。これについての情報は、構成ファイルの中に保持されています。

3. バックアップ・データをこれから入れるディレクトリーを空にしてください。
4. バックアップされたキュー・マネージャーのデータおよびログ・ファイルを、正しい場所にコピーします。

できあがったディレクトリー構造を検査して、必要なディレクトリーがすべて入っていることを確認してください。

MQSeries のディレクトリーおよびサブディレクトリーの詳細については、401ページの『付録C. ディレクトリー構造』を参照してください。

ログ・ファイルだけでなく、ログ制御ファイルもあることを確認してください。また、MQSeries が復元データを見つけるために正しい場所を調べられるように、MQSeries およびキュー・マネージャーの構成ファイルが一貫しているかどうか検査してください。

データが正しくバックアップされ復元されていれば、キュー・マネージャーは開始します。

注: キュー・マネージャーのデータおよびログ・ファイルが異なるディレクトリーに保持されているとしても、それらのディレクトリーのバックアップおよび復元を同時に行う必要があります。キュー・マネージャーのデータとログ・ファイルの経過日数が異なっていると、キュー・マネージャーは有効な状態ではなく、おそらく開始しません。たとえ開始したとしても、データはほぼ間違いなく破壊されます。

リカバリーの手順

この節では、考えられるいくつかの問題を示すと共に、それらの問題からリカバリーする方法について説明します。

ディスク・ドライブの障害

キュー・マネージャーのデータ、ログ、あるいはその両方を含んでいるディスク・ドライブに関する問題が生じる場合があります。それらの問題には、データの損失や破壊なども含まれています。これら 3 つのケースの相違点は、データのどの部分が損失や破壊を免れたか (ある場合) という点だけです。

すべての ケースで、まず損傷がないかどうかディレクトリー構造を検査し、必要ならば、そうした損傷を修理する必要があります。キュー・マネージャーのデータを失った場合には、キュー・マネージャーのディレクトリー構造が損傷している危険性があります。そのような場合には、キュー・マネージャーを再

リカバリーの手順

始動する前に、手動でディレクトリー・ツリーを再作成する必要があります。構造上の損傷がないかどうかを検査した上で、使用しているロギングのタイプに応じて、次のように実行できる事柄がいくつかあります。

- ディレクトリー構造の大きな損傷またはログの多少の損傷がある場合には、すべての古いファイルを (構成ファイル、ログ、およびキュー・マネージャーのディレクトリーを含めて) QMgrNameレベルに戻し、最新のバックアップを復元し、キュー・マネージャーの再始動を試みます。
- メディア・リカバリーが行われるリニア・ロギングの場合には、ディレクトリー構造が完全であることを確認した上で、キュー・マネージャーの再始動を試みます。キュー・マネージャーが再始動しない場合は、バックアップを復元してください。キュー・マネージャーが再始動した場合は、MQSC を使用して、他に損傷しているオブジェクトがないか検査してください。見つかった損傷オブジェクトは、**rcrmqobj** コマンドを使用してリカバリーさせます。たとえば、次のようにします。

```
rcrmqobj -m QMgrName -t * *
```

ここで、QMgrNameは、リカバリーさせられるキュー・マネージャーです。-t * * は、すべてのタイプすべてのオブジェクトがリカバリーされるということを示しています。損傷しているオブジェクトが 1 つまたは 2 つのみなら、それらのオブジェクトの名前を指定して、ここに入力できます。

注: これらのコマンドは、チャンネルには適用できません。

- メディア・リカバリーが行われるリニア・ロギングであり、ログが損傷していない場合には、キュー・マネージャーのデータのバックアップを復元しますが、既存のログ・ファイルおよびログ制御ファイルは変更しないままにすることができる場合があります。キュー・マネージャーを開始すると、ログからの変更が適用され、キュー・マネージャーは障害が起きた時点の状態に戻されます。

この方法は、次の 2 つの点に依存しています。まず、チェックポイント・ファイルが、キュー・マネージャーのデータの一部として復元できなければなりません。このファイルには、一貫性のあるキュー・マネージャーを得るために、ログ内のどれだけのデータを適用する必要があるかを決定するための情報が含まれています。

2 番目に、バックアップ時にキュー・マネージャーの開始に必要とされた最も古いログ・ファイル、およびそれ以降のすべてのログ・ファイルがあり、それらのファイルがログ・ファイル・ディレクトリーに入っている必要があります。

これが不可能なら、キュー・マネージャーのデータとログの両方のバックアップ (両方同時にとられたもの) を復元してください。

- 循環ロギング、またはメディア・リカバリーが行われないリニア・ロギングの場合には、所有している最新のバックアップからキュー・マネージャーを復元する必要があります。バックアップを復元したら、キュー・マネージャーを再始動し、損傷オブジェクトがないかどうかを上述のとおり検査します。ただし、メディア・リカバリーが行われないため、損傷オブジェクトを再作成するための別の方法を見つける必要があります。

キュー・マネージャーのオブジェクトの損傷

通常の操作時に、キュー・マネージャーのオブジェクトの損傷が報告された場合、キュー・マネージャーは優先シャットダウンを実行します。こうした状況では、使用しているロギングのタイプに応じて、リカバリー方法が 2 通りあります。

- リニア・ロギングのみの場合、損傷オブジェクトを含んでいるファイルを手動で削除し、キュー・マネージャーを再始動します。損傷オブジェクトのメディア・リカバリーは、自動的に行われます。
- 循環ロギングまたはリニア・ロギングの場合、キュー・マネージャーのデータおよびログの最新のバックアップを復元し、キュー・マネージャーを再始動します。

単独オブジェクトの損傷

通常の操作時に、単独オブジェクトの損傷が報告された場合、使用しているロギングのタイプに応じて、リカバリー方法が 2 通りあります。

- リニア・ロギングの場合、メディア・イメージからオブジェクトを再作成します。
- 循環ロギングの場合、キュー・マネージャーのデータおよびログの最新のバックアップを復元し、キュー・マネージャーを再始動します。

自動メディア・リカバリーの障害

リニア・ログによるキュー・マネージャーの始動に必要なローカル・キューが損傷している場合で、自動メディア・リカバリーが失敗した場合には、キュー・マネージャーのデータおよびログの最新のバックアップを復元し、キュー・マネージャーを再始動します。

dmpmqlog コマンドを使用したログの内容のダンプ

キュー・マネージャーのログの内容をダンプするには、**dmpmqlog** コマンドを使用します。デフォルトでは、すべてのアクティブなログ・レコードがダンプされます。つまり、このコマンドは、ログのヘッドからダンプを開始します。通常、最後に終了したチェックポイントの最初からです。

キュー・マネージャーが実行されていない場合にのみログをダンプできます。終了中にキュー・マネージャーはチェックポイントを取るため、通常ログのアクティブ部分には、ログ・レコードの小さい数が入ります。ただし、次のオプションのいずれかを使用すると、**dmpmqlog** コマンドによりログ・レコードをより多くダンプするよう指示され、ダンプの開始位置を変更することができます。

- 最も簡単なオプションは、ログの ベース からダンプを開始するものです。ログのベースとは、ログのヘッドを含むログ・ファイルの中の最初のログ・レコードです。この場合、ダンプされるデータ追加量は、ログのヘッドがログ・ファイルの中で位置付けされる場所によって異なります。ログ・ファイルの最初に近い場合には、少量の追加データしかダンプされません。ログ・ファイルの終わりに近い場合には、極めて多量のデータがダンプされます。
- もう 1 つのオプションは、ダンプの開始位置を個々のログ・レコードとして指定できます。各ログ・レコードは、固有の *ログ順序番号 (LSN)* により識別されます。循環ログの場合、この開始ログ・レコードをログのベースの前に置くことはできません。この制限は、リニア・ログには適用されません。非アクティブ・ログ・ファイルは、コマンドを実行する前に、リカバリーさせる必要がある場合があります。このオプションでは、開始位置として有効な LSN を指定しなければなりません。これは、直前の **dmpmqlog** 出力から行う必要があります。

たとえば、リニア・ロギングでは、最後の **dmpmqlog** 出力から `nextlsn` を指定します。すると、次の LSN が Log File Header に表示され、次のログ・レコードの LSN が書き込まれることを示します。したがって、ログが最後にダンプされた後に、書き込まれるログ・レコードをすべてフォーマットするため、開始位置としてこれを使用することができます。

- 第 3 のオプションはリニア・ログ専用です。指定のログ・ファイル・エクステンツからログ・レコードのフォーマット化を開始するようダンパーに指示できます。この場合、ログ・ダンパーにより、このログ・ファイル、およびアクティブ・ログ・ファイルと同じディレクトリーにある次のファイルが検索されます。このオプションは循環ログには適用されません。それは、ログ・ダンパーがログのベース前のログ・レコードにアクセスできないためです。

dmpmqlog コマンドからの出力は、ログ・ファイル・ヘッダー および一連のフォーマット済みログ・レコードです。キュー・マネージャーは、いくつかのログ・レコードを使用して、データに対する変更を記録します。

フォーマットされる一部の情報は、内部的に使用されるものです。次のリストには、最も役立つログ・レコードを示しています。

Log File Header

各ログには単一のログ・ファイル・ヘッダーがあり、**dmpmqlog** コマンドにより必ず最初にフォーマットされます。これには、次のようなフィールドがあります。

<i>logactive</i>	1 次ログ・エクステントの数
<i>loginactive</i>	2 次ログ・エクステントの数
<i>logsize</i>	エクステントあたり 4 KB ページ数
<i>baselsn</i>	ログのヘッドを含むログ・エクステントの中の最初の LSN
<i>nextlsn</i>	書き込まれる次のログ・レコードの LSN
<i>headlsn</i>	ログのヘッドでのログ・レコードの LSN
<i>tailsn</i>	ログのテール位置を識別する LSN
<i>hflag1</i>	ログが CIRCULAR であるか LOG RETAIN (リニア) であるかどうかを識別する
<i>HeadExtentID</i>	ログのヘッドを含むログ・エクステント

Log Record Header

ログ内の各ログ・レコードには以下の情報を含む固定のヘッダーがあります。

<i>LSN</i>	ログの順序番号
<i>LogRecdType</i>	ログ・レコードのタイプ
<i>XTranid</i>	このログ・レコード (ある場合) と関連したトランザクション ID
	MQI の <i>TranType</i> は MQ トランザクションのみを示します。XA の <i>TranType</i> は、他のリソース・マネージャーに関連するトランザクションを示します。同じ作業単位内に含まれる更新には同じ <i>XTranid</i> があります。
<i>QueueName</i>	このログ・レコード (ある場合) と関連したキュー
<i>Qid</i>	キュー用の固有の内部 ID

dmpmqlog の使用

PrevLSN 同じトランザクション内の前のログ・レコード (ある場合) の LSN

Start Queue Manager

これは、キュー・マネージャーを開始した記録を取ります。

StartDate キュー・マネージャーを開始した日付

StartTime キュー・マネージャーを開始した時刻

Stop Queue Manager

これは、キュー・マネージャーを停止した記録を取ります。

StopDate キュー・マネージャーを停止した日付

StopTime キュー・マネージャーを停止した時刻

ForceFlag 使用した終了のタイプ

Start Checkpoint

これはキュー・マネージャーのチェックポイントの開始を示します。

End Checkpoint

これはキュー・マネージャーのチェックポイントの終了を示します。

ChkPtLSN このチェックポイントを開始したログ・レコードの LSN

Put Message

これはキューに書き込んだ持続メッセージの記録を取ります。メッセージが同期点で書き込まれた場合、ログ・レコード・ヘッダーには非空文字 *XTranid* が入ります。レコードの残りには以下のものが入ります。

SpcIndex キュー上のメッセージの ID。これは、キューからこのメッセージを得るのに使用した該当する MQGET を突き合わせる場合に使用できます。この場合、同じ *QueueName* および *SpcIndex* が含まれている次の *Get Message* ログ・レコードを検出できます。この時点で、キューにメッセージを続けて書き込むために *SpcIndex ID* を再使用できます。

Data このログ・レコードに 16 進ダンプで含まれているものは、メッセージ記述子 (目印 MD) およびメッセージ・データ自体が後に続く内部データです。

Put Part

単一のログ・レコードとしては長すぎる持続メッセージは、複数の *Put Part* ログ・レコードが後に続く単一の *Put Message* レコードとして記録されます。

Data 前のログ・レコードが終了した場所のメッセージ・データが続きます。

Get Message

持続メッセージの読み取りのみが記録されます。メッセージが同期点で読み取られた場合、ログ・レコード・ヘッダーには非空文字 *XTranid* が入ります。レコードの残りには以下のものが入ります。

SpcIndex キューから得られたメッセージを識別します。同じ *QueueName* および *SpcIndex* を含む最新の *Put Message* ログ・レコードは得られたメッセージを識別します。

QPriority キューから得られたメッセージの優先順位。

Start Transaction

新規トランザクションの開始を示します。MQI の *TranType* は MQ トランザクションのみを示します。XA の *TranType* は、他のリソース・マネージャーに関連するトランザクションを示します。このトランザクションが行ったすべての更新には同じ *XTranid* があります。

Prepare Transaction

キュー・マネージャーは、指定した *XTranid* と関連した更新をコミットする準備が整っていることを示します。このログ・レコードは、他のリソース・マネージャーに関連する 2 フェーズ・コミットの一部として書き込まれます。

Commit Transaction

トランザクションが行ったすべての更新をキュー・マネージャーがコミットしたことを示します。

Rollback Transaction

このログ・レコードは、キュー・マネージャーがトランザクションをロールバックしようとしていることを表します。

End Transaction

このログ・レコードはロールバック・トランザクションの終了を表します。

Transaction Table

このレコードは同期点の最中に書き込まれます。これは持続更新を行ったトランザクションそれぞれの状態を記録します。トランザクションごとに、次の情報が記録されます。

XTranid トランザクション ID

FirstLSN トランザクションと関連した最初のログ・レコードの LSN

LastLSN トランザクションと関連した最後のログ・レコードの LSN

Transaction Participants

このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。これは、トランザクションに関連する外部リソース・マネージャーを記録します。関連するプログラムごとに次の情報が記録されます。

RMName リソース・マネージャーの名前

RMId リソース・マネージャー ID。これは後続の *Transaction Prepared* ログ・レコードにもログインします。このレコードはリソース・マネージャーが関連しているグローバル・トランザクションを記録するものです。

SwitchFile このリソース・マネージャー用のスイッチ・ロード・ファイル

XAOpenString このリソース・マネージャー用の XA オープン・ストリング

XACloseString このリソース・マネージャー用の XA オープン・ストリング

Transaction Prepared

このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。指定したグローバル・トランザクションが正常に作成されたことを示します。関連するリソース・マネージャーのそれぞれにコミットが指示されます。準備済みの各リソース・マネージャーの *RMId* は、ログ・レコードに記録されます。キュー・マネージャー自体がトランザクションに関連している場合、*RMID* がゼロである *Participant Entry* が示されます。

Transaction Forget

このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。コミットの決定が関連する各プログラムに出された場合に *Transaction Prepared* ログ・レコードの後に続くものです。

Purge Queue

これは、キュー上のすべてのメッセージが、たとえば `RUNMQSC CLEAR` コマンドを使用して除去されたという事実の記録を取ります。

Queue Attributes

これは、キューの属性の初期化または変更の記録を取ります。

Create Object

MQSeries オブジェクトの作成記録を取ります。

ObjName 作成したオブジェクトの名前

UserId 作成を実行するユーザー ID

Delete Object

MQSeries オブジェクトの削除記録を取ります。

ObjName 削除したオブジェクトの名前

200ページの図18 は、**dmpmqlog** コマンドからの出力例を示したものです。特定のログ・レコードの LSN で開始したダンプは、次のコマンドを使用して実行されます。

```
dmpmqlog -m "testqm" -s 0:0:0:44162
```

dmpmqlog の使用

```
AMQ7701: DMPMQLOG command is starting.
LOG FILE HEADER
*****

counter1 . . . . : 23                counter2 . . . . : 23
FormatVersion . . : 2                logtype . . . . . : 10
logactive . . . . : 3                loginactive . . . : 2
logsize . . . . . : 1024             pages
baselsn . . . . . : <0:0:0:0>
nextlsn . . . . . : <0:0:0:60864>
lowtranlsn . . . . : <0:0:0:0>
minbufflsn . . . . : <0:0:0:58120>
headlsn . . . . . : <0:0:0:58120>
taillsn . . . . . : <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
firstDlteFile . . . : 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: lreclsize 212, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM Start Checkpoint (1025)
Eyecatcher . . . . : ALRH                Version . . . . . : 1
LogRecdLen . . . . : 192                LogRecdOwnc . . . : 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
```

図 18. dmpmqlog 出力の例 (1/16)

```

LOG RECORD - LSN <0:0:0:44374>
*****

HLG Header: lrecsize 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: lrecsize 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 . . :

```

図 18. dmpmqlog 出力の例 (2/16)

dmpmqlog の使用

```
LOG RECORD - LSN <0:0:0:46448>
*****

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 . . . . : <0:0:0:44162>
OldestLSN . . . . : <0:0:0:0>
MediaLSN . . . . : <0:0:0:0>

LOG RECORD - LSN <0:0:0:52262>
*****

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 1}
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . . : 1
SoftLogLimit . . . : 10000
```

図 18. dmpmqlog 出力の例 (3/16)

```

LOG RECORD - LSN <0:0:52482>
*****

HLG Header: lrecsize 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
SpcIndex . . . . : 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 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  .....
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 22 00 00 00 00 00 00 00 00  .....
00144: 00 00 00 00 00 00 00 C9 2C 85 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 20 74 65 73 74  .....
00256: 71 6D 20 20 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 20 20  .....
00288: 20 20 20 20 20 20 20 20 20 20 20 20 20 73 62 6F 6C  .....
00304: 61 6D 20 20 20 20 20 20 04 37 34 38 30 00 00 00  am .....sbol
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....7480...

```

図 18. dmpmqlog 出力の例 (4/16)

dmpmqlog の使用

```
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
LOG RECORD - LSN <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
SpcIndex . . . . : 1
PrevLink.Locn . . : 36 PrevLink.Length : 8
PrevDataLink . . : {High 0, Low 2048}
Data.Locn . . . . : 2048 Data.Length . . . : 490
Data . . . . . :
```

図 18. dmpmqlog 出力の例 (5/16)


```

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 3€-&#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 00 01 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 73 62 6F 6C ..... sbol
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 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 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 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

```

図 18. dmpmqlog 出力の例 (6/16)

dmpmqlog の使用

```
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>
*****
```

図 18. dmpmqlog 出力の例 (7/16)

```
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
SpcIndex . . . . : 1                      QPriority . . . . : 0
PrevLink.Locn . . : 36                    PrevLink.Length : 8
PrevDataLink . . . : {High 4294967295, Low 4294967295}
```

図 18. dmpmqlog 出力の例 (8/16)

dmpmqlog の使用

```
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
SpcIndex . . . . : 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>
*****
```

図 18. dmpmqlog 出力の例 (9/16)

```
HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Start Transaction (769)
Eyecatcher . . . : ALRH                               Version . . . . : 1
LogRecdLen . . . : 200                               LogRecdOwnc . . : 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
```

図 18. dmpmqlog 出力の例 (10/16)

dmpmqlog の使用

```

LOG RECORD - LSN <0:0:0:55544>
*****

HLG Header: lreclsize 738, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Put Message (257)
Eyecatcher . . . : ALRH                               Version . . . . . : 1
LogRecdLen . . . : 718                               LogRecdOwnc . . . : 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
Spcln . . . . . : 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   3€-&#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
00256: 71 6D 20 20 20 20 20 20 20 20 20 20 20 20 20 20   qm          test
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
00304: 61 6D 20 20 20 20 20 04 37 34 38 30 00 00 00 00   am          sbol
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .7480...
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
00384: 70 69 20 20 20 20 20 20 20 20 20 20 20 20 20 20   pi          utza
00400: 20 20 20 20 20 20 20 31 39 39 37 30 35 31 39
00416: 31 30 34 34 35 38 37 32 20 20 20 20 00 00 00 00   pi          19970519
00432: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   10445872 ....
00448: 41 6E 6F 74 68 65 72 20 70 65 72 73 69 73 74 65   .....
00464: 6E 74 20 6D 65 73 73 61 67 65 20 70 75 74 20 75   Another persiste
00480: 6E 64 65 72 20 73 79 6E 63 70 6F 69 6E 74         nt message put u
                                                nder syncpoint

```

図 18. dmpmqlog 出力の例 (11/16)

```

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                               LogRecdOwnc . . . : 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                               LogRecdOwnc . . . : 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

```

図 18. dmpmqlog 出力の例 (12/16)

dmpmqlog の使用

```
LOG RECORD - LSN <0:0:0:57206>
*****

HLG Header: lreclsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Commit Transaction (774)
Eyecatcher . . . : ALRH                               Version . . . . : 1
LogRecdLen . . . : 196                               LogRecdOwnc . . . : 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: lreclsize 224, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : Transaction Forget (1539)
Eyecatcher . . . : ALRH                               Version . . . . : 1
LogRecdLen . . . : 204                               LogRecdOwnc . . . : 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
```

図 18. dmpmqlog 出力の例 (13/16)


```

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                      LogRecdOwnc . . . : 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                      LogRecdOwnc . . . : 768 (ATM)
XTranid . . . . : TranType: NULL
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . : 1
TranCount . . . : 0

```

図 18. dmpmqlog 出力の例 (14/16)

dmpmqlog の使用

```
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                               LogRecdOwnc . . . : 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                               LogRecdOwnc . . . : 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>
```

図 18. dmpmqlog 出力の例 (15/16)

```

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.

```

図 18. *dmpmqlog* 出力の例 (16/16)

200ページの図18 の注:

1. *Log File Header* の *headlsn* の値は <0:0:0:58120> です。これは、別の開始 LSN を要求しなかった場合にダンプが開始する場所です。
2. *nextlsn* は <0:0:0:60864> で、キュー・マネージャーを次に再始動する場合にプログラムが書き込む最初のログ・レコードの LSN です。
3. *HeadExtentID* は 1 で、ログのヘッドが現在ログ・ファイル S0000001.LOG に常駐していることを示します。
4. フォーマットされる最初のログ・レコードは *Start Checkpoint* ログ・レコードです。 <0:0:0:46448> の *End CheckPoint* まで、チェックポイントはログ・レコード数を測ります。
5. チェックポイント中に保管されるレコードの 1 つは、 <0:0:0:44594> の *Transaction Participants* ログ・レコードです。これは、キュー・マネージャーが調整したグローバル・トランザクションに関連するリソース・マネージャーの詳細を示すものです。
6. <0:0:0:52262> の *Start Transaction* ログ・レコードは、トランザクションの開始を表しています。 *XTranid* は MQI の *TranType* を示しており、MQ 更新のみを含むローカル・トランザクションであることを表しています。
7. 次のログ・レコードは *Put Message* ログ・レコードで、トランザクションを開始した同期点の下の持続 MQPUT を記録します。 MQPUT がキュー *Queue1* に対して行われ、メッセージ・データが Persistent message put

dmpmqlog の使用

under syncpoint として記録されます。このメッセージに 1 の *SpIndex* が割り振られますが、これはこのメッセージの後の MQGET と一致しません。

8. LSN <0:0:0:53458> の次のログ・レコードも *Put Message* レコードです。この持続メッセージは別のキュー *Queue2* に書き込まれますが、*XTranid* が *NULL* であるため、同期点の下では書き込みが行われませんでした。*SpIndex* も 1 で、この特定のキューに固有の ID です。
9. LSN <0:0:0:54192> の次のログ・レコードが同期点の下に書き込まれたメッセージをコミットします。
10. ログ・レコード <0:0:0:54408> および <0:0:0:54628> では、キュー *Queue1* 用の同期点の下で、MQGET により新規トランザクションが開始されます。*Get Message* ログ・レコードの *SpIndex* は 1 であり、<0:0:0:52262> の *Queue1* に書き込まれたメッセージと同じであることを示します。
11. 次のログ・レコードには、他の *Put Message* ログ・レコードにより *Queue2* に書き込まれたメッセージが入ります。
12. <0:0:0:55108> の *Commit Transaction* が示したように、同期点の下の MQGET がコミットされました。
13. 最終的には、MQBEGIN を使用して、<0:0:0:55324> の *Start Transaction* ログ・レコードにあるグローバル・トランザクションを開始します。このログ・レコードの *XTranid* には、XA の *TranType* があります。
14. 次の *Put Message* が *Queue2* に書き込まれた持続メッセージを記録します。これは、前のログ・レコードと同じ *XTranid* を共用します。
15. *Transaction Prepared* ログ・レコードがこの *Xtranid* に書き込まれる場合は、トランザクションを全体的にコミットしなければなりません。このようなログ・レコードがない場合、トランザクションがロールバックされたと見なすことができます。この場合、*Transaction Prepared* ログ・レコードが <0:0:0:56498> で検出されます。これは、*RMId* がゼロである関連するプログラムとしてキュー・マネージャー自体を記録します。さらに、2 つの関連するプログラムがあり、1 と 2 の *RMIds* を、前の *Transaction Participants* ログ・レコードと一致させることができます。
16. コミット中に、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントは、関連するプログラムから個々の応答を記録しません。このログは、キュー・マネージャーの更新がコミットされたかどうかを示すだけです。<0:0:0:57206> の *Commit Transaction* ログ・レコードは、メッセージが実際に *Queue2* にコミットされることを示しています。

17. <0:0:0:57440> の *Transaction Forget* ログ・レコードは、コミットの決定が、他の 2 つのリソース・マネージャーに対しても出されることを示しています。2 つのリソース・マネージャーが更新のコミットに失敗する場合は、キュー・マネージャーのエラー・ログで診断が実行されます。

dmpmqlog の使用

第12章 名前サービスの使用

名前サービスはインストール可能なサービスであり、あるキュー・マネージャーに接続されたアプリケーションが、ローカル・キューであると判断するものをオープンできるようにします。これらのキューは、別のキュー・マネージャー上 (別のマシン上) で定義されている実際のキューで、SCOPE 属性がCELL に設定されています。

アプリケーションは、この方法でオープンされたキューに対して、リモート・キューに許可されたすべての操作を実行できます。提供されているインプリメンテーションでは、DCE (分散コンピューティング環境) が使用されますが、DCE を使用しない独自のコンポーネントを自由に作成することができます。

提供されている名前サービス・コンポーネントを使用するには、名前サービスとそのインストール済みのコンポーネントをキュー・マネージャーに定義する必要があります。これは、適切なスタanzasをキュー・マネージャーの構成ファイル (qm.ini) ファイルに挿入することによって行います。詳細については、*MQSeries プログラム式システム管理* を参照してください。いくつかの DCE 構成を行う必要もあります。

さまざまなキュー・マネージャー上のキューを共用するために DCE を使用する

キュー・マネージャーが分散コンピューティング環境 (DCE) セル内にある場合は、キューを共用するようにキュー・マネージャーを構成することができます。このようにすると、アプリケーションはあるキュー・マネージャーに接続して、別のノード上の別のキュー・マネージャーにあるキューをオープンできます。アプリケーションにとっては、これは透過的です。つまり、アプリケーションは、キューが実際には別のキュー・マネージャーにあることを識別しません。(通常、キュー・マネージャーは、キューがキュー・マネージャー上に存在しない場合は、ローカル・アプリケーションからのオープン要求を拒否します。)

共用キューのための構成タスク

この節では、DCE セル内のノードにあるキュー・マネージャー上の共用キューをセットアップする方法について説明します。

各キュー・マネージャーについて、次のようにします。

キューの共用

1. 必要な名前サービス・スタンザをキュー・マネージャーの構成ファイルに追加することによって、名前サービスを構成します。このスタンザの内容については、*MQSeries プログラム式システム管理* で説明されています。名前サービスを起動するには、キュー・マネージャーを再始動する必要があります。
2. キュー・マネージャーが実行中の場合は、**endmqm** コマンドを使用してキュー・マネージャーを停止します。
3. **strmqm** コマンドを使用して、キュー・マネージャーを再始動します。
4. キュー・マネージャー間のメッセージング用にチャンネルをセットアップします。87ページの『リモート管理のためにチャンネルおよび伝送キューを作成する』を参照してください。

共用するすべてのキューについて、SCOPE 属性を CELL に指定してください。次のような MQSC コマンドを使用してください。

```
DEFINE QLOCAL(GREY.PUBLIC.QUEUE) SCOPE(CELL)
```

または、

```
ALTER QLOCAL(PINK.LOCAL.QUEUE) SCOPE(CELL)
```

作成または更新されたキューは、DCE セル内のノードにあるキュー・マネージャーに属している必要があります。

DCE 構成

提供されている名前サービス・コンポーネントを使用するには、OSF 分散コンピューティング環境 (DCE) をインストールしておく必要があります。このサービスによって、あるキュー・マネージャーに接続しているアプリケーションが、同じ DCE セル内の別のキュー・マネージャーに属しているキューをオープンできるようになります。

提供されている名前サービスを実行できるようにする DCL シェル・スクリプトの例は、`mqc_examples:dcesetu.com` にあります。

第13章 MQSeries の構成

この章では、個々のキュー・マネージャーの動作、つまりノードの動作を、使用しているシステムの要件に合わせて変更する方法について説明します。

MQSeries 構成を変更するには、MQSeries の動作を決定する一連の構成属性 (またはパラメーター) に指定されている値を変更します。

この構成情報を変更する方法、および MQSeries が変更を保管する場所は、プラットフォームによって異なります。MQSeries for Compaq OpenVMS のユーザーは、**MQSeries 構成ファイル**を編集することにより、構成情報を変更します。

この章の内容は次のとおりです。

- 224ページの『MQSeries 構成情報を変更するための属性』では、MQSeries 構成情報を変更するために使用できる属性について説明します。
- 231ページの『キュー・マネージャー構成情報の変更』では、キュー・マネージャー構成情報を変更するために使用できる属性について説明します。
- 242ページの『mqs.ini および qm.ini ファイルの例』では、MQSeries for Compaq OpenVMS の mqs.ini および qm.ini ファイルの例を提供します。

MQSeries 構成ファイル

MQSeries for Compaq OpenVMS のユーザーは、次のファイル内で MQSeries 構成属性を変更します。

- ノード上の MQSeries 全体に変更するには、MQSeries 構成ファイル (mqs.ini)。ノードあたり mqs.ini ファイルは1つです。
- 特定のキュー・マネージャーについて変更を加えるには、キュー・マネージャー構成ファイル (qm.ini)。qm.ini ファイルは、ノード上のそれぞれのキュー・マネージャーごとに1つずつあります。

構成ファイル (**スタンザ**・ファイルまたは **.ini** ファイルとも呼ばれます) には、1つまたは複数のスタンザが含まれています。スタンザは、単にファイル内の行のグループで、全体で1つの機能を表しているか、ログ機能、チャネル機能、およびインストール可能サービスなどのシステムの一部を定義するものです。

構成ファイル

構成ファイルに対する変更の効力が発生するのは、次回にキュー・マネージャーを始動したときです。

構成ファイルを編集する

構成ファイルを編集する前にバックアップをとって、必要な場合にコピーを使って復元できるようにしておいてください。

構成ファイルの編集方法には次の 2 つがあります。

- 自動。これには、ノード上のキュー・マネージャーの構成を変更するコマンドを使用します。
- 手動。これには標準テキスト・エディターを使用します。

MQSeries 構成ファイル内のデフォルト値は、インストール後に変更することができます。

構成ファイル属性のどれかに誤った値を設定した場合、その値は無視され、問題を示すオペレーター・メッセージが表示されます (その結果、その属性をまったく指定しなかった場合と同じになります)。

新規のキュー・マネージャーを作成するときは、必ず次の操作をしてください。

- MQSeries 構成ファイルのバックアップをとる
- 新しいキュー・マネージャー構成ファイルのバックアップをとる

どのようなときに構成ファイルの編集が必要か

構成ファイルを編集する必要があるのは、たとえば次のような場合です。

- 構成ファイルが失われた場合。バックアップがある場合は、それから回復します。
- 1 つまたは複数のキュー・マネージャーを新しいディレクトリーに移す必要がある場合。
- デフォルトのキュー・マネージャーを変更する必要がある場合。これは、既存のキュー・マネージャーを誤って削除してしまった場合などに必要です。
- S.E. から編集するよう勧められた場合。

構成ファイルの優先順位

構成ファイルの属性値は、次の優先順位に従って設定されます。

- コマンド行に入力されたパラメーターは、構成ファイル内で定義されている値より優先されます。

- `qm.ini` ファイル内で定義されている値は、`mqs.ini` ファイル内で定義されている値より優先されます。

構成ファイルへの変更のインプリメント

構成ファイルを編集する場合は、キュー・マネージャーにより変更が即時にインプリメントされることはありません。MQSeries 構成ファイルになされる変更は、MQSeries の始動時にのみインプリメントされます。キュー・マネージャー構成ファイルになされる変更は、キュー・マネージャーの始動時にインプリメントされます。変更の際にキュー・マネージャーが実行している場合には、この変更をシステムが認識するように、このキュー・マネージャーを停止してから再始動しなければなりません。

MQSeries 構成ファイル、`mqs.ini`

MQSeries 構成ファイル `mqs.ini` は、ノード上のすべてのキュー・マネージャーに関連する情報を含んでいます。これは、インストール時に自動的に作成されます。特に、`mqs.ini` ファイルは、各キュー・マネージャーに関連したデータを見つけるために使用されます。

`mqs.ini` ファイルは、デフォルトでは、データ・ディレクトリー **MQS_ROOT:[MQM]** に保管されます。

`mqs.ini` ファイルには次のものが含まれています。

- キュー・マネージャーの名前
- デフォルト・キュー・マネージャーの名前
- 各キュー・マネージャーに関連したファイルの位置

`mqs.ini` の内容についての詳細は、224ページの『MQSeries 構成情報を変更するための属性』を参照してください。

キュー・マネージャー構成ファイル `qm.ini`

キュー・マネージャー構成ファイル `qm.ini` は、特定のキュー・マネージャーに関係のある情報を含んでいます。各キュー・マネージャーごとに、1つのキュー・マネージャー構成ファイルがあります。`qm.ini` ファイルは、関連したキュー・マネージャーが作成されるときに、自動的に作成されます。

`qm.ini` ファイルは、キュー・マネージャーが占有するディレクトリー・ツリーのルートに保持されます。

たとえば、MQSeries for Compaq OpenVMS では、`QMNAME` という名前のキュー・マネージャーの構成ファイルのパスと名前は、次のとおりです。

```
MQS_ROOT:[MQM.QMGRS.QMNAME]QM.INI
```

キュー・マネージャー構成ファイル。

注: キュー・マネージャーの名前の長さは 48 文字までです。ただし、この名前が有効である、または固有であるかは保証されません。このため、キュー・マネージャー名に基づいてディレクトリー名が生成されます。このプロセスは**名前変換**と呼ばれています。その説明については、28ページの『MQSeries のファイル名についての理解』を参照してください。

qm.ini についての詳細は、231ページの『キュー・マネージャー構成情報の変更』を参照してください。

MQSeries 構成情報を変更するための属性

mqc.ini には、以下の属性のグループが表示されます。

- AllQueueManagers スタンザ
- 226ページの『ClientExitPath スタンザ』
- 226ページの『DefaultQueueManager スタンザ』
- 226ページの『ExitProperties スタンザ』
- 227ページの『LogDefaults スタンザ』
- 230ページの『QueueManager スタンザ』

mqc.ini のサンプルを 242ページの『mqc.ini および qm.ini ファイルの例』に示してあります。

AllQueueManagers スタンザ

AllQueueManagers スタンザには次のものを指定できます。

- キュー・マネージャーに関連したファイルが保管されている qmgrs ディレクトリーへのパス
- EBCDIC 形式のデータを ASCII 形式に変換するメソッド

DefaultPrefix=directory_name

この属性には、キュー・マネージャー・データが保管されている qmgrs ディレクトリーへのパスを指定します。

キュー・マネージャーのデフォルトの接頭部を変更する場合は、インストール時に作成されているディレクトリー構造を複製する必要があります (401ページの『付録C. ディレクトリー構造』を参照)。

特に、qmgrs 構造体の作成が必要になります。デフォルト接頭部を変更する前に MQSeries を停止してください。MQSeries の再始動は、構造が新しい位置に移され、デフォルト接頭部が変更された後に行ってください。

デフォルト接頭部を変更する代わりに、論理 MQSPREFIX を使用して、**crtmqm** コマンドの DefaultPrefix を指定変更することもできます。

ConvEBCDICNewline=NL_TO_LF|TABLE|ISO

EBCDIC コード・ページには、ASCII コード・ページではサポートされない改行 (NL) 文字が含まれています。ただし、ISO 仕様の ASCII には、この改行文字と同等の文字が含まれているものもいくつかあります。

ConvEBCDICNewline 属性には、EBCDIC の NL 文字を ASCII 形式に変換するときに MQSeries が使用するメソッドを指定します。

NL_TO_LF

EBCDIC から ASCII へのすべての変換について、EBCDIC の NL 文字 (X'15') が ASCII の改行 (LF) 文字 (X'0A') に変換されるようにするには、NL_TO_LF を指定します。

NL_TO_LF はデフォルトです。

TABLE

ご使用のプラットフォームで EBCDIC から ASCII への変換に使用している変換テーブルに従って、EBCDIC の NL 文字が変換されるようにするには、TABLE を使用します。

このタイプの変換の結果は、プラットフォームと言語によって異なる場合があるので、注意が必要です。同じプラットフォーム上でも、使用する CCSID が異なれば動作が違ってくる場合があります。

ISO

次の場合は ISO を指定します。

- TABLE メソッドを使用して ISO CCSID を変換したい場合
- NL_TO_LF メソッドを使用してその他のすべての CCSID を変換したい場合

使用できる ISO CCSID は、表7 に示されています。

表7. 使用できる ISO CCSID のリスト

CCSID	コード・セット
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

MQSeries 構成ファイルの変更

ASCII CCSID が ISO サブセットの 1 つではない場合は、デフォルトにより、ConvEBCDICNewline は NL_TO_LF に設定されます。

データ変換の詳細については、MQSeries アプリケーション・プログラミング・ガイド または 99ページの『データ変換』を参照してください。

ClientExitPath スタンザ

ClientExitPath スタンザには、クライアント上のチャンネル出口の位置のデフォルト・パスを指定します。

ExitsDefaultPath=*defaultprefix*

ExitsDefaultPath 属性には、プラットフォームのデフォルト接頭部を指定します。

DefaultQueueManager スタンザ

DefaultQueueManager スタンザには、ノードのデフォルト・キュー・マネージャーを指定します。

Name=*default_queue_manager*

デフォルト・キュー・マネージャーは、キュー・マネージャー名が明示的に指定されていないすべてのコマンドを処理します。

DefaultQueueManager 属性は、新規のデフォルト・キュー・マネージャーを作成すると、自動的に更新されます。誤って新規のデフォルト・キュー・マネージャー作成した場合、それを元のプログラムに戻したいときは、DefaultQueueManager 属性を手動で変更する必要があります。

ExitProperties スタンザ

ExitProperties スタンザには、キュー・マネージャーの出口プログラムが使用する構成オプションを指定します。

CLWLMode=SAFE|FAST

クラスター作業負荷出口 CLWL では、MQAPI 呼び出し (MQOPEN または MQPUT など) に応えて、クラスター内のどのクラスター・キューをオープンするかを指定できます。CLWL 出口は、CLWLMode 属性に指定する値に応じて、FAST モードまたは SAFE モードで実行されます。

CLWLMode 属性を指定しなかった場合は、クラスター作業負荷出口は SAFE モードで実行されます。

SAFE

SAFE オプションは、キュー・マネージャーから独立したプロセスで CLWL 出口を実行することを指定します。これはデフォルトです。

SAFE モードでの実行中にユーザー作成の CLWL に問題が起きた場合は、次のような結果が生じます。

- CLWL サーバー・プロセス (amqzlw0) は失敗します。
- キュー・マネージャーは CLWL サーバー・プロセスを再始動します。
- エラーがエラー・ログに記録されます。進行中の MQAPI 呼び出しがある場合は、不成功を示す戻りコードが戻されます。

キュー・マネージャーの健全性は維持されます。

注: CLWL 出口を独立したプロセスで実行すると、オーバーヘッドが発生し、パフォーマンスが影響を受けることがあります。

FAST

クラスター出口をキュー・マネージャー内でインラインで実行したい場合は、FAST を指定します。

このオプションを指定すると、SAFE モードで実行した場合のオーバーヘッドを回避できるのでパフォーマンスは向上しますが、代わりにキュー・マネージャーの健全性が保証されないことになります。したがって、CLWL 出口を FAST モードで実行するのは、CLWL 出口にまったく問題がないという確信があり、パフォーマンス・オーバーヘッドが特に重要な要素である場合のみにしてください。

CLWL 出口を FAST モードで実行しているときに問題が起きると、キュー・マネージャーは失敗し、キュー・マネージャーの健全性が損なわれる恐れがあります。

LogDefaults スタンザ

LogDefaults スタンザには、ノードのデフォルトのログ属性を指定します。ログ属性は、ユーザーがキュー・マネージャーを作成するときにデフォルト値として使用されますが、**crtmqm** コマンドにログ属性を指定することにより、指定変更することができます。このコマンドの詳細については、320ページの『crtmqm (キュー・マネージャーの作成)』を参照してください。

キュー・マネージャーが作成されたあとは、そのキュー・マネージャーのログ属性は、qm.ini ファイル内のログ・スタンザから読み取られます。

AllQueueManagers スタンザの *DefaultPrefix* 属性と、LogDefaults スタンザの中の *LogPath* 属性を使用すると、キュー・マネージャーとそのログを別々の物理ドライブに入れておくことができます。デフォルトではこの 2 つは同じドライブに入りますが、別々のドライブを使用する方をお勧めします。

MQSeries 構成ファイルの変更

ログ・サイズを計算する方法については、180ページの『ログのサイズの計算』を参照してください。

注: 次のパラメーター・リストに示す制限は、MQSeries により設定される制限です。オペレーティング・システムの制限により、設定可能なログ・サイズが削減されることがあります。

LogPrimaryFiles=32-62

1 次ログ・ファイルは、将来の利用のために作成時に割り振られるログ・ファイルです。

1 次ログ・ファイルの最小数は 2 で、最大数は 62 です。デフォルトは 3 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 63 を超えてはならず、3 より小さくてもなりません。

キュー・マネージャーが作成される際には、**crtmqm** コマンドの **-lp** パラメーターによりこの値が上書きされます。

LogSecondaryFiles=21-61

2 次ログ・ファイルは、1 次ファイルが足りなくなったときに割り振られるログ・ファイルです。

2 次ログ・ファイルの最小数は 1 であり、最大数は 61 です。デフォルトの数は 2 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 63 を超えてはならず、3 より小さくてもなりません。

キュー・マネージャーが作成される際には、**crtmqm** コマンドの **-ls** パラメーターによりこの値が上書きされます。

LogFilePages=number

ログ・データは、ログ・ファイルと呼ばれる一連のファイルに保持されません。ログ・ファイル・サイズは、4 MB ページ単位で指定します。

MQSeries for Compaq OpenVMS の場合は、ログ・ファイルのデフォルトのページ数は 1024 で、その場合のログ・ファイルのサイズは 4 MB になります。ログ・ファイルの最小ページ数は 64 ページであり、最大ページ数は、16 384 ページです。

キュー・マネージャーが作成される際には、**crtmqm** コマンドの **-lf** パラメーターによりこの値が上書きされます。

LogType=CIRCULAR|LINEAR

LogType 属性は、使用するタイプを定義するために使用します。デフォルトは CIRCULAR です。

CIRCULAR

ログを使用して、システムの停止時に処理中だったトランザクションをロールバックするために再始動リカバリーを開始したい場合は、この値を指定します。

循環ロギングの詳細については、175ページの『循環ロギング』を参照してください。

LINEAR

再始動リカバリーと、メディア・リカバリーまたは順方向リカバリー(ログの内容を再生することにより、消失または損傷したデータを作成する)の両方を使用したい場合は、この値を指定します。

リニア・ロギングの詳細については、175ページの『リニア・ロギング』を参照してください。

デフォルトのログ・タイプを変更したい場合は、mqs.ini ファイルで LogType 属性を編集できます。または、**crtmqm** コマンドで **-ll** パラメーターを使用してリニア・ロギングを指定することにより、デフォルトを変更することができます。キュー・マネージャーを作成したあとにロギングの方法を変更することはできません。

LogBufferPages=17|4-32

書き込み用としてバッファ・レコードに割り振るメモリーの量は、構成可能です。バッファのサイズは、4 KB ページ単位で指定します。

バッファの最小ページ数は 4 であり、最大ページ数は 32 です。バッファが大きいとスループットが高くなりますが、これは特に大きなメッセージに適用されます。

デフォルトのバッファ・ページ数は 17 であり、これは 68 KB と同じ大きさです。

この値は、キュー・マネージャーの作成時または開始時に調べられ、その時に増やしたり減らしたりすることができます。ただし、この値の変更は、キュー・マネージャーが再始動されるまで有効になりません。

LogDefaultPath=directory_name

キュー・マネージャーのログ・ファイルが置かれるディレクトリーを指定できます。ディレクトリーは、キュー・マネージャーが書き込みを行える

MQSeries 構成ファイルの変更

ローカル装置に置く必要があり、できればメッセージ・キューとは異なるドライブに置く必要があります。異なるドライブを指定すると、システム障害時の保護機能が加わります。

MQSeries for Compaq OpenVMS の場合のデフォルトは `MQS_ROOT:[MQM.LOG]` です。

代わりに、`crtmqm` コマンドで `-ld` フラグを使用して、ディレクトリーの名前を指定することもできます。キュー・マネージャーが作成されると、キュー・マネージャー・ディレクトリーの下にもう 1 つディレクトリーが作成され、ログ・ファイルの保存に使用されます。このディレクトリーの名前はキュー・マネージャー名に基づいて付けられます。これによってログ・ファイル・パスが確実に固有になり、ディレクトリー名の長さに関する制限にも抵触しないこととなります。

`crtmqm` コマンドで `-ld` を指定しなかった場合は、`mqs.ini` ファイル内の `LogDefaultPath` 属性の値が使用され、これは、`MQS_ROOT:[MQM.LOG]` になります。

複数のキュー・マネージャーで別々のログ・ディレクトリーが使用されるように、キュー・マネージャー名がログ・ファイル・ディレクトリー名に付加されます。

キュー・マネージャーが作成されると、`qm.ini` ファイル内のログ・スタンザの中に `LogPath` 値が作成されて、キュー・マネージャーのログ・ファイル用の完全ディレクトリー名が与えられます。この値は、キュー・マネージャーの開始時または削除時に、ログ・ファイルを見つけるために使用されます。

QueueManager スタンザ

各キュー・マネージャーごとに、`QueueManager` スタンザが 1 つずつあります。これらの属性は、キュー・マネージャーの名前と、そのキュー・マネージャーに関連したファイルが含まれているディレクトリーの名前を指定します。ディレクトリーの名前はキュー・マネージャー名に基づいていますが、キュー・マネージャー名が有効なファイル名でない場合は、変換されます。

名前変換の詳細については、28ページの『MQSeries のファイル名についての理解』を参照してください。

Name=*queue_manager_name*

この属性には、キュー・マネージャーの名前を指定します。

Prefix=*prefix*

この属性には、キュー・マネージャーのファイルが保管される場所を指定

します。デフォルトでは、`mqs.ini` ファイル内の `AllQueueManager` スタンザの `DefaultPrefix` 属性に指定されている値と同じです。

Directory=name

この属性は、キュー・マネージャーのファイルの記憶先のサブディレクトリの名前を指定します。これは通常、代替接頭部値が指定されない限り、`MQS_ROOT:[MQM.QMGRS]` の下にあります。この名前はキュー・マネージャー名に基づくものですが、重複する名前がある場合、またはキュー・マネージャー名が無効なファイル名の場合は、変換される場合があります。

キュー・マネージャー構成情報の変更

次の各属性グループは、特定のキュー・マネージャーに固有の `qm.ini` ファイルに含まれるもので、`mqs.ini` に設定されている値を指定変更するためにも使用されます。

- 『Service スタンザ』
- 232ページの『ServiceComponent スタンザ』
- 233ページの『Log スタンザ』
- 236ページの『XAResourceManager スタンザ』
- 238ページの『Channels スタンザ』
- 240ページの『LU62 および TCP スタンザ』
- 242ページの『ExitPath スタンザ』

Service スタンザ

`Service` スタンザには、インストール可能なサービスの名前、およびそのサービスの入り口点の数を指定します。使用する各サービスについて、`Service` スタンザが 1 つずつ必要です。

1 つのサービス内の各コンポーネントについて、`ServiceComponent` スタンザが 1 つずつ必要です。このスタンザは、そのコンポーネントのコードを含むモジュールの名前とパスを識別します。詳細については、『`ServiceComponent` スタンザ』を参照してください。

Name=AuthorizationService|NameService

必要なサービスの名前を指定します。

AuthorizationService

MQSeries の場合は、`Authorization Service` コンポーネントはオブジェクト権限管理プログラム (OAM) と呼ばれるものです。

MQSeries 構成ファイルの変更

MQSeries for Compaq OpenVMS では、キュー・マネージャーの作成時に、AuthorizationService スタンザとそれに関連した ServiceComponent スタンザが自動的に追加されますが、これは、キュー・マネージャーを作成する前に mqsnoaut を使用して mqsnoaut 論理を設定することにより、指定変更できます。(詳細については、108ページの『オブジェクト権限マネージャーを使用不可にする』を参照してください。) その他の ServiceComponent スタンザは、すべて手動で追加する必要があります。

NameService

提供されている名前サービスを使用可能にするには、NameService スタンザを手動で qm.ini ファイルに追加する必要があります。

EntryPoints=number-of-entries

このサービス用として定義する入り口点の数を指定します。これには、初期化入り口点と終了入り口点も含まれます。

インストール可能なサービスおよびコンポーネントの詳細については、*MQSeries プログラム式システム管理* を参照してください。

一般的なセキュリティー・サービスに関する詳細については、103ページの『第7章 MQSeries オブジェクトの保護』を参照してください。

ServiceComponent スタンザ

ServiceComponent スタンザは、このコンポーネントのコードを含むモジュールの名前とパスを識別します。

1 つのサービスについて複数の ServiceComponent スタンザがあってもかまいませんが、各 ServiceComponent スタンザはそれぞれ対応する Service スタンザに一致していることが必要です。

MQSeries for Compaq OpenVMS の場合、デフォルトでは、許可サービス・スタンザが事前に送られて、関連するコンポーネントである OAM がアクティブになっています。

Service=service_name

必要なサービスの名前を指定します。この名前は、Service スタンザの Name 属性に指定されている値に一致していることが必要です。

Name=component_name

サービス・コンポーネントの記述名を指定します。この名前は、固有のものであること、および MQSeries オブジェクトの名前 (たとえばキュー名) として有効な文字のみを含むものであることが必要です。この名前は、サ

ービスにより生成されるオペレーター・メッセージの中で使用されます。したがって、この名前は、企業の商標やそれに類似した識別性の高いストリングで始めることをお勧めします。

Module=*module_name*

このコンポーネントのコードを含むモジュールの名前を指定します。

注: 絶対パス名を指定してください。

ComponentDataSize=*size*

各呼び出しでコンポーネントに渡されるコンポーネント・データ域のサイズをバイト数で指定します。コンポーネント・データが不要な場合は、ゼロを指定します。

インストール可能なサービスおよびコンポーネントの詳細については、*MQSeries* プログラム式システム管理 を参照してください。

Log スタンザ

Log スタンザには、特定のキュー・マネージャーのログ属性を指定します。デフォルトでは、これらの属性は、**crtmqm** コマンドの特定のパラメーターに上書きされない限り、キュー・マネージャーの作成時に *mqs.ini* ファイル内の **LogDefaults** スタンザで指定されている設定から継承されます。詳細については、227ページの『**LogDefaults** スタンザ』および 320ページの『**crtmqm** (キュー・マネージャーの作成)』の両方を参照してください。

このスタンザの属性は、このキュー・マネージャーを他のキュー・マネージャーと違う構成にしたい場合に限り、変更するようにしてください。

qm.ini ファイル内の属性に指定されている値は、キュー・マネージャーの始動時に読み取られます。この構成ファイルは、キュー・マネージャーの作成時に作成されます。

ログ・サイズを計算する方法については、180ページの『ログのサイズの計算』を参照してください。

注: 次のパラメーター・リストに示す制限は、MQSeries により設定される制限です。オペレーティング・システムの制限により、設定可能なログ・サイズが削減されることがあります。

LogPrimaryFiles=3²⁻⁶²

1 次ログ・ファイルは、将来の利用のために作成時に割り振られるログ・ファイルです。

MQSeries 構成ファイルの変更

1 次ログ・ファイルの最小数は 2 で、最大数は 62 です。デフォルトは 3 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 63 を超えてはならず、3 より小さくてもなりません。

この値は、キュー・マネージャーの作成時または開始時に調べられます。キュー・マネージャーが作成された後に、この値を変更することができます。ただし、この変更された値は、キュー・マネージャーが再始動されるまで有効にはならず、効果はただちには現れません。

LogSecondaryFiles=2|1-6|

2 次ログ・ファイルは、1 次ファイルが足りなくなったときに割り振られるログ・ファイルです。

2 次ログ・ファイルの最小数は 1 であり、最大数は 61 です。デフォルトの数は 2 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 63 を超えてはならず、3 より小さくてもなりません。

この値は、キュー・マネージャーの始動時に検査されます。この値は変更することができます。ただし、変更された値は、キュー・マネージャーが再始動されるまでは有効にはならないし、有効になった場合でも効果がただちに現れるとは限りません。

LogFilePages=number

ログ・データは、ログ・ファイルと呼ばれる一連のファイルに保持されます。ログ・ファイル・サイズは、4 MB ページ単位で指定します。

MQSeries for Compaq OpenVMS では、ログ・ファイルのデフォルトのページ数は 1024 です。これは、4 MB のログ・ファイル・サイズと同じです。ログ・ファイルの最小ページ数は 64 ページであり、最大ページ数は、16 384 ページです。

注: キュー・マネージャーの作成時に指定したログ・ファイルのサイズを、既存のキュー・マネージャーについて変更することはできません。

LogType=CIRCULAR|LINEAR

LogType 属性は、キュー・マネージャーにより使用されるロギングのタイプを定義します。ただし、キュー・マネージャーの作成後に、使用するロギングのタイプを変更することはできません。必要とするタイプのロギングを持つキュー・マネージャーを作成する方法については、227ページの『LogDefaults スタンザ』の LogType 属性の説明を参照してください。

CIRCULAR

ログを使用して、システムの停止時に処理中だったトランザクションをロールバックするために再始動リカバリーを開始したい場合は、この値を指定します。

循環ロギングの詳細については、175ページの『循環ロギング』を参照してください。

LINEAR

再始動リカバリーと、メディア・リカバリーまたは順方向リカバリー(ログの内容を再生することにより、消失または損傷したデータを作成する)の両方を使用したい場合は、この値を指定します。

リニア・ロギングの詳細については、175ページの『リニア・ロギング』を参照してください。

LogBufferPages=174-32

書き込み用としてバッファ・レコードに割り振るメモリーの量は、構成可能です。バッファのサイズは、4 KB ページ単位で指定します。

バッファの最小ページ数は4であり、最大ページ数は32です。バッファが大きいとスループットが高くなりますが、これは特に大きなメッセージに適用されます。

デフォルトのバッファ・ページ数は17であり、これは68 KBと同じ大きさです。

この値は、キュー・マネージャーの始動時に検査され、その時点で増減される場合があります。ただし、この値の変更は、キュー・マネージャーが再始動されるまで有効になりません。

LogPath=directory_name

キュー・マネージャーのログ・ファイルが置かれるディレクトリーを指定できます。ディレクトリーは、キュー・マネージャーが書き込みを行えるローカル装置に置く必要があります、できればメッセージ・キューとは異なるドライブに置く必要があります。異なるドライブを指定すると、システム障害時の保護機能が加わります。

デフォルトは、MQS_ROOT:[MQM.LOG] です。

ディレクトリーの名前は、-ld フラグを使用して **crtmqm** コマンドに指定できます。キュー・マネージャーが作成されると、キュー・マネージャー・ディレクトリーの下にもう1つディレクトリーが作成され、ログ・ファイルの保存に使用されます。このディレクトリーの名前はキュー・マネ

MQSeries 構成ファイルの変更

ージャー名に基づいて付けられます。これによってログ・ファイル・パスが確実に固有になり、ディレクトリー名の長さに関する制限にも抵触しないこととなります。

crtmqm コマンドで **-ld** を指定しなかった場合は、**mqm.ini** ファイル内の **LogDefaultPath** 属性の値が使用されます。

注: MQSeries for Compaq OpenVMS では、ユーザー ID **mqm** およびグループ **mqm** はログ・ファイルについてすべての許可を持っている必要があります。これらのファイルの位置を変更する場合は、それらの許可を取得する必要があります。ログ・ファイルがプロダクトで提供されたデフォルト位置にある場合、これは必要ありません。

XAResourceManager スタンザ

XAResourceManager スタンザには、キュー・マネージャーで調整するグローバル作業単位に関するリソース・マネージャーを指定します。

グローバル作業単位に関するリソース・マネージャーのインスタンスごとに **qm.ini** で 1 つの XAResourceManager スタンザが必要となります。 **mqm.ini** を介してこのスタンザに提供されるデフォルト値はありません。

XAResourceManager 属性を **qm.ini** に追加する方法の詳細については、154ページの『データベースの調整』を参照してください。

Name=name (必須)

この属性は、リソース・マネージャーのインスタンスを識別します。

Name 値は長さが 31 文字までで、**qm.ini** の中で固有の名前でなければなりません。XA スイッチ構造で定義されているリソース・マネージャーの名前を使用することができます。ただし、同じリソース・マネージャーの複数のインスタンスを使用している場合は、各インスタンスごとに固有の名前を作成する必要があります。たとえば、次のように、データベースの名前を **Name** ストリングに組み込むことによって、それぞれの名前を固有に保つことができます。

MQSeries は、メッセージおよび **dspmqtrn** コマンドからの出力で **Name** 値を使用します。

関連するキュー・マネージャーが開始され、リソース・マネージャー名が有効になったら、リソース・マネージャーのインスタンスの名前を変更したり、**qm.ini** からの項目を削除したりしないことをお勧めします。

SwitchFile=name (必須)

この属性は、リソース・マネージャーの XA スイッチ構造が含まれているロード・ファイルの完全修飾名です。

XAOpenString=string (オプション)

この属性には、リソース・マネージャーの `xa_open` 入り口点に渡されるデータのストリングを指定します。このストリングの内容は、リソース・マネージャーによって異なります。たとえば、このリソース・マネージャーのインスタンスがアクセスするデータベースを識別します。この属性の定義方法については、161ページの『Oracle 用の XAResourceManager 構成情報の追加』を参照し、リソース・マネージャーの資料で適切なストリングを調べてください。

XACloseString=string (オプション)

この属性は、リソース・マネージャーの `xa_close` 入り口点に渡されるデータのストリングを指定します。このストリングの内容は、リソース・マネージャーによって異なります。この属性の定義方法については、161ページの『Oracle 用の XAResourceManager 構成情報の追加』を参照し、データベースの資料で適切なストリングを調べてください。

ThreadOfControl=THREAD|PROCESS

キュー・マネージャーは、自身のマルチスレッド・プロセスの 1 つからリソース・マネージャーを呼び出す必要がある場合に、`ThreadOfControl` 属性に設定されている値を使用して、逐次化の処理をします。

THREAD

これは、リソース・マネージャーが完全に「スレッド認識型」であることを意味します。マルチスレッド MQSeries プロセスでは、XA 関数呼び出しを外部のリソース・マネージャーに対して複数の論理経路から同時に行うことができます。

PROCESS

これは、リソース・マネージャーが「スレッド・セーフ」ではないことを意味します。マルチスレッド MQSeries プロセスでは、リソース・マネージャーに対して行える XA 関数呼び出しは一度に 1 つだけです。

`ThreadOfControl` 項目は、マルチスレッド・アプリケーション・プロセスのキュー・マネージャーによって発行される XA 関数呼び出しには適用されません。通常、別々の論理経路に並行する複数の作業単位を持つアプリケーションは、各リソース・マネージャーでサポートされるためにこのモードの操作が必要となります。

Channels スタンザ

Channels スタンザにはチャンネルに関する情報が入っています。

MaxChannels=100|number

この属性には、使用できるチャンネルの最大数を指定します。デフォルトは 100 です。

MaxActiveChannels=MaxChannels_value

この属性には、任意のある時点でアクティブにできるチャンネルの最大数を指定します。デフォルトは、MaxChannels 属性に指定されている値です。

MaxInitiators=3|number

この属性には、開始プログラムの最大数を指定します。

MQIBINDTYPE=FASTPATH|STANDARD

この属性には、アプリケーションのバインディングを指定します。

FASTPATH

チャンネルは MQCONNX FASTPATH を使用して接続されます。つまり、エージェント・プロセスはありません。

STANDARD

チャンネルは STANDARD を使用して接続されます。

AdoptNewMCA=NO|SVR|SDR|RCVR|CLUSRCVR|ALL|FASTPATH

MQSeries がチャンネル開始の要求を受け取ったときに、同じチャンネルについてすでに amqcrsta プロセスが存在している場合は、その既存のプロセスを停止してからでなければ、新しいプロセスを開始することはできません。AdoptNewMCA 属性を使用することにより、指定したチャンネル・タイプについて、既存のプロセスの終了と新規プロセスの開始を制御することができます。

特定のチャンネル・タイプについて AdoptNewMCA 属性を指定してあるときに、そのチャンネルがすでに実行中であるために新規チャンネルが始動できない場合は、次のようになります。

1. 新規チャンネルは、規定の手順に基づく終了要求を出して、既存のチャンネルを停止しようとしています。
2. 既存のチャンネル・サーバーが、AdoptNewMCATimeout に指定されている待ち時間が満了するまでにこの要求に応じない場合は、既存のチャンネル・サーバーのプロセス (またはスレッド) は強制終了されます。
3. ステップ 2 を実行したあとも既存のチャンネル・サーバーが終了していない場合は、AdoptNewMCATimeout の待ち時間が 2 回目に満了した時点で、MQSeries は、そのチャンネルを終了し、「CHANNEL IN USE」エラーを戻します。

下記のリストから 1 つまたは複数の値を選んで、コンマまたは空白で区切って指定します。

NO AdoptNewMCA 機能が不要であることを意味します。これはデフォルトです。

SVR サーバー・チャンネルが採用されます。

SDR 送信側チャンネルが採用されます。

RCVR 受信側チャンネルが採用されます。

CLUSRCVR

クラスター受信側チャンネルが採用されます。

ALL FASTPATH チャンネルを除くすべてのチャンネル・タイプが採用されます。

FASTPATH

該当チャンネルが FASTPATH チャンネルのときは、そのチャンネルが採用されます。これが起きるのは、

AdoptNewMCA=RCVR,SVR,FASTPATH などのように該当のチャンネル・タイプも指定されている場合だけです。

重要!

FASTPATH チャンネルの場合は、キュー・マネージャーの内部設計が原因で、AdoptNewMCA 属性が予期しない動作をすることがあります。したがって、FASTPATH チャンネルについて AdoptNewMCA 属性を使用可能にするときは、十分な注意が必要です。

AdoptNewMCATimeout=60|1--3600

この属性は、新プロセスが旧プロセスの終了を待つ時間を秒数で指定します。1 ~ 3600 秒の範囲内の値を指定してください。デフォルトは 60 です。

AdoptNewMCACheck=QM|ADDRESS|NAME|ALL

AdoptNewMCACheck 属性には、AdoptNewMCA属性を使用可能にするときに必要なタイプ検査を指定します。故意または不注意によりチャンネルが遮断されるのを防ぐためには、次の 3 種類の検査をすべて行うことが重要です。最低限、チャンネル名が一致することだけは確認してください。

MQSeries 構成ファイルの変更

下記から 1 つまたは複数の値を選んで、コンマまたはブランクで区切って指定します。

QM リスナー・プロセスが、キュー・マネージャー名が一致していることを確認しなければならないことを示します。

ADDRESS

リスナー・プロセスが通信アドレスを検査しなければならないことを示します。たとえば、TCP/IP アドレス。

NAME リスナー・プロセスが、チャンネル名が一致していることを確認しなければならないことを示します。

ALL リスナー・プロセスは、キュー・マネージャー名の一致、通信アドレス、およびチャンネル名の一致を検査します。

デフォルトは、FAP1、FAP2、および FAP3 の場合は AdoptNewMCACheck=NAME,ADDRESS で、FAP4 以降の場合は AdoptNewMCACheck=NAME,ADDRESS,QM です。

LU62 および TCP スタンザ

これらのスタンザは、ネットワーク・プロトコル構成パラメーターを指定します。これらのスタンザは、チャンネルのデフォルトの属性を指定変更します。

注: 指定する必要があるのは、デフォルト値に対する変更を表す属性だけです。

LU62

次の属性を指定できます。

TPName

この属性には、リモート・サイトで始動する TP 名を指定します。

LocalLU

これは、ローカル・システムで使用する論理装置の名前です。

TCP

次の属性を指定できます。

Port=1414|port_number

この属性には、TCP/IP セッション用のデフォルトのポート番号を 10 進表記で指定します。MQSeries の場合の「事前割り当て」ポート番号は 1414 です。

KeepAlive=YES|NO

この属性は、KeepAlive 機能をオンまたはオフに切り替えるために使用

します。KeepAlive=YES を指定すると、TCP/IP は、接続の相手側がまだ使用可能かどうかを定期的に検査します。使用可能でない場合は、チャンネルはクローズされます。

ListenerBacklog=number

TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。これは、TCP/IP ポート上でリスナーからの受け入れを待っている要求のバックログと考えることができます。デフォルトのリスナー・バックログ値は、表8 に示すとおりです。

表8. デフォルトの未解決接続要求数 (TCP)

プラットフォーム	ListenerBacklog のデフォルト値
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 以降	100
AIX V4.1 以前	10
その他のすべてのプラットフォーム	5

バックログが表8 に示されている値に達すると、TCP/IP 接続は拒否され、チャンネルは始動できなくなります。

MCA チャンネルの場合は、結果としてチャンネルは RETRY 状態になり、あとで接続を再試行することになります。

クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC_Q_MGR_NOT_AVAILABLE を受け取り、あとで接続を再試行することが必要になります。

ListenerBacklog 属性を使用すると、TCP/IP リスナーの場合のデフォルトの未解決要求数を指定変更することができます。

注: オペレーティング・システムの中には、表に示されているデフォルトより大きい値をサポートするものもあります。必要なら、この特性を利用して、接続限界数に達するのを防ぐことができます。

MQSeries 構成ファイルの変更

ExitPath スタンザ

ExitDefaultPath=string

ExitDefaultPath 属性は、下記の出口の位置を指定します。

- クライアント用のチャネル出口
- サーバー用のチャネル出口およびデータ変換出口

出口パスは、クライアントの場合は `mqs.ini` ファイル内の `ClientExitPath` スタンザから読み取られ、サーバーの場合はこのスタンザ (`ExitPath`) から読み取られます。

mqs.ini および qm.ini ファイルの例

243ページの図19 に、MQSeries for Compaq OpenVMS の `mqs.ini` ファイルの例を示します。

```

*****#
** 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=saturn.queue.manager
  Prefix=mqs_root:[mqm]
  Directory=saturn$queue$manager
DefaultQueueManager:
  Name=saturn.queue.manager
QueueManager:
  Name=pluto.queue.manager
  Prefix=mqs_root:[mqm]
  Directory=pluto$queue$manager

```

図 19. MQSeries for Compaq OpenVMS システム用の MQSeries 構成ファイルの例

244ページの図20 は、MQSeries for Compaq OpenVMS のキュー・マネージャ構成ファイル内で属性のグループをどのように配置するかを示しています。

MQSeries 構成ファイルの変更

```
#####  
#* 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.saturn$queue$manager]
```

図 20. キュー・マネージャー構成ファイル (MQSeries for Compaq OpenVMS) の例 (1/2)


```

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

```

図 20. キュー・マネージャー構成ファイル (MQSeries for Compaq OpenVMS) の例 (2/2)

注:

ノード上の MQSeries は、キュー・マネージャー用のデフォルト位置とログ用のデフォルト位置を使用しています。

キュー・マネージャー saturn.queue.manager は、ノードでのデフォルト・キュー・マネージャーです。このキュー・マネージャーに関連したファイルのディレクトリーは、OpenVMS ファイル・システムで有効なファイル名に自動的に変換されています。

MQSeries 構成ファイルは、キュー・マネージャーに関連したデータを見つけるために使用されるので、構成ファイルが存在しなかったり、正しくなかったりすると、一部または全部の MQSeries コマンドが失敗します。また、アプリケーションは、MQSeries 構成ファイルに定義されていないキュー・マネージャーには接続できません。

第14章 問題判別

この章では、MQSeries for Compaq OpenVMS の使用中に生じる問題を処理する方法を提案します。

問題の中には、すぐには解決できないものもあります。たとえば、パフォーマンスの問題は、ハードウェアの限界が原因となる場合があります。また、問題の原因が MQSeries コードにあると考えられる場合は、IBM® の S.E. に連絡ください。この章には、次の節があります。

- 『予備的な検査』
- 252ページの『共通プログラミング・エラー』
- 252ページの『次に何を行うか』
- 256ページの『アプリケーション設計の考慮事項』
- 258ページの『不正出力』
- 262ページの『エラー・ログ』
- 267ページの『送達不能キュー』
- 267ページの『構成ファイルと問題判別』
- 268ページの『MQSeries トレースの使用』
- 269ページの『第 1 障害サポート・テクノロジー (FFST)』
- 275ページの『クライアント問題判別』

予備的な検査

MQSeries の問題は、通常、以下のいずれかのコンポーネントで発生します。

- MQSeries
- ネットワーク
- アプリケーション
- 基礎となるオペレーティング・システム。

以下のセクションでは、問題を調査する際に考慮しなければならない基本的な質問をいくつか提供します。

以前に MQSeries は正常に実行されたか

もし、MQSeries が以前に正常に実行されていないならば、正しくセットアップしていない可能性があります。MQSeries for Compaq OpenVMS Alpha バージョン 5.1 スタートアップ・ガイド を参照して、MQSeries が正しくインストールおよびセットアップされたかどうかを確認してください。

エラー・メッセージが出されたか

MQSeries は、エラー・ログを使用して、MQSeries 自体の操作、ユーザーが開始したキュー・マネージャー、および使用中のチャネルからのエラー・データに関係したメッセージを取り込みます。問題に関連したメッセージが記録されていないかどうか、エラー・ログを検査してください。

エラー・ログの内容と位置については、262ページの『エラー・ログ』を参照してください。

問題について説明した戻りコードが戻されていないか

メッセージ・キュー・インターフェース (MQI) 呼び出しが失敗したことを示す戻りコードをアプリケーションが受け取っている場合、MQSeries アプリケーション・プログラミング・リファレンス で、戻りコードの説明を参照してください。

問題を再現できるか

問題を再現できる場合は、問題再現が可能な次の条件を考慮してください。

- 問題は、コマンドまたはそれと同等の管理要求によって引き起こされますか。

別の方法で入力した場合に、操作はうまくいきますか。コマンド行から入力した場合にはコマンドは作動するが、コマンド行以外からの場合には作動しない場合、コマンド・サーバーが停止していないかどうか検査してください。またキュー定義の `SYSTEM.ADMIN.COMMAND.QUEUE` が変更されていないかどうか検査してください。

- 問題はプログラムによって引き起こされますか。すべての MQSeries システムおよびすべてのキュー・マネージャーで失敗しますか、それとも一部のものだけで失敗しますか。
- 問題が生じる時にいつもシステムで実行されていると思われるアプリケーションを識別することができますか。識別できる場合、そのアプリケーションにエラーがないかどうか調べてください。

最後に正常に実行された後、変更が行われたか

最近行われた可能性のある変更について考慮する場合は、MQSeries システムについて考えてください。また、MQSeries システムがインターフェースする他のプログラムや、ハードウェアおよび新しいアプリケーションについても考えてください。さらに、ユーザーの気付いていない新しいアプリケーションがシステムで実行されている可能性はないかどうか考慮してください。

- キュー定義を変更、追加、または削除しましたか。

- チャネル定義を変更または追加しましたか。変更は、MQSeries チャネル定義、またはアプリケーションが必要とする基本的な通信定義に行われた可能性があります。
- アプリケーションは、ユーザーが行った変更の結果としてアプリケーションが受け取る可能性のある戻りコードを処理しますか。

アプリケーションは以前に正常に実行されたか

ある特定のアプリケーションが関係する問題であると思われる場合には、そのアプリケーションが以前に正常に実行されたかどうか考えてください。

この質問に対してはいと答える前に、次の点を考慮してください。

- そのアプリケーションが最後に正常に実行された後、そのアプリケーションに何らかの変更が行われましたか。

変更が行われた場合、アプリケーションの新しい部分または修正部分のどこかにエラーがある可能性があります。変更部分を調べて、問題の明白な理由があるかどうか見てください。アプリケーションの前のバージョンを使用して再試行できますか。

- アプリケーションのすべての機能が以前に完全に動作したか。

以前に呼び出されたことのないアプリケーションの一部が初めて使用されたときに問題が起きた可能性はありませんか。もしそうなら、アプリケーションのその部分にエラーがある可能性があります。障害が起きたときにアプリケーションが何を実行していたかを突きとめ、プログラムのその部分のソース・コードにエラーがないかどうか検査してください。

以前はたいていプログラムが正常に実行されていたという場合は、エラーが起きたときに処理されていた現行のキューの状態およびファイルを検査してください。プログラムでめったに使用されないパスを呼び出すようなごくまれなデータ値が、それに含まれている可能性があります。

- アプリケーションはすべての戻りコードを検査しますか。

MQSeries システムは、おそらく小さな点で変更されている可能性があります。その変更の結果出される戻りコードをアプリケーションは検査しません。たとえば、アプリケーションは自分がアクセスするキューは共用できると想定しています。もしキューが排他的として再定義された場合には、そのアプリケーションは、そのキューがアクセス不能になったことを示す戻りコードを処理できますか。

- アプリケーションは他の MQSeries システムで実行されますか。

予備的な検査

この MQSeries システムのセットアップの仕方に異なった点があり、それが問題を引き起こしている可能性がありますか。たとえば、キューは同じメッセージ長または優先順位で定義されていますか。

アプリケーションが以前に正常に実行されなかった場合

アプリケーションがこれまでに正常に実行されたことがない場合は、そのアプリケーションにエラーがないかどうか注意深く調べる必要があります。

コードを調べる前に、コードに使用されているプログラム言語に応じて、エラーが報告されていないかどうか、変換プログラムからの出力、またはコンパイラーとリンケージ・エディター (該当する場合) からの出力を調べてください。

アプリケーションが、ロード・ライブラリーに変換、コンパイル、またはリンク・エディットされない場合には、そのアプリケーションを呼び出そうとした場合、そのアプリケーションは実行できません。アプリケーションの作成の詳細については、MQSeries アプリケーション・プログラミング・リファレンスを参照してください。

出力文書が、それぞれのステップがエラーなしで行われたことを示している場合は、アプリケーションのコーディング論理を検討する必要があります。問題の症状は、機能の障害を示しており、その原因がコード内のエラーである場合があります。MQSeries アプリケーションで問題の原因となる共通エラーの例については、252ページの『共通プログラミング・エラー』を参照してください。

問題がネットワークの特定部分に影響するか

問題の影響を受けるネットワークの特定の部分 (たとえば、リモート・キュー) を識別できる場合があります。リモート・メッセージ・キュー・マネージャーへのリンクが働いていない場合、メッセージはリモート・キューに送ることができません。

2 つのシステム間の接続が使用可能かどうか、また MQSeries の相互通信コンポーネントが始動しているかどうかを検査してください。

メッセージが伝送キューに到達しているかどうかを検査し、伝送キューのローカル・キュー定義およびリモート・キューを検査してください。

ネットワーク関連の変更を行ったり、問題の理由の説明となる可能性がある MQSeries 定義を変更しましたか。

1 日の特定の時刻に問題が起こるか

1 日の特定の時刻に問題が起こる場合は、問題にシステム負荷が関与している可能性があります。普通、システム負荷のピークは、午前中ごろと午後の中ごろです。そのため、この時間帯に負荷に依存する問題が起こる可能性が最も高くなります (MQSeries ネットワークが複数の時間帯にまたがっている場合には、システム負荷のピークは、1 日の別の時間帯に起こる可能性があります)。

問題は断続的に起こるか

断続的な問題は、プロセスが互いに独立して実行できるということを考慮に入れておかなかったことが原因となっている可能性があります。たとえば、プログラムは、前のプロセスが完了する前に、待機オプションを指定せずに MQGET 呼び出しを出すことができます。メッセージを書き込む呼び出しが未確定であるとき (つまり、コミットされるかバックアウトされる前) に、アプリケーションがキューからメッセージを読み取ろうとする場合にも、断続的な問題が起こる可能性があります。

サービス更新を適用したか

サービス更新が MQSeries に適用された場合、更新処置が正常に完了したかどうか、またエラー・メッセージが生成されていないかどうかを検査してください。

- 更新には、特別の指示がありませんでしたか。
- 更新が正しく完全に適用されたことを確認するためのテストが行われましたか。
- MQSeries が前のサービス・レベルに復元されても、まだ問題が解消しませんか。
- インストールが正常に行われている場合、パッチ・エラーがないかどうか、IBM の S.E. に問い合わせてください。
- パッチが他のプログラムに適用されている場合には、MQSeries がそのプログラムとインターフェースをとる際の影響を考慮してください。

構成を適用する必要があるか

MQSeries は、基礎となるオペレーティング・システム (OpenVMS)、および TCP/IP などの各種のネットワーキング製品に依存します。それぞれのベンダーに連絡を取り、これらの製品で必要なサービス更新をすべて適用したかを確認してください。

共通プログラミング・エラー

次のリストのエラーでは、MQSeries プログラムの実行中に検出される問題に共通する原因の大部分が示されています。MQSeries システムの問題が、次のエラーの 1 つまたは複数によって引き起こされる可能性があることを考慮する必要があります。

- キューが共用可能であると想定しているが、実際は排他的である場合
- MQI 呼び出しで誤ったパラメーターを渡す場合
- MQI 呼び出しで不十分なパラメーターを渡す場合。これは、MQI が、アプリケーションが処理できるような完了コードおよび理由コードをセットアップできないことを意味します。
- MQI 要求からの戻りコードを検査できない場合
- 誤った長さを指定した変数を渡す場合
- 間違った順序でパラメーターを渡す場合
- *MsgId* および *CorrelId* を正しく初期化しなかった場合

次に何を行うか

予備的な検査を行えば、問題の原因を見つけることができます。見つけたら、今度ではできる限り MQSeries ライブラリー (参考文献を参照)、および他のライセンス・プログラムのライブラリーのその他の資料を使って、その問題を解決する必要があります。

原因がまだ分からない場合には、さらに詳しく問題を調査してください。

この節は、予備的な検査で原因を識別できなかった場合、問題の原因を突き止めるのに役立ちます。

システムに変更が行われていないこと、アプリケーション・プログラムに問題がないことを確認したら、問題の症状を最も的確に表している項目を選んでください。

- 253ページの『不正な出力を得たか』
- 253ページの『PCF コマンドからの応答を受け取れなかったか』
- 255ページの『問題はリモート・キューにのみ影響するか』

これらの症状がいずれもユーザーの問題を表していない場合には、システムの別のコンポーネントが原因となっていないかどうか考慮してください。

不正な出力を得たか

本書で「不正出力」とは、次のことを指します。

- アプリケーションが、予期していたメッセージを受け取らないこと
- アプリケーションが、予期しない情報または破壊された情報を含んでいるメッセージを受け取ること
- アプリケーションが、予期していなかったメッセージ (たとえば、別のアプリケーションに向けられたメッセージ) を受け取ること

いずれの場合にも、アプリケーションが使用しているキューまたはキュー・マネージャーの別名が正しく指定されているかどうかを検査し、ネットワークに対して行われた変更を配慮してください。

MQSeries エラー・メッセージ (すべて先頭に「AMQ」の文字が付いている) が生成されたら、エラー・ログを見てください。詳細については、262ページの『エラー・ログ』を参照してください。

PCF コマンドからの応答を受け取れなかったか

コマンドを出したが応答を受け取れなかった場合には、次の質問を考慮してください。

- コマンド・サーバーが実行されていますか。

dspmqcsv コマンドを使用して、コマンド・サーバーの状況を検査してください。

- このコマンドに対する応答が、コマンド・サーバーが実行されていないことを示したら、**strmqcsv** コマンドを使用してそれを開始してください。
- コマンドに対する応答が、MQGET 要求に対して **SYSTEM.ADMIN.COMMAND.QUEUE** が使用できないことを示したなら、そのキューを MQGET 要求に対して使用できるようにしてください。
- 応答は送達不能キューへ送られましたか。

送達不能キューのヘッダー構造は、問題を説明する理由またはフィードバック・コードを含んでいます。送達不能キューのヘッダー構造 (MQDLH) については、*MQSeries* アプリケーション・プログラミング・リファレンス を参照してください。

送達不能キューにメッセージが入っている場合は、提供されているブラウザのサンプル・アプリケーション (amqsbcbg) を使用し、MQGET 呼び出しを用いてメッセージをブラウザで見ます。サンプル・アプリケーションは、命名されたキュー・マネージャーの指定されたキューのすべてのメッセージを処

次に

理し、指定されたキューのすべてのメッセージのメッセージ記述子フィールドとメッセージ・コンテキスト・フィールドの両方を表示します。

- メッセージはエラー・ログに送られましたか。
詳細については、262ページの『エラー・ログ』を参照してください。
- キューは、読み書き操作に対して使用可能ですか。
- *WaitInterval* の長さは充分ですか。

MQGET 呼び出しがタイムアウトの場合は、完了コード MQCC_FAILED および理由コード MQRC_NO_MSG_AVAILABLE が戻されます。

(*WaitInterval* フィールドについて、および MQGET からの完了コードと理由コードについては、MQSeries アプリケーション・プログラミング・リファレンスを参照してください。)

- ユーザー自身のアプリケーション・プログラムを使用して、コマンドを SYSTEM.ADMIN.COMMAND.QUEUE に入れる場合には、同期点をとる必要がありますか。
要求メッセージを同期点から特別に除外しない限り、応答メッセージを受け取ろうとする前に同期点をとる必要があります。
- キューの MAXDEPTH 属性および MAXMSGL 属性は、十分高い値に設定されていますか。
- *CorrelId* フィールドおよび *MsgId* フィールドを正しく使用していますか。
キューからのすべてのメッセージを受け取ることができるように、アプリケーションで *MsgId* および *CorrelId* の値を設定してください。

コマンド・サーバーを停止し、その後再始動するよう試みて、生成されたエラー・メッセージに応答してください。

それでもシステムが応答しなければ、問題は、キュー・マネージャーか、または MQSeries システム全体のいずれかに関係している可能性があります。まず、個々のキュー・マネージャーを停止してみて、障害を起こしたキュー・マネージャーを分離してみてください。これによって問題が明らかにならないときは、エラー・ログに生成されているいずれかのメッセージに応じて、MQSeries を停止してから再始動してみてください。

再始動後にもまだ問題が起きる場合は、IBM の S.E. に連絡してください。

キューの一部が障害を起こしているか

キューのサブセットのみで問題が起きている疑いがある場合には、疑いのあるローカル・キューに問題があるかどうか検査してください。

1. 各キューについての情報を表示します。MQSC コマンド DISPLAY QUEUE を使用すればこの情報を表示できます。
2. 表示されたデータを使用して、次の検査を行います。
 - CURDEPTH が MAXDEPTH になっている場合、それは、そのキューが処理されていないことを示します。すべてのアプリケーションが正常に実行されているかどうか検査してください。
 - CURDEPTH が MAXDEPTH になっていない場合、次のキュー属性を検査してそれらが正しいかどうか確認してください。
 - トリガー操作が使用されている場合
 - トリガー・モニターは実行されていますか。
 - トリガーのサイズが大き過ぎませんか。つまり、トリガー操作によってトリガー・イベントは十分な頻度で生成されますか。
 - プロセス名は正しいですか。
 - プロセスは入手でき、操作可能ですか。
 - キューは共用可能ですか。共用可能でなければ、別のアプリケーションがすでにそのキューを入力の目的でオープンしている可能性があります。
 - キューは、読み取り (GET) および書き込み (PUT) が適切に行えるようになっていますか。
 - キューからメッセージを入手するアプリケーション・プロセスがない場合、その理由を判別してください。アプリケーションを開始する必要がある、接続が中断している、あるいは MQOPEN 呼び出しが何らかの理由で障害を起こしているなどの可能性があります。

キュー属性の IPPROCS および OPPROCS を検査してください。これらの属性は、キューが入力および出力のためにオープンされているかどうかを示します。値がゼロの場合、該当するタイプの操作は行われなことを示します。それらの値は変更されている可能性があること、およびオープンされていたキューがクローズされてしまうことに注意してください。

メッセージの書き込みまたは読み取りを予期している時の状況を検査する必要があります。

問題を解決できない場合、IBM の S.E. に連絡してください。

問題はリモート・キューにのみ影響するか

問題がリモート・キューにのみ影響する場合には、次の検査を行ってください。

次に

- 必要とされるチャンネルは開始されているかどうか、トリガーが可能かどうか、および必要とされる開始プログラムが実行されているかどうかを検査します。
- リモート・キューにメッセージを書き込む必要のあるプログラムが問題を報告していないかを検査します。
- トリガー操作を使用して分散キューイング・プロセスを開始する場合、伝送キューのトリガー操作がオンに設定されているかを検査します。また、チャンネル開始プログラムが実行しているかを検査します。
- チャンネル・エラーや問題を示すようなメッセージがないかエラー・ログを検査します。
- 必要ならば、チャンネルを手動で開始します。この方法については、 *MQSeries 相互通信* を参照してください。

チャンネルの定義方法については、 *MQSeries 相互通信* を参照してください。

アプリケーション設計の考慮事項

設計の悪いプログラムは、いろいろな方法でパフォーマンスに影響を与える可能性があります。どのような方法で影響するかを見つけるのは難しい場合があります。そのプログラムはうまく実行されているように見えても、他のタスクのパフォーマンスに影響していることがあるためです。 *MQSeries* 呼び出しを行うプログラムに特有のいくつかの問題を、以降の各節で解説します。

アプリケーション設計の詳細については、 *MQSeries アプリケーション・プログラミング・ガイド* を参照してください。

メッセージ長の影響

MQSeries では、100MB のデータを保持できることになっていますが、メッセージ内のデータの量はメッセージを処理するアプリケーションのパフォーマンスに影響します。アプリケーションの最高のパフォーマンスを達成するためには、メッセージ内の重要なデータのみを送信する必要があります。たとえば、銀行預金口座の借方勘定に記入する要求では、クライアントからサーバー・アプリケーションに渡す必要のある情報は、口座番号と借方の金額だけです。

メッセージ持続性の影響

持続メッセージはログに記録されます。メッセージをログに記録すると、アプリケーションのパフォーマンスは低下します。したがって、重要なデータの持続メッセージのみを使用する必要があります。キュー・マネージャーが停止するか障害を起こした場合にメッセージ内のデータが破棄されても構わない場合には、非持続性メッセージを使用してください。

特定のメッセージの検索

MQGET 呼び出しでは、通常、キューの最初のメッセージが取り出されます。しかし、メッセージ記述子内のメッセージ ID と相関 ID (*MsgId* および *CorrelId*) を使用して特定のメッセージを指定すると、キュー・マネージャーは、そのメッセージを見つけるまでキューを検索しなければなりません。このような方法で MQGET 呼び出しを使用すると、アプリケーションのパフォーマンスに影響します。

長さが異なるメッセージを含んでいるキュー

キュー上のメッセージの長さが様々である場合、メッセージのサイズを決めるために、アプリケーションでは *BufferLength* フィールドをゼロに設定した MQGET 呼び出しを使用できます。このようにすると、その呼び出しは失敗するが、メッセージ・データのサイズが戻されます。その後、アプリケーションは最初の呼び出しで調べたメッセージの ID と正しいサイズのバッファーを指定して、呼び出しをもう一度行うことができます。ただし、同じキューをサービスしている別のアプリケーションがある場合には、ユーザーのアプリケーションのパフォーマンスは低下します。2 番目に出す MQGET 呼び出しは、最初の呼び出しと 2 番目の呼び出しの間に別のアプリケーションによって取り出されたメッセージを見つけるために時間を費やしてしまうからです。

アプリケーションが固定長のメッセージを使用できない場合、この問題の別の解決方法は、MQINQ 呼び出しを使用して、キューが受け入れることのできるメッセージの最大サイズを調べ、その後 MQGET 呼び出しでその値を使用することです。キューのメッセージの最大サイズは、キューの *MaxMsgLength* 属性に保管されています。ただし、このキュー属性の値は 100 MB (MQSeries for Compaq OpenVMS で許容されている最大値) という大きな値である場合があるので、この方法は大きなストレージを使用する可能性があります。

同期点の頻度

1 つの同期点内で多数の MQPUT 呼び出しをコミットなしで出すプログラムは、パフォーマンス効率の問題を起こす可能性があります。影響を受けるキューは、現在アクセス不能なメッセージで満杯になり、他のタスクはそれらのメッセージを読み取るために待機することがあります。これには、ストレージの観点、およびメッセージを読み取ろうとしているタスクと関連付けられているスレッドの観点から、考慮すべき点があります。

MQPUT1 呼び出しの使用

キューに 1 つのメッセージを入れる場合にのみ MQPUT1 呼び出しを使用します。複数のメッセージを書き込みたい場合は、MQOPEN 呼び出しと、その後続けて一連の MQPUT 呼び出しおよび 1 つの MQCLOSE 呼び出しを使用します。

不正出力

「不正出力」という用語は、見方により異なった解釈ができます。本書で問題判別するため、その意味は 253 ページの『不正な出力を得たか』で説明しています。

この節では、次の 2 つのタイプの不正な出力について解説します。

- 予期していたときに現れないメッセージ
- 不正な情報または破壊された情報を含んでいるメッセージ

アプリケーションが分散キューの使用を含んでいる場合に起こる可能性のある別の問題についても解説します。

キューにメッセージが現れない場合

予期していたときにメッセージが現れない場合、次の点を検査してください。

- メッセージはキューに正常に書き込まれましたか。
 - キューは正しく定義されていますか。たとえば、MAXMSGL の大きさは充分ですか。
 - キューは書き込みが行えるようになっていますか。
 - キューが満杯になっていませんか。満杯になっていると、アプリケーションは必要なメッセージをキューに書き込めません。
- メッセージをキューから読み取ることができますか。
 - 同期点をとる必要がありますか。

同期点内でメッセージが書き込まれたり取り出されたりしている場合、リカバリー単位がコミットされるまで他のタスクはそれらのメッセージを使用できません。
 - 待機間隔の長さは十分ですか。

待機間隔は、MQGET のオプションとして設定できます。応答を待つ時間を十分に長くする必要があります。
 - メッセージ識別子または相関識別子 (*MsgId* または *CorrelId*) で識別された特定のメッセージを待っていますか。

正しい *MsgId* または *CorrelId* のメッセージを待っているかどうか検査してください。正常な MQGET 呼び出しでは、これらの値は取り出されたメッセージの値に設定されます。したがって、別のメッセージを正常に読み取るためにこれらの値をリセットする必要があります。

他のメッセージをそのキューから取得できるかどうかも検査してください。

- 他のアプリケーションは、キューからメッセージを取得できますか。
- 予期しているメッセージは、持続メッセージとして定義されましたか。
持続メッセージとして定義されていないと、MQSeries が再始動した場合には、メッセージは失われています。
- 別のアプリケーションがそのキューを排他的にアクセスしていましたか。

キューに問題が見いだされず、MQSeries が実行されている場合には、ユーザーが考えていたキューへのメッセージの書き込み処理について次の検査を行ってください。

- アプリケーションは開始されていましたか。
トリガーで開始されたと思われる場合には、正しいトリガー・オプションが指定されていたかどうか検査してください。
- アプリケーションは停止しましたか。
- トリガー・モニターは実行されていますか。
- トリガー・プロセスは正しく定義されていましたか。
- アプリケーションは正しく完了しましたか。
ジョブ・ログに異常終了の記録がないかどうか調べてください。
- アプリケーションは変更をコミットしましたか、あるいは変更はバックアウトされましたか。

複数のトランザクションがキューをサービスしている場合、それらは互いに対立する可能性があります。たとえば、あるトランザクションは、バッファ長ゼロを指定した MQGET 呼び出しを出してメッセージの長さを調べ、その後、そのメッセージの *MsgId* を指定した特定の MQGET 呼び出しを出すつもりです。しかし、一方で、別のトランザクションは、そのメッセージについて正常な MQGET 呼び出しを出すため、最初のアプリケーションは理由コード **MQRC_NO_MSG_AVAILABLE** を受け取ることになります。複数サーバー環境で実行されることが予期されるアプリケーションは、この状況に対処できるよう設計されている必要があります。

メッセージは受信されたが、アプリケーションがある点でそれを処理できなかった場合を考えてください。たとえば、予期した形式のメッセージにエラーが

あったためプログラムがそれを拒否しましたか。そのような場合は、『メッセージが予期しない情報または破壊された情報を含んでいる場合』を参照してください。

メッセージが予期しない情報または破壊された情報を含んでいる場合

メッセージに含まれている情報が、アプリケーションの予期していたものではない場合、あるいはその情報がある点で破壊されていた場合、次の点を考慮してください。

- ユーザーのアプリケーション、つまりメッセージをキューに入れるアプリケーションは変更されましたか。

変更はすべて、その変更を認識している必要のあるすべてのシステムで同時に反映されるようにしてください。

たとえば、メッセージ・データの形式が変更された可能性がある場合、どちらのアプリケーションもその変更を取り入れるために再コンパイルする必要があります。一方のアプリケーションが再コンパイルされていなければ、他方のアプリケーションにとってはデータが破壊されているように見えます。

- アプリケーションは、誤ったキューにメッセージを送信していますか。

ユーザーのアプリケーションが受け取っているメッセージは、実際には別のキューをサービスしているアプリケーションに向けられたものではないかどうか検査してください。無許可のアプリケーションが誤ったキューにメッセージを書き込むのを防ぐために、必要に応じてセキュリティー定義を変更してください。

アプリケーションが別名キューを使用していた場合は、別名が正しいキューを指し示しているかどうか検査してください。

- トリガー情報は、このキューについて正しく指定されていますか。

ユーザーのアプリケーションが開始されていたのか、または別のアプリケーションが開始されていたのかを検査してください。

上記の検査を行っても問題を解決できない場合には、メッセージを送信するプログラムとメッセージを受信するプログラムの両方のアプリケーション論理を検査しなければなりません。

分散キューを使用している場合の不正出力の問題

アプリケーションが分散キューを使用する場合には、次の点を考慮してください。

- MQSeries は、送信側のシステムと受信側のシステムの両方に正しくインストールされ、分散キューイング用に正しく構成されていますか。

- 2 つのシステム間でリンクは使用できますか。
両方のシステムが使用可能で、かつ MQSeries に接続されているかどうか検査してください。2 つのシステムの間接続、および 2 つのキュー・マネージャーの間のチャンネルがアクティブかどうかを検査してください。
- トリガー操作が、送信側のシステムでオンに設定されていますか。
- 待機の対象となっているメッセージは、リモート・システムからの応答メッセージですか。
トリガー操作が、リモート・システムでアクティブにされているかを検査してください。
- キューが満杯になっていませんか。
満杯になっていると、アプリケーションは必要なメッセージをキューに書き込めません。その場合は、メッセージが送達不能キューに書き込まれているかを検査してください。
送達不能キューのヘッダーは、メッセージが宛先キューに書き込まれなかった理由を示す理由コードまたはフィードバック・コードを含んでいます。送達不能キューのヘッダー構造については、MQSeries アプリケーション・プログラミング・リファレンス を参照してください。
- 送信側のキュー・マネージャーと受信側のキュー・マネージャーの間に不整合がありませんか。
たとえば、メッセージ長が、受信側のキュー・マネージャーの取り扱える長さより長い場合があります。
- 送信側のチャンネルと受信側のチャンネルのチャンネル定義に互換性がありますか。
たとえば、順序番号折り返しに不整合があると、分散キューイング・コンポーネントは停止します。分散キューイングの詳細については、MQSeries 相互通信 を参照してください。
- データ変換が関係していますか。送信側のアプリケーションと受信側のアプリケーションの間でデータ形式が異なっている場合には、データ変換が必要です。データ形式が、組み込まれている形式の 1 つとして認識される場合は、MQGET が出された時に変換が自動的に行われます。
データ・セットが変換を行えるものとして認識されない場合には、ユーザー自身のルーチンで変換を行えるように、データ変換出口が取られます。
例外は、データを MQSeries for MVS/ESA に送信している場合に発生しません。
データ変換の詳細については、MQSeries 相互通信 を参照してください。

エラー・ログ

MQSeries は、MQSeries 自体の操作、ユーザーが開始したキュー・マネージャー、および使用中のチャンネルからのエラー・データに関係したメッセージを取り込むために、多数のエラー・ログを使用します。

エラー・ログが保管される位置は、キュー・マネージャー名が知られているかどうか、およびエラーがクライアントに関連しているかどうかによって異なります。

- キュー・マネージャー名が知られていて、キュー・マネージャーが使用可能である場合は、次の位置になります。

```
MQS_ROOT:[MQM.QMGRS.QMgrName.ERRORS]AMQERR01.LOG
```

- キュー・マネージャーが使用可能でない場合には、次の位置になります。

```
MQS_ROOT:[MQM.QMGRS.$SYSTEM.ERRORS]AMQERR01.LOG
```

- エラーがクライアント・アプリケーションで発生した場合には、次の位置になります。

```
MQS_ROOT:[MQM.ERRORS]AMQERR01.LOG
```

- 第 1 障害サポート・テクノロジー (Support Technology[®]) (FFST) - 269ページの『FFST の読み方』を参照してください。

注: クライアントの場合、エラーは、クライアントのルート・ドライブに保管されます。

ログ・ファイル

インストール時に、[MQM.QMGRS.\$SYSTEM.ERRORS] ディレクトリーが QMGRS ファイル・パスに作成されます。この errors サブディレクトリーには、次のような名前のエラー・ログ・ファイルを 3 つまで入れることができます。

- AMQERR01.LOG
- AMQERR02.LOG
- AMQERR03.LOG

キュー・マネージャーの作成後、キュー・マネージャーが必要とするときに 3 つのエラー・ログ・ファイルが作成されます。これらのファイルの名前は、\$SYSTEM の名前と同じです。つまり、AMQERR01、AMQERR02、AMQERR03 であり、それぞれの容量は 256 KB です。これらのファイルは、作成した各キュー・マネージャーのエラー・サブディレクトリーに置かれます。

エラー・メッセージが生成されるたびに、AMQERR01 に入れられます。AMQERR01 が 256 KB よりも大きなメッセージを受け取ると、それは

AMQERR02 にコピーされます。そのコピーの前に、AMQERR02 が AMQERR03.LOG にコピーされます。 AMQERR03 の前の内容 (ある場合) は、破棄されます。

このようにして、最新のエラー・メッセージは常に AMQERR01 に入れられ、他のファイルは、エラー・メッセージの履歴を保持する目的で使用されます。

チャンネルに関連したすべてのメッセージも、該当のキュー・マネージャーのエラー・ファイルに入れられます。ただし、キュー・マネージャーの名前が知られていない場合や、キュー・マネージャーが使用できない場合は別です。キュー・マネージャー名が入手できない場合やその名前を判別できない場合には、チャンネル関連メッセージは [MQM.QMGRS.\$SYSTEM.ERRORS] サブディレクトリーに入ります。

エラー・ログ・ファイルの内容を調べるためには、通常の OpenVMS エディターを使用します。

早期エラー

上記のエラー・ログがまだ確立されていないときにエラーが起こるような特別のケースがあります。 MQSeries は、そのようなエラーをエラー・ログに記録しようとします。ログの位置は、どの程度のキュー・マネージャーが確立されたかによって異なります。

構成ファイルの破壊などのため位置情報を判別できない場合は、エラーは、インストール時にルート・ディレクトリーに作成された errors ディレクトリー mqm にログ記録されます。

MQSeries 構成ファイルが読み取り可能で、 AllQueueManagers スタンザの DefaultPrefix 属性が読み取り可能である場合、エラーは DefaultPrefix[.errors] ディレクトリーにログ記録されます。

構成ファイルの詳細については、221ページの『第13章 MQSeries の構成』を参照してください。

オペレーター・メッセージ

MQSeries for Compaq OpenVMS では、オペレーター・メッセージの通常エラーを識別します。通常エラーは、普通、有効でないパラメーターをコマンドに指定するといった事柄をユーザーが行うと直接生成されます。オペレーター・メッセージは、各国語 (NLS) が有効であり、メッセージ・カタログが標準位置にインストールされています。

エラー・ログ

これらのメッセージは関連したウィンドウ (ある場合) に書き込まれ、キュー・マネージャー・ディレクトリーにあるエラー・ログ AMQERR01.LOG にも書き込まれます。たとえば、次のとおりです。

```
MQS_ROOT:[MQM.QMGRS.QUEUE$MANAGER.ERRORS]
```

さらに、キュー・マネージャーのディレクトリーの AMQERR01.LOG ファイルにログ記録されるエラー、エラー・ログの \$SYSTEM ディレクトリー・コピーに記録されるエラーもあります。

エラー・ログの例

以下の例は、MQSeries for Compaq OpenVMS エラー・ログの一部を示します。

```
...
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.
-----
06/29/00 09:42:06 AMQ8003: MQSeries queue manager started.

EXPLANATION: MQSeries queue manager BK1 started.
ACTION: None.
-----
06/29/00 09:42:06 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 09:42:06 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKMI 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: None.
-----
06/29/00 09:46:43 AMQ7467: The oldest log file required to start queue
manager BKMI 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 BKMI 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 BKMI ended.
ACTION: None.
-----
06/29/00 09:46:59 AMQ7467: The oldest log file required to start queue
manager BKMI 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 BKMI 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.
-----
...

```

送達不能キュー

何らかの理由で送達できないメッセージは、送達不能キューに入れられます。MQSC DISPLAY QUEUE コマンドを出すことにより、キューが何らかのメッセージを含んでいるかどうかを検査できます。キューがメッセージを含んでいる場合、提供されているブラウザのサンプル・アプリケーション (amqsbcg) を使用し、MQGET 呼び出しを使用してキュー上のメッセージをブラウザできます。サンプル・アプリケーションは、命名されたキュー・マネージャーの指定されたキューのすべてのメッセージを処理し、指定されたキューのすべてのメッセージのメッセージ記述子フィールドとメッセージ・コンテキスト・フィールドの両方を表示します。

送達不能キューにメッセージが入れられた理由に応じて、そのキューにあるメッセージの後処理の方法を決める必要があります。

使用しているそれぞれのキュー・マネージャーに送達不能エラーがない場合に、問題が発生することがあります。 **crtmqm** コマンドを使用してキュー・マネージャーが作成される際に、デフォルト・オブジェクトとして、送達不能レターのキュー SYSTEM.DEAD.LETTER.QUEUE が自動的に作成されます。ただし、このキューは、キュー・マネージャーの送達不能キューとしては定義されません。60ページの『送達不能キューを定義する』を参照してください。

構成ファイルと問題判別

構成ファイルにエラーがあると、通常、キュー・マネージャーが見つからず、「キュー・マネージャーは利用不能」というタイプのエラーが起こります。

構成ファイルで実行する検査がいくつかあります。

- 構成ファイルが存在することを確認します。
- 適切な許可があるかどうかを確認します。たとえば、次のようにします。

```

MQS.INI;1 MQM (RWED, RWED, RW, R)
(identifier=MQM, ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)

```

- MQSeries 構成ファイルが確実に正しいキュー・マネージャーとログ・ディレクトリを参照するようにします。

MQSeries トレースの使用

MQSeries for Compaq OpenVMS は、トレース機能に以下のコマンドを使用します。

- **strmqtrc** - 387ページの『strmqtrc (MQSeries トレースの開始)』を参照
- **dspmqrtrc** - 340ページの『dspmqrtrc (MQSeries 定様式トレース出力の表示)』を参照
- **endmqtrc** - 349ページの『endmqtrc (MQSeries トレースの終了)』を参照

トレース機能では、トレースされているそれぞれのエンティティごとに 1 つのファイルを使用し、トレース情報は適切なファイルに記録されます。

トレースに関連したファイルは、ディレクトリー `MQS_ROOT:[MQM.TRACE]` に作成されます。

このディレクトリーのファイルは、キュー・マネージャーの詳細、およびすべての初期トレースやすべての `$SYSTEM` トレースを組み込みます。

トレース・ファイル名

トレース・ファイル名は、以下のように構成されます。

`AMQppppppppp.TRC`

この `ppppppppp` は、トレースを生成するプロセスのプロセス識別子 (PID) です。

注:

1. MQSeries for Compaq OpenVMS では、プロセス ID の値の長さは、常に 8 文字です。
2. トレースされるエンティティの一部として実行されるプロセスごとに 1 つのトレース・ファイルが作成されます。

トレース・データのサンプル

以下のサンプルは、OpenVMS トレースの抜粋です。


```

...
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
...

```

図 21. MQSeries for Compaq OpenVMS トレースのサンプル

注:

1. この例では、データの切り捨てが行われていますが、実際のトレースには完全な機能名と戻りコードが提示されます。
2. 戻りコードはリテラルではなく値として戻されます。

第 1 障害サポート・テクノロジー (FFST)

通常 FFST ログに記録される情報は、MQSeries for Compaq OpenVMS では、MQS_ROOT:[MQM.ERRORS] ディレクトリーにあるファイルに記録されません。

記録されるエラーは、通常、重大な回復不能エラーであって、システム関連の構成問題か MQSeries の内部エラーを示します。

FFST の読み方

ファイル名の形式は AMQnnnnnnnnn_mm.FDC です。ここで、
nnnnnnnn

エラーを報告するプロセス ID です。

mm は順序番号であり、通常、0 です。

プロセスが FFST を作成すると、システム・エラー・ログにエントリーを書き込むこともします。問題の自動追跡に役立つように、このレコードには FFST ファイルの名前が格納されています。

```
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
WSMAX=65536
NPAGEVIR=9437184
VIRTUALPAGECNT=2147483647
PQL_DASTLM=24
PQL_DBIOLM=32
PQL_DBYTLM=65536
PQL_DCPULM=0
PQL_DDIOLM=32
PQL_DFILLM=128
PQL_DPGFLQUOTA=65536
PQL_DPRCLM=32
PQL_DTQELM=16
PQL_DWSDEFAULT=1392
PQL_DWSQUOTA=2784
PQL_DWSEXTENT=65536
PQL_DENQLM=128
PQL_DJTQUOTA=4096
CLISYMTBL=750
DEFMBXBUFQUO=1056
DLCKEXTRASTK=2560
CTLPAGES=256
BALSETCNT=100
NPAGEDYN=2269184
PAGEDYN=1597440
LOCKIDTBL_MAX=109437
PQL_MASTLM=100
PQL_MBIOLM=100
PQL_MBYTLM=100000
PQL_MCPULM=0
PQL_MDIOLM=100
PQL_MFILLM=100
PQL_MPGFLQUOTA=32768
PQL_MPRCLM=10
PQL_MTQELM=0
PQL_MWSDEFAULT=1392
PQL_MWSQUOTA=2784
PQL_MWSEXTENT=65536
PQL_MENQLM=300
PQL_MJTQUOTA=0
DEFMBXMXMSG=256
CHANNELCNT=5000
PIOPAGES=575
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
--> xcDeleteMutexMem
--> xllCloseMutex
--> 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
<-- 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
6A91C0 03000000 E8030000 03220000 0100DA01 ZXEА
6A91D0 2C05A500 D4040000 0A00DA01 01000000
6A91E0 B0000000 0A00DA01 03000000 E8030000
6A91F0 03220000 0100DA01 07000000 00000000
6A9200 283F9700 03000000 F0030000 08410000
6A9210 0300DA01 03000000 F0030000 08410000
6A9220 0300DA01 01000000 B4000000 0200DA01
6A9230 03000000 E8030000 03220000 0100DA01
6A9240 00000000 00000000 00000000 00000000
6A9250 00000000 00000000 00000000 00000000
6A9260 00000000 01000000 B4000000 0200DA01
6A9270 03000000 E8030000 03220000 0100DA01
6A9280 B41F0000 0200DA01 01000000 B4000000
6A9290 0200DA01 03000000 E8030000 03220000
6A92A0 0100DA01 DA012020 00000000 EFC03000
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 /mqс_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
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 .....

```

「Function Stack」と「Trace History」は、問題判別を支援するために IBM が使用します。ほとんどの場合、FFST が生成されても、システム管理者が行うことは、SE を通じて問題を提示する以外にはほとんどありません。

ただし、解決可能な問題もあります。内部機能の 1 つを呼び出そうとしたとき、FFST に「out of resource」または「out of space on device」が示された場合は、該当する SYSGEN パラメーターの限界を超えたためと考えられます。

この問題を解決するには、システム・パラメーターを調整して内部限度を大きくしてください。詳細については、221 ページの『第13章 MQSeries の構成』を参照してください。

クライアント問題判別

MQI クライアント・アプリケーションは、非クライアントの MQI アプリケーションと同じ方法で MQRC_* 理由コードを受け取ります。しかし、クライアントに関連したエラー条件による理由コードが追加されました。たとえば、次のとおりです。

- リモート・マシンが応答しない
- 通信回線エラー
- 無効な機械アドレス

最も共通してエラーが起こるのは、アプリケーションが MQCONN を出して、応答 MQRC_Q_MQR_NOT_AVAILABLE を受け取った場合です。エラーの原因を説明するエラー・メッセージが、クライアント・ログ・ファイルに書き込まれます。障害の性質によっては、メッセージはサーバーにもログ記録される場合があります。

クライアントの終了

クライアントが終了しても、サーバーではプロセスがまだキューをオープンしておくことが可能です。通常、これは短い時間だけであり、通信層によりパートナーが終了したことが検出されるまでです。

クライアントによるエラー・メッセージ

クライアント・システムでエラーが発生したときには、可能であれば、サーバーに関連するエラー・ファイルにエラー・メッセージが書き込まれます。エラー・メッセージをそこに入れることができない場合は、クライアント・コードは、クライアント・マシンのルート・ディレクトリへのエラー・ログにエラー・メッセージを入れようとしています。

OS/2、UNIX、および OpenVMS システム・クライアント

OS/2、UNIX、および OpenVMS システムのクライアントに関するエラー・メッセージは、それぞれの MQSeries サーバー・システムと同じ方法でエラー・ログに書き込まれます。通常、これらのファイルは、OpenVMS システムでは MQS_ROOT:[MQM.ERRORS] ディレクトリに、UNIX システムでは /var/mqm/errors に表示されます。

DOS および Windows® クライアント

ログ・ファイル AMQERR01.LOG の位置は、MQDATA 環境変数で設定されます。MQDATA によって指定変更されていない場合は、デフォルト位置は次のようになります。

C:¥

DOS 環境で作業する場合には、環境変数 MQDATA が重要な働きをします。

これは、クライアント・コードがトレースおよびエラー情報を保管するために使用するデフォルト・ライブラリーです。これは、qm.ini ファイルが保管されるディレクトリ名も保持します (NetBIOS セットアップに必要)。指定しなかった場合は、デフォルトで C ドライブが使用されます。

このライブラリー内に保持されるデフォルト・ファイルの名前は次のとおりです。

AMQERR01.LOG

エラー・メッセージ用

AMQERR01.FDC

第 1 障害データ検知メッセージ用

第15章 パフォーマンスのチューニング

この章では、OpenVMS システムを調整して、MQSeries で最善のパフォーマンスを得る方法を説明します。

MQSeries のような製品では、あらゆるチューニング OpenVMS パラメーターに関して、すべての状況に渡って正しい値を定義することは不可能です。最も効果的な値は、MQSeries 自体、そして OpenVMS システム全体のワークロードにより判別されます。MQSeries for Compaq OpenVMS Alpha バージョン 5.1 スタートアップ・ガイド に記載されているパラメーター設定では、道理にかなった最小値または初期値を挙げていますが、キュー・マネージャーのワークロードが大きくなるにしたがって、これらの値を大きくする必要があります。これを行うプロセスのことを、「チューニング」と呼びます。

OpenVMS システムのパフォーマンス・チューニングについては、*OpenVMS Performance Management* で説明されています。このマニュアルの情報、そして、特に MQSeries に関係のある以下の点に注意してください。

- チューニングの中には、システム全体 (たとえば、GBLPAGES) に適用されるものがあります。これらのパラメーターは SYSGEN ユーティリティにより制御されるため、SYSGEN パラメーターと呼ばれる場合もあります。AUTOGEN FEEDBACK 機構を正しく使うことにより、システムが使用するリソースをモニターし、SYSGEN パラメーターを自動的に調整して、変化するワークロードをトラックできるようになります。これにより、システムを適正に調整するのに必要な手操作による介入を大幅に削減することができ、リソースの消耗が原因のエラーを避ける助けになります。MQSeries に関係のある SYSGEN パラメーターは、以下のとおりです。

GBLPAGES、GBLSECTIONS、および GBLPAGFIL

キュー・マネージャーは、共用 (グローバル) メモリーを介して通信する協調プロセスのセットとしてインプリメントされます。したがって、グローバル・メモリー (GBLPAGES、GBLSECTIONS、および GBLPAGFIL) を制御する SYSGEN パラメーターに十分大きな値を指定することが重要になります。MQSeries for Compaq OpenVMS Alpha バージョン 5.1 スタートアップ・ガイド では、これらのパラメーターについて、道理にかなった初期値を挙げています。

MQSeries のユーザーの数が増えると、グローバル・メモリーへの要求も大きくなるため、対応する SYSGEN パラメーターも大きくする必要があります。

CHANNELCNT

キュー・マネージャー・プロセスは、プロセス間通信および同期機構として OpenVMS メールボックスを使用します。これらのメールボックスは、チャンネルを介してアクセスされるため、SYSGEN パラメーター CHANNELCNT に十分大きな値を指定しなければなりません。たいていの場合、インストール時に設定された値で十分でしょう。ただし、アクティブな MQSeries プロセスの多い負荷の高いシステムでは、この値を大きくしなければならないかもしれません。

SYSGEN パラメーターを明示的にセットするには、*OpenVMS Performance Management* で説明されているように、ファイル MODPARAMS.DAT を変更してください。

- OpenVMS チューニング・パラメーターの中には、個々のユーザー名またはプロセス（たとえば、PGFLQUOTA）に適用されるものがあります。これらのパラメーターは、（常にではないとしても）通常、AUTHORIZE ユーティリティーにより制御されます。これらのパラメーターを調整するための自動メソッドはありません。ただし、これらのパラメーターは、特定のプロセスが使用できるいくつかのリソースの量の限度を示すため、必要最低限より高く値を設定して、時折起きるロードのピークに備えて余分に能力を提供したいと思われるでしょう。MQSeries に関係のあるプロセス固有のパラメーターは、以下のとおりです。

PGFLQUOTA

これは、プロセスが使用できるページ・ファイルのスペースの量を制御します。MQSeries の典型的な処理はメッセージを移動することですが、メッセージは非常に大きかったり、数が多かったりすることがあります。したがって、大量のページ・ファイル・スペースを消費する可能性があります。

PRCLM

このパラメーターは、指定されるプロセスが作成できるサブプロセスの数を制御します。たいていの MQSeries プロセスは、実行コントローラーのサブプロセスとして作成されているため、システムでは、PRCLM に高い値を使用することが必要になります。

ENQLM、ASTLM、TQELM

すでに述べたとおり、キュー・マネージャーは、協調プロセスのセットとしてインプリメントされます。これらのプロセスは、OpenVMS ロック・マネージャー、非同期システム・トラップ (AST) およびタイマーを使用して、これらのアクティビティーを同期化し

ます。これらのリソースの使用を制限する 3 つのパラメーターは、キュー・マネージャーの必要に対応できるように、十分に大きな値に設定しなければなりません。

プロセス固有パラメーターの値の設定

これらのパラメーターを設定する最も一般的な方法は、AUTHORIZE ユーティリティを使用して、適切なユーザー名の値 (MQSeries には通常、MQM) を調整することです。

ただし、OpenVMS では、プロセス割り当て量の中には、同じジョブのすべてのプロセス、つまり、親プロセス、または親のサブプロセスである他のすべてのプロセスにより共用されるものがあります。このカテゴリーのプロセス割り当て量には、BYTLM、FILLM、PGFLQUOTA、PRCLM、TQELM、および ENQLM が含まれ、これらはプール割り当て量と呼ばれます。

大半のキュー・マネージャーは、実行コントローラーのサブプロセスとして作成されるため、結果として、プール割り当て量は、すべてのキュー・マネージャー・プロセスにより共用されることとなります。したがって、通常、これらの割り当て量を、単一のプロセスには大きすぎると思える値に設定することが必要となります。これは、PGFLQUOTA の場合には、特にそう言えます。このパラメーターは、キュー・マネージャー・プロセスが**集合的に**作成できる仮想メモリーの量を制限するためです。この理由により、実行コントローラーが起動されると、AUTHORIZE により保守される許可ファイルから初期の割り当て量の値を取得せずに、コントローラー自体が適当な値にこれらを設定します。この結果、これらの値を変更するのに AUTHORIZE を使用することはできなくなります。代わりに、以下の論理名を使用して、明示的な割り当て量の値を変更することができます。

```
MQS_ASTLM
MQS_BIOLM
MQS_BYTLM
MQS_DIOLM
MQS_ENQLM
MQS_FILLM
MQS_PGFLQUOTA
MQS_PRCLM
MQS_TQELM
```

プロセス固有パラメーターの設定

これらの論理は、ファイル `SYSDMANAGER:MQS_SYSTARTUP.COM` を使用して設定できます。MQSeries には、編集および名前変更が可能な、`SYSDMANAGER:MQS_SYSTARTUP.TEMPLATE` というファイルが用意されています。たとえば、`PGFLQUOTA` パラメーターに異なる値を提供するには、以下のようにします。

1. `MQS_SYSTARTUP.TEMPLATE` という `.TEMPLATE` ファイルを、`.COM` ファイルにコピーします。
2. 上で作成した `MQS_SYSTARTUP.COM` ファイルを編集して、`mqs_pgflquota` などのプロセス割り当て量に対応する論理を定義する行をコメント解除 (活性化) します。
3. 新しい値を定義します。
たとえば、次のとおりです。

```
$! DEFINE/SYSTEM MQS_PGFLQUOTA 1000000
```

は、以下のようになります。

```
$ DEFINE/SYSTEM MQS_PGFLQUOTA 5000000
```

4. 論理を定義するために `mqs_systartup` ファイルを起動します。たとえば、次のとおりです。

```
$ @sys$manager:mqs_systartup
```

通常、これは、システム始動プロシーチャーの一部として実行されます。

プール割り当て量が不足しているのが明らかになるのは、新しいクライアント・アプリケーションがキュー・マネージャーの接続に失敗した場合や、新しい接続でリソースを消費しすぎたために別のアプリケーションに障害が発生した場合などです。

また、実行コントローラーは、この割り当て量の多くについて明示設定を使用して始動されるため、`SYSGEN PQL_D*` パラメーターは EC に適用されないことに注意してください。

第16章 MQSeries for OpenVMS およびクラスター化

OpenVMS クラスター と *MQSeries* キュー・マネージャー・クラスター は、互いに独立した 2 つの異なるものです。

注: クラスター という語が用いられる場合、それは *MQSeries* キュー・マネージャー・クラスターを指します。 *OpenVMS* クラスターは、常に *OpenVMS* クラスター と呼ばれます。

MQSeries キュー・マネージャー・クラスターは、*OpenVMS* クラスター相互通信プロトコル、 *OpenVMS* クラスター分散ロック・マネージャー、あるいは *OpenVMS* クラスター・ファイル・システムを使用するとは限りません。*MQSeries* クラスターでのキュー・マネージャー間の通信はすべて、サポートされるプロトコルの 1 つを使用して *MQSeries* チャンネルを介して実行されます。したがって、同じ *OpenVMS* クラスターの一部ではない *OpenVMS* システム上で実行するキュー・マネージャーを使用して、 *MQSeries* キュー・マネージャー・クラスターを実行することが可能です。

MQSeries キュー・マネージャーが *OpenVMS* クラスター内で構成されている場合、この *MQSeries* キュー・マネージャーは、*OpenVMS* 内では一度に 1 つの *OpenVMS* ノード (この章では、この先、ノードと呼びます) でしか実行できません。単一の *MQSeries* キュー・マネージャーの機能は、 *OpenVMS* クラスター内の複数の *OpenVMS* ノードに配布することはできません。複数の *OpenVMS* ノードで *MQSeries* キュー・マネージャーを始動しようとする、エラーが戻されます。ただし、*OpenVMS* クラスター内で複数の *MQSeries* キュー・マネージャーが構成されている場合には、これらは、*OpenVMS* クラスター内の異なる *OpenVMS* ノード上で実行することができます。

OpenVMS クラスターでの *MQSeries* キュー・マネージャーの可用性をさらに高水準なものにするため、フェールオーバー・セットと呼ばれる新しい機能が *MQSeries* V5.1 に導入されました。これにより、障害が発生した場合には、別の *OpenVMS* クラスター・ノードでキュー・マネージャーが自動的に再始動されるようになります。この機能は、 *MQSeries* キュー・マネージャー・クラスターで使用することも、 *MQSeries* キュー・マネージャー・クラスターなしで使用することもできます。(283ページの『*OpenVMS* クラスター・フェールオーバー・セット』を参照してください)。

OpenVMS クラスタへの MQSeries のインストール

OpenVMS クラスタへの MQSeries for Compaq OpenVMS Alpha V5.1 のインストールは、スタンドアロン OpenVMS システムへの MQSeries のインストールに大変類似しています。ただし、インストールの前に、以下のことを考慮する必要があります。

- OpenVMS クラスタに複数のシステム・ディスクがある場合には、ノードのブート元であり、かつ MQSeries を実行しなければならないそれぞれのシステム・ディスクに MQSeries をインストールする必要があります。MQSeries は、ノードごとにではなく、システム・ディスクごとに一度インストールするだけで構いません。
- MQS_ROOT ディレクトリー構造を保持するディスクは、ディレクトリー構造に含まれているキュー・マネージャーを実行する OpenVMS ノード上に、システム全体のレベルでマウントされなければなりません。それぞれのノードに異なる MQS_ROOT ディレクトリー構造を使用することができます。しかし、フェールオーバー・セットが構成される場合には、1 つのフェールオーバー・セットにあるそれぞれの OpenVMS ノードは、同じ MQS_ROOT ディレクトリー構造を参照しなければなりません。MQSeries をインストールする際には、インストールを行うたびに、(‘Enter the root device for the MQSeries datafiles:’ の質問に回答する形で) MQS_ROOT ディレクトリーを指定しなければなりません。
- キュー・マネージャーのログ・ファイルを含むディスクが、MQS_ROOT を含むディスクと異なる場合には、フェールオーバー・セットにあるすべてのノード上に、このログ・ファイルを含むディスクをマウントしなければなりません。
- MQSeries は、デフォルト・ディレクトリー SYS\$SPECIFIC:[MQS_SERVER] のある MQM アカントを使用します。このディレクトリーは、MQSeries がインストールされているノードにのみ作成されます。ディレクトリーは、同じシステム・ディスクからブートし、かつ MQSeries を実行するそれぞれの追加ノードに実行されなければなりません。これは、それぞれの追加ノード上で以下の DCL コマンドを実行することにより可能です。

```
$create/directory sys$specific:[mqserver]/owner=[mqserver] -  
/protection=(s:rwed,o:rwed,g,w)  
$set sec/acl=(identifier=mqm,options=default,access=r+w+e+d+c) -  
sys$specific:[000000]mqserver.dir  
$set sec/acl=(identifier=mqm,access=r+w+e+d+c) -  
sys$specific:[000000]mqserver.dir
```

OpenVMS クラスタ・フェールオーバー・セット

OpenVMS クラスタ・フェールオーバー・セットの概説

OpenVMS クラスタ・フェールオーバー・セットは、MQSeries for Compaq OpenVMS V5.1 で使用可能な新機能です。これにより、MQSeries キュー・マネージャーに障害が起きた場合には、OpenVMS クラスタにある別の OpenVMS ノードで MQSeries キュー・マネージャーを自動的に再始動することができるようになります。OpenVMS クラスタ・フェールオーバー・セットでは、以下の障害のタイプがサポートされています。

- MQSeries キュー・マネージャーを実行する OpenVMS ノードの一時停止
- MQSeries キュー・マネージャーを実行する OpenVMS ノードのシステムの破損
- MQSeries キュー・マネージャーの完全終了がなされていない、MQSeries キュー・マネージャーを実行する OpenVMS ノードのシャットダウン
- MQSeries キュー・マネージャーの Execution Controller プロセスの障害。

OpenVMS クラスタ・フェールオーバー・セットでは、以下の障害のタイプはサポートされていません。

- MQSeries キュー・マネージャーを実行している OpenVMS ノードでの障害で、ノードまたは MQSeries キュー・マネージャーに障害を引き起こさないもの。
- Execution Controller プロセス以外の MQSeries キュー・マネージャー・プロセスの障害。MQSeries キュー・マネージャーが、同じノードで自動的に再始動されることはありません。
- MQSeries キュー・マネージャーのキュー・ファイルおよびログ・データを保持するディスクのソフトウェアまたはハードウェアの障害。
- MQSeries キュー・マネージャー・キュー・ファイルまたはログ・データの破壊。

OpenVMS クラスタ・フェールオーバー・セットは、MQSeries チャネルに TCP/IP プロトコルを使用するキュー・マネージャーにのみサポートされています。以下の TCP/IP スタックがサポートされています。

- Digital[®] TCP/IP Services for OpenVMS V5.0A
- Process Software の TCPware[®] for OpenVMS V5.4
- Process Software の Multinet[®] for OpenVMS 4.3

OpenVMS クラスタ・フェールオーバー・セットの概念

OpenVMS クラスタ・フェールオーバー・セットは、もしかすると MQSeries キュー・マネージャーを実行するかもしれない OpenVMS ノードのコレクションです。OpenVMS クラスタ・フェールオーバー・セットには、1 つから 4 つの OpenVMS ノードを入れることができ、すべての OpenVMS ノードは、同じ OpenVMS クラスタのメンバーでなければなりません。1 つの OpenVMS クラスタ・フェールオーバー・セットは、1 つの MQSeries キュー・マネージャーに限定されます。1 つの OpenVMS クラスタに構成される OpenVMS クラスタ・フェールオーバー・セットは複数あります。OpenVMS クラスタ・フェールオーバー・セットによりサポートされるキュー・マネージャー名の最大長は、25 文字であることに注意してください。

フェールオーバーとは、サポートされる障害の発生時に MQSeries キュー・マネージャーが別の OpenVMS ノード上で再始動されるプロセスのことです。このプロセスが完了すると、MQSeries キュー・マネージャーはフェールオーバーしたということになります。

フェイルバックとは、障害の解決後に MQSeries キュー・マネージャーが元の OpenVMS ノード上で再始動されるプロセスのことです。OpenVMS クラスタ・フェールオーバー・セットは、自動フェイルバックをサポートしませんが、手動で実行することはできます。このプロセスが完了すると、MQSeries キュー・マネージャーはフェイルバックしたということになります。

フェールオーバー・モニターは、OpenVMS クラスタ・フェールオーバー・セットのそれぞれのメンバーで実行するプロセスです。フェールオーバー・モニターは、フェールオーバー・セットのすべての機能の実行を担当します。OpenVMS クラスタ・フェールオーバー・セット内のフェールオーバー・モニターは、互いに協調し、これらの機能を提供します。フェールオーバー・モニターは、`runmqfm` コマンドを使用して始動されます。(このコマンドの詳細については、367ページの『`runmqfm` (フェールオーバー・モニターの開始)』を参照してください。)

1 つのフェールオーバー・モニターが監視プログラム・フェールオーバー・モニターに指定されます。このフェールオーバー・モニターを、監視状態にあるといいます。フェールオーバー・セットで最初に始動するフェールオーバー・モニターが、最初の監視プログラム・フェールオーバー・モニターです。最初のフェールオーバー・モニターが始動されると、フェールオーバー・セットはライブになります。この監視プログラム・フェールオーバー・モニターに障害が起きたり、これが実行している OpenVMS ノードに障害が起きた場合には、他のフェールオーバー・モニターが、自動的に監視プログラム・フェールオー

OpenVMS クラスタ・フェールオーバー・セットの概念

バー・モニターに指定されます。監視プログラム・フェールオーバー・モニターは、MQSeries キュー・マネージャーが実行していることを検査したり、サポート対象の障害が発生した場合にフェールオーバー操作を始動したりすることを担当します。別の OpenVMS ノードで実行しなければならない操作は、監視プログラム・フェールオーバー・モニターにより、関係する OpenVMS ノード上の実際に操作を実行するフェールオーバー・モニターに転送されます。

OpenVMS クラスタ・フェールオーバー・セットは、DCL コマンド **failover** を使用して管理されます。**failover** コマンドは、OpenVMS クラスタ・フェールオーバー・セットの任意のノードから使用できます。すべてのコマンドは、監視プログラム・フェールオーバー・モニターに送信され、ここで、このコマンドを処理するフェールオーバー・モニターはどれか、また、別のフェールオーバー・モニターにこれを転送する必要があるかどうかを決定します。

OpenVMS クラスタ・フェールオーバー・セット構成ファイルは、OpenVMS ノードの番号および名前を含む、OpenVMS クラスタ・フェールオーバー・セットの詳細を含みます。ファイルの名前は FAILOVER.INI であり、ディレクトリー MQS_ROOT:[MQM.QMGRS.queueanagername] にあります。これはテキスト・ファイルであり、テキスト・エディターで変更されます。最初のフェールオーバー・モニターを始動する前に、このファイルを作成しなければなりません。テンプレート構成ファイル FAILOVER.TEMPLATE がディレクトリー MQS_EXAMPLES に用意されています。構成ファイルにあるパラメーターは、動的に変更することはできません。変更を有効にするには、すべてのフェールオーバー・モニターを停止してから再始動しなければなりません。フェールオーバー・モニターが始動されていない場合には、MQSeries キュー・マネージャーの自動フェールオーバーが起こることはないため、これを実行する際には、注意が必要です。

フェールオーバー・セット内の MQSeries キュー・マネージャーでは、**strmqm** および **endmqm** コマンド以外のすべての MQSeries コマンドは、通常通りに処理されます。これら 2 つのコマンドは、ライブ・フェールオーバー・セットに MQSeries キュー・マネージャーがある場合にエラーを戻します。MQSeries キュー・マネージャーを始動および終了するには、**failover** コマンドを使用しなければなりません。

OpenVMS ノードの優先順位は、OpenVMS クラスタ・フェールオーバー・セットにあるそれぞれの OpenVMS ノードに指定される優先順位であり、障害が発生した後にどの OpenVMS ノードでキュー・マネージャーを始動するかを判別するのに使用されます。優先順位の数値が最も小さい OpenVMS ノードが、最も高い優先順位になります。

OpenVMS クラスタ・フェールオーバー・セットの概念

OpenVMS クラスタ・フェールオーバー・セット TCP/IP アドレスは、フェールオーバー・セットに割り当てられる TCP/IP アドレスです。フェールオーバー・セット・キュー・マネージャーを参照するすべてのチャンネルは、接続名にこの TCP/IP アドレスを指定するように構成されなければなりません。それぞれの OpenVMS クラスタ・フェールオーバー・セットは、固有の TCP/IP アドレスを使用しなければなりません。OpenVMS クラスタ・フェールオーバー・セット内のすべての OpenVMS ノードには、同じサブネットのインターフェース TCP/IP アドレスがなければならず、OpenVMS クラスタ・フェールオーバー・セット TCP/IP アドレスは、同じサブネットになければなりません。

OpenVMS クラスタ・フェールオーバー・セットの構成の準備

OpenVMS クラスタ・フェールオーバー・セットを構成する前に、以下のステップを実行しなければなりません。

1. **crtmqm** を使用して、キュー・マネージャーを作成します (これが存在していない場合)。
2. OpenVMS クラスタ・フェールオーバー・セットの TCP/IP アドレスを取得します。
3. OpenVMS クラスタ TCP/IP フェールオーバー・セット TCP/IP アドレスを使用するための MQSeries チャンネルを作成または変更します。
4. OpenVMS クラスタ・フェールオーバー・セットに入れる OpenVMS ノードを決定し、それらの優先順位を決定します。
5. MQS_ROOT 論理が、OpenVMS クラスタ・フェールオーバー・セット内のすべての OpenVMS ノードで同じディレクトリを参照していること、およびすべてのノード上でディスクがシステム全体のレベルでマウントされていることを確認します。MQS_ROOT ディレクトリおよびログ・ファイルを含むディスクは、フェールオーバー・セット内のあるノードから別のノードに処理される MSCP にすることはできません。ディスクを処理しているノードが使用不可になると、ディスクが処理されているノードがディスクにアクセスできなくなってしまうためです。

OpenVMS クラスタ・フェールオーバー・セットの構成

OpenVMS クラスタ・フェールオーバー・セットを構成するには、以下のステップを実行してください。

1. MQS_EXAMPLES:FAILOVER.TEMPLATE ファイルを
MQS_ROOT:[MQM.QMGRS.queuemanagername]FAILOVER.INI にコピーします。

2. `MQS_ROOT:[MQM.QMGRS.queuemanagename]FAILOVER.INI` ファイルを編集し、この OpenVMS クラスタ・フェールオーバー・セット構成用に変更します。(288ページの『FAILOVER.INI 構成ファイルの編集』を参照してください。)
3. **START_QM.COM, END_QM.COM** および **TIDY_QM.COM** コマンド・プロシージャーを編集します。(289ページの『フェールオーバー・セットにより使用されるコマンド・プロシージャー』を参照してください。)
4. フェールオーバー・モニターにより使用される ICC アソシエーションの ICC セキュリティーをセットアップします(299ページの『ICC アソシエーションのセキュリティの設定』を参照)。
5. **runmqfm -m queuemanagename** コマンドを使用して、OpenVMS クラスタ・フェールオーバー・セットのそれぞれのノードでフェールオーバー・モニターを始動します。
6. **failover -m queuemanagename -n nodename -s** コマンドを使用して、キュー・マネージャーを始動します。
7. サイト固有シャットダウンを以下のように変更します。
 - シャットダウン時にノード上でキュー・マネージャーが実行している場合には、これを終了または移動します。
 - フェールオーバー・モニターを停止します。

OpenVMS クラスタ・フェールオーバー・セットの構成後のタスク

以下は、クラスタ・フェールオーバー・セットが構成された後に実行できるタスクです。

- **runmqfm** コマンドを使用して、システム始動ファイルを編集し、OpenVMS クラスタ・フェールオーバー・セットにあるそれぞれのノード上でフェールオーバー・モニター・プロセスを始動します。**runmqfm** コマンドは、MQSeries を始動するためのコマンドの後に置いてください。
- システムの始動時にキュー・マネージャーを自動的に始動する必要がある場合には、関連するノード上でシステム始動にコマンドを配置して、フェールオーバー・モニターの始動後に、キュー・マネージャーを始動するようにしてください。ノード上でキュー・マネージャーを始動するためのコマンドは、**failover -m queuemanagename -n nodename -s** です。
- サイト固有のシャットダウンを変更して、システムのシャットダウン時にフェールオーバー・モニターを終了するようにします。また、シャットダウン時にノード上でキュー・マネージャーが実行している場合には、これを終了または移動します。

FAILOVER.INI 構成ファイルの編集

FAILOVER.INI ファイルは、それぞれの OpenVMS クラスタ・フェールオーバー・セットごとにカスタマイズしなければなりません。それぞれのフィールドの意味は、表9 にリストされています。MQS_EXAMPLES で提供されるテンプレート構成ファイルは、417ページの『付録F. OpenVMS クラスタ・フェールオーバー・セット・テンプレート』にあります。フェールオーバー・モニター・プロセスがこれを読み取る際、先頭に '#' 文字がある行はすべて無視されます。ファイル内にあるフィールド名の太文字と小文字は、テンプレート・ファイルで指定されたのと同じでなければなりません。それぞれのフィールド名の後には '=' 文字と、関連した値を続けます。テンプレートにあるフィールドはすべて必須であるため、どのフィールドも除去してはなりません。

表9. FAILOVER.INI ファイル内のフィールドの説明

フィールド名	説明
IpAddress	フェールオーバー・セットにより使用される TCP/IP アドレス
PortNumber	キュー・マネージャーのリスナーにより使用される TCP/IP ポート番号
TimeOut	このタイムアウト値は EndCommand プロシージャに渡されます。289ページの『フェールオーバー・セットにより使用されるコマンド・プロシージャ』を参照してください。
StartCommand	キュー・マネージャーを始動するのに使用されるコマンド・プロシージャ
EndCommand	キュー・マネージャーを終了するのに使用されるコマンド・プロシージャ
TidyCommand	キュー・マネージャーの障害後に OpenVMS ノードが存続しているノードの整理に使用されるコマンド・プロシージャ
LogDirectory	StartCommand、EndCommand、および TidyCommand プロシージャで作成されるログ・ファイルを保持するディレクトリ。
NodeCount	フェールオーバー・セットにあるノードの数。このフィールドの後に定義されるノード・トリプレットの数は、この値に対応しなければなりません。サポートされるノードの最大数は4つです。
NodeName	ノードのノード名。これは、SCSNODE OpenVMS システム・パラメーターに指定される値です。

表9. FAILOVER.INI ファイル内のフィールドの説明 (続き)

フィールド名	説明
Interface	Digital TCP/IP Services for OpenVMS TCP/IP スタックを使用する場合の、ノードの TCP/IP インターフェース名。これは、 <code>\$tcpip show interface</code> コマンドの出力から取得できます。TCPware for OpenVMS TCP/IP または Multinet for OpenVMS TCP/IP スタックを使用する際には、このフィールドが使用されることはないものの、デフォルト値 <code>we0</code> を指定しなければなりません。(構成ファイルからこのフィールドを除去しないでください。)
Priority	これは、フェールオーバー・セット内でこのノードに指定された優先順位です。この値の範囲は、1 ~ 10 です。値 1 は、優先順位が最も高くなります。複数のノードの同じ優先順位を指定することができます。障害が発生した場合、あるいは、 <code>failover -s</code> または <code>-f</code> コマンドで特定のノードが指定されていない場合には、キュー・マネージャーは、使用可能な優先順位の最も高いノードで始動されます。

フェールオーバー・セットにより使用されるコマンド・プロシージャー

フェールオーバー・セットは、3 つのコマンド・プロシージャーを使用して、いくつかの機能をインプリメントします。これらのコマンド・プロシージャーの位置は、`FAILOVER.INI` 構成ファイルのフィールド名 `StartCommand`、`EndCommand`、および `TidyCommand` で指定されています。これらのコマンド・プロシージャーのそれぞれのテンプレート・ファイル、**START_QM.TEMPLATE**、**END_QM.TEMPLATE**、および **TIDY_QM.TEMPLATE** は、`MQS_EXAMPLES` に提供されています。これらのファイルは、417ページの『付録F. OpenVMS クラスタ・フェールオーバー・セット・テンプレート』にリストされています。

コマンド・プロシージャーには、5 つまたは 6 つのパラメーターが渡されます。これらは、表10 にリストされています。

表10. コマンド・プロシージャーに渡されるパラメーター

パラメーター	値
P1	キュー・マネージャー名
P2	キュー・マネージャー・ディレクトリー名
P3	クラスタの TCP/IP アドレス
P4	ノードのインターフェース名
P5	listener のポート番号

コマンド・プロシージャー

表 10. コマンド・プロシージャーに渡されるパラメーター (続き)

パラメーター	値
P6	キュー・マネージャーの終了のタイムアウト (EndCommand プロシージャーのみ)

StartCommand プロシージャーは、以下の状況でキュー・マネージャーを始動するのに使用されます。

- **failover** コマンドの `-s` フラグで明示的に指定される場合
- キュー・マネージャーが、**failover** コマンドの `-f` フラグを使用して、別の OpenVMS ノードに移動される場合。
- キュー・マネージャーの障害の後に、自動的に再始動される場合

デフォルトでは、StartCommand プロシージャーは、フェールオーバー・セット TCP/IP アドレスを構成してキュー・マネージャーを実行し、**strmqm -m queuemanagername** コマンドを使用して、キュー・マネージャーを始動します。システム要件によっては、コマンド・プロシージャーは以下の方法で変更できます。

- **strmqm** コマンドを変更する。
- コマンドを追加して、listener などの追加の MQSeries プロセスを始動する。
- アプリケーション・プロセスを始動するコマンドを追加する。

キュー・マネージャーの始動後このキュー・マネージャーがモニターされるようにするには、StartCommand プロシージャーは、状況 1 で終了しなければなりません。

EndCommand プロシージャーは、以下の状況でキュー・マネージャーを終了するのに使用されます。

- **failover** コマンドの `-e` フラグで明示的に指定される場合
- キュー・マネージャーが、**failover** コマンドの `-f` フラグを使用して、別の OpenVMS ノードに移動される場合

デフォルトでは、EndCommand プロシージャーは、**endmqm -i queuemanagername** コマンドを使用してキュー・マネージャーを終了しようとします。構成ファイルで指定されているタイムアウト期間内にキュー・マネージャーが終了しなかった場合、プロシージャーは、**endmqm -p queuemanagername** コマンドを使用してキュー・マネージャーを終了しようとします。さらにタイムアウト期間が経過してもキュー・マネージャーが終了しない場合には、このキュー・マネージャーは、Execution Controller プロセスを

削除することにより終了されます。キュー・マネージャーが終了すると、フェールオーバー・セット TCP/IP アドレスは構成解除されます。 **endmqm** コマンドを使用してキュー・マネージャーが正常に終了すると、状況 **SS\$_NORMAL** が戻されます。 Execution Controller の削除によりキュー・マネージャーが終了した場合、状況 **SS\$_ABORT** が戻されます。3 度目のタイムアウト期間が経過してもキュー・マネージャーが終了しなかった場合には、状況 **SS\$_TIMEOUT** が戻されます。これらの状況は、監視プログラム・フェールオーバー・モニターにより、EndCommand プロシージャーの結果を判別したり、それに従ってフェールオーバー・セットの状態を設定したりするのに使用されます。システム要件によっては、コマンド・プロシージャーは以下の方法で変更できます。

- コマンドを追加して、リスナーなどの追加の MQSeries プロセスを終了します。
- コマンドを追加して、アプリケーション・プロセスを終了します。

TidyCommand プロシージャーは、キュー・マネージャーに障害が発生したものの、OpenVMS の実行が継続する場合に、OpenVMS ノード上を整理するために使用されます。

デフォルトでは、TidyCommand プロシージャーは、フェールオーバー・セット TCP/IP アドレスを構成解除します。システム要件によっては、コマンド・プロシージャーは以下の方法で変更できます。

- コマンドを追加して、リスナーなどの依然として実行中の任意の MQSeries プロセスを終了します。
- コマンドを追加して、依然として実行中のアプリケーション・プロセスを終了します。

テンプレート・ファイルは、デフォルトで、TCP/IP アドレスの構成と構成解除を Digital TCP/IP Services for OpenVMS のコマンドを使って行うように設定されています。TCPware for OpenVMS または MultiNet for OpenVMS を使用している場合、Digital TCP/IP Services for OpenVMS コマンドをコメント化 (非活動化) し、TCPware for OpenVMS または MultiNet for OpenVMS コマンドをコメント解除 (活動化) してください。

フェールオーバー・セットの管理

フェールオーバー・セットは、SYSTEM アカウントまたは MQSeries 管理アカウントから管理されなければなりません。フェールオーバー・セットは、2 つのコマンド、DCL **runmqfm** および **failover** を使用して管理されます。**runmqfm** コマンドは **failover** モニターを始動するのに使用され、**failover**

コマンド・プロシージャ

コマンドは、その他のすべての管理タスクを実行します。これらのコマンドについては、367ページの『`runmqfm` (フェールオーバー・モニターの開始)』および351ページの『フェールオーバー (フェールオーバー・セットの管理)』で説明しています。

フェールオーバー・モニターの始動

フェールオーバー・モニターは、フェールオーバー・モニターの始動が必要な OpenVMS ノード上で `runmqfm` コマンドを実行することにより始動されます。たとえば、キュー・マネージャー TESTQM についてフェールオーバー・モニターを始動するには、以下のコマンドを使用します。

```
$ runmqfm -m TESTQM
```

これにより、キュー・マネージャー名に基づき、末尾に `_FM` が付いた切り離されたプロセスが作成されます。この例では、プロセス名は `TESTQM_FM` です。このプロセスは、`monmq` アクティブ表示でリストされます。

ログ・ファイルが必要な場合には、`runmqfm` コマンドの出力をリダイレクトすることによりこれを指定することができ、`-d` フラグを指定すると追加のデバッグ情報をログ・ファイルに表示できます。たとえば、次のとおりです。

```
$ runmqm -m TESTQM -d > sys$manager:fm.log
```

`runmqfm` コマンドは、フェールオーバー・モニター・プロセスを始動するだけであり、キュー・マネージャーは始動しないことに注意してください。

フェールオーバー・セット内のキュー・マネージャーの始動

フェールオーバー・セット内でキュー・マネージャーを始動するには、少なくとも 1 つのフェールオーバー・モニターが実行していなければならず、キュー・マネージャーを始動しようとしているノードにフェールオーバー・モニターが実行していなければなりません。キュー・マネージャーは、`failover` コマンドの `-s` フラグを使用して始動されます。このコマンドは、フェールオーバー・セット内にある任意の OpenVMS ノードから実行できます。たとえば、ノード BATMAN でキュー・マネージャー TESTQM を始動したい場合には、以下のコマンドを使用してください。

```
$ failover -m TESTQM -n BATMAN -s
```


使用可能な最高の優先順位の OpenVMS ノード上でキュー・マネージャーを始動する必要がある場合、コマンドから `-n` フラグを除去してください。たとえば、次のとおりです。

```
$ failover -m TESTQM -s
```

(任意のノード上で) キュー・マネージャーにフェールオーバー・モニターがいったん始動されると、`strmqm` コマンドを使用してキュー・マネージャーを始動しようとする試みは失敗します。ただし、キュー・マネージャーについてすべてのフェールオーバー・モニターが停止されたら、`strmqm` コマンドは普通どおりに使用できます。

フェールオーバー・セット内のキュー・マネージャーの終了

フェールオーバー・セット内でキュー・マネージャーを終了するには、キュー・マネージャーが実行しているノードにフェールオーバー・モニターが実行していなければなりません。キュー・マネージャーは、`failover` コマンドの `-e` フラグを使用して終了されます。このコマンドは、フェールオーバー・セット内にある任意の OpenVMS ノードから実行できます。たとえば、キュー・マネージャー TESTQM を終了したい場合には、以下のコマンドを使用してください。

```
$ failover -m TESTQM -e
```

(任意のノード上で) キュー・マネージャーにフェールオーバー・モニターがいったん始動されると、`endmqm` コマンドを使用してキュー・マネージャーを終了しようとする試みは失敗します。ただし、キュー・マネージャーについてすべてのフェールオーバー・モニターが停止されたら、`endmqm` コマンドは普通どおりに使用できます。

フェールオーバー・セット内のキュー・マネージャーの移動

フェールオーバー・セット内でキュー・マネージャーを移動するとは、現在キュー・マネージャーが実行しているノードでこれを停止し、それから、フェールオーバー・セット内の別のノードでこれを始動し直すことを意味します。フェールオーバー・セット内でキュー・マネージャーを移動するには、少なくとも 1 つのフェールオーバー・モニターが現在実行していなければならず、キュー・マネージャーを移動しようとしているノードにフェールオーバー・モニターが実行していなければなりません。

キュー・マネージャーの移動

キュー・マネージャーは、**failover** コマンドの **-f** フラグを使用して移動されます。このコマンドは、フェールオーバー・セット内にある任意の OpenVMS ノードから実行できます。たとえば、キュー・マネージャー TESTQM をノード ROBIN に移動させたい場合には、以下のコマンドを使用してください。

```
$ failover -m TESTQM -n ROBIN -f
```

使用可能な最高の優先順位の OpenVMS ノードにキュー・マネージャーを移動する必要がある場合、コマンドから **-n** フラグを除去してください。たとえば、次のとおりです。

```
$ failover -m TESTQM -f
```

フェールオーバー・セットの状態の表示

フェールオーバー・セットの全体の状態を説明するのに、3 つのタイプの状態があります。

- フェールオーバー・セット・キュー・マネージャーの状態
- フェールオーバー・セット・ノード・キュー・マネージャーの状態 (それぞれのノードに 1 つ)
- フェールオーバー・セット・ノード・モニターの状態 (それぞれのノードに 1 つ)

それぞれの状態で指定可能な値は、以下の 3 つのテーブルで説明されています。

表 II. フェールオーバー・セット・キュー・マネージャーの状態

状態	説明
STOPPED	フェールオーバー・セットでキュー・マネージャーが始動されたことはありません。あるいは、キュー・マネージャーは、完全にシャットダウンされています。
STARTED	フェールオーバー・セットでキュー・マネージャーが始動されました。キュー・マネージャーの障害が発生した場合、フェールオーバー・セットは、キュー・マネージャーを再始動しようとしません。

表 12. フェールオーバー・セット・ノード・キュー・マネージャーの状態

状態	説明
AVAILABLE	他のノードで障害が発生した場合には、このノードでキュー・マネージャーを始動できます。
RUNNING	キュー・マネージャーは、このノードで実行しています。
EXCLUDED	ノード自体に障害はないものの、このノードにおけるキュー・マネージャーの停止の方法が不完全です。他のノードでキュー・マネージャーに障害が起きた場合には、このノードで再始動されることはありません。

表 13. フェールオーバー・セット・ノード・モニター状態

状態	説明
STARTED	フェールオーバー・モニターはこのノード上で実行中ですが、これは監視プログラムではありません。
WATCHING	フェールオーバー・モニターはこのノード上で実行中であり、これは監視プログラムです。
STOPPED	このノード上で実行しているフェールオーバー・モニターはありません。

フェールオーバー・セットの状態は、**failover** コマンドの **-q** フラグを使用して表示されます。少なくとも 1 つのフェールオーバー・モニター・プロセスが実行していなければならない、このコマンドは、フェールオーバー・セット内にある任意のノードから実行できます。たとえば、キュー・マネージャー TESTQM についてフェールオーバー・セットの状態を表示するには、以下のコマンドを使用してください。

```
$ failover -m TESTQM -q
```

このコマンドの出力例を以下に示します。

DCL シンボルの設定

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

フェールオーバー・セットの状態への DCL シンボルの設定

フェールオーバー・セットを制御するために DCL コマンド・プロシージャを作成しなければならない場合があります。 **failover** コマンドの **-I** フラグは、3つのローカル DCL シンボルを設定して、フェールオーバー・セットの状態を示します。これらのシンボルは、キュー・マネージャーの状態に基づいた条件に従ったアクションを実行するのに使用されます。少なくとも1つのフェールオーバー・モニター・プロセスが実行していなければならず、このコマンドは、フェールオーバー・セット内にある任意のノードから実行できます。セットされるシンボルは、表14に示されています。

表 14. DCL シンボルおよび説明

DCL シンボル名	説明
MQS\$QMGR_NODE	キュー・マネージャーを実行している OpenVMS に設定されます。どのキュー・マネージャーも実行していない場合には、ヌル・ストリングになります。

表 14. DCL シンボルおよび説明 (続き)

DCL シンボル名	説明
MQS\$AVAILABLE_NODES	キュー・マネージャーを実行するのに使用できる OpenVMS ノードのリストに設定されます。これは、キュー・マネージャーの状態が AVAILABLE で、フェールオーバー・モニターが実行中のノードです。
MQS\$MONITOR_NODES	フェールオーバー・モニターが実行している OpenVMS ノードのリストに設定されます。

たとえば、キュー・マネージャー TESTQM についてフェールオーバー・セットの状態にシンボルを設定するには、以下のコマンドを使用してください。

```
$ failover -m TESTQM -l
```

シンボルの設定の結果例を以下に示します。

```
MQS$AVAILABLE_NODES = ""
MQS$MONITOR_NODES = "BATMAN,ROBIN"
MQS$QMGR_NODE = "BATMAN"
```

フェールオーバー・モニター・プロセスの停止

OpenVMS ノード上のフェールオーバー・モニター・プロセスは、**failover** コマンドの **-h** フラグを使用して停止できます。このコマンドは、フェールオーバー・セット内にある任意のノードから実行できます。たとえば、ノード **BATMAN** でキュー・マネージャー **TESTQM** のフェールオーバー・モニターを停止するには、以下のコマンドを使用してください。

```
$ failover -m TESTQM -n BATMAN -h
```

停止しようとしているフェールオーバー・モニターが監視プログラム・フェールオーバー・モニターである場合、別のフェールオーバー・モニターが存在するなら、それが監視プログラムになります。停止しようとしているフェールオーバー・モニターがフェールオーバー・セットで最後のフェールオーバー・モニターである場合、このフェールオーバー・セットは、ライブではなくなります。この場合、キュー・マネージャーは **strmqm** および **endmqm** コマンドを使用して、始動したり停止したりできるようになります。フェールオーバ

フェールオーバー・モニターの停止

ー・コマンドの `-h` フラグで、キュー・マネージャーを終了することはありません。停止しようとしているフェールオーバー・モニターのある OpenVMS ノード上でキュー・マネージャーが実行中の場合には、このキュー・マネージャーは実行を継続します。

更新の進行中のコマンドの実行

failover コマンドでフラグ `-s`、`-e`、`-f`、および `-c` が指定されると、これは、更新と見なされます。これらのコマンドが進行中の際には、進行フラグでの更新は、監視プログラム・フェールオーバー・モニターにより設定されます。このフラグが設定されると、同時更新は許可されないため、他の更新およびフェールオーバー・モニター停止コマンドは失敗します。`-q` および `-l` フラグなどの非更新コマンドは、更新が進行中の場合には、作業を継続します。

まれなことですが、障害の起きた更新が、進行フラグ・セットに更新を残したままにしてしまう可能性があります。**failover** コマンドに `-u` フラグを指定すると、進行フラグでの更新をクリアできます。このコマンドを使用する際には、十分注意してください。たとえば、キュー・マネージャー TESTQM の進行フラグの更新をクリアするには、以下のコマンドを使用してください。

```
$ failover -m TESTQM -u
```

フェールオーバー・セットの状態の変更

状況によっては、フェールオーバー・セットの状態を変更することが必要な場合もあります。これは、**failover** コマンドの `-c` フラグを使用して実行されます。障害が発生した後にノードのキュー・マネージャー状態が EXCLUDED になっており、ノードを終結処理した後でこの状態を AVAILABLE に戻そうとしている場合などに、これが必要になるでしょう。たとえば、ノード BATMAN 上のキュー・マネージャー TESTQM について、状況を AVAILABLE に変更するには、以下のコマンドを使用してください。

```
$ failover -m TESTQM -n BATMAN -c -qmgr available
```

また、あるノードを、キュー・マネージャーを実行できる候補から一時的に除外したい場合には、ノード・キュー・マネージャー状態を AVAILABLE から EXCLUDED に変更することもできます。たとえば、ノード BATMAN 上のキュー・マネージャー TESTQM について、状況を EXCLUDED に変更するには、以下のコマンドを使用してください。

```
$ failover -m TESTQM - n BATMAN -c -qmgr excluded
```

また、他の状態すべてを変更することも可能ですが、要求される変更が、実行中のシステムと一貫性のある場合にしか、変更は有効になりません。たとえば、ノード上でフェールオーバー・モニターが実行中の場合に、このモニターの状態の STOPPED に変更しようとしても、この変更は有効になりません。監視プログラム・フェールオーバー・モニターは、30 秒ごとに保全性を検査し、実行中のシステムの間には矛盾がある場合には、状態に変更と実行するため、ノード・キュー・マネージャーの状態を EXCLUDED と AVAILABLE との間で変更する以外は、状態の変更コマンドを使用する必要はないでしょう。

ICC アソシエーションのセキュリティの設定

フェールオーバー・セット・モニターおよびクライアント・プログラムは、OpenVMS Intra Cluster Communication (ICC) 呼び出しを使用して、メッセージを渡します。許可のないユーザーが、フェールオーバー・モニター・プロセスにメッセージを送信しないようにするには、

SYS\$STARTUP:ICC\$SYSTARTUP.COM コマンド・プロシージャーで ICC アソシエーションを構成しなければなりません。

それぞれのフェールオーバー・セットは、2 つのアソシエーション名を使用します。1 つは、監視プログラム・フェールオーバー・モニターとの通信に使用されるキュー・マネージャーの名前が付いたもの、そして、もう 1 つは、それぞれのフェールオーバー・モニターとの通信に使用されるキュー・マネージャーの名前に `_MQ_FM` が付加されたものです。

フェールオーバー・セットのそれぞれのノードについて、**ICC\$SYSTARTUP.COM** で必要なエントリーの例を 300 ページの図 22 に示します。フェールオーバー・セットには BATMAN および ROBIN という 2 つのノードがあり、キュー・マネージャー名は TESTQM です。

セキュリティーの設定

```
$! ----- 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 nodes in the
$! cluster
$!
$!
$ @SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT ICC$::"TESTQM" -
"/owner=MQM/acl=((id=MQM,access=open+access),(id=*,access=none))"
$!
$ @SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT 'nodename'::"TESTQM" -
"/owner=MQM/acl=((id=MQM,access=open+access),(id=*,access=none))"
$!
$ @SYS$MANAGER:ICC$CREATE_SECURITY_OBJECT 'nodename'::"TESTQM_MQ_FM" -
"/owner=MQM/acl=((id=MQM,access=open+access),(id=*,access=none))"
$!
$ set security/class=logicial_name_table icc$registry_table -
/acl=(id=MQM,access=read+write)
$!
$ GOTO EXIT
$!
```

図 22. ICC\$SYSTARTUP.COM で必要なエントリーのサンプル

ICC アソシエーション名の長さが 31 文字に制限されている場合、フェールオーバー・セットで使用される際に、MQSeries キュー・マネージャー名にサポートされている最大長は 25 文字であることに注意してください。ICC アソシエーションのセキュリティーを設定することについての詳細は、*OpenVMS System Manager's Manual* に説明されています。

フェールオーバー・セットのトラブルシューティング

start_qm.com、**end_qm.com**、および **tidy_qm.com** プロシージャが実行されると、**failover.ini** 構成ファイルで指定される **LogDirectory** にログ・ファイルが書き込まれます。ログ・ファイルの名前は、**qmgrname_procedurename.log** です。たとえば、キュー・マネージャー名が **TESTQM** の場合、**start_qm.com** コマンド・プロシージャは、ログ・ファイル **testqm_start_qm.log** を生成します。

デフォルトでは、フェールオーバー・モニターはログ・ファイルを生成しません。が、**runmqfm** コマンドでリダイレクト・パラメーターを使用してログ・ファイルを指定することができます。**runmqfm** コマンドに **-d** パラメーターを指定すると、追加のデバッグ情報をファイルに書き込むことができます。

FDC ファイルが生成されたかどうかは、**MQS_ROOT:[MQM.ERRORS]** で検索してください。

フェールオーバー・セットと MultiNet for OpenVMS の使用

フェールオーバー・セットと共に Multinet for OpenVMS を使用するには、クラスター別名サービスが使用可能になっていなければなりません。クラスター別名サービスを使用可能にするには、以下のコマンドを使用します。

```
$ MULTINET CONFIGURE/SERVERS
SERVER-CONFIG> ENABLE CLUSTERALIAS
SERVER-CONFIG> EXIT
```

テンプレート・コマンド・ファイルでは、クラスター別名アドレスは 1 つしかなく、これはフェールオーバー・セットにより使用されることを前提としています。ただし、他のクラスター別名アドレスが使用されている場合には、コマンド・プロシージャーを変更して、**MULTINET_CLUSTER_IP_ALIASES** 論理名に他のアドレスを使用する必要があります。

フェールオーバー・セットの使用例

以下は、OpenVMS クラスターにある 2 つのノード、**BATMAN** および **ROBIN** を、キュー・マネージャー **TESTQM** のフェールオーバー・セットに構成する例です。フェールオーバー・セットの **TCP/IP** アドレスは **10.20.30.40** であり、**TCP/IP** リスナー・ポートは **1414** です。ノード **BATMAN** は 1 次ノード、**ROBIN** は 2 次ノードに指定されています。最初は **BATMAN** でキュー・マネージャーが始動され、このキュー・マネージャーが失敗した場合には、**ROBIN** で再始動されます。キュー・マネージャーが **ROBIN** で実行している場合、この **ROBIN** がリブートされても、キュー・マネージャーが **BATMAN** にフェイルバックされることはありません。キュー・マネージャーを実行しているノードがシャットダウンされると、このキュー・マネージャーは終了し、フェールオーバー・モニターは停止します。ノードが実行していない場合でも、キュー・マネージャーがシャットダウンすることなく、フェールオーバー・モニターのみ停止されます。

フェールオーバー・セットの例

failover.template のカスタマイズ

failover.template ファイルは以下のように変更され、
mq_s_root:[mqm.qmgrs.testqm] failover.ini としてコピーされます。

```
# 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
#
```

図 23. FAILOVER.INI 構成ファイルを作成するための failover.template (1/2)

```

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

```

図 23. *FAILOVER.INI* 構成ファイルを作成するための *failover.template* (2/2)

フェールオーバー・セット・コマンド・プロシージャーの変更

このコマンド・プロシージャーは、テンプレート・ファイルからコピーされます。変更された点は、**start_qm.com** でのリスナーの始動、および **end_qm.com** でのリスナーの終了がコメント解除 (活動化) されていることです。停止および始動するアプリケーションがある場合には、適切なコマンドをコマンド・プロシージャーに追加することができます。

フェールオーバー・セット始動コマンド・プロシージャーの例、

start_failover_set.com

start_failover_set.com コマンド・プロシージャーは、各ノードでフェールオーバー・モニターを始動するのに使用され、条件によってはキュー・マネージャーを始動します。このプロシージャーは、**MQS_STARTUP.COM** コマンド・プロシージャーが実行された後に、システム始動から呼び出されます。プロシージャーには、2 つのパラメーター、キュー・マネージャー名および 1 次ノード名が渡されます。この場合、以下のように呼び出されます。

```
$@start_failover_set testqm batman
```

start_failover_set.com コマンド・プロシージャーは、フェールオーバー・モニターを始動してから、**failover** コマンドで -1 パラメーターを使用して、フェールオーバー・セットの状態を検出します。**failover** コマンドの実行時にフェールオーバー・モニターが完全に始動されない場合もあり、その場合には、1

フェールオーバー・セットの例

秒間隔でこのコマンドが最大 3 回再試行されることに注意してください。したがって、このノードが 1 次ノードであり、キュー・マネージャーが始動されていない場合には、キュー・マネージャーは **failover** コマンドの **-s** パラメーターを使用して始動されます。

```

$on error then exit
@$ysy$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
$!

```

図 24. start_failover_set コマンド・プロシージャ (1/2)

フェールオーバー・セットの例

```
$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
```

図 24. *start_failover_set* コマンド・プロシージャー (2/2)

フェールオーバー・セット終了コマンド・プロシージャーの例、

end_failover_set.com

end_failover_set.com コマンド・プロシージャーは、各ノードで、キュー・マネージャー、そしてフェールオーバー・モニターを条件に応じて終了するの

に使用されます。このプロシージャーは、 **MQS_SHUTDOWN.COM** コマンド・プロシージャーが実行される前に、サイト固有のシャットダウンから呼び出されます。プロシージャーには、1つのパラメーター、キュー・マネージャー名が渡されます。この場合、以下のように呼び出されます。

```
$@start_failover_set testqm
```

end_failover_set.com コマンド・プロシージャーは、 **failover** コマンドの **-I** パラメーターを使用して、フェールオーバー・セットの状態を取得します。このノード上でキュー・マネージャーが実行している場合には、これは終了します。それから、フェールオーバー・モニターが停止します。

フェールオーバー・セットの例

```
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
$!
$this_node=f$getsysi("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
```

図 25. *end_failover_set* コマンド・プロシージャ

第2部 リファレンス

第17章 MQSeries 制御コマンド	313	strmqtrc (MQSeries トレースの開始).	387
MQSeries オブジェクトを命名する際の規則	313		
オブジェクト・ファイルの表示	314		
構文図の見方	314		
構文ヘルプ	316		
例	316		
MQSeries 戻りコード	317		
crtmqcvx (データ変換)	318		
crtmqm (キュー・マネージャーの作成)	320		
dltmqm (キュー・マネージャーの削除)	326		
dmpmqlog (ダンプ・ログ)	328		
dspmqaout (許可の表示)	330		
dspmqcsv (コマンド・サーバーの表示)	335		
dspmqls (MQSeries ファイルの表示)	337		
dspmqtrc (MQSeries 定様式トレース出力の表 示)	340		
dspmqtrn (MQSeries トランザクションの表 示)	341		
endmqcsv (コマンド・サーバーの終了)	343		
endmqlsr (リスナーの終了)	345		
endmqm (キュー・マネージャーの終了)	346		
endmqtrc (MQSeries トレースの終了)	349		
フェールオーバー (フェールオーバー・セッ トの管理)	351		
rcdmqimg (メディア・イメージの記録)	355		
rcrmqobj (オブジェクトの再作成)	358		
rsvmqtrn (MQSeries トランザクションの解決)	361		
runmqchi (チャンネル開始プログラムの実行)	363		
runmqchl (チャンネルの実行)	364		
runmqdlq (送達不能キュー・ハンドラーの実 行)	365		
runmqfm (フェールオーバー・モニターの開 始)	367		
runmqlsr (リスナーの実行)	369		
runmqsc (MQSeries コマンドの実行)	371		
runmqtmc (クライアント・トリガー・モニタ ーの開始)	375		
runmqtrm (トリガー・モニターの開始)	376		
setmqaut (許可の設定 / リセット)	377		
strmqcsv (コマンド・サーバーの開始)	384		
strmqm (キュー・マネージャーの開始)	385		

第17章 MQSeries 制御コマンド

この章は、MQSeries for Compaq OpenVMS で使用する制御コマンドについての参照資料です。この章に示すコマンドは、すべて OpenVMS DCL プロンプトから発行できます。

コマンド名およびフラグは大文字小文字の区別をしません。コマンド名およびフラグを大文字、小文字または大文字と小文字の組み合わせで入力することができます。しかし、制御コマンドのパラメーター (キュー名など) は大文字小文字の区別をする場合があります。詳細については、30ページの『制御コマンドでの大文字小文字の区別』を参照してください。

最後にリポートした後で `mqs_startup` を実行していなければ、制御コマンドは使用できません。

MQSeries オブジェクトを命名する際の規則

一般に、MQSeries オブジェクトの名前は、最高 48 文字までです。この規則は、次のすべてのオブジェクトに当てはまります。

- キュー・マネージャー (ただし、キュー・マネージャーが OpenVMS クラスター・フェールオーバー・セットによりサポートされている場合には、最大長は 25 文字です。)
- キュー
- プロセス定義
- 名前リスト
- クラスター

チャンネル名の最大長は 20 文字です。

すべての MQSeries の名前に使用できる文字は、次のとおりです。

- 大文字 A-Z
- 小文字 a-z
- 数値 0 - 9
- ピリオド (.)
- 下線 (_)
- 斜線 (/) (注 1を参照)
- パーセント記号 (%) (注 1を参照)

注:

1. 斜線とパーセント記号は、特殊文字です。ただし、名前の先頭に斜線およびパーセント記号を使用することはできません。名前の中でこの 2 つの文字のいずれかを使用する場合、名前は必ず二重引用符で囲む必要があります。
2. 先行空白も組み込み空白も許されていません。
3. 各国語文字は許されていません。
4. 名前は二重引用符で囲むことができますが、これは、特殊文字が名前に含まれている場合、あるいは大文字と小文字が区別される場合にのみ必要です。

オブジェクト・ファイルの表示

MQSeries キュー、キュー・マネージャー、またはプロセス・オブジェクトは、それぞれファイルによって表されます。これらのオブジェクト名は必ずしも有効なファイル名ではないので、キュー・マネージャーは、必要に応じてそのオブジェクト名を有効なファイル名に変換します。これについては、28ページの『MQSeries のファイル名についての理解』で説明しています。

オブジェクトの実ファイル名を表示する方法については、337ページの『dspmqfls (MQSeries ファイルの表示)』を参照してください。

構文図の見方

この章では、構文図（「幹線」図ともいう）を示しています。

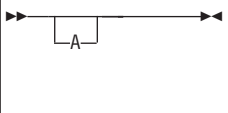
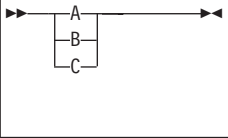
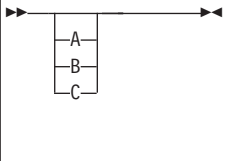
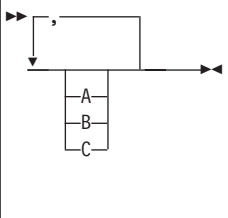
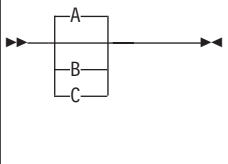
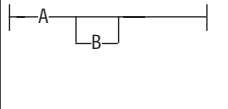
各構文図は 二重の右矢印で始まり、右矢印と左矢印の対で終了します。1 つの右矢印で始まる行は、継続行です。構文図を見る場合は左から右、上から下へ、矢印の方向に従って進みます。

構文図で使用される他の規則は次のとおりです。

表 15. 構文図の見方

規則	意味
▶▶—A—B—C————▶▶	値 A、B、および C を指定しなければなりません。必須の値は、構文図の主線上に示されます。

表 15. 構文図の見方 (続き)

規則	意味
	<p>値 A を指定できます。オプションの値は構文図の主線の下に示されます。</p>
	<p>値 A、B、および C は代替であり、いずれかの値を指定しなければなりません。</p>
	<p>値 A、B、および C は代替であり、いずれかの値を指定することができます。</p>
	<p>値 A、B、および C のいずれかまたは複数の値を指定することができます。複数値または繰り返し値の必須の区切り記号 (この例ではコンマ (,)) はどれも矢印の上に表示されます。</p>
	<p>値 A、B、および C は代替であり、いずれかの値を指定することができます。値をどれも指定していない場合は、デフォルト値 A (主線の上に示される値) が使用されます。</p>
<p>▶ Name ▶▶</p> <p>Name:</p> 	<p>構文断片 Name は、主構文図とは別に示されます。</p>

構文ヘルプ

この章に示されているコマンドの構文についてのヘルプは、該当のコマンドとその後疑問符を入力すると得られます。MQSeriesは、選択したコマンドに必要な構文をリストすることにより、応答します。構文には、該当のコマンドに関連したすべてのパラメーターと変数が示されます。パラメーターが必須かどうかを指示するために、各種の括弧を使用します。たとえば、次のとおりです。

```
CmdName [-x OptParam ] ( -c | -b ) { -p principal } argument
```

説明

CmdName	ヘルプが要求されたコマンドの名前です。
[-x OptParam]	大括弧は、これがオプション・パラメーターであることを示します。
(-c -b)	必須のフィールド。この場合、フラグ c または b のいずれかを選択しなければなりません。
{ -p principal }	提供できるオプションの変数のリスト。ただし、これが表示される場合には、コマンドの入力時に少なくとも 1 つの変数を提供しなければなりません。
argument	このコマンドに提供する必要のある引き数。照会の応答として表示される場合には、これは必須です。

例

1. `endmqm ?` を入力すると、次のような応答があります。

```
endmqm [-z] [-c | -i | -p] QMgrName
```

2. `rcdmqimg ?` を入力すると、次のような応答があります。


```
rcdmqimg [-z] [-m QMgrName] -t ObjType [GenericObjName]
```

MQSeries 戻りコード

`crtmqm` などの、大半の MQSeries コマンドは、終了時に状況表示行を表示し、コマンドが正常に実行されたか、障害が発生したかを示します。

DCL コマンド・ファイルでコマンドの状況を検査する場合には、MQSeries プログラムから戻される状況値を解釈する必要があります。

MQSeries 戻りコードは、メッセージ・ファイル `SYS$MESSAGE:MQS_MSG.EXE` に定義されています。

ファイルで戻りコードに関連したメッセージ・テキストにアクセスするには、DCL `SET MESSAGE` コマンドを使用しなければなりません。このコマンドは、プロセスのメッセージ・テーブルにメッセージ・コードをロードします。たとえば、次のとおりです。

```
$ SET MESSAGE SYS$MESSAGE:MQS_MSG.EXE
```

この後、`F$MESSAGE` 字句機能を使用して、MQSeries の戻りコードのテキストを出力できます。たとえば、次のとおりです。

```
$ strmqm)(*bad-qm-name&##
The queue manager name is either not valid or not known
$ WRITE SYS$OUTPUT F$MESSAGE($STATUS)
%MQS-F-CSPRC_Q_MGR_NAM, Queue manager name error
```

OpenVMS 戻りコードを、MQSeries for OS/2 または UNIX システムで使用される戻りコードに変換するには、以下の DCL 式を使用できます。

```
$ RC = $STATUS / 8 .AND. %xFFF
```

たとえば、次のとおりです。

crtmqcvx

```
$ crtmqm &*)*(  
The queue manager name is either not valid or not known  
$ RC = $STATUS / 8 .AND. %xFFF  
$ SHOW SYMBOL RC  
RC = 72 Hex = 00000048 Octal = 0000000110
```

crtmqcvx (データ変換)

目的

crtmqcvx コマンドは、データ・タイプ構造のデータ変換を実行するコード断片を作成するために使用します。このコマンドは、C 構造体を変換するために出口で利用できるC関数を生成します。

このコマンドは、変換される構造体を含んでいる入力ファイルを読み取ってから、それらの構造を変換するコード断片を含んでいる出力ファイルを書き込みます。

このコマンドおよび使用方法の詳細については、*MQSeries アプリケーション・プログラミング・ガイド*を参照してください。

構文

```
►►—crtmqcvx—SourceFile—TargetFile—◄◄
```

必須パラメーター

SourceFile

変換されるC構造体を含んでいる入力ファイルを指定します。

TargetFile

構造体を変換するために生成されたコード断片を含んでいる出力ファイルを指定します。

戻りコード

- 0 コマンドは正常に終了しました。
- 10 コマンドは終了しましたが、予期しない結果がでました。
- 20 処理中にエラーが起きました。

例

次の例は、ソースC構造体に対してデータ変換コマンドを使用した結果を示しています。次のコマンドが出されました。

```
crtmqcvx source.tmp target.c
```

入力ファイル `source.tmp` は、次のようなものです。

```
/* This is a test C structure which can be converted by the */  
/* crtmqcvx utility                                         */  
  
struct my_structure  
{  
    int    code;  
    MQLONG value;  
};
```

コマンドによって生成される出力ファイル `target.c` を次に示します。アプリケーションでこれらのコード断片を使用すれば、データ構造体を変換できます。ただし、これを行う場合には、コード断片が MQSeries ヘッダー・ファイル `amqsvmha.h` に与えられているマクロを使用することを理解しておく必要があります。

crtmqm

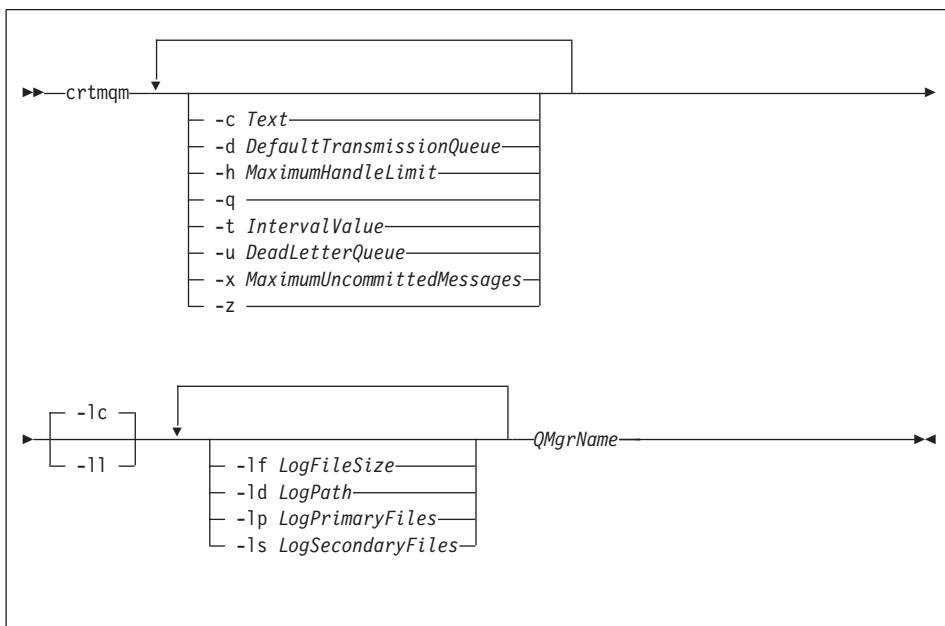
```
MQLONG Convertmy_structure(  
    PMQBYTE *in_cursor,  
    PMQBYTE *out_cursor,  
    PMQBYTE in_lastbyte,  
    PMQBYTE out_lastbyte,  
    MQHCONN hConn,  
    MQLONG  opts,  
    MQLONG  MsgEncoding,  
    MQLONG  ReqEncoding,  
    MQLONG  MsgCCSID,  
    MQLONG  ReqCCSID,  
    MQLONG  CompCode,  
    MQLONG  Reason)  
{  
    MQLONG ReturnCode = MQRC_NONE;  
  
    ConvertLong(1); /* code */  
  
    AlignLong();  
    ConvertLong(1); /* value */  
  
    Fail:  
        return(ReturnCode);  
}
```

crtmqm (キュー・マネージャーの作成)

目的

crtmqm コマンドは、ローカル・キュー・マネージャーを作成します。キュー・マネージャーが作成された場合は、**strmqm** コマンドを使ってそれを開始します。

構文



必須パラメーター

QMgrName

作成するキュー・マネージャーの名前を指定します。この名前は、最高 48 文字までです。これは、このコマンドの最後の項目にする必要があります。

オプション・パラメーター

-c *Text*

このキュー・マネージャーの記述テキストをいくつか指定します。デフォルトは、すべてブランクです。

最高 64 文字まで使用できます。大文字と小文字を混合して指定する必要がある場合には、記述を二重引用符で囲まなければなりません。

-d *DefaultTransmissionQueue*

伝送キューが宛先について明示的に定義されなかったときにリモート・メッセージが置かれるローカル伝送キューの名前を指定します。デフォルトはありません。

-h *MaximumHandleLimit*

どのアプリケーションでも同時にオープンできるハンドルの最大数を指定します。

1 から 999 999 999 までの範囲の値を指定します。デフォルト値は 256 です。

-q このキュー・マネージャーが、デフォルト・キュー・マネージャーになることを指定します。新しいキュー・マネージャーは、デフォルトとして既存のキュー・マネージャーと置き換わります。

誤ってこのフラグを使用した場合で、既存のキュー・マネージャーをデフォルト・キュー・マネージャーに戻したい場合には、MQSeries 構成ファイルの中の *DefaultQueueManager* スタンザを編集することができます。構成ファイルについては、221ページの『第13章 MQSeries の構成』を参照してください。

-t *IntervalValue*

このキュー・マネージャーが制御するすべてのキューについて、トリガー時間間隔 (ミリ秒単位) を指定します。この値は、トリガー生成メッセージを受け取った後、トリガー操作が中断される時間を指定します。つまり、あるメッセージがキューに到着してトリガー・メッセージが開始キューに入れられると、指定された時間間隔内に同じキューにメッセージが到着しても、別のトリガー・メッセージは生成されません。

このトリガー時間間隔を使用すれば、アプリケーションは、同じキューの別のメッセージを取り扱うよう警告されるまでに、トリガー条件を取り扱うための十分な時間の余裕が与えられます。生じるすべてのトリガー・イベントを見たい場合があります。その場合には、このフィールドに小さな値かまたはゼロを設定してください。

0 から 999 999 999 までの範囲の値を指定します。デフォルトは、999 999 999 ミリ秒 (11 日以上) です。効果的にデフォルトが使用されるようにすると、トリガー操作は最初のトリガー・メッセージの後、使用不可になることとなります。ただし、そのキューをサービスするアプリケーションは、キュー更新コマンドを使用してトリガー属性をリセットし、トリガー操作を使用可能にすることができます。

-u *DeadLetterQueue*

送達不能 (未配布メッセージ) キューとして使用されるローカル・キューの名前を指定します。メッセージが正しい宛先に送られない場合は、メッセージはこのキューに書き込まれます。

属性が省略された場合のデフォルトは、送達不能キューなしです。

-x *MaximumUncommittedMessages*

同期点においてコミットされないメッセージの最大数を指定します。つまり、これは次のものの合計です。

- キューから取り出すことができるメッセージの数
- キューに書き込むことができるメッセージの数
- この作業単位内で生成されたトリガー・メッセージ

この限界は、同期点以外で取り出したり書き込まれたりするメッセージには当てはまりません。

値は、1 から 10 000 までの範囲で指定します。デフォルト値は、1000 個のコミットされていないメッセージになります。

-z エラー・メッセージを抑制します。

このフラグは、通常、MQSeries の中で不要なエラー・メッセージを抑制するために使用します。このフラグを使用することによって情報が失われる可能性があるため、コマンド行からコマンドを入力する場合には、これは使用しないようお勧めします。

以下のフラグのセットは、作成中のキュー・マネージャーにより使用されるロギングを定義するのに使用されます。ログについての詳細は、185ページの『ログをリカバリーに使用する』を参照してください。

-lc 循環ロギングが使用されます。これは、デフォルトのロギングの方法です。

-ll リニア・ロギングが使用されます。

-lf *LogFileSize*

ログ・ファイルのサイズを 4 KB の倍数として指定します。最小値は 64、最大値は 16384 です。デフォルト値は 1024 であり、これは、4 MB のデフォルト・ログ・サイズになります。

-ld *LogPath*

ログ・ファイルを保持するために使用されるディレクトリーを指定します。デフォルトは、MQS_ROOT:[MQM.LOG] です。MQSeries がカスタマイズされる際にも、デフォルトを変更することができます。

ユーザー ID mqm およびグループ mqm はログ・ファイルについてすべての許可を持っている必要があります。これらのファイルの位置を変更する場合は、それらの許可を取得する必要があります。ログ・ファイルがデフォルトの位置にある場合、これは自動的に行われます。

-lp *LogPrimaryFiles*

割り振る 1 次ログ・ファイルの数を指定します。デフォルト値は 3、最小値は 2、最大値は 62 です。

-ls *LogSecondaryFiles*

割り振る 2 次ログ・ファイルの数を指定します。デフォルト値は 2、最小値は 1、最大値は 61 です。

注: ログ・ファイルの総数は、要求された数にかかわらず 63 に制限されます。

直前のパラメーターの記述に指定する制限は、MQSeries により設定される制限です。オペレーティング・システムの制限により、設定可能なログ・サイズが削減されることがあります。

戻りコード

- 0** キュー・マネージャーが作成されました。
- 8** キュー・マネージャーはすでに存在しています。
- 49** キュー・マネージャーは停止中です。
- 69** ストレージが利用不能です。
- 70** キュー・スペースが利用不能です。
- 71** 予期しないエラーです。
- 72** キュー・マネージャー名のエラーです。
- 100** ログの位置が無効です。
- 111** キュー・マネージャーが作成されました。ただし、プロダクト構成ファイル内のデフォルト・キュー・マネージャー定義の処理中に問題がありました。デフォルト・キュー・マネージャーの指定が誤りである可能性があります。
- 115** ログ・サイズが無効です。

例

1. 次のコマンドでは、`Paint.queue.manager` という名前のデフォルト・キュー・マネージャーが作成されます。これには、`Paint shop` という説明が与えられています。これは、リニア・ロギングが使用されることも指定します。

```
crtmqm -c "Paint shop" -ll -q "Paint.queue.manager"
```


- この例は、あらゆるログ・ファイルを要求します。2つの1次ログ・ファイルおよび3つの2次ログ・ファイルが指定されます。

```
crtmqm -c "Paint shop" -ll -lp 2 -ls 3 -q "Paint.queue.manager"
```

- この例では、別のキュー・マネージャー `travel` が作成されます。トリガー間隔は 5000 ミリ秒 (5 秒) に定義され、送達不能レター・キューは、`SYSTEM.DEAD.LETTER.QUEUE` になります。

```
crtmqm -t 5000 -u SYSTEM.DEAD.LETTER.QUEUE "travel"
```

トリガー・イベントが一度生成されると、追加のトリガー・イベントは 5 秒間使用不可になります。

関連コマンド

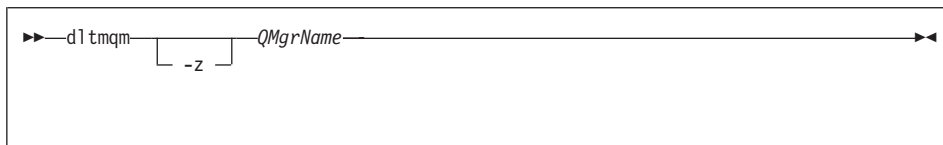
strmqm	キュー・マネージャーを開始します。
endmqm	キュー・マネージャーを終了します。
dltmqm	キュー・マネージャーを削除します。

dltmqm (キュー・マネージャーの削除)

目的

dltmqm コマンドは、指定したキュー・マネージャーを削除するために使用します。削除されるキュー・マネージャーに関連したすべてのオブジェクトも削除されます。キュー・マネージャーを削除するためには、その前に **endmqm** コマンドでそれを終了しておく必要があります。

構文



必須パラメーター

QMgrName

削除するキュー・マネージャーの名前を指定します。

オプション・パラメーター

-z エラー・メッセージを抑制します。

戻りコード

- 0** キュー・マネージャーは削除されました。
- 3** キュー・マネージャーは作成中です。
- 5** キュー・マネージャーは実行中です。
- 16** キュー・マネージャーは存在していません。
- 49** キュー・マネージャーは停止中です。
- 69** ストレージが利用不能です。
- 71** 予期しないエラーです。
- 72** キュー・マネージャー名のエラーです。
- 100** ログの位置が無効です。
- 112** キュー・マネージャーは削除されました。ただし、プロダクト構成ファイル内のデフォルト・キュー・マネージャー定義の処理中に問題がありました。デフォルト・キュー・マネージャーの指定が誤りである可能性があります。

例

1. 次のコマンドは、キュー・マネージャー saturn.queue.manager を削除します。

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

2. 次のコマンドは、キュー・マネージャー travel を削除し、コマンドによって引き起こされるメッセージを抑制します。

```
dltmqm -z "travel"
```

関連コマンド

crtmqm	キュー・マネージャーを作成します。
strmqm	キュー・マネージャーを開始します。
endmqm	キュー・マネージャーを終了します。

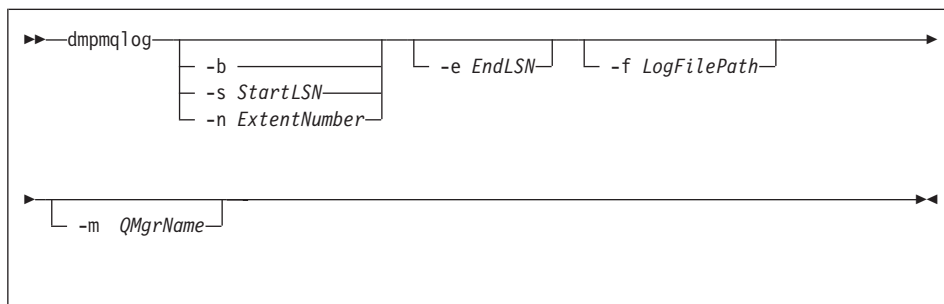
dmpmqlog (ダンプ・ログ)

目的

dmpmqlog コマンドは、MQSeries システム・ログの形式化されたバージョンのダンプをとるために使用されます。

ダンプがとられるログは、このコマンドを出すのに使用されるオペレーティング・システムと同じタイプで作成されなければなりません。

構文



オプション・パラメーター

ダンプの開始点

次のパラメーターのいずれかを使用して、ダンプを開始するログの順序番号 (LSN) を指定します。開始点が指定されていない場合、デフォルトでは、アクティブなログの最初のレコードの LSN からダンプが開始します。

-b 基本 LSN からダンプを開始すべきであることを指定します。基本 LSN は、アクティブなログの開始を含むログ・エクステントの開始を識別します。

-s *StartLSN*

ダンプを指定した LSN から開始すべきであることを指定します。LSN は、形式 *nnnn:nnnn:nnnn:nnnn* で指定されます。

循環ログを使用している場合、LSN 値は、ログの基本 LSN 値以上でなければなりません。

-n ExtentNumber

ダンプを指定したエクステント番号から開始すべきであることを指定します。エクステント番号の範囲は 0-9 999 999 でなければなりません。

このパラメーターは、*LogType* (構成ファイル *qm.ini* に記録される場合) が **LINEAR** であるキュー・マネージャーにのみ有効です。

-e EndLSN

ダンプが指定した LSN で終了すべきことを指定します。LSN は、形式 *nnnn:nnnn:nnnn:nnnn* で指定されます。

-f LogFilePath

これは、ログ・ファイルの相対ディレクトリー・パス名ではなく、絶対ディレクトリー・パス名です。指定したディレクトリーには、ログ・ヘッダー・ファイル (*amqh1ct1.lfh*) および アクティブというサブディレクトリーが入っていないなければなりません。アクティブなサブディレクトリーにはログ・ファイルが入っている必要があります。デフォルトでは、*mq5.ini* および *qm.ini* ファイルに指定したディレクトリーにログ・ファイルがあると想定しています。このオプションを使用すると、キュー・マネージャー名に明示的に **-m** オプションを指定し、しかもキュー・マネージャーのディレクトリー・パスにオブジェクト・カタログ・ファイルがある場合に限り、キュー ID と関連するキュー名がダンプに示されます。

長いファイル名をサポートするシステムでは、このファイルの名前は *qmqmobjcat* となります。キュー ID をキュー名にマップするため、ログ・ファイルの作成時に使用したファイルでなければなりません。 *qm1* というキュー・マネージャーを例にすると、オブジェクト・カタログ・ファイルはディレクトリー *MQS_ROOT:[MQM.QMGRS.QM1.QMANAGER]* にあります。マッピングを実行するには、たとえば *tmpq* という一時キュー・マネージャーを作成し、そのオブジェクト・カタログを特定のログ・ファイルと関連するファイルと置き換え、次に **dmpmqlog** を開始して、絶対ディレクトリー・パス名を持つ **-m tmpq** および **-f** をログ・ファイルに指定する必要があります。

-m QMgrName

キュー・マネージャーの名前です。このパラメーターを省略すると、デフォルトのキュー・マネージャーの名前が使用されます。

dmpmqlog コマンドが出される場合は、指定するキュー・マネージャーまたはデフォルトのキュー・マネージャーを実行してはなりません。また、**dmpmqlog** の実行中は、キュー・マネージャーを開始してはなりません。

dspmqaout (許可の表示)

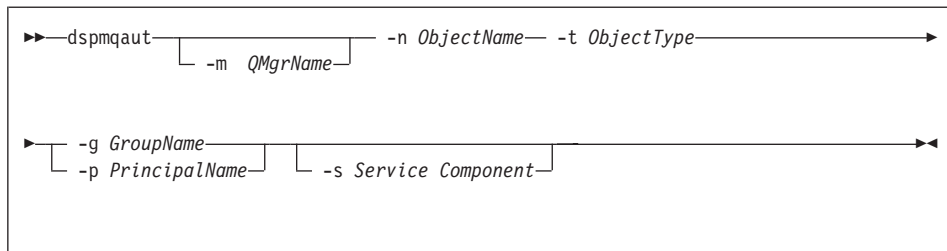
目的

dspmqaout コマンドは、指定したオブジェクトに対する現在の許可を表示するために使用します。

1 つのグループのみを指定できます。

ユーザー ID が複数のグループのメンバーである場合、このコマンドは、すべてのグループの許可を組み合わせ表示します。

構文



必須パラメーター

-n ObjectName

照会が行われる対象となるオブジェクトの名前を指定します。

照会の対象がキュー・マネージャー自身でない限り、これは必須パラメーターです。どの場合も組み込んでではありません。

キュー・マネージャー、キュー、またはプロセス定義の名前を指定しなければなりません。

-t ObjectType

照会が行われる対象となるオブジェクトのタイプを指定します。可能な値は次のとおりです。

queue または **q** オブジェクト・タイプ・パラメーターに一致するキュー (1 つまたは複数)

qmgr キュー・マネージャー・オブジェクト

process または **prcs**
プロセス

namelist または **nl**
名前リスト

オプション・パラメーター

-mQMgrName

照会が行われる対象となるキュー・マネージャーの名前を指定します。

-g GroupName

照会が行われる対象となるユーザー・グループの名前を指定します。指定できるのは **1 つ** の名前のみであり、既存の権限 ID の名前である必要があります。

-p PrincipalName

表示しようとしている指定のオブジェクトに対する許可を持っているユーザーの名前を指定します。

-s ServiceComponent

このパラメーターは、インストール可能な許可サービスを使用している場合にのみ適用されます。それ以外は、無視されます。

インストール可能な許可サービスがサポートされている場合、このパラメーターは、許可が適用される許可サービスの名前を指定します。このパラメーターはオプションです。これを指定しないと、サービスの最初のインストール可能なコンポーネントに対して許可照会が行われます。

戻りパラメーター

このコマンドは、許可リストを戻します。その中には、許可パラメーターが何も含まれていないこともあれば、1 つまたはそれ以上の許可パラメーターが含まれていることもあります。戻される各許可パラメーターは、指定のグループの中のユーザー ID が、そのパラメーターで定義された操作を実行する許可を持っていることを意味します。

表16 は、種々のオブジェクト・タイプに与えることができる許可を示したものです。

表 16. *dspmqaut* コマンドからのセキュリティ許可

許可	キュー	プロセス	キュー・マネージャー	名前リスト
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓

表 16. dspmqaout コマンドからのセキュリティ許可 (続き)

許可	キュー	プロセス	キュー・マネージャー	名前リスト
clr	✓			
connect			✓	
cpy	✓	✓	✓	✓
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

次のリストは、各パラメーターに関連した許可の定義を示したものです。

- all** オブジェクトに関係のあるすべての操作を使用する。
- alladm** オブジェクトに関係のあるすべての管理操作を実行する。
- allmqi** オブジェクトに関係のあるすべての MQI 呼び出しを使用する。
- altusr** MQI 呼び出しで代替ユーザー ID を指定する。
- browse** BROWSE オプションを指定した MQGET 呼び出しを出して、キューからメッセージを取り出す。
- chg** 指定したオブジェクトの属性を、該当のコマンド・セットを使用して変更する。
- clr** キューをクリアする (PCF コマンド「キューのクリア」のみ)。
- connect** MQCONN 呼び出しを出して、指定のキュー・マネージャーにアプリケーションを接続する。
- cpy** オブジェクトの定義をコピーする。たとえば、PCF キューのコピー・コマンド。

crt	指定のタイプのオブジェクトを、該当のコマンド・セットを使用して作成する。
dlt	指定のオブジェクトを、該当のコマンド・セットを使用して削除する。
dsp	指定のオブジェクトの属性を、該当のコマンド・セットを使用して表示する。
get	MQGET 呼び出しを出して、キューからメッセージを取り出す。
inq	MQINQ 呼び出しを出して、特定のキューの照会を行う。
passall	すべてのコンテキストを渡す。
passid	識別コンテキストを渡す。
put	MQPUT 呼び出しを出して、特定のキューにメッセージを書き込む。
set	MQSET 呼び出しを出して、MQI からキューに属性を設定する。
setall	キューにすべてのコンテキストを設定する。
setid	キューの識別コンテキストを設定する。

管理操作の許可は、サポートされている場合には、次のコマンド・セットに適用されます。

- 制御コマンド
- MQSC コマンド
- PCF コマンド

戻りコード

0	正常な操作です。
36	与えられた引き数が無効です。
40	キュー・マネージャーは利用不能です。
49	キュー・マネージャーは停止中です。
69	ストレージが利用不能です。
71	予期しないエラーです。
72	キュー・マネージャー名のエラーです。
133	オブジェクト名が不明です。
145	予期しないオブジェクト名です。
146	オブジェクト名が脱落しています。
147	オブジェクト・タイプが脱落しています。

dspmqaut

- 148 オブジェクト・タイプが無効です。
- 149 エンティティ名が脱落しています。

例

次の例は、ユーザー・グループ `staff` に関連したキュー・マネージャー `saturn.queue.manager` に関する許可を表示するためのコマンドを示しています。

```
dspmqaut -m "saturn.queue.manager" -t qmgr -g staff
```

このコマンドの結果を次に示します。

```
Entity staff has the following authorizations for object:  
  get  
  browse  
  put  
  inq  
  set  
  connect  
  altusr  
  passid  
  passall  
  setid
```

関連コマンド

setmqaut 許可の設定またはリセット

dspmqsrv (コマンド・サーバーの表示)

目的

dspmqsrv コマンドは、指定したキュー・マネージャーのコマンド・サーバーの状況を表示するために使用します。

状況は次のいずれかです。

- 開始中
- 実行中
- SYSTEM.ADMIN.COMMAND.QUEUE で実行中 (読み取り不可)
- 終了中
- 停止

構文

```
▶▶ dspmqsrv [QMGrName] ▶▶
```

オプション・パラメーター

QMGrName

コマンド・サーバーの状況を要求する対象となるローカル・キュー・マネージャーの名前を指定します。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

例

次のコマンドは、venus.q.mgr: に関連するコマンド・サーバーの状況を表示します。

```
dspmqsrv "venus.q.mgr"
```

関連コマンド

strmqcsv コマンド・サーバーを開始します。

dspmqcsv

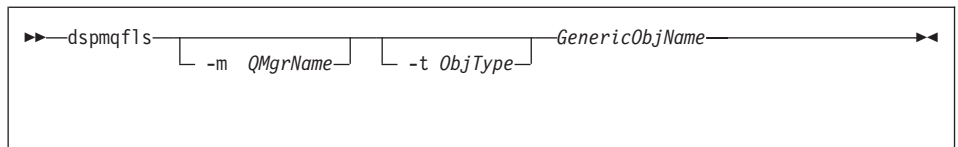
endmqcsv コマンド・サーバーを終了します。

dspmqls (MQSeries ファイルの表示)

目的

dspmqls コマンドは、指定した基準に一致するすべての MQSeries オブジェクトの実際のファイル・システム名を表示するために使用します。このコマンドを使用すれば、特定の MQSeries オブジェクトに関連したファイルを識別することができます。これは、特定のオブジェクトのバックアップをとるのに役立ちます。名前変換の詳細については、28ページの『MQSeries のファイル名についての理解』を参照してください。

構文



必須パラメーター

GenericObjName

MQSeries オブジェクトの名前を指定します。名前は、フラグなしのストリングで、必須パラメーターです。名前を省略すると、エラーが戻されます。

このパラメーターは、ストリングの最後にワイルドカード文字 `*` を指定できます。

オプション・パラメーター

-m *QMgrName*

ファイルを調べる対象となるキュー・マネージャーの名前を指定します。これを省略すると、コマンドは、デフォルト・キュー・マネージャーに対して操作を実行します。

-t *ObjType*

MQSeries オブジェクト・タイプを指定します。有効なオブジェクト・タイプを以下に示します。省略名を最初に示し、その後完全な名前を示しています。

* または **all** 全てのオブジェクト・タイプ。これはデフォルトです。

dspmqls

q または queue

オブジェクト名パラメーターに一致するキュー (1 つまたは複数)

ql または qlocal

ローカル・キュー

qa または qalias

別名キュー

qr または qremote

リモート・キュー

qm または qmodel

モデル・キュー

qmgr

キュー・マネージャー・オブジェクト

prcs または process

プロセス

ctlg または catalog

オブジェクト・カタログ

nl または namelist

名前リスト

注: **dspmqls** コマンドは、キュー自身の名前ではなく、キューが格納されているディレクトリーを表示するコマンドです。

戻りコード

- 0 コマンドは正常に終了しました。
- 10 コマンドは完了しましたが、すべて予期したとおりではありません。
- 20 処理中にエラーが起きました。

例

1. 次のコマンドは、デフォルト・キュー・マネージャーに定義されている、SYSTEM.ADMIN で始まる名前を持つすべてのオブジェクトの詳細を表示します。

```
dspmqls SYSTEM.ADMIN*
```

2. 次のコマンドは、キュー・マネージャー RADIUS に定義された、PROC で始まる名前を持つすべてのプロセスのファイルの詳細を表示します。

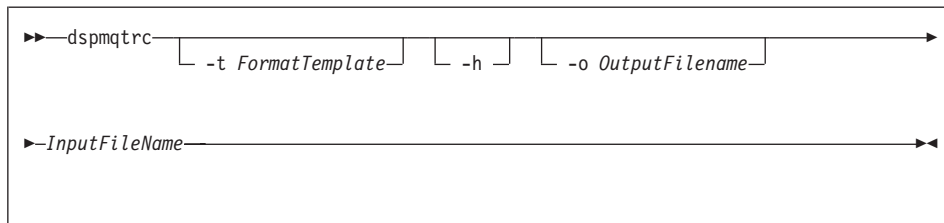
```
dspmqfls -m RADIUS -t prcs PROC*
```

dspmqtrc (MQSeries 定様式トレース出力の表示)

目的

dspmqtrc コマンドは、MQSeries の定様式トレース出力を表示する場合に使用します。

構文



必須パラメーター

InputFileName

不定様式トレースが格納されているファイルの名前を指定します。たとえば、MQS_ROOT:[MQM.TRACE]AMQ20202345.TRC などです。

オプション・パラメーター

-t *FormatTemplate*

トレースの表示方法の詳細を含んでいるテンプレート・ファイルの名前を指定します。デフォルト値は SYS\$SHARE:AMQTRC.FMT です。

-h レポートからヘッダー情報を省略します。

-o *output_filename*

定様式データを書き込むファイルの名前。

例

1. 以下のコマンドは、出力のリダイレクトを示します。

```
dspmqtrc mqs_root:[mqm.trace]amq20202345.trc > mqs_root:[mqm.trace]amq20202345.fmt
```

関連コマンド

endmqtrc MQSeries トレースの終了
strmqtrc MQSeries トレースの開始

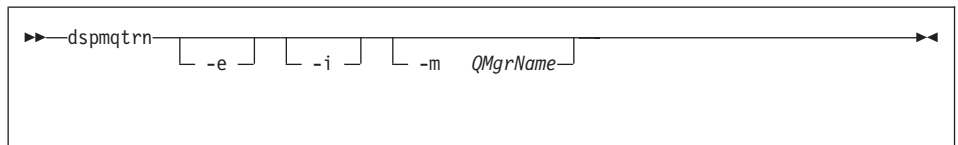
dspmqtrn (MQSeries トランザクションの表示)

目的

dspmqtrn コマンドを使用すると、2 フェーズ・コミット・プロシージャで準備済み状況にあり、キュー・マネージャーにより認識されるトランザクションを表示することができます (下記のアテンションを参照してください)。

各トランザクションは、トランザクション番号 (人間が理解できるトランザクション ID)、トランザクションの状態、およびトランザクション ID として表示されます。トランザクション ID の長さは、最高 128 文字です。そのため、トランザクション番号が必要です。

構文



アテンション: このコマンドを使用できるのは、外部トランザクション・マネージャーを使用しており、2 フェーズ・コミットメント・プロシージャが関係している場合だけです。2 フェーズ・コミットを使用しない場合には、このコマンドを使用しないでください。このコマンドは、同期点マネージャーがトランザクションの解決に失敗した場合にのみ使用してください。

オプション・パラメーター

-m*QMgrName*

トランザクションを調査する対象となるキュー・マネージャーの名前を指定します。これを省略すると、コマンドは、デフォルト・キュー・マネージャーに対して操作を実行します。

- e** 外部的に調整した未確定トランザクションの詳細を要求します。このようなトランザクションは、MQSeries がコミットの準備を求められたが、まだ、トランザクションの結果を知らされていないものです。
- i** 内部的に調整した未確定トランザクションの詳細を要求します。このようなトランザクションは、各リソース・マネージャーがコミットの準備を求められたが、MQSeries がまだリソース・マネージャーにトランザクションの結果を知らせていないものです。

dspmqtrn

関連しているリソース・マネージャーのそれぞれにおいて、トランザクションの推論状態の情報が表示されます。この情報は、特定のリソース・マネージャーの障害の結果を査定するのに役立ちます。

注: `-e` も `-i` も指定しない場合は、内部および外部的に調整された未確定トランザクション両方の詳細が表示されます。

戻りコード

- 0** 正常な操作です。
- 36** 与えられた引き数が無効です。
- 40** キュー・マネージャーは利用不能です。
- 49** キュー・マネージャーは停止中です。
- 69** ストレージが利用不能です。
- 71** 予期しないエラーです。
- 72** キュー・マネージャー名のエラーです。
- 102** トランザクションが見つかりません。

関連コマンド

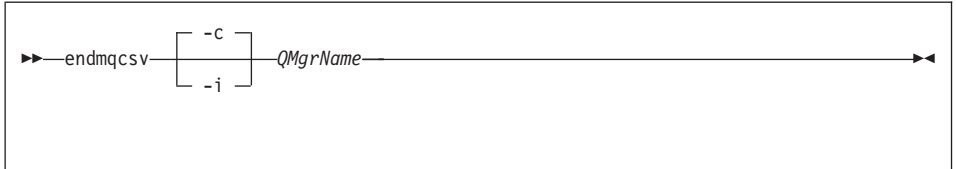
rsvmqtrn MQSeries トランザクションの解決

endmqcsv (コマンド・サーバーの終了)

目的

endmqcsv コマンドは、指定したキュー・マネージャーのコマンド・サーバーを停止するために使用します。

構文



必須パラメーター

QMgrName

コマンド・サーバーを終了させる対象となるキュー・マネージャーの名前を指定します。

オプション・パラメーター

-c コマンド・サーバーが制御された状態で停止することを指定します。コマンド・サーバーは、すでに開始されているコマンド・メッセージの処理があれば、その処理を完了することができます。新しいメッセージが、コマンド・キューから読み取られることはありません。

これはデフォルトです。

-i コマンド・サーバーが即時に停止することを指定します。現在処理されているコマンド・メッセージに関連したアクションは、完了しない可能性があります。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

endmqcsv

例

1. 次のコマンドは、キュー・マネージャー saturn.queue.manager のコマンド・サーバーを停止します。

```
endmqcsv -c "saturn.queue.manager"
```

コマンド・サーバーは、停止する前に、すでに開始しているコマンドの処理を完了することができます。新しく受け取ったコマンドは、コマンド・サーバーが再始動されるまで、処理されずにコマンド・キューに残ります。

2. 次のコマンドは、キュー・マネージャー pluto のコマンド・サーバーを即時に停止します。

```
endmqcsv -i "pluto"
```

関連コマンド

- | | |
|-----------------|---------------------|
| strmqcsv | コマンド・サーバーを開始します。 |
| dspmqcsv | コマンド・サーバーの状況を表示します。 |

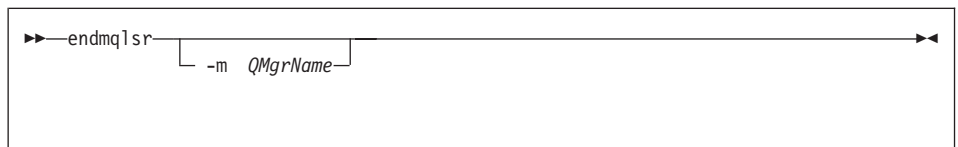
endmqlsr (リスナーの終了)

目的

endmqlsr コマンドは、指定したキュー・マネージャーのリスナー・プロセスをすべて終了します。

endmqlsr コマンドを発行する前に、キュー・マネージャーを停止しなければなりません。

構文



オプション・パラメーター

-m*QMgrName*

キュー・マネージャーの名前を指定します。名前を指定しない場合、デフォルトのキュー・マネージャーに対する処理が実行されます。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

endmqm (キュー・マネージャーの終了)

目的

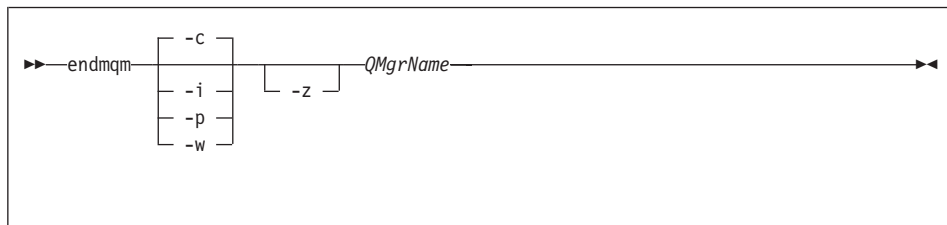
endmqm コマンドは、指定したローカル・キュー・マネージャーを終了 (停止) するために使用します。このコマンドは、次の 3 つのモードのいずれかでキュー・マネージャーを停止します。

- 正常または静止シャットダウン
- 即時シャットダウン
- 優先シャットダウン

キュー・マネージャーの属性および関連したオブジェクトの属性は、影響を受けません。**strmqm** (キュー・マネージャーの開始) コマンドを使用すれば、キュー・マネージャーを再始動できます。

キュー・マネージャーを削除するためには、それを停止し、その後 **dltmqm** (キュー・マネージャーの削除) コマンドを使用する必要があります。

構文



必須パラメーター

QMgrName

停止させるメッセージ・キュー・マネージャーの名前を指定します。

オプション・パラメーター

-c 制御された (または静止) シャットダウン。キュー・マネージャーは、すべてのアプリケーションが切断された後でのみ停止します。現在処理されている MQI 呼び出しはすべて完了します。これはデフォルトです。

制御は即時にその管理者に戻され、キュー・マネージャーが停止した時点は通知されません。

-w 待機シャットダウン。

このタイプのシャットダウンは、キュー・マネージャーが停止した後でのみ制御がユーザーに戻されるということを除けば、制御されたシャットダウンと同じです。シャットダウンの進行中に "Waiting for queue manager *QMGrName* to end" というメッセージを受け取ります。

- i 即時シャットダウン。キュー・マネージャーは、現在処理されている MQI 呼び出しをすべて完了してから停止します。このコマンドが出された後に出された MQI 要求はすべて失敗します。完了しなかった作業単位は、キュー・マネージャーが次に開始されるときに、ロールバックされます。
- p 優先シャットダウン

このタイプのシャットダウンは、例外的な状況でのみ使用してください。たとえば、キュー・マネージャーが通常の **endmqm** コマンドで停止しない場合などです。

キュー・マネージャーは、アプリケーションが切断されるのを待たずに、あるいは MQI 呼び出しが完了するのを待たずに停止します。これによって、MQSeries アプリケーションにとって予測不能な結果が生じる可能性があります。キュー・マネージャー内の停止できなかったすべてのプロセスは、このコマンドが出された後 30 秒してから終了します。

注: 強制シャットダウンまたは優先シャットダウンの後、またはキュー・マネージャーが失敗した場合は、キュー・マネージャーは、所有する共有メモリーをクリーンアップせずに終了した可能性があります。この場合には、問題が再び生じるようになることがあります。MONMQ ユーティリティーを使用して、この種の異常終了をクリーンアップする方法についての詳細は、458ページの『共有メモリーを MONMQ で管理する』を参照してください。

- z コマンドでのエラー・メッセージを抑制します。

戻りコード

- 0 キュー・マネージャーは終了しました。
- 3 キュー・マネージャーは作成中です。
- 16 キュー・マネージャーは存在していません。
- 40 キュー・マネージャーは利用不能です。
- 49 キュー・マネージャーは停止中です。
- 69 ストレージが利用不能です。
- 71 予期しないエラーです。
- 72 キュー・マネージャー名のエラーです。

endmqm

例

次の例は、指定されたキュー・マネージャーを終了 (停止) させるコマンドを示しています。

1. 次のコマンドは、`mercury.queue.manager` という名前のキュー・マネージャーを制御された状態で終了させます。現在接続されているすべてのアプリケーションは、切断することが可能です。

```
endmqm "mercury.queue.manager"
```

2. 次のコマンドは、`saturn.queue.manager` という名前のキュー・マネージャーを即時に終了させます。現行の MQI 呼び出しはすべて完了しますが、新しい呼び出しは完了できません。

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

関連コマンド

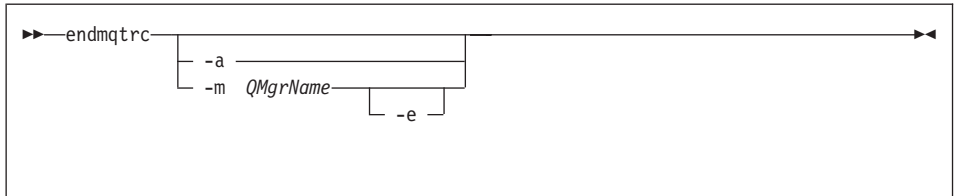
crtmqm	キュー・マネージャーを作成します。
strmqm	キュー・マネージャーを開始します。
dltmqm	キュー・マネージャーを削除します。

endmqtrc (MQSeries トレースの終了)

目的

endmqtrc コマンドは、指定したエンティティまたはすべてのエンティティに関するトレースを終了する場合に使用します。

構文



オプション・パラメーター

-m*QMgrName*

トレースが終了されるキュー・マネージャーの名前です。

このコマンドには、**-m** フラグおよび関連するキュー・マネージャー名を最大 1 つしか指定できません。

キュー・マネージャー名および **-m** フラグは、**-e** フラグと同じコマンドで指定できます。

-e このフラグを指定すると、早期トレースが終了します。

-a このフラグを指定すると、すべてのトレースが終了します。

このフラグは、**必ず** 単独で指定してください。

戻りコード

AMQ5611 このメッセージは、コマンドに無効な引き数を指定した場合に出されます。

例

このコマンドは、QM1 というキュー・マネージャーに関するデータのトレースを終了させます。

```
endmqtrc -m QM1
```

endmqtrc

関連コマンド

dspmqtrc	定様式トレース出力の表示
strmqtrc	MQSeries トレースの開始

フェールオーバー (フェールオーバー・セットの管理)

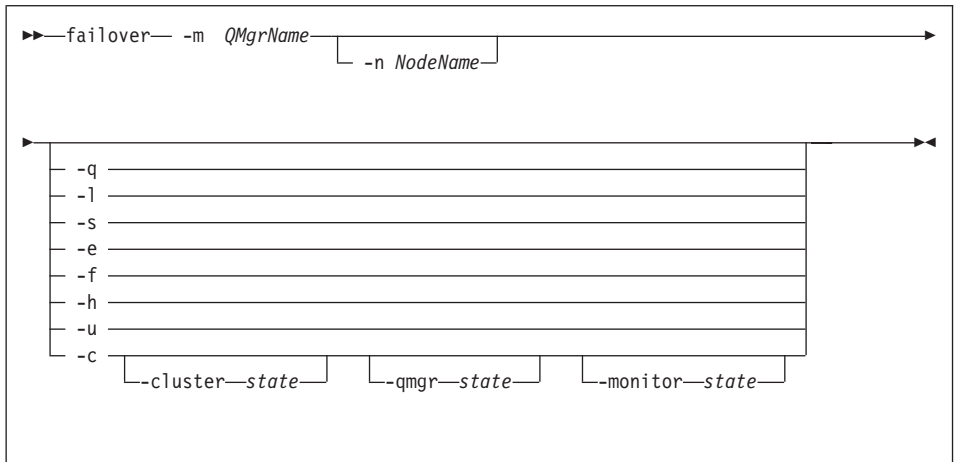
目的

failover コマンドは、フェールオーバー・セットを管理するのに使用します。

failover コマンドは、更新および照会パラメーターの両方を組み込みます。

failover コマンドは、フェールオーバー・セット内の任意の OpenVMS ノードから実行できます。

構文



必須パラメーター

-m *QMgrName*

failover コマンドを適用する対象となるキュー・マネージャーの名前を指定します。 *QMgrName* 名にサポートされている最大長は 25 文字です。

-n *NodeName*

コマンドの適用対象となる OpenVMS ノードを指定します。このパラメーターは、-h および -c パラメーターに必須です。

オプション・パラメーター

-q フェールオーバー・セットの状態を照会し、出力を表示します。

-l フェールオーバー・セットの状態を照会し、以下の DCL シンボルをセットします。

フェールオーバー

DCL シンボル名	説明
MQS\$QMGR_NODE	キュー・マネージャーを実行している OpenVMS に設定されます。どのキュー・マネージャーも実行していない場合には、ヌル・ストリングが出されます。
MQS\$AVAILABLE_NODES	キュー・マネージャーを実行するのに使用できる OpenVMS ノードのリストに設定されます。これは、キュー・マネージャーの状態が AVAILABLE で、フェールオーバー・モニターが実行中のノードです。
MQS\$MONITOR_NODES	フェールオーバー・モニターが実行している OpenVMS ノードのリストに設定されます。

- s フェールオーバー・セットでキュー・マネージャーを開始します。 -n パラメーターが指定されている場合には、指定される OpenVMS ノードでキュー・マネージャーが開始されます。指定されていない場合には、優先順位の最も高いノードで開始されます。
- e フェールオーバー・セットでキュー・マネージャーを終了します。
- f フェールオーバー・セットでキュー・マネージャーを別のノードに移動します。 -n パラメーターが指定されている場合には、キュー・マネージャーは指定されたノードに移動されます。指定されていない場合には、優先順位の最も高いノードに移動されます。
- h -n パラメーターで指定されるノード上で実行しているフェールオーバー・モニターを停止します。
- u 進行フラグにある更新をクリアします。
- c フェールオーバー・セットの状態を変更します。変更される状態は、以下の 3 つのパラメーターによって決定されます。変更は、フェールオーバー・セットの実行状態と一貫性がある場合にのみ有効になります。
 - cluster started | stopped
 - c パラメーターと共に、全体のフェールオーバー・セット状態を変更するために使用されます。
 - qmgr available | running | excluded
 - c パラメーターと共に、-n パラメーターで指定されるノードのノード・キュー・マネージャー状態を変更するために使用されます。

-monitor started | stopped | watching

-c パラメーターと共に、-n パラメーターで指定されるノードのノード・フェールオーバー・モニター状態を変更するために使用されます。

戻りコード

- 0** コマンドは正常に終了しました。
- 5** キュー・マネージャーは実行中です。
- 36** コマンドに提供される引き数は、有効ではありません。
- 326** キュー・マネージャーは実行していません。
- 1925** キュー・マネージャー用に開始されているフェールオーバー・モニターがありません。
- 1926** フェールオーバー・セットの更新操作が進行中です。
- 1937** キュー・マネージャーを始動するのに使用できるノードがありません。
- 1939** キュー・マネージャーの終了が強制されました。
- 1940** キュー・マネージャーの終了が完了する前にタイムアウトになりました。

OpenVMS エラー・コード

- 36** %SYSTEM-F-NOPRIV. 特権が不十分であるか、あるいはオブジェクト保護に違反しています。
- 652** %SYSTEM-F-NOSUCHNODE、リモート・ノードが不明です。
- 660** %SYSTEM-F-REJECT、ネットワーク・オブジェクトへの接続が拒否されました。

例

1. この例は、キュー・マネージャー testqm のフェールオーバー・セットの状態を照会します。

```
failover -m "testqm" -q
```

2. この例は、ノード batman でキュー・マネージャー testqm を開始します。

```
failover -m "testqm" -n batman -s
```

3. この例は、キュー・マネージャー testqm を使用可能な最高の優先順位のノードに移動します。

フェールオーバー

```
failover -m "testqm" -f
```

関連コマンド

runmqfm フェールオーバー・モニターの開始

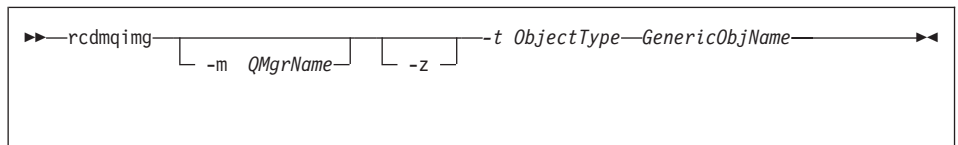
rcdmqimg (メディア・イメージの記録)

目的

rcdmqimg コマンドは、1 つの MQSeries オブジェクトまたはオブジェクトのグループのイメージを、メディア・リカバリー用にログに書き込む場合に使用します。これに関連したコマンド **rcrmqobj** は、そのイメージからオブジェクトを再作成するために使用します。

このコマンドは、アクティブ・キュー・マネージャーに関して使用します。その後のキュー・マネージャーでの活動は、ログに記録されます。これは、イメージが最新のものでなくなったとしても、ログ・レコードがオブジェクトに対する変更を表すことができるようにするためです。

構文



必須パラメーター

GenericObjName

記録されるオブジェクトの名前を指定します。このパラメーターには、後ろにアスタリスクが付いている場合があります。このアスタリスクは、アスタリスクの前にくる名前部分と一致する名前のオブジェクトが記録されることを示すものです。

キュー・マネージャーのオブジェクトまたはチャンネル同期ファイルを記録しているのであれば、このパラメーターが必要です。チャンネル同期ファイルのオブジェクト名を指定する場合、無視されます。

-t *ObjectType*

イメージが記録される対象となるオブジェクトのタイプを指定します。有効なオブジェクト・タイプは次のとおりです。

prcs または **process**

プロセス

q または **queue**

すべてのタイプのキュー

- ql** または **qlocal**
ローカル・キュー
- qa** または **qalias**
別名キュー
- qr** または **qremote**
リモート・キュー
- qm** または **qmodel**
モデル・キュー
- qmgr**
キュー・マネージャー・オブジェクト
- syncfile**
チャンネル同期ファイル
- nl** または **namelist**
名前リスト
- ctlg** または **catalog**
オブジェクト・カタログ
- * または **all** 上記のすべて

オプション・パラメーター

-mQMgrName

イメージが記録される対象となるキュー・マネージャーの名前を指定します。これを省略すると、コマンドは、デフォルト・キュー・マネージャーに対して操作を実行します。

-z エラー・メッセージを抑制します。

戻りコード

- 0** 正常な操作です。
- 36** 与えられた引き数が無効です。
- 40** キュー・マネージャーは利用不能です。
- 49** キュー・マネージャーは停止中です。
- 68** メディア・リカバリーはサポートされていません。
- 69** ストレージが利用不能です。
- 71** 予期しないエラーです。
- 72** キュー・マネージャー名のエラーです。
- 119** ユーザーは許可を与えられていません。
- 128** オブジェクトは処理されませんでした。
- 131** リソースの問題です。
- 132** オブジェクトが損傷しました。

135 一時オブジェクトを記録できません。

例

次のコマンドは、キュー・マネージャー・オブジェクト saturn.queue.manager のイメージをログに記録します。

```
rcdmqimg -t qmgr -m "saturn.queue.manager"
```

関連コマンド

rcrmqobj キュー・マネージャー・オブジェクトを再作成します。

rcrmqobj (オブジェクトの再作成)

目的

rcrmqobj コマンドは、オブジェクトまたはオブジェクト・グループを、ログに含まれているそれらのイメージから再作成するために使用します。これに関連したコマンド **rcdmqimg** は、オブジェクトのイメージをログに記録するために使用します。

このコマンドは、実行中のキュー・マネージャーに対して使用する必要があります。イメージが作成された後のキュー・マネージャーの活動はすべてログに記録されます。オブジェクトを再作成するためには、ログを再生して、オブジェクト・イメージが取り込まれた後に起きたイベントを再作成する必要があります。

構文

```

▶▶rcrmqobj [-m QMgrName] [-z] -t ObjectTypeGenericObjName▶▶

```

必須パラメーター

GenericObjName

再作成するオブジェクトの名前を指定します。このパラメーターには、後ろにアスタリスクが付いている場合があります。このアスタリスクは、アスタリスクの前にくる名前部分と一致する名前のオブジェクトが再作成されることを示すものです。

オブジェクト・タイプがチャンネル同期ファイルでない限り、このパラメーターが必要です。オブジェクト名がこのオブジェクト・タイプに指定される場合、無視されます。

-t *ObjectType*

再作成するオブジェクトのタイプを指定します。有効なオブジェクト・タイプは次のとおりです。

procs または **process**

プロセス

q または **queue**

すべてのタイプのキュー

ql または qlocal	ローカル・キュー
qa または qalias	別名キュー
qr または qremote	リモート・キュー
qm または qmodel	モデル・キュー
nl または namelist	名前リスト
ctlg または catalog	オブジェクト・カタログ
* または all	上記のものすべて
syncfile	チャンネル同期ファイル

注: このフラグを使用すると、指定されるキュー・マネージャーにチャンネル同期化ファイルを再生成できます。
rcdmqing コマンドによりファイルが保管されることはないため、これが必要になります。

オプション・パラメーター

-m*QMgrName*

オブジェクトが再作成される対象となるキュー・マネージャーの名前を指定します。これを省略すると、コマンドは、デフォルト・キュー・マネージャーに対して操作を実行します。

-z エラー・メッセージを抑制します。

戻りコード

0	正常な操作です。
36	与えられた引き数が無効です。
40	キュー・マネージャーは利用不能です。
49	キュー・マネージャーは停止中です。
66	メディア・イメージは利用不能です。
68	メディア・リカバリーはサポートされていません。
69	ストレージが利用不能です。
71	予期しないエラーです。
72	キュー・マネージャー名のエラーです。

rcrmqobj

- 119 ユーザーは許可を与えられていません。
- 128 オブジェクトは処理されませんでした。
- 135 一時オブジェクトを回復できません。
- 136 オブジェクトは使用中です。

例

1. 次のコマンドは、デフォルト・キュー・マネージャーのすべてのローカル・キューを再作成します。

```
rcrmqobj -t ql *
```

2. 次のコマンドは、キュー・マネージャー store に関連したすべてのリモート・キューを再作成します。

```
rcrmqobj -m "store" -t qr *
```

関連コマンド

rcdmqimg MQSeries オブジェクトのログへの記録

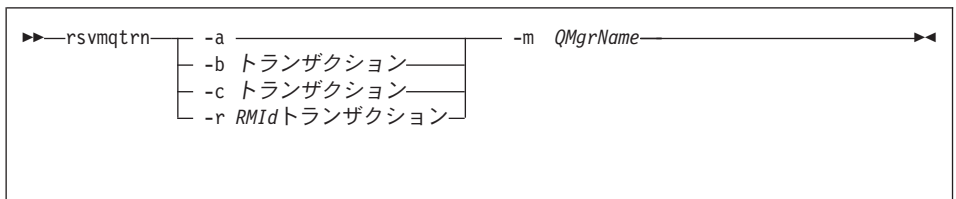
rsvmqtrn (MQSeries トランザクションの解決)

目的

rsvmqtrn コマンドは、内部的または外部的に調整された未確定トランザクションをコミットまたはバックアウトするために使用します。

通常のプロトコルではトランザクションを解決できないことが確かな場合にのみ、このコマンドを使用します。このコマンドを出すと、分散トランザクションの場合、リソース・マネージャー相互間でのトランザクションの健全性が失われる可能性があります。

構文



必須パラメーター

-m *QMgrName*

キュー・マネージャーの名前を指定します。このパラメーターは必須です。

オプション・パラメーター

- a** キュー・マネージャーが、内部的に調整した未確定トランザクションすべて(つまり、グローバル作業単位すべて)の解決を試みる必要があることを指定します。
- b** バックアウトされる名前付きトランザクションを指定します。このフラグは、外部的に調整されたトランザクション(つまり、外部の作業単位)に対してのみ有効です。
- c** コミットされる名前付きトランザクションを指定します。このフラグは、外部的に調整されたトランザクション(つまり、外部の作業単位)に対してのみ有効です。
- r** *RMIId*
コミットまたはバックアウトの決定が下される対象となるリソース・マネージャーを識別します。このフラグは、内部的に調整されたトランザクシ

rsvmqtrn

ョンのみ、およびキュー・マネージャーの `qm.ini` ファイル内で構成されなくなっているリソース・マネージャーに有効です。送達される結果は、トランザクションに対し MQSeries が下す決定と一貫性があります。

Transaction

コミットまたはバックアウトされるトランザクションのトランザクション番号です。関連のあるトランザクション番号を調べるには、**dspmqtrn** コマンドを使用します。このパラメーターには、`-b`、`-c`、および `-r RMIId` パラメーターが必要です。

戻りコード

- 0** 正常な操作です。
- 32** トランザクションを解決できません。
- 34** リソース・マネージャーは認識されていません。
- 35** リソース・マネージャーは永続的に利用できません。
- 36** 与えられた引き数が無効です。
- 40** キュー・マネージャーは利用不能です。
- 49** キュー・マネージャーは停止中です。
- 69** ストレージが利用不能です。
- 71** 予期しないエラーです。
- 72** キュー・マネージャー名のエラーです。
- 85** トランザクションは認識されていません。

関連コマンド

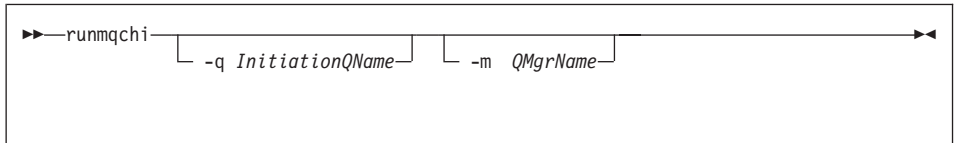
dspmqtrn 準備済みトランザクションのリストを表示します。

runmqchi (チャンネル開始プログラムの実行)

目的

runmqchi コマンドは、チャンネル開始プログラムのプロセスを実行するために使用します。このコマンドの使用の詳細については、*MQSeries 相互通信* を参照してください。

構文



オプション・パラメーター

-q *InitiationQName*

このチャンネル開始プログラムが処理する開始キューの名前を指定します。これを指定しなければ、**SYSTEM.CHANNEL.INITQ** が使用されます。

-m *QMgrName*

開始キューが存在しているキュー・マネージャーの名前を指定します。この名前を省略した場合、デフォルト・キュー・マネージャーが使用されません。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

戻りコードの 10 か 20 が戻されるようなエラーが起きた場合、チャンネルが関連付けられているキュー・マネージャーのエラー・ログを見て、エラー・メッセージを調べてください。また、**\$SYSTEM** エラー・ログも見てください。チャンネルがキュー・マネージャーと関連付けられる前に起きた問題はそこに記録されています。エラー・ログの詳細については、262ページの『エラー・ログ』を参照してください。

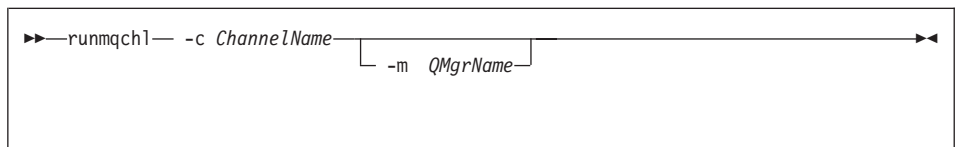
runmqchl (チャネルの実行)

目的

runmqchl コマンドは、送信側 (SDR) または要求側 (RQSTR) を実行する場合に使用します。

チャネルは同期して実行されます。チャネルを停止するには、MQSC コマンド `STOP CHANNEL` を出します。

構文



必須パラメーター

-c ChannelName

実行するチャネルの名前を指定します。

オプション・パラメーター

-m QMgrName

このチャネルが関連付けられているキュー・マネージャーの名前を指定します。名前を指定しない場合、デフォルト・キュー・マネージャーが使用されます。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

戻りコードの 10 または 20 が生成された場合、関連したキュー・マネージャーのエラー・ログを見て、エラー・メッセージを調べてください。また、`$$SYSTEM` エラー・ログも見てください。チャネルがキュー・マネージャーと関連付けられる前に起きた問題はそこに記録されています。

runmqdlq (送達不能キュー・ハンドラーの実行)

目的

runmqdlq コマンドは、送達不能キュー (DLQ) ハンドラーを開始するために使用します。このハンドラーは、送達不能キューのメッセージのモニターと処理を行うために実行できるユーティリティーです。

メッセージの選択、およびそのメッセージに関して実行される処理の定義の両方を行える一組のルールを指定することによって、送達不能キュー・ハンドラーを使用して、選択したメッセージに関するさまざまな処理を実行できます。

runmqdlq コマンドは、SYS\$INPUT から入力を受け取ります。コマンドが処理されると、結果と要約がレポートに書き込まれ、SYS\$OUTPUT に送られます。

SYS\$INPUT をキーボードから受け取ることによって、**runmqdlq** ルールを対話式に入力できます。

入力をファイルからリダイレクトすることによって、指定したキューにルール・テーブルを適用できます。ルール・テーブルには、ルールが少なくとも 1 つはなければなりません。

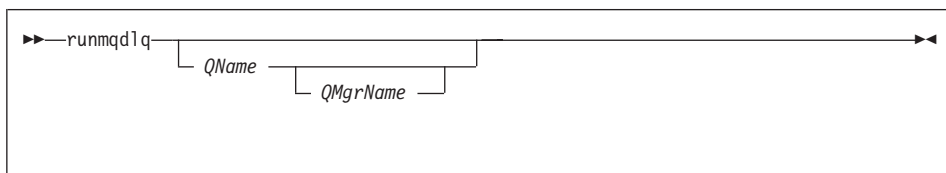
ファイル (規則表) から SYS\$INPUT をリダイレクトしないで DLQ ハンドラーをフォアグラウンド・モードで使用する場合には、DLQ ハンドラーは次のようになります。

- キーボードから入力を読み取ります。
- end_of_file (Ctrl+D) 文字を受け取るまで、指定されたキューの処理は開始しません。

ルール・テーブルおよびルール・テーブルの構成方法の詳細については、133ページの『DLQ ハンドラーのルール・テーブル』を参照してください。

runmqdlq

構文



オプション・パラメーター

注釈行と行結合に関する MQSC のルールは、DLQ ハンドラーの入力パラメーターにも適用されます。

QName

処理されるキューの名前を指定します。

名前を指定しない場合、ローカル・キュー・マネージャーに定義した送達不能キューが使用されます。1 つまたは複数のブランク (' ') を使用した場合は、ローカル・キュー・マネージャーの送達不能キューが明示的に割り当てられます。

DLQ ハンドラーを使用すると、特定の処理を行うために、送達不能キューの特定のメッセージを選択できます。たとえば、複数のメッセージを別々の送達不能キューにリダイレクトすることができます。DLQ ハンドラーの別のインスタンスによるその後の処理では、異なるルール・テーブルに従ってメッセージを処理できます。

QMgrName

処理するキューを所有するキュー・マネージャーの名前。

名前を指定しない場合、システムでのデフォルト・キュー・マネージャーが使用されます。1 つまたは複数のブランク (' ') を使用した場合は、このシステムでのデフォルト・キュー・マネージャーが明示的に割り当てられます。

runmqfm (フェールオーバー・モニターの開始)

目的

runmqfm コマンドを使用すると、OpenVMS ノードでフェールオーバー・モニターを開始できます。フェールオーバー・モニターは、**runmqfm** コマンドが実行される OpenVMS ノード上で実行します。

構文

```
▶▶—runmqfm— -m QMgrName —————▶▶  
└── -d ─┘
```

必須パラメーター

-mQMgrName

runmqfm コマンドを開始させる対象となるキュー・マネージャーの名前を指定します。 *QMgrName* 名にサポートされている最大長は 25 文字です。

オプション・パラメーター

-d ログ・ファイルにログ記録する追加のデバッグ情報を指定します。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

例

以下の例は、キュー・マネージャー `testqm` についてフェールオーバー・モニターを開始し、ログ・ファイル `test.log` にデバッグ情報を書き込みます。

```
runmqfm -m "testqm" -d > test.log
```

runmqfm

関連コマンド

failover

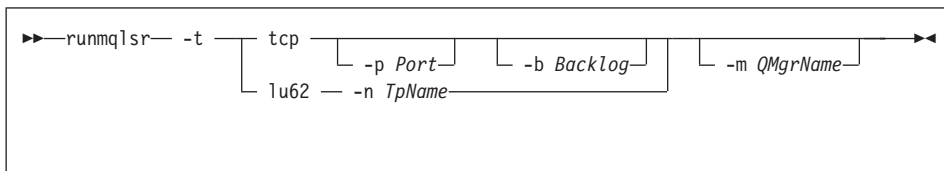
フェールオーバー・セットを管理します。

runmqslsr (リスナーの実行)

目的

runmqslsr (リスナーの実行) コマンドは、リスナー・プロセスを実行します。

構文



必須パラメーター

-t 使用する伝送プロトコルを指定します。

tcp 伝送制御プロトコル / インターネット・プロトコル (TCP/IP)

lu62 SNA LU 6.2。 (このパラメーターの使用方法についての最新情報は、`sys$help:mqseries0510.release_notes` のリリース・ノートを参照してください。)

オプション・パラメーター

-p Port

TCP/IP の場合のポート番号。このフラグは、TCP と UDP に有効です。値を指定しなければ、キュー・マネージャー構成ファイルから、またはプログラムの中のデフォルトから値が取られます。デフォルト値は 1414 です。

-n TpName

LU 6.2 トランザクション・プログラム名。このフラグは、LU 6.2 伝送プロトコルの場合にのみ有効です。値を指定しなければ、キュー・マネージャー構成ファイルから値が取られます。値が与えられなければ、コマンドは失敗します。

-m QMgrName

キュー・マネージャーの名前を指定します。名前を指定しない場合、コマンドはデフォルト・キュー・マネージャーに対して操作を実行します。

runmqtsr

-b Backlog

リスナーがサポートする並行接続要求の数を指定します。デフォルト値のリストおよび補足情報については、240ページの『LU62 および TCP スタンザ』を参照してください。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

例

次のコマンドでは、デフォルト・キュー・マネージャー上で TCP/IP プロトコルを使用するリスナーを実行します。このコマンドは、リスナーがポート 4321 を使用するよう指定します。

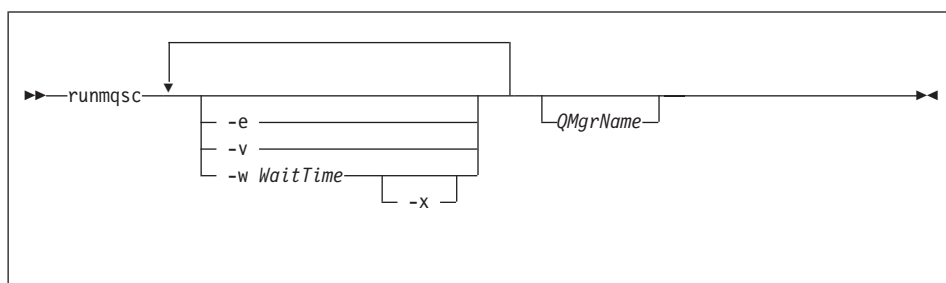
```
runmqtsr -t tcp -p 4321
```

runmqsc (MQSeries コマンドの実行)

目的

runmqsc コマンドは、キュー・マネージャーに対して MQSC コマンドを出すために使用します。MQSC コマンドを使用すると、ローカル・キュー・オブジェクトの定義、更新、または削除などの管理タスクを実行できます。MQSC コマンドとその構文については、*MQSeries* アプリケーション・プログラミング・ガイド で説明しています。

構文



説明

runmqsc コマンドは、次の 3 つのモードで呼び出すことができます。

検証モード

MQSC コマンドは検査されますが、実際には実行されません。各コマンドが正常に実行されるか失敗するかを示す出力レポートが生成されます。このモードは、ローカル・キュー・マネージャーでのみ使用できません。

直接モード

MQSC コマンドは、ローカル・キュー・マネージャーに直接送られます。

間接モード

MQSC コマンドは、リモート・キュー・マネージャーで実行されます。これらのコマンドは、リモート・キュー・マネージャーのコマンド・キューに書き込まれ、キューに入れられた順序で実行されます。コマンドからのレポートは、ローカル・キュー・マネージャーに戻されません。

runmqsc

注: リモート・キュー・マネージャーを実行しているユーザー ID は、ローカルに存在すること、および正しい権限を持っていることが必要です。

runmqsc コマンドは、SYS\$INPUT から入力を取ります。コマンドが処理されると、結果と要約がレポートに書き込まれ、そのレポートは SYS\$OUTPUT に送られます。

SYS\$INPUT がキーボードである場合は、MQSC コマンドを対話式で入力できません。

ファイルからの入力をリダイレクトすることにより、ファイルに入っている使用頻度の高い一連のコマンドを実行できます。出力レポートをファイルにリダイレクトすることもできます。

オプション・パラメーター

- e MQSC コマンドのソース・テキストがレポートにコピーできないようにします。これは、対話式でコマンドを入力する場合に便利です。
- v 検証モードを指定します。アクションを実行しないで、指定のコマンドを確認します。このモードを使用できるのは、ローカルのみです。-w および -x フラグは、同時に指定された場合には無視されます。

-w *WaitTime*

間接モードを指定します。つまり、MQSC コマンドは、別のキュー・マネージャーで実行されることとなります。このためには、必要なチャンネルと伝送キューがセットアップされている必要があります。詳細については、87ページの『リモート管理のためにチャンネルおよび伝送キューを作成する』を参照してください。

WaitTime

runmqsc が応答を待つ時間を秒単位で指定します。この時間が経過した後に受け取る応答は破棄されますが、MQSC コマンドはそのまま実行されます。1 から 999 999 秒までの時間を指定します。

各コマンドは、Escape PCF として、宛先キュー・マネージャーのコマンド・キュー (SYSTEM.ADMIN.COMMAND.QUEUE) へ送られます。

応答キューは SYSTEM.MQSC.REPLY.QUEUE に入れられ、結果はレポートに追加されます。これは、ローカル・キューまたはモデル・キューとして定義できます。

間接モード操作は、デフォルト・キュー・マネージャーを介して実行されます。

フラグが指定されている場合には、このフラグは無視されます。

- x 宛先キュー・マネージャーが MVS/ESA の下で実行されることを指定します。このフラグは、間接モードでしか適用されません。-w フラグも指定する必要があります。間接モードでは、MQSC コマンドは、MQSeries (OS/390 版) のコマンド・キューに適した形式で書き込まれます。

QMgrName

MQSC コマンドが実行される宛先キュー・マネージャーの名前を指定します。これを省略すると、MQSC コマンドは、デフォルト・キュー・マネージャーに対して実行されます。

戻りコード

- 0 MQSC コマンド・ファイルは正常に処理されました。
- 10 MQSC コマンド・ファイルは処理されましたが、エラーが起きました。レポートの中にコマンドの失敗の理由が示されています。
- 20 エラー。MQSC コマンドは実行されませんでした。

例

1. 次のコマンドを OpenVMS コマンド・プロンプトに入力します。

```
runmqsc
```

これで、OpenVMS コマンド・プロンプトに MQSC コマンドを直接入力できるようになります。キュー・マネージャー名が指定されなかったため、MQSC コマンドは、デフォルト・キュー・マネージャーで処理されます。

2. MQSC コマンドの検査のみが行われることを指定するには、次のコマンドを使用します。

```
runmqsc -v BANK < DKA0:[USERS]COMMFIL.IN
```

これは、ディレクトリー DKA0:[USERS] の MQSC コマンド・ファイル、COMMFIL.IN を検査します。キュー・マネージャー名は BANK です。出力は、現行ウィンドウに表示されます。

3. 次のコマンドは、デフォルト・キュー・マネージャーに対して、MQSC コマンド・ファイル MQS_ROOT:[MQM.MQSC]MQSCFILE.IN を実行します。

runmqsc

```
runmqsc < MQS_ROOT:[MQM.MQSC]MQSCFILE.IN > MQS_ROOT:[MQM.MQSC]MQSCFILE.OUT
```

この例では、出力先はファイル MQS_ROOT:[MQM.MQSC]MQSCFILE.OUT です。

runmqtmc (クライアント・トリガー・モニターの開始)

目的

runmqtmc コマンドは、クライアントのトリガー・モニターを呼び出すために使用します。トリガー・モニターの詳細については、*MQSeries アプリケーション・プログラミング・ガイド* を参照してください。

注: このコマンドは、OpenVMS、OS/2、および AIX クライアントでしか使用できません。

構文

```
runmqtmc [-m QMgrName] [-q InitiationQName]
```

オプション・パラメーター

-m *QMgrName*

クライアントのトリガー・モニターが操作を行う対象となるキュー・マネージャーの名前を指定します。これを省略すると、クライアントのトリガー・モニターは、デフォルト・キュー・マネージャーに対して操作を行います。

-q *InitiationQName*

処理される開始キューの名前を指定します。省略すると、SYSTEM.DEFAULT.INITIATION.QUEUE が使用されます。

戻りコード

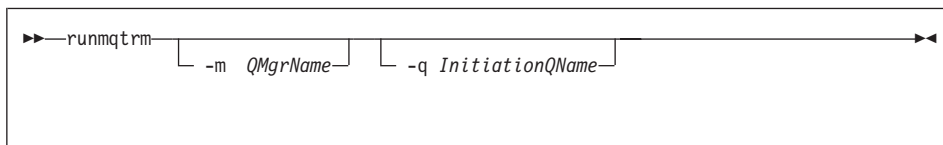
- 0** 使用されません。クライアントのトリガー・モニターは、連続的に実行されるよう設計されているので、終了しません。この値は予約されています。
- 10** クライアントのトリガー・モニターは、エラーによって割り込まれました。
- 20** エラー。クライアントのトリガー・モニターは実行されませんでした。

runmqtrm (トリガー・モニターの開始)

目的

runmqtrm コマンドは、トリガー・モニターを呼び出すために使用します。トリガー・モニターの詳細については、*MQSeries アプリケーション・プログラミング・ガイド* を参照してください。

構文



オプション・パラメーター

-m *QMgrName*

トリガー・モニターが操作を行う対象となるキュー・マネージャーの名前を指定します。これを省略すると、トリガー・モニターは、デフォルト・キュー・マネージャーに対して操作を行います。

-q *InitiationQName*

処理される開始キューの名前を指定します。省略すると、SYSTEM.DEFAULT.INITIATION.QUEUE が使用されます。

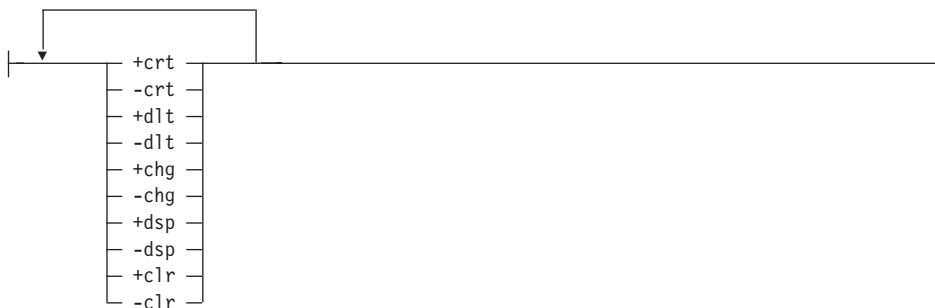
戻りコード

- 0** 使用されません。トリガー・モニターは、連続で実行するよう設計されているので、終了しません。この値は予約されています。
- 10** トリガー・モニターは、エラーによって割り込まれました。
- 20** エラー。トリガー・モニターは実行されませんでした。

setmqaut



管理許可:



総称許可:



説明

このコマンドは、許可の設定（つまり、操作の実行許可をユーザー・グループまたはプリンシパルに与えること）および許可のリセット（つまり、操作の実行許可を取り除くこと）の両方を行う場合に使用できます。許可が適用されるユーザー・グループおよびプリンシパルを指定し、同時にキュー・マネージャー、オブジェクト・タイプ、およびオブジェクトのオブジェクト名も指定する必要があります。単一のコマンドで、グループおよびプリンシパルをいくつでも指定できます。

重要: プリンシパルについて一組の許可を指定すると、同じ基本グループ内のすべてのプリンシパルにも同じ許可が与えられます。

与えることができる許可は、次のように分類されます。

- MQI 呼び出しの発行許可
- MQI コンテキストの許可
- 管理タスク用コマンドの発行許可
- 総称許可

変更するそれぞれの許可は、コマンドの一部として許可リストに指定します。そのリスト内の各項目は、接頭部「+」または「-」が付いた文字列です。たとえば、+put を許可リストに含めると、キューに対する MQPUT の発行許可を与えることになります。一方、-put を許可リストに含めると、MQPUT の発行許可を取り除くことになります。

複数の許可を、どのような順序でも指定できますが、それらの許可が対立しないようにする必要があります。たとえば、allmqi と set を同時に指定すると、対立が生じます。

必要な数のグループまたは許可を単一のコマンドに指定できます。

あるユーザー ID が複数のグループに属するメンバーである場合には、そのユーザー ID が属している各グループの許可を合併したものが、適用される許可になります。

必須パラメーター

-mQMgrName

許可変更の対象となるオブジェクトのキュー・マネージャーの名前を指定します。この名前は、最高 48 文字までです。

-n ObjectName

許可変更の対象となるオブジェクトの名前を指定します。

これは、それ自体がキュー・マネージャーでない限り、必須パラメーターです。キュー・マネージャー、キュー、またはプロセスの名前を指定する必要がありますが、総称名であってはなりません。

-t ObjectType

許可変更の対象となるオブジェクトのタイプを指定します。

可能な値は次のとおりです。

- **q** または **queue**
- **prcs** または **process**

- qmgr

オプション・パラメーター

-p *PrincipalName*

許可変更の対象となるプリンシパルの名前を指定します。

最低 1 人のプリンシパルまたは 1 つのグループが必要です。

-g *GroupName*

許可変更の対象となるユーザー・グループを表す権限 ID の名前を指定します。複数の権限 ID 名を指定できますが、それぞれの名前の前に -g フラグを付ける必要があります。

-s *ServiceComponent*

このパラメーターは、インストール可能な許可サービスを使用している場合にのみ適用されます。それ以外は、無視されます。

インストール可能な許可サービスがサポートされている場合、このパラメーターは、許可が適用される許可サービスの名前を指定します。このパラメーターはオプションです。これを指定しなければ、サービスの最初のインストール可能なコンポーネントに対して許可更新が行われます。

Authorizations

付与または除去される許可を指定します。リスト中の各項目には接頭部として、許可が与えられることを示す「+」または許可が除去されることを示す「-」が付きます。たとえば、MQI から MQPUT 呼び出しを発行する許可を与えるには、リストに +put を指定します。MQPUT 呼び出しを発行する許可を除去するには、-put を指定します。

表17は、種々のオブジェクト・タイプに与えることができる許可を示したものです。

表 17. 種々のオブジェクト・タイプについての許可の指定

許可	キュー	プロセス	キュー・マネージャー	名前リスト
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
clr	✓			

表 17. 種々のオブジェクト・タイプについての許可の指定 (続き)

許可	キュー	プロセス	キュー・マネージャー	名前リスト
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

MQI 呼び出しについての許可

altusr メッセージで代替ユーザー ID を使用する。

代替ユーザー ID の詳細については、*MQSeries アプリケーション・プログラミング・ガイド* を参照してください。

browse

BROWSE オプションを指定した MQGET 呼び出しを出して、キューからメッセージを取り出す。

connect

MQCONN 呼び出しを出して、指定のキュー・マネージャーにアプリケーションを接続する。

get MQGET 呼び出しを出して、キューからメッセージを取り出す。

inq MQINQ 呼び出しを出して、特定のキューの照会を行う。

put MQPUT 呼び出しを出して、特定のキューにメッセージを書き込む。

set MQSET 呼び出しを出して、MQI からキューに属性を設定する。

注: 複数のオプションを適用するようにキューをオープンする場合は、各オプションについての許可を持っている必要があります。

コンテキストについての許可

passall

すべてのコンテキストを指定のキューに渡す。すべてのコンテキスト・フィールドが元の要求からコピーされます。

passid

アイデンティティ・コンテキストを指定のキューに渡す。識別コンテキストは、要求の識別コンテキストと同じです。

setall すべてのコンテキストを指定のキューに設定する。これは特別なシステム・ユーティリティーによって使用されます。

setid アイデンティティ・コンテキストを指定のキューに設定する。これは特別なシステム・ユーティリティーによって使用されます。

コマンドについての許可

chg 指定のオブジェクトの属性を変更する。

clr 指定のキューをクリアする (PCF の Clear queue コマンドのみ)。

crt 指定のタイプのオブジェクトを作成する。

dlt 指定のオブジェクトを削除する。

dsp 指定のオブジェクトの属性を表示する。

一般操作についての許可

all オブジェクトに適用可能なすべての操作を使用する。

alladm

オブジェクトに適用可能なすべての管理操作を実行する。

allmqi オブジェクトに適用可能なすべての MQI 呼び出しを使用する。

戻りコード

- 0** コマンドは正常に終了しました。
- 36** 与えられた引き数が無効です。
- 40** キュー・マネージャーは利用不能です。
- 49** キュー・マネージャーは停止中です。
- 69** ストレージが利用不能です。
- 71** 予期しないエラーです。
- 72** キュー・マネージャー名のエラーです。
- 133** オブジェクト名が不明です。
- 145** 予期しないオブジェクト名です。
- 146** オブジェクト名が脱落しています。
- 147** オブジェクト・タイプが脱落しています。
- 148** オブジェクト・タイプが無効です。

- 149 エンティティ名が脱落しています。
- 150 許可の指定が脱落しています。
- 151 許可の指定が無効です。

例

1. 次の例は、許可を与える対象となるオブジェクトが、キュー・マネージャー saturn.queue.manager のキュー orange.queue であることを指定するコマンドを示しています。

```
setmqaut -m "saturn.queue.manager" -n "orange.queue" -t queue -g "tango" +inq +alladm
```

許可は、ユーザー・グループ tango に与えられ、関連した許可リストは、ユーザー・グループ tango について次のことを指定します。

- MQINQ 呼び出しを出すことができる。
 - 該当のオブジェクトに対してすべての管理操作を実行する許可がある。
2. 次の例では、許可リストはユーザー・グループ foxy について次のことを指定します。
 - 指定のキューに対して、MQI から呼び出しを出すことができない。
 - 指定のキューに対してすべての管理操作を実行する許可がある。

```
setmqaut -m "saturn.queue.manager" -n "orange.queue" -t queue -g "foxy" -allmqi +alladm
```

3. 次の例では、許可リストは、ユーザー・グループ waltz にはキュー・マネージャー saturn.queue.manager を作成および削除する許可があることを指定します。

```
setmqaut -m "saturn.queue.manager" -t qmgr -g "waltz" +crt +dlt
```

関連コマンド

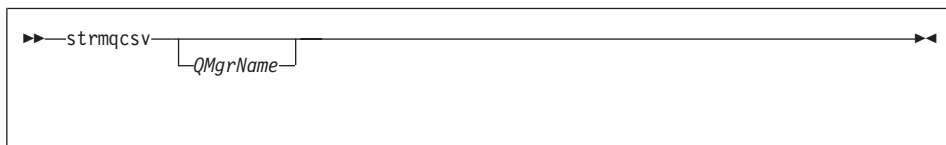
dspmqaout 許可の表示

strmqcsv (コマンド・サーバーの開始)

目的

strmqcsv コマンドは、指定したキュー・マネージャーのコマンド・サーバーを開始するために使用します。これにより、MQSeries はコマンド・キューに送られるコマンドを処理できます。

構文



オプション・パラメーター

QMgrName

コマンド・サーバーを開始させる対象となるキュー・マネージャーの名前を指定します。

戻りコード

- 0** コマンドは正常に終了しました。
- 10** コマンドは終了しましたが、予期しない結果がでました。
- 20** 処理中にエラーが起きました。

例

次のコマンドは、キュー・マネージャー `earth` のコマンド・サーバーを開始します。

```
strmqcsv "earth"
```

関連コマンド

- endmqcsv** コマンド・サーバーを終了します。
- dspmqcsv** コマンド・サーバーの状況を表示します。

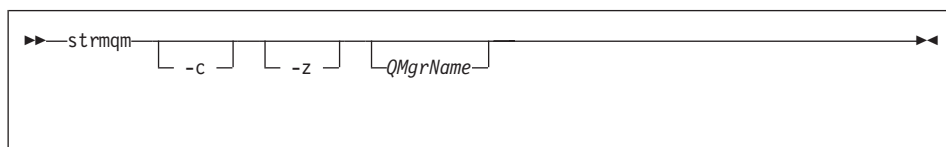
strmqm (キュー・マネージャーの開始)

目的

strmqm コマンドは、ローカル・キュー・マネージャーを開始するために使用します。

注: キュー・マネージャーを正常に始動および実行するには、最後にリブートした後、**strmqm** コマンドまたは他の制御コマンドを使用する前に、**mqc_startup** コマンドを一度実行しておく必要があります。

構文



オプション・パラメーター

-c キュー・マネージャーを開始し、デフォルトおよびシステム・オブジェクトを再定義してから、キュー・マネージャーを停止します。(キュー・マネージャーのデフォルトおよびシステム・オブジェクトは最初に **crtmqm** コマンドにより作成されます。) キュー・マネージャーに属する既存のシステムおよびデフォルト・オブジェクトは、このフラグを指定すると置き換えられます。

QMgrName

開始させるローカル・キュー・マネージャーの名前を指定します。これを省略すると、デフォルト・キュー・マネージャーが開始します。

-z エラー・メッセージを抑制します。

このフラグは、不要なエラー・メッセージを抑制するために MQSeries 内で使用します。このフラグを使用することによって情報が失われる可能性があるため、コマンド行からコマンドを入力する場合には、これは使用しないでください。

戻りコード

- 0 キュー・マネージャーは開始しました。
- 3 キュー・マネージャーは作成中です。
- 5 キュー・マネージャーは実行中です。

strmqm

- 16 キュー・マネージャーは存在していません。
- 23 ログは利用不能です。
- 49 キュー・マネージャーは停止中です。
- 69 ストレージが利用不能です。
- 71 予期しないエラーです。
- 72 キュー・マネージャー名のエラーです。
- 100 ログの位置が無効です。

例

次のコマンドは、キュー・マネージャー `account` を開始します。

```
strmqm "account"
```

関連コマンド

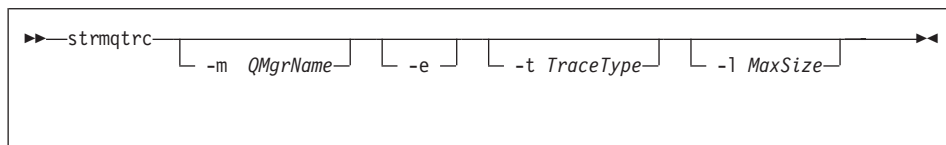
- crtmqm** キュー・マネージャーを作成します。
- dltmqm** キュー・マネージャーを削除します。
- endmqm** キュー・マネージャーを終了します。

strmqtrc (MQSeries トレースの開始)

目的

strmqtrc コマンドは、トレースを有効にする場合に使用します。このコマンドは、トレースが使用可能であってもなくても使用できます。トレースがすでに有効である場合は、有効なトレース・オプションがこのコマンドの最新の呼び出しで指定したオプションに変更されます。

構文



説明

詳細トレースの異なるレベルを要求できます。フローの `tracetype` 値 (`-t all` を含む) を指定するたびに、何らかの特定のトレース・タイプを表す `-t params` または `-t detail` のどちらかを指定してください。さもないと、そのトレース・タイプに関してはデフォルトの詳細なトレースしか生成されません。

このコマンドにより生成されるトレース・データの例については、268ページの『MQSeries トレースの使用』を参照してください。

オプション・パラメーター

-m*QMgrName*

トレースされるキュー・マネージャーの名前です。

キュー・マネージャー名および `-m` フラグは、`-e` フラグと同じコマンド上に指定できます。トレース対象のあるエンティティに対して複数のトレース指定が適用される場合、実際のトレースでは、指定されたすべてのオプションが有効になります。

`-e` フラグを指定しない場合に、`-m` フラグおよびキュー・マネージャー名を省略するとエラーになります。

- e** このフラグの指定により早期トレースを要求します。このため、キュー・マネージャーの作成や開始をトレースすることが可能です。これは、プロセスが所属する MQSeries コンポーネントを認識する前にトレース情報が書き込まれるということです。このフラグを指定すると、すべてのキュー

ー・マネージャーの任意のコンポーネントに属するどのプロセスでも早期処理がトレースされます。このフラグを指定しない場合、デフォルトでは早期トレースは実行されません。

-t *TraceType*

処理中にトレースできるポイントを定義します。このフラグを省略した場合は、すべてのトレース・ポイントが使用可能になり、完全なトレースが生成されます。

その他に、以下のリストにある 1 つまたは複数のオプションを指定することもできます。

注: 複数のトレース・タイプを指定する場合は、それぞれのトレース・タイプに**必ず** 1 つずつ **-t** フラグを付ける必要があります。 **-t** フラグは、それぞれ有効なトレース・タイプが関連付けられている限り、いくつでも指定できます。

複数の **-t** フラグに同じトレース・タイプを指定しても、エラーにはなりません。

all	システム内のすべてのトレース・ポイントについてデータを出力します。 -t フラグを指定しない場合も、これがデフォルトになります。
api	MQI および主なキュー・マネージャー・コンポーネントに関連するトレース・ポイントについてデータを出力します。
comms	通信ネットワークを介して流れるデータに関連するトレース・ポイントについてデータを出力します。
csflows	共通サービス内の処理フローに関連するトレース・ポイントについてデータを出力します。
lqmflows	ローカル・キュー・マネージャー内の処理フローに関連するトレース・ポイントについてデータを出力します。
remoteflows	通信コンポーネント内の処理フローに関連するトレース・ポイントについてデータを出力します。
otherflows	その他のコンポーネント内の処理フローに関連するトレース・ポイントについてデータを出力します。
csdata	共通サービス内の内部データ・バッファーに関連するトレース・ポイントについてデータを出力します。
lqmdata	ローカル・キュー・マネージャー内の内部データ・バッファーに関連するトレース・ポイントについてデータを出力します。

remotedata	通信コンポーネント内の内部データ・バッファーに関連するトレース・ポイントについてデータを出力します。
otherdata	その他のコンポーネント内の内部データ・バッファーに関連するトレース・ポイントについてデータを出力します。
versiondata	実行中の MQSeries のバージョンに関連するトレース・ポイントについてデータを出力します。
commentary	MQSeries コンポーネント内の注釈に関連するトレース・ポイントについてデータを出力します。

-l *MaxSize*

MaxSize の値は、100 万バイト単位でトレース・ファイル (AMQnnnn.TRC) の最大サイズを指定します。たとえば、*MaxSize* を 1 に指定した場合、トレースのサイズは 100 万バイトに制限されます。

トレース・ファイルは、指定した最大値に達すると、その名前を AMQnnnn.TRC から AMQnnnn.TRS に変更され、新しい AMQnnnn.TRC ファイルが開始されます。最大値の制限に達すると、すべてのトレース・ファイルが再始動されます。AMQnnnn.TRS ファイルの直前のコピーが存在している場合、そのファイルは削除されます。

戻りコード

AMQ7024	このメッセージは、コマンドに無効な引き数を指定した場合に出されます。
AMQ8304	最大数である 9 個の並行トレースがすでに実行されています。

例

このコマンドは、QM1 というキュー・マネージャーについて、共通サービスおよびローカル・キュー・マネージャーからのデータのトレースを有効にします。

```
strmqtrc -m QM1 -t csdata -t lqmdata
```

関連コマンド

dspmqtrc	定様式トレース出力の表示
endmqtrc	MQSeries トレースの終了

第3部 付録

付録A. MQSeries for Compaq OpenVMS の概要

プログラムと部品番号

- 5724-A38 MQSeries for Compaq OpenVMS Alpha バージョン 5 リリース 1、部品番号 0790997 (日本語版: 29P3587)。

ハードウェア要件

MQSeries サーバーとして使用できるのは、最低 128 MB のシステム・ディスク・スペースを持つ Compaq Alpha マシンです。

ソフトウェア要件

ソフトウェア要件は、特に注記がない限り、サーバーおよびクライアント Compaq OpenVMS 環境と同一です。

最低のサポート・レベルは次のとおりです。

- Compaq OpenVMS Alpha バージョン 7.2-1

接続性

MQSeries for Compaq OpenVMS は、次のネットワーク・プロトコルとハードウェアをサポートします。

ネットワーク・プロトコル:

- SNA LU6.2
- TCP/IP
- DECnet Phase V

さらに、DECnet か TCP/IP、または DIGITAL DECnet/SNA Gateway をサポートする通信ハードウェアでもかまいません。

DECnet 接続性の場合:

- DECnetPLUS Version 7.1 for OpenVMS Version 7.2-1

TCP/IP 接続性の場合:

- DIGITAL TCP/IP Services for OpenVMS AlphaV5.0.a および V5.1

ソフトウェア要件

- Process Software の TCPWare V5.4
- Process Software の Multinet V4.3

SNA 接続性の場合: SNA APPC LU6.2 ソフトウェア、およびライセンスをインストールする必要があります。適切に構成された SNA ゲートウェイへのアクセスが必要です。

- DECnet SNA Gateway ST V1.3
- DECnet SNA LU6.2 API V2.4

セキュリティ

MQSeries for Compaq OpenVMS は、MQSeries for Compaq OpenVMS 用のオブジェクト権限マネージャー (OAM) のセキュリティ機能を使用します。

すべての MQSeries リソースは、VMS 権限 ID MQM を使って実行します。この権限 ID は MQSeries インストール中に作成されたもので、このリソース属性をもつ権限 ID を、MQSeries リソースを制御する必要のあるすべてのユーザーに付与しなければなりません。

メンテナンス機能

MQSeries は、次のものを使用して機能します。

- **runmqsc** コマンド行インターフェース。

互換性

MQI for MQSeries for Compaq OpenVMS Alpha V5.1 は、バージョン 2.2.1.1 を実行している、既存のアプリケーションと互換性があります。

サポートされるコンパイラー

プログラムは、C、C++、COBOL、または Java で作成できます。

- C プログラムは DEC C コンパイラーを使用可能
- C++ プログラムは DEC C++ コンパイラーを使用可能
- COBOL プログラムは DEC COBOL コンパイラーを使用可能
- Java プログラムは Java コンパイラーを使用可能

言語の選択

提供されるメッセージ・テキスト・ファイルは、OpenVMS オペレーティング・システムに固有の、7 ビットの文字セットでエンコードされます。

国際化対応

MQSeries for Compaq OpenVMS によって、キュー・マネージャー・インスタンスの作成時に CCSID を指定できます。キュー・マネージャー CCSID は 819 をデフォルトとします。MQSeries for Compaq OpenVMS は、キュー・マネージャーの構成済み CCSID への文字セット変換をサポートします。ユーロ文字のサポートを提供するものを含め、MQSeries for Compaq OpenVMS キュー・マネージャーに指定できる CCSID についての詳細は、*MQSeries アプリケーション・プログラミング・リファレンス* を参照してください。

付録B. システム・デフォルト

crtmqm 制御コマンドを使用してキュー・マネージャーを作成する場合、システム・オブジェクトおよびデフォルト・オブジェクトが自動的に作成されます。

- システム・オブジェクトは、キュー・マネージャーまたはチャネルの操作に必要な MQSeries オブジェクトです。
- デフォルト・オブジェクトは、オブジェクトのすべての属性を定義します。ローカル・キューなどのオブジェクトを作成する場合、明示的に指定していない属性はすべてデフォルト・オブジェクトから継承されます。

表 18. キューのシステムおよびデフォルト・オブジェクト

オブジェクト名	説明
SYSTEM.DEFAULT.ALIAS.QUEUE	デフォルト別名キュー
SYSTEM.DEFAULT.LOCAL.QUEUE	デフォルト・ローカル・キュー
SYSTEM.DEFAULT.MODEL.QUEUE	デフォルト・モデル・キュー
SYSTEM.DEFAULT.REMOTE.QUEUE	デフォルト・リモート・キュー
SYSTEM.DEAD.LETTER.QUEUE	サンプルの送達不能 (未配布メッセージ) キュー
SYSTEM.DEFAULT.INITIATION.QUEUE	デフォルト開始キュー
SYSTEM.CICS.INITIATION.QUEUE	デフォルト CICS [®] 開始キュー
SYSTEM.ADMIN.COMMAND.QUEUE	管理者コマンド・キュー。リモート MQSC コマンドおよび PCF コマンドに使用されます。
SYSTEM.MQSC.REPLY.QUEUE	MQSC 応答先キュー。これは、リモート MQSC コマンドに対する応答の一時動的キューを作成するモデル・キューです。
SYSTEM.ADMIN.QMGR.EVENT	キュー・マネージャー・イベントのイベント・キュー
SYSTEM.ADMIN.PERFM.EVENT	パフォーマンス・イベントのイベント・キュー
SYSTEM.ADMIN.CHANNEL.EVENT	チャネル・イベントのイベント・キュー
SYSTEM.CHANNEL.INITQ	チャネル開始キュー
SYSTEM.CHANNEL.SYNCQ	チャネルの同期データを保持するキュー

デフォルト

表 18. キューのシステムおよびデフォルト・オブジェクト (続き)

オブジェクト名	説明
SYSTEM.CLUSTER.COMMAND.QUEUE	リポジトリ・キュー・マネージャーにメッセージを転送するために使用されるキュー
SYSTEM.CLUSTER.REPOSITORY.QUEUE	すべてのリポジトリ情報を保管するために使用されるキュー
SYSTEM.CLUSTER.TRANSMIT.QUEUE	クラスターに対するすべてのメッセージのための伝送キュー

表 19. チャネルのシステムおよびデフォルト・オブジェクト

オブジェクト名	説明
SYSTEM.DEF.SENDER	デフォルト送信側チャネル
SYSTEM.DEF.SERVER	デフォルト・サーバー・チャネル
SYSTEM.DEF.RECEIVER	デフォルト受信側チャネル
SYSTEM.DEF.REQUESTER	デフォルト要求側チャネル
SYSTEM.DEF.SVRCONN	デフォルト・サーバー接続チャネル
SYSTEM.DEF.CLNTCONN	デフォルト・クライアント接続チャネル
SYSTEM.AUTO.RECEIVER	動的受信側チャネル
SYSTEM.AUTO.SVRCONN	動的サーバー接続チャネル
SYSTEM.DEF.CLUSRCVR	クラスターのためのデフォルト受信側チャネル。クラスター内のキュー・マネージャー上に CLUSRCVR チャネルを作成するときに指定しなかった属性のデフォルト値を提供するために使用されます。
SYSTEM.DEF.CLUSSDR	クラスターのためのデフォルト送信側チャネル。クラスター内のキュー・マネージャー上に CLUSSDR チャネルを作成するときに指定しなかった属性のデフォルト値を提供するために使用されます。

表 20. 名前リストのシステムおよびデフォルト・オブジェクト

オブジェクト名	説明
SYSTEM.DEFAULT.NAMELIST	デフォルト名前リスト

表 21. プロセスのシステムおよびデフォルト・オブジェクト

オブジェクト名	説明
SYSTEM.DEFAULT.PROCESS	デフォルト・プロセス定義

デフォルト

付録C. ディレクトリー構造

402ページの図26 は、特定のキュー・マネージャーに関連したデータおよびログのディレクトリーの全体的なレイアウトを示したものです。この図に示したディレクトリーは、デフォルトのインストールに適用されます。これを変更する場合は、ファイルおよびディレクトリーの位置もそれに応じて修正されます。

ディレクトリー構造

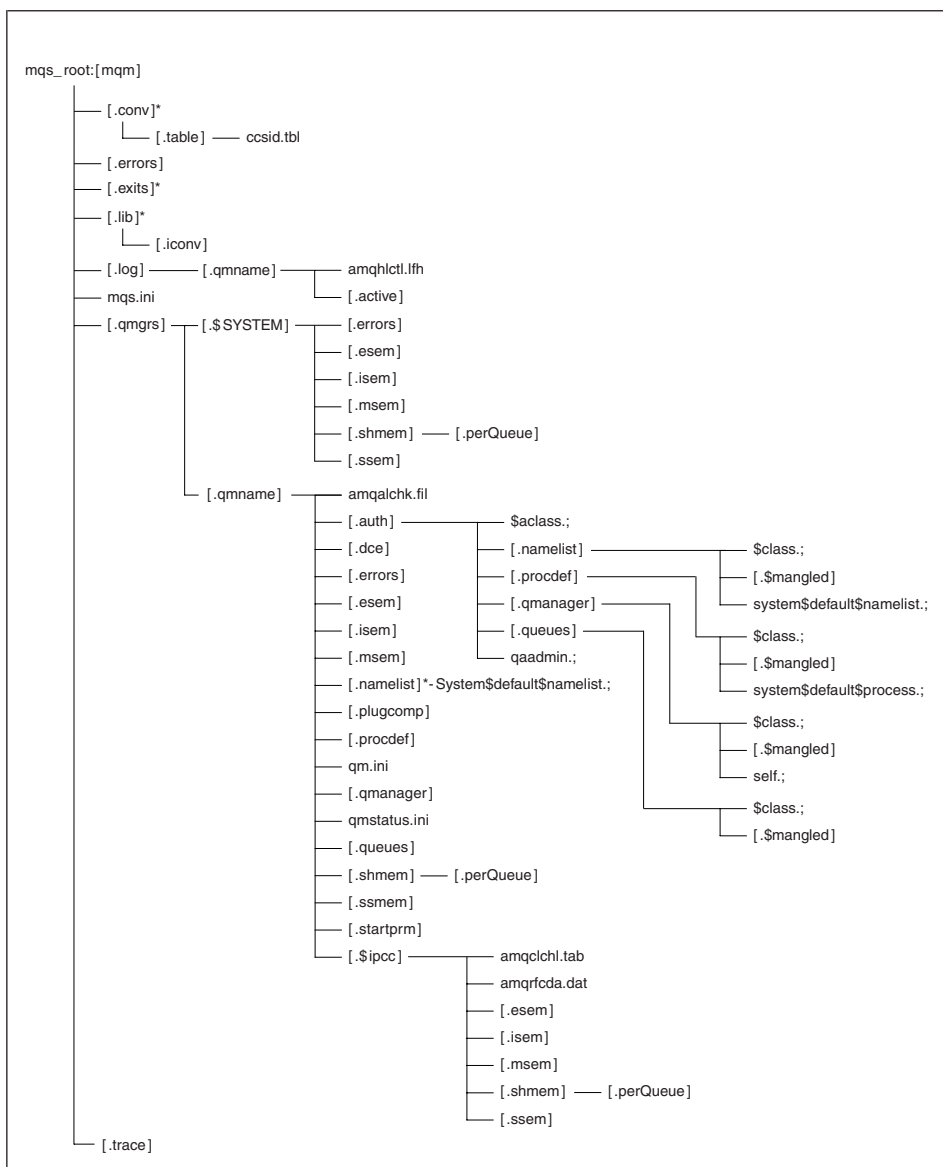


図 26. キュー・マネージャーが開始された後のデフォルト・ディレクトリー構造

図26 では、このレイアウトは、キュー・マネージャーがしばらく使用された後の代表的なMQSeriesです。実際の構造は、キュー・マネージャーに対して行われた操作によって異なります。

MQS_ROOT:[MQM] 中のディレクトリーとファイル

デフォルトでは、次のディレクトリーおよびファイルがディレクトリー MQS_ROOT:[MQM] にあります。

.conv このディレクトリーには、データ変換に使用されるすべてのファイルが含まれます。

.table このディレクトリーには ccsid.tbl. ファイルが含まれます。

.errors

このディレクトリーには、最新のものから最古のものまでのオペレーター・メッセージ・ファイルが含まれます。

AMQERR01.LOG

AMQERR02.LOG

AMQERR03.LOG

.exits ユーザー作成出口を入れる空のディレクトリー。

.lib このディレクトリーには subdirectory .iconv が含まれます。サブディレクトリーには、すべてのコード・セット変換テーブルが含まれます。

.iconv コード・セット変換テーブル (002501B5.TBL から 44B031A8.TBL など) が含まれるディレクトリー。

.log MQSeries をインストールし、キュー・マネージャーを作成して、これを開始してからしばらく使用すると、このディレクトリーには以下のサブディレクトリーとファイルが含まれることとなります。

amqhlctl.lfh

ログ制御ファイル。

active このディレクトリーには、次のように番号が付いたログ・ファイルが含まれます。

S0000000.LOG

S0000001.LOG

S0000002.LOG

... 以下同様に続く。

mqs.ini

MQSeries 構成ファイル。

.qmgrs

このディレクトリーには、各キュー・マネージャーごとに、サブディレクトリー `.$system` およびサブディレクトリー `.qmname` が入っています。`.$system` ディレクトリーには、MQSeries が内部的に使用するディレクトリーおよびファイルが含まれます。`.qmname` サブディレク

ディレクトリー構造

トリーの詳細については、『MQS_ROOT:[MQM.QMGRS.QMNAME] サブディレクトリーの中のディレクトリーとファイル』を参照してください。

.trace このディレクトリーには、**strmqtrc** コマンドによって作成されるトレース・ファイルが入っています。

MQS_ROOT:[MQM.QMGRS.QMNAME] サブディレクトリーの中のディレクトリーとファイル

デフォルトでは、次のディレクトリーおよびファイルがディレクトリー MQS_ROOT:[MQM.QMGRS.QMNAME] にあります。システム上で作成および実行されるすべてのキュー・マネージャーのために、.QMNAME が作成されません。

amqalchk.fil

最新のチェックポイントの情報を含んでいるチェックポイント・ファイル。

.auth このディレクトリーには、許可に関連したサブディレクトリーおよびファイルが入っています。

\$aclass.;

このファイルには、すべてのクラスの許可スタanzasが入っています。

.namelist

このディレクトリーには、各名前リストのファイルが入っています。各ファイルには、関連した名前リストに関する許可スタanzasが入っています。

\$class.;

このファイルには、名前リスト・クラスの許可スタanzasが入っています。

.\$mangled

無効な OpenVMS 文字が含まれている名前リスト名は、自動的に有効な OpenVMS 名に変換されます。有効な OpenVMS 名はこのファイルに入っています。28ページの『MQSeries のファイル名についての理解』を参照してください。

system\$default\$namelist

このファイルには、システム・デフォルト名前リストの許可スタanzasが入っています。

.procdef

各 MQSeries プロセス定義は、このディレクトリーのファイルの 1 つに関連付けられています。

\$class.;

このファイルには、プロセス定義クラスの許可スタanzasが入っています。

.\$mangled

無効な OpenVMS 文字が含まれているプロセス定義名は、自動的に有効な OpenVMS 名に変換されます。有効な OpenVMS 名はこのファイルに入っています。28ページの『MQSeries のファイル名についての理解』を参照してください。

.system\$default\$process.;

このファイルには、システム・デフォルト・プロセスの許可スタanzasが入っています。

.qmanager

このディレクトリーには、各キュー・マネージャーのファイルが入っています。各ファイルには、関連したキュー・マネージャーに関する許可スタanzasが入っています。

\$class.;

このファイルには、キュー・マネージャー・クラスの許可スタanzasが入っています。

.\$mangled

無効な OpenVMS 文字が含まれているキュー・マネージャー定義名は、自動的に有効な OpenVMS 名に変換されます。有効な OpenVMS 名はこのファイルに入っています。28ページの『MQSeries のファイル名についての理解』を参照してください。

self.; このファイルには、キュー・マネージャー・オブジェクトの許可スタanzasが入っています。

.queues

このディレクトリーには、各キューのファイルが入っています。各ファイルには、関連したキューに関する許可スタanzasが入っています。

\$CLASS

このファイルには、キュー・クラスの許可スタンプが入っています。

.\$mangled

無効な OpenVMS 文字が含まれているキュー名は、自動的に有効な OpenVMS 名に変換されます。有効な OpenVMS 名はこのファイルに入っています。28ページの『MQSeries のファイル名についての理解』を参照してください。

キューの定義ファイル

各ファイルは、キュー・マネージャーに事前定義されたオブジェクトに対応します。

```
system$admin$channel$event.;
system$admin$command$queue.;
system$admin$perfm$event.;
system$admin$qmgr$event.;
system$channel$initq.;
system$channel$syncq.;
system$scics$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.;

許可を制御するために内部で使用されるファイル。

.dce DCE サポート用に予約されている空のディレクトリー。

.errors

このディレクトリーには、最新のものから最古のものまでのオペレーター・メッセージ・ファイルが含まれます。

```
amqerr01.log
amqerr02.log
amqerr03.log
```

- .esem** 内部で使用されるファイルを含んでいるディレクトリー
- .isem** 内部で使用されるファイルを含んでいるディレクトリー
- .msem**
 - 内部で使用されるファイルを含んでいるディレクトリー
- .namelist**
 - このディレクトリーには、各キュー・マネージャーの名前リストが入っています。
- .plugcomp**
 - この空のディレクトリーは、インストール可能なサービス用に予約されています。
- .procdef**
 - 各 MQSeries プロセス定義は、このディレクトリーのファイルの 1 つに関連付けられています。ファイル名は、プロセス定義名と一致します。
- qm.ini** キュー・マネージャー構成ファイル。
- .qmanager**
 - キュー・マネージャー・オブジェクト。
- qmstatus.ini**
 - このファイルには、キュー・マネージャーの状況を記述するテキストが含まれます。
- .queues**
 - 各キューには、「q」 という単一のファイルを含んでいるディレクトリーがあります。
 - ファイル名は、制限付きでキュー名に一致します。それらの制限については、28ページの『MQSeries のファイル名についての理解』を参照してください。
- .shmem**
 - .perQueue**
 - 内部で使用されるファイルを含んでいるディレクトリー
- .ssem** 内部で使用されるファイルを含んでいるディレクトリー
- .startprm**
 - 内部で使用される一時ファイルを含んでいるディレクトリー。
- .\$ipcc**

ディレクトリー構造

amqclchl.tab

クライアント・チャンネル・テーブル・ファイル。

amqrfcda.dat

チャンネル・テーブル・ファイル。

.esem 内部で使用されるファイルを含んでいるディレクトリー

.isem 内部で使用されるファイルを含んでいるディレクトリー

.msem 内部で使用されるファイルを含んでいるディレクトリー

.shmem

.perQueue

内部で使用されるファイルを含んでいるディレクトリー

.ssem 内部で使用されるファイルを含んでいるディレクトリー

付録D. コマンド・セットの比較

次の表は、異なる管理コマンド・セットから使用することの可能な機能を比較しています。

- 『キュー・マネージャーの管理用のコマンド』
- 410ページの『コマンド・サーバーの管理用のコマンド』
- 410ページの『キューの管理用のコマンド』
- 411ページの『プロセスの管理用のコマンド』
- 411ページの『チャンネルの管理用のコマンド』
- 412ページの『その他の制御コマンド』

注: MQSeries for Compaq OpenVMS に当てはまる MQSC コマンドだけが示されています。

キュー・マネージャーの管理用のコマンド

表 22. キュー・マネージャーの管理用のコマンド

PCF	MQSC	制御
Change Queue Manager	ALTER QMGR	-
(キュー・マネージャーの作成)*	-	crtmqm
(キュー・マネージャーの削除) *	-	dltmqm
Inquire Queue Manager	DISPLAY QMGR	-
(キュー・マネージャーの停止)*	-	endmqm
Ping Queue Manager	PING QMGR	-
(キュー・マネージャーの開始)*	-	strmqm
注: * PCF コマンドにはありません。		

コマンド・セットの比較

コマンド・サーバーの管理用のコマンド

表 23. コマンド・サーバーの管理用のコマンド

説明	制御
コマンド・サーバーの表示	dspmqcsv
コマンド・サーバーの開始	strmqcsv
コマンド・サーバーの停止	endmqcsv
注: このグループの機能は、制御コマンドとしてのみ使用可能です。このグループに相当する MQSC コマンドまたは PCF コマンドはありません。	

キューの管理用のコマンド

表 24. キューの管理用のコマンド

PCF	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
注: このグループに相当する制御コマンドはありません。	

プロセスの管理用のコマンド

表 25. プロセスの管理用のコマンド

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
注: このグループに相当する制御コマンドはありません。	

チャンネルの管理用のコマンド

表 26. チャンネルの管理用のコマンド

PCF	MQSC	制御
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	-	runmqlsr
Stop Channel	STOP CHANNEL	-

その他の制御コマンド

表 27. その他の制御コマンド

説明	制御
MQSeries 変換出口の作成	crtmqcvx
許可の表示	dspmqaout
オブジェクトが使用するファイルの表示	dspmqls
MQSeries 定様式トレース出力の表示コマンド	dspmqtrc
MQSeries トレースの終了	endmqtrc
フェールオーバー・セットの管理	failover
メディア・イメージの記録	rcdmqimg
メディア・オブジェクトの再作成	rcrmqobj
MQSeries トランザクションの解決	rsvmqtrn
MQSC コマンドの実行	runmqsc
トリガー・モニターの実行	runmqtrm
クライアントのトリガー・モニターの実行	runmqtrmc
許可の設定またはリセット	setmqaut
フェールオーバー・モニターを開始	runmqfm
MQSeries トレースを開始	strmqtrc
注: このグループの機能は、制御コマンドとしてのみ使用可能です。これに直接的に相当する PCF または MQSC コマンドはありません。	

付録E. MQI プログラムおよび MQSC ファイルのサンプル

MQSeries for Compaq OpenVMS には、使用して試してみることのできる MQI プログラム、および MQSC コマンド・ファイルのサンプルのセットが用意されています。これらについては、次のセクションで説明しています。

- 『MQSC コマンド・ファイルのサンプル』
- 『C および COBOL プログラム・サンプル』
- 414ページの『各種ツール』

MQSC コマンド・ファイルのサンプル

表28 は、MQSC コマンド・ファイルのサンプルを示しています。これらは、MQSC コマンドを含む単純な ASCII テキスト・ファイルです。各ファイルに対して順番に `runmqsc` コマンドを呼び出し、ファイルで指定されるオブジェクトを作成できます。55ページの『システムに提供された MQSC コマンド・ファイルを実行する』を参照してください。

デフォルトでは、これらのファイルはディレクトリー `MQS_EXAMPLES:` にあります。

表28. MQSC コマンド・ファイル

ファイル名	目的
AMQSCOS0.TST	C および COBOL プログラム・サンプルで使用するための、MQI オブジェクトのセットを作成します。

C および COBOL プログラム・サンプル

414ページの表29 は、MQI ソース・ファイルのサンプルを示しています。デフォルトでは、ソース・ファイルはディレクトリー `MQS_EXAMPLES:`、およびコンパイル済みバージョンは `MQS_EXAMPLES:` の下の `[.BIN]` ディレクトリーにあります。プログラムの実行内容とその使用方法についての詳細は、*MQSeries アプリケーション・プログラミング・ガイド* を参照してください。

例

表 29. サンプル・プログラム - ソース・ファイル

C	COBOL	目的
AMQSBCG0.C	-	指定されたキュー上のすべてのメッセージの、メッセージ記述子およびメッセージ・コンテキスト・フィールドの両方を読み取ってから出力します。
AMQSECHA.C	AMQVECHX.COB	メッセージを、メッセージ・キューから応答先キューにエコーします。起動されるアプリケーション・プログラムとして実行できます。
AMQSGBR0.C	AMQ0GBR0.COB	メッセージをキューから SYSS\$OUTPUT に書き込みます。メッセージはキューに残ります。ブラウザ・オプションを指定して MQGET を使用します。
AMQSGET0.C	AMQ0GET0.COB	名前付きキューからメッセージを除去し (MQGET を使用)、SYSS\$OUTPUT に書き込みます。
AMQSINQA.C	AMQVINQX.COB	起動されるキューを読み取ります。各要求はキュー名として読み取られ、応答としてそのキューに関する情報が戻されます。
AMQSPUT0.C	AMQ0PUT0.COB	SYSS\$INPUT をメッセージにコピーしてから、このメッセージを指定されたキューに書き込みます。
AMQSREQ0.C	AMQ0REQ0.COB	要求メッセージを指定されたキューに書き込み、応答メッセージを表示します。
AMQSSETA.C	AMQVSETX.COB	名前付きキューへの書き込みを禁止し、応答として結果のステートメントが戻されます。起動されるアプリケーションとして実行します。
AMQSTRG0.C	-	名前付き開始キューを読み取ってから、各トリガー・メッセージと関連したプログラムを開始するトリガー・モニター。提供される runmqtrm コマンドの全トリガー機能のサブセットを提供します。
AMQSVFCX.C	-	データ変換出口ルーチンの C スケルトンのサンプル。

各種ツール

これらのツール・ファイルは、フォーマッターおよびコード変換をサポートするために提供されています。

表 30. 各種ファイル

ファイル名	位置	目的
AMQTRC.FMT	SYSS\$LIBRARY	MQSeries トレース・フォーマット設定を定義します。

表 30. 各種ファイル (続き)

ファイル名	位置	目的
CCSID.TBL	MQS_ROOT:[MQM.CONV.TABLE]	新たにサポートされる CCSID 値を MQSeries システムに追加するには、このファイルを編集します。CCSID についての詳細は、CDRA (Character Data Representation Architecture) に関する文書を参照してください。

例

付録F. OpenVMS クラスタ・フェールオーバー・セット・テンプレート

この付録には、以下のフェールオーバー・セット・テンプレートが含まれます。

- 『テンプレート構成ファイル FAILOVER.TEMPLATE』
- 419ページの『テンプレート StartCommand プロシージャ START_QM.TEMPLATE』
- 421ページの『テンプレート EndCommand プロシージャ END_QM.TEMPLATE』
- 425ページの『テンプレート TidyCommand プロシージャ TIDY_QM.TEMPLATE』

テンプレート構成ファイル 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
```

FAILOVER.TEMPLATE

```
#
# 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
```

図 27. テンプレート構成ファイル: *failover.template*

テンプレート StartCommand プロシージャー 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
$!
$! runmqtsr -t tcp -p 'p5' -m 'p1'

```

START_QM.TEMPLATE

```
$!  
$! Insert commands to start any applications  
$!  
$exit
```

図 28. テンプレート *StartCommand* プロシージャ: *Start_QM.template*

テンプレート EndCommand プロシージャ 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_QM.TEMPLATE

```

$! 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
$!
$! 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

```

END_QM.TEMPLATE

```
$else  
$ exit SS$_TIMEOUT  
$endif
```

図 29. テンプレート *EndCommand* プロシージャー: *END_QM.template*

テンプレート TidyCommand プロシージャー 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

```

TIDY_QM.TEMPLATE

```
$!  
$! @multinet:start_server  
$!  
$exit
```

図 30. テンプレート *TidyCommand* プロシージャ: *TIDY_QM.template*

付録G. MQSeries for Compaq OpenVMS でのコード・セットのサポート

MQSeries for Compaq OpenVMS は、ロケール (特定の文化のための規則を定義する、ユーザーの環境のサブセット) で使用されるコード・セットの大部分をサポートします。そのようなコード・セットは、標準で MQSeries for Compaq OpenVMS に用意されています。

ロケールを設定しないと、使用される CCSID は 819、つまり ISO8859-1 コード・セットになります。

メッセージおよびメッセージ・ヘッダー・データに使用されるコード・セットを識別するために MQSeries で使用される CCSID (Coded Character Set Identifier) は、ロケール構成の LC_CTYPE カテゴリを分析することによって取得されます。

表31 は、ロケール、およびロケールが使用するコード・セットに登録される CCSID を示します。

表 31. ロケールと CCSID

ロケール	言語	コード・セット	CCSID
C	英語	ISO8859-1	819
CS_CZ_ISO8859-2	チェコ語	ISO8859-2	912
DA_DK_ISO8859-1	デンマーク語	ISO8859-1	819
DE_DE_ISO8859-1	ドイツ語	ISO8859-1	819
DE_CH_ISO8859-1	ドイツ語 (スイス)	ISO8859-1	819
EL_GR_ISO8859-7	ギリシャ語	ISO8859-7	813
EN_GB_ISO8859-1	英語 (英国)	ISO8859-1	819
EN_US_ISO8859-1	英語 (米国)	ISO8859-1	819
ES_ES_ISO8859-1	スペイン語	ISO8859-1	819
FI_FI_ISO8859-1	フィンランド語	ISO8859-1	819
FR_FR_ISO8859-1	フランス語 (フランス)	ISO8859-1	819
FR_BE_ISO8859-1	フランス語 (ベルギー)	ISO8859-1	819

サポートされるコード・セット

表 31. ロケールと CCSID (続き)

ロケール	言語	コード・セット	CCSID
FR_CA_ISO8859-1	フランス語 (カナダ)	ISO8859-1	819
FR_CH_ISO8859-1	フランス語 (スイス)	ISO8859-1	819
HU_HU_ISO8859-2	ハンガリー語	ISO8859-2	912
IS_IS_ISO8859-1	アイスランド語	ISO8859-1	819
IT_IT_ISO8859-1	イタリア語 (イタリア)	ISO8859-1	819
IW_IL_ISO8859-8	ヘブライ語	ISO8859-8	916
JA_JP_EUCJP	日本語	eucJP	954
JA_JP_SDECKANJI	日本語	SDECKANJI	954**
JA_JP_SJIS	日本語	SJIS	932
KO_KR_DECKOREAN	韓国語	DECKOREAN	970**
NL_NL_ISO8859-1	オランダ語 (オランダ)	ISO8859-1	819
NL_BE_ISO8859-1	オランダ語 (ベルギー)	ISO8859-1	819
NO_NO_ISO8859-1	ノルウェー語	ISO8859-1	819
PL_PL_ISO8859-2	ポーランド語	ISO8859-2	912
PT_PT_ISO8859-1	ポルトガル語	ISO8859-1	819
SK_SK_ISO8859-2	スロバキア語	ISO8859-2	912
RU_RU_ISO8859-5	キリル文字	ISO8859-5	915
SV_SE_ISO8859-1	スウェーデン語	ISO8859-1	819
TR_TR_ISO8859-9	トルコ語	ISO8859-9	920
ZH_CN_DECHANZI	中国語 (簡体字)	DECHANZI	1383**
ZH_HK_DECHANZI	中国語 (簡体字)	DECHANZI	1383**
ZH_HK_EUCTW	中国語 (繁体字)	eucTW	964
ZH_HK_EUCTW	中国語 (繁体字)	eucTW	964
ZH_HK_DECHANYU	中国語 (繁体字)	DECHANYU	964**
ZH_TW_DECHANYU	中国語 (繁体字)	DECHANYU	964**
ZH_HK_BIG5	中国語 (繁体字)	big5	950
ZH_TW_BIG5	中国語 (繁体字)	big5	950

注:

** 使用される CCSID は、登録済みの IBM CCSID に最も近いものです。

これらのロケールのプラットフォーム相互のサポートの詳細については、*MQSeries* アプリケーション・プログラミング・リファレンス を参照してください。

付録H. MONMQ 診断ユーティリティー

MONMQ ユーティリティーは、MQSeries for Compaq OpenVMS で発生する問題の診断および解決を援助するためのツールです。MONMQ ユーティリティーは、コマンド行から、または DCL スクリプト内から、対話式に使用できます。

MONMQ ユーティリティーは、一般的に次の用途に使われます。

- 共用メモリーの管理
- OpenVMS リソース使用法に関する情報収集の支援
- 実行中のキュー・マネージャーからのトレース出力の取得

MONMQ には、パラメーターを補助するヘルプ・システムがあり、MONMQ コマンドのスクリプトも実行できます。MONMQ が開始すると、デフォルト・スクリプト `sys$manager:mqs_trace_startup.mqt` が実行され、初期構成を提供します。

```
$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]
```

概説

OpenVMS での MQSeries のトレースは、グローバル・セクションおよびメールボックスを使ってインプリメントされます。MQSeries がインストールされた 1 つのノードに、最高 10 までのトレース・セクション (LU) が共存できます。しかし、いつでも、トレース・セッションが 1 つの LU を使用するよう強くお勧めします。また、複数のユーザーが、一度に同じ LU をオープンすることはあまりお勧めできません。これらの条件の結果は予測不能なものとなります。

各共用セクション (LU) には、チャンネル定義および LU 定義自体が入っています。各チャンネル定義には、接続されたスレッドの詳細、スレッド私用スタッ

MONMQ の概要

ク、およびスレッド循環バッファが含まれます。さらに、共用セクションには、MONMQ および接続されたスレッド間のプロセス間通信に使用されるフラグのセットが入っています。

各 LU ごとに、リアルタイムでトレース・メッセージを受け取るのに使用される、関連したメールボックスがあります。リアルタイム・トレースを実行するには、クライアント・プロセスを、**TRACE START** コマンドで開始する必要があります。この専用の切り離されたプロセスは、LU メールボックスにメッセージが受信されると、各メッセージを読み取り、フォーマットし、表示します。接続された各スレッドは同じメールボックスに書き込みを行い、したがって MQSeries プロセス / スレッド間の相互通信を物理的に表示する機能が提供されます。

MONMQ は、正しく駆動されれば、プロセス間のタイミングの問題、使い果たされたオペレーティング・システム・リソース、またはコーディングの問題などの問題を診断するための包括的な方法を備えるものとなります。

MONMQ コマンドについては、この付録で説明します。

MONMQ 内の変数

MONMQ 内の多くのコマンドは、変数を利用します。変数は、コマンド内で指定されていない場合は、**set** コマンドで定義されるデフォルト値を使用します。変数が **set** 以外のコマンドで使用されると、その変数用のデフォルト値は変更されません。

変数は、次のものを含むことができます。

- 整数 (10 進または 16 進)
16 進値は、先頭に 0x を付けて入力するか、または 16 進値が期待される場所に文字 a-f を付けた値を入力することによって、入力可能です。
- テキスト。ただし引用符で囲む。
- 範囲。最小:最大を書き込むことによって入力する。
範囲を使用することにより、たとえばコマンドの適用されるチャンネルの範囲を指定できます。

たとえば、次のとおりです。

```

MQT> set lu=2
MQT> set pid=0x223
MQT> set pid=2fa
MQT> set buffile="filename.buf"
MQT> set chl=0:20

```

変数用の現行のデフォルト値は、変数コマンドを使用して表示できます。

```

MQT> variables
defined variables
lu=0:0                nochls=20                buffer=1000
chl=0:20              component=0(HEX)        line=0
mask=0(HEX)           pid=0(HEX)              node=(null)
function=0(HEX)       div=0                    depth=32
resource=0             wait=1(BOOL)            timestamp=0(BOOL)
listfile=(null)       buffile=(null)          step=0(BOOL)
active=0(BOOL)        fname=(null)            delay=100
post=0(BOOL)

defined constants
fent=1(HEX)           fout=2(HEX)             ferr=4(HEX)
fxxx=8(HEX)           dgn=10(HEX)            shm=20(HEX)
spl=40(HEX)           evt=80(HEX)            mtx=100(HEX)
prc=200(HEX)          msc=400(HEX)           inf=800(HEX)
log=2000(HEX)         shl=4000(HEX)          memory=3
mutex=4                mailbox=5               nanoseconds=1
microseconds=2         milliseconds=3         seconds=4

```

コマンドを単純化するため、デフォルト値を設定することをお勧めします。たとえば、以下のコマンド・シーケンスは、どちらも機能的に同一です。

```

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

```

または、

```

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

```

MONMQ コマンドは、コマンドの固有性を保つのに必要な最小の文字数まで省略できます。上記の一連のコマンドは、さらに次のように短縮できます。

MONMQ 内の変数

```
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
```

また、MONMQ は変数を使って単純な算術演算を実行することもできます。したがって、`set lu=lu+1` などのコマンドが使用可能です。

新しい変数は、**declare** コマンドで宣言できます。パラメーターは次のとおりです。

- 変数名
- 変数タイプ
- ヘルプ・テキスト。これにより、ヘルプ・コマンドがヘルプ・テキストを検索できるようになります。

```
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  AMQZXMA0.EXE
MQT> help ec

VARIABLE ec:
channel number for execution controller

MQT>
```

デフォルト値の割り当て

`DEFAULT variable=<expression> [variable=expression] ...`

このコマンドによって、MONMQ 内で定義されるすべての変数に、デフォルト値を割り当てることができます。いったん設定されると、デフォルト変数名はコマンド行から省略できるようになります。たとえば、次のとおりです。

```
MQT>default lu=2 chl=3:6
```

このコマンドは、`lu` 変数にデフォルト値 2 を設定し、チャンネル変数に値 3 から 6 までを設定します。今後、**show channels** などの典型的なコマンドを使用する際、`lu 2` のチャンネル 3 から 6 までが表示されます。

デフォルト値は、変数がコマンド行から省略される場合にのみ使用されます。すべてのデフォルト値は、始動スクリプト・ファイル `MQS_TRACE_STARTUP.MQT` で設定されます。このファイルは、必要に合わせて編集することができます。

トレース・セクションおよび関連するメールボックスのオープンまたは作成

`OPEN [lu=number] [nochls=number] [buffer=number]`

このコマンドは、トレース・セクションおよび関連するメールボックスをオープンまたは作成します。**open** コマンドは、MQSeries プロセスのトレースに必要な基本リソースを作成します。各 LU には、MQSeries プロセスと通信するために使用される、関連したセクションとメールボックスがあります。

このコマンドは、3 つのオプション・パラメーターを使用します。最初のパラメーター [LU] は、トレース・セクション / メールボックスに割り当てられる数値で、他のほとんどの MONMQ コマンドによって参照として使用されます。最高 10 の LU を、単一のノード上で作成することができます。デフォルト値はゼロです。指定された LU がすでに存在する場合、MONMQ は既存のトレース・セクションに接続します。セクションが存在しない場合、新しいセクションが作成されます。

2 番目のパラメーター [nochls] は、この LU が持つチャンネルの数を指定します。各チャンネルは、単一の MQSeries プロセス/スレッド接続を表します。デフォルト値は 20 です。

3 番目のパラメーター [buffer] は、各チャンネルごとのトレース・ヒストリー・バッファの最大サイズを指定します。デフォルトは 1000 です。

他の MONMQ コマンドを実行できるようにする前に、最低 1 つの LU をオープンする必要があります。

論理装置定義の表示

`SHOW SEGMENT [lu=range]`

このコマンドは、論理装置定義を表示します。以下に、出力例と、コマンド **show segment lu=0** のタイプ時に各フィールドの横に表示される説明を示します。

CLOSE LU

```
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      : MBAI065:    /* 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)
```

LU のクローズおよび削除

CLOSE [lu=*number*]

このコマンドは、**OPEN** の反対を実行し、指定された LU をクローズおよび削除します。LU は、制御された順序でクローズします。まず、接続されている各プロセスに切断のシグナルが送られ、次に各チャンネルがリセットされ、最後にトレース・メールボックスの割り当て解除と共用セクションの削除が行われます。このコマンドは、トレース・セッションが完了した後にだけ実行してください。

チャンネル詳細の表示

SHOW CHANNELS [full] [connected] [chl=*range*]

このコマンドは、指定されたチャンネルの詳細を表示します。[connected] パラメーターを指定すると、スレッドを接続されているチャンネルだけが表示されます。たとえば、show channels connected コマンドを実行すると、次のように表示されます。

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

[full] パラメーターは、指定されたチャンネルの完全な定義を表示します。たとえば、show channels full connected chl=0:3 コマンドを実行すると、次のように表示されます。


```

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       : On           /* 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

```

チャネルの現行のトレース・マスクを表示する

SHOW MASK [chl=*range*]

このコマンドは、チャネルの現行のトレース・マスクを表示します。強調表示された行は、btrace マスク・ビットが使用可能であることを示します。たとえば、コマンド **show mask chl=1** を実行すると、次のように表示されます。

```

Trace Mask for Channel 1
Bit 00 - (fent) function entry      Function entry messages
Bit 01 - (fout) function exit       Function exit messages
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
Bit 06 - (spl) spinlocks             OVMS spinlock messages
Bit 07 - (evt) events                OVMS event messages
Bit 08 - (mtx) mutexes               OVMS mutex messages
Bit 09 - (prc) process msgs          OVMS thread messages
Bit 10 - (msc) miscellaneous         OVMS kernel miscellaneous messages
Bit 11 - (inf) informational         Internal data messages as requested by show command
Bit 12 -                             Reserved for user defined messages

```

この出力は、このスレッドについて、関数入り口、関数出口、スピンロック、およびイベント・メッセージがトレースされることを示します。他のすべてのタイプのメッセージはブロックされます。

ターゲット・スレッド・スタックの内容を表示する

SHOW STACK [chl=*range*]

このコマンドはターゲット・スレッド・スタックの内容を表示します。たとえば、コマンド **show stack chl=0:1** を実行すると、次のように表示されます。

SHOW PROCESSES

```
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
```

アクティブな MQSeries 関連プロセスおよびメモリー使用法の表示

SHOW PROCESSES

このコマンドは、現行ノード上のすべての MQSeries 関連プロセス、さらにそのメモリーの使用状況を表示します。たとえば、コマンド **show process** を実行すると、次のように表示されます。

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	AMQHASMx	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

チャンネルに保留されるすべてのメッセージの表示

SHOW HISTORY [chl=range]

このコマンドは、チャンネル循環ヒストリー・バッファに保留されるすべてのメッセージを表示します。各メッセージはフォーマットされ、出力は生成時のスタックの深さにしたがって字下げされます。たとえば、コマンド **show history chl=3** を実行すると、次のように表示されます。

```

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

```

このサンプル出力は、ヒストリー・バッファ内の行番号、メッセージ生成時刻、チャンネル番号、メッセージ生成時のスタックの深さ、および関数の名前を示します。LU のオープン時、ヒストリー・メッセージ入りの最大数が定義されています。このバッファがいっぱいである場合、MONMQ は最初の入り口まで折り返して戻り、最初のメッセージおよび続くメッセージを上書きします。トレース中に FFST が生成される場合、障害が起こった時点で、失敗したスレッドについてトレースが使用不可になります。これは、バッファの埋め込みから、エラー・ルーチンによってトレース・メッセージが生成されるのを防ぐためです。したがって、ヒストリー・バッファで表示される最後のメッセージは、FFST が生成された時点になります。

現行ノード上のすべての MQSeries 関連グローバル・セクションを表示する

SHOW GLOBALS

このコマンドは、現行ノード上のすべての MQSeries 関連グローバル・セクションを表示します。

SHOW MUTEX

```
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
```

ターゲット・スレッドに対し、クライアント・トレース・プロセスに mutex テーブルを送信するようシグナルを送る

SHOW MUTEX [chl=*range*]

このコマンドは、ターゲット・スレッドに対して、内部 mutex テーブルの内容を、クライアント・トレース・プロセスに送信するようにシグナルを出します。このタイプの情報データをクライアントが表示するためには、正しいトレース・マスク・ビットの設定が重要であることに注意してください。ビット INF および DGN は、このチャンネル用にトレース・マスクで使用可能にする必要があります。(451ページの『マスク・ビットを使用可能または使用不可にする』を参照してください。)たとえば、コマンド **show mutex chl=3** を実行すると、次のように表示されます。

```

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

```

データは、ヒストリー・ファイル中の行番号、mutex 名、およびシステム・ロック ID を示します。

ターゲット・スレッドに対し、クライアント・トレース・プロセスに内部イベント・テーブルを送信するようシグナルを送る

SHOW EVENTS [chl=range]

このコマンドは、ターゲット・スレッドに対して、内部イベント・テーブルの内容を、クライアント・トレース・プロセスに送信するようシグナルを出します。このタイプの情報データをクライアントが表示するためには、正しいトレース・マスク・ビットの設定が重要であることに注意してください。ビット INF および DGN は、このチャンネル用にトレース・マスクで使用可能にする必要があります。(451ページの『マスク・ビットを使用可能または使用不可にする』を参照してください。) たとえば、コマンド **show events chl=3** を実行すると、次のように表示されます。

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: BKM3/@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 ***
```

データは、ヒストリー・ファイル中の行番号、メールボックス・チャンネル番号、およびイベント名を示します。

ターゲット・スレッドに対し、クライアント・トレース・プロセスに内部マップ共用メモリー・テーブルを送信するようシグナルを送る

SHOW MEMORY [chl=range]

このコマンドは、ターゲット・スレッドに対して、内部マップ共用メモリー・テーブルの内容を、クライアント・トレース・プロセスに送信するようにシグナルを出します。このタイプの情報データをクライアントが受信するためには、正しいトレース・マスク・ビットの設定が重要であることに注意してください。ビット INF および DGN は、このチャンネル用にトレース・マスクで使用可能にする必要があります。(451ページの『マスク・ビットを使用可能または使用不可にする』を参照してください。) たとえば、コマンド **show memory chl=2** を実行すると、次のように表示されます。

```

*** 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 ***

```

データは、ヒストリー・ファイル中の行番号、共用メモリー ID、仮想マップ・アドレス範囲、セクションの作成 / マッピングで使用されるフラグ、およびセクションに指定される内部 MQSeries 名を示します。

名前および 16 進 ID によるアクティブ MQSeries コンポーネントの表示

SHOW COMPONENTS

このコマンドは、名前および関連する 16 進 ID によって、すべてのアクティブな MQSeries コンポーネントを表示します。これらの 16 進 ID は、show functions や select component などの、他の MONMQ show コマンドで使用します。たとえば、コマンド **show components** を実行すると、次のように表示されます。

SHOW FUNCTIONS

```
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
```

指定されたコンポーネント内の関数を表示する

```
SHOW FUNCTIONS [comp=hex]
```

このコマンドは、指定されたコンポーネント内のすべての関数を表示します。コンポーネントは、16 進数で入力されなければなりません。SHOW COMPONENT を使って、すべてのアクティブな MQSeries コンポーネントを表示します。たとえば、コマンド **show functions component=0x1f** を実行すると、次のように表示されます。


```

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
.....

```

プロセスの開始点からトレースをアクティブにする

ONSTARTUP [start] [lu=number] [chl=range]

このコマンドによって、プロセスの開始点からトレースをアクティブにすることができます。実行時、論理名 `MQS_DEF_TRACE` がシステム論理名テーブルで定義され、`lu channel` という形式の等価の名前を持ちます。任意の MQSeries プロセスの開始時、この論理名はプロセス初期化ルーチンの内部で検査され、存在していれば、指定された LU とチャンネル番号に接続します。チャンネル ID がすでに割り振られている場合、次に使用可能なチャンネルが使用されます。このコマンドは、MQSeries プロセス / スレッド作成の早期段階中に、トレースが必要である場合に役立ちます。

MQSeries プロセスの始動直後のトレースを回避する

ONSTARTUP [stop]

このコマンドは、システム論理名テーブルから `MQS_DEF_TRACE` 論理の割り当てを解除するので、MQSeries プロセスが始動直後にトレースを行わないようにします。

ターゲット・スレッドを指定されたチャンネルに接続する

```
CONNECT pid number [tid=number] [chl=range]
```

このコマンドは、ターゲット・スレッドに、指定されたチャンネルに接続するようシグナルを出します。チャンネルが指定されないと、最初に使用可能なチャンネルが使用されます。

ターゲット・スレッドを指定されたチャンネルから切断する

```
DISCONNECT [chl=range]
```

このコマンドは、ターゲット・スレッドを、指定されたチャンネルから切断するようシグナルを出します。

LU トレース・メールボックスに書き込まれたリアルタイム・トレース・メッセージを表示する

```
TRACE START [node=string]
```

このコマンドは、クライアント・トレース・プロセスを立ち上げて、LU トレース・メールボックスに書き込まれたリアルタイム・トレース・メッセージを表示します。オプション・ノード・パラメーターは、指定されたノードにウィンドウを作成し、出力をそのウィンドウに誘導します。

現行のクライアント・プロセスの切り離しと終了

```
TRACE STOP
```

このコマンドによって、現行のクライアント・プロセスがトレース・メールボックスおよび終了から切り離されます。このメールボックスに現在書き込んであるすべてのスレッドは、メッセージの書き込みができなくなります。

```
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 ***
```

上記の出力は、トレース・クライアント・ウィンドウに表示されます。

トレース・データの指定

```
SELECT [component] AND/OR [function] OR [fname]
```

このコマンドによって、トレースされるコンポーネント / 関数の組み合わせを 8 つまで指定できます。他のすべてのトレース・データはフィルターに掛けられて除外されます。関数名か、コンポーネントか、またはコンポーネント / 関数のいずれかが指定できます。選択されたコンポーネント / 関数が有効であれば、フィルター・テーブルに書き込まれます。エントリが存在しない場合、すべての関数コンポーネント / 関数がデフォルトとしてトレースされます。

パラメーターを指定しないで **SELECT** コマンドを入力すると、フィルター・テーブルの内容が表示されます。出力の各行に対して、テーブル索引入力、および 16 進数のコンポーネントと関数、および関数のテキスト名が表示されます。コンポーネントだけが入力される場合、このコンポーネント内のすべての関数がトレースされます。これは、関数値に対して 0xffff として表示されます。

たとえば、独自のコマンド **SELECT** を実行すると、次のように表示されま

```
Ch1:0 - Cmp/fnc selection criteria
ALL component/functions
```

以下のコマンドのセットの場合、

```
MQT>select fname="kill"
MQT>select comp=0x1f
MQT>select comp=0x20 func=0x3
MQT>select
```

次のように表示されます。

```
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
```

トレース・フィルター・テーブルから単一記入項目を除去する

```
DESELECT INDEX=<0:7>
```

DESELECT INDEX

このコマンドは、テーブル索引パラメーターに指定されたとおり、トレース・フィルター・テーブルから単一記入項目を除去します。すべてのコンポーネント / 関数は、8 つの項目すべてが空である場合にトレースされます。たとえば、**select** コマンドはその索引と共に、次の項目を表示します。

```
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
```

次の **deselect** コマンドは、指定されたプロセスまたは関数を除去します。

```
MQT> desel index=0
MQT> desel 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
-----
```

クライアント・プロセスがトレース・メッセージをバイナリー・ファイルに書き込む

OPEN BINARY [filename=*string*]

このコマンドはトレース・メッセージ・バイナリー・ファイルをオープンし、クライアント・プロセスがリアルタイム・トレース・メッセージをこのファイルに書き込むようにします。このファイルは、後で MQSeries アプリケーションのパフォーマンスの分析に使用できます。デフォルトのファイル名は、`mqs_root:[mqm.errors]mqs_buffer_xx.bin` (`xx` は LU 番号) です。

バイナリー・トレース・メッセージ・ファイルのクローズ

CLOSE BINARY

このコマンドは、指定された LU バイナリー・トレース・ファイルをクローズします。

クライアント・プロセスがトレース・メッセージをテキスト・ファイルに書き込む

OPEN TEXT [filename=*string*]

このコマンドは読み取り可能なテキスト・ファイルをオープンし、クライアント・プロセスがフォーマット済みバイナリー・トレース・メッセージをこのファイルに書き込むようにします。このファイルは、単に DCL タイプ・コマンドを使用することによって後で表示したり、編集したりすることができます。デフォルトのファイル名は、`mqs_root:[mqm.errors]mqs_buffer_xx.lis` (`xx` は LU 番号) です。このタイプの出力ファイルを使用することの利点は、これを読むためにプリプロセスが必要ないということです。しかし、バイナリー・ファイルよりもディスク・スペースを消費するという欠点もあります。

テキスト・トレース・メッセージ・ファイルのクローズ

CLOSE TEXT

このコマンドは、指定された LU テキスト・トレース・ファイルをクローズします。

タイム・スタンプ・メッセージ

ENABLE TIMESTAMP [`chl=range`]

このコマンドは、チャンネル定義テーブルで `Timestamp` フラグを設定します。このコマンドを使って、MQSeries プロセスが、各メッセージに現在時間をスタンプするように強制します。このフラグは、バイナリー・トレース・ファイルをパフォーマンス分析に使用する場合に設定する必要があります。

メッセージのタイム・スタンプの停止

DISABLE TIMESTAMP [`chl=range`]

このコマンドは、チャンネル定義テーブルで `Timestamp` フラグを設定解除します。(『タイム・スタンプ・メッセージ』を参照してください。)

トレースを使用可能にする

ENABLE TRACE [`chl=range`]

このコマンドは、チャンネル定義テーブルで `RTime` トレース・フラグを設定します。このフラグを使用して、トレース・メッセージのトレース・クライアントへの送信を、使用可能にしたり使用不可にしたりします。たとえばスレッドが接続されており、トレース・クライアントが存在する場合、このスレッドを切断するのではなく、TRACE START を使ってチャンネルを切り替えることができます。

DISABLE TRACE

トレースを使用不可にする

DISABLE TRACE [chl=*range*]

このコマンドは、チャンネル定義テーブルで RTime トレース・フラグを設定解除します。(449ページの『トレースを使用可能にする』を参照してください。)

メッセージ・ヒストリーの保管

ENABLE HISTORY [chl=*range*]

このコマンドは、チャンネル定義テーブルで History フラグを指定されたチャンネルに設定します。このコマンドは、接続されたスレッドに、トレース・メッセージを LU 循環バッファに書き込むようシグナルを出します。メッセージの書き込みがトレースされるプロセスによって実行される場合、クライアント・プロセスが存在する必要はありません。トレース循環バッファのサイズは、**open** コマンドによる LU 作成中に定義されます。バッファは、最後の項目が書き込まれると最初にラップして戻ります。LU 定義テーブルの [Next Log] レコード・フィールドは、バッファ中の次のレコードが書き込まれる位置を指定します。

メッセージ・ヒストリーを使用不可にする

DISABLE HISTORY [chl=*range*]

このコマンドは、チャンネル定義テーブルで History フラグを指定されたチャンネルに設定解除します。『メッセージ・ヒストリーの保管』を参照してください。

メッセージ・ヒストリーの削除

DELETE HISTORY [chl=*range*]

このコマンドは、循環ヒストリー・バッファ中のすべてのメッセージを削除します。このコマンドは、バッファに書き込んでいるプロセスがある場合でも実行できるので、削除前にヒストリーを使用不可にする必要はありません。

ヒストリーの深さの設定

SET [depth]

このコマンドは、トレース・クライアント・ウィンドウへの出力の、スタックの深さの最大値を制御します。この値より深いメッセージは出力されませんが、バイナリー・トレース・ファイルおよびヒストリー・バッファーが使用可能ならそこに表示されます。ゼロのデフォルト値だと、すべてのメッセージがスタックの深さに関係なく出力されます。分析データをバイナリー・ファイルに書き込んでいる場合、この値を非常に低く（たとえば 1）に設定することをユーザーにお勧めします。全スタック表示は、クライアント・プロセスのパフォーマンスに逆効果です。

チャンネルのスタックおよびヒストリー・データをリセットする

```
SET [free] [chl=range]
```

このコマンドは、指定されたチャンネルをリセットします。すべての既存のスタックおよびヒストリー・データは削除され、チャンネルは割り振りを解除されて再利用可能になります。

マスク・ビットを使用可能または使用不可にする

```
SET [mask=var] [chl=range]
```

このコマンドは、接続されたスレッド・ビット・マスク・フィールド内で、マスク・ビットを使用可能または使用不可にします。各ビットは、MQSeries プロセスが生成するメッセージ・タイプを表します。このコマンドを使って、トレースする必要のあるメッセージのタイプをフィルターに掛けることができます。メッセージ・タイプは次のとおりです。

```
MQT>set mask = 0xffffffff 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 -
```

これらのメッセージ・タイプの組み合わせを指定するには、各マスク・タイプを OR 記号で区切ります。次のとおりです。

```
MQT>set mask =0x0 ch1=1 MQT>show mask ch1=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 ch1=1
MQT>show mask ch1=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 -

```

各マスクは、特定のメッセージ・タイプを使用可能または使用不可に切り替えることができる、8つのマスク・タイプから成っています。たとえば、関数入り口点のみに関心がある場合、コマンド **set mask = fent** を入力します。

チャンネルにカラーを設定する

```
SET COLOR [chl=range]
```

このコマンドは、カラーを指定されたチャンネルと関連付けます。このチャンネルに関連するすべての出力は、カラーが変更されるか、チャンネルがリセットされるまで、このカラーで表示されます。このコマンドは、単一の出力ストリーム内で、異なるスレッドのメッセージ間で強調表示したり、認識したりするのに役立ちます。たとえば、以下のコマンドを実行すると、

```

MQT> set color=yellow chl=2
MQT> set color=blue chl=0
MQT> sho chan chl=0:3 connected

```

次のように表示されます。

Chl	Pid/Tid	Mailbox	Stack	History	RTime	Time	Mask	Process Name
0	00000245/1	800b0330	4	0	0	0	ffffffff	AMQZLAA0.EXE
1	00000244/1	800b0200	8	0	0	0	ffffffff	RUNMQCHI.EXE
2	00000243/1	800b01e0	10	0	0	0	ffffffff	AMQRRMFA.EXE
3	00000242/1	800b01c0	4	0	0	0	ffffffff	AMQZLLP0.EXE

ここで、チャンネル 0 は青、チャンネル 2 は黄色です。

出力をファイルに宛先変更する

```
SET OUTPUT [filename=string]
```

SET OUTPUT

このコマンドは、すべての出力を指定されたファイルに宛先変更し、表示する出力を使用付加にします。 **output** コマンドは、他のコマンドを持つパラメーターとして使用される場合、そのコマンドのみに有効です。エラーは、ファイルではなく、引き続きディスプレイ装置に報告されることに注意してください。有効なトレース・データのみが指定されたファイルに書き込まれます。

トレース・バイナリー・ファイルの分析

ANALYSE [component] [function] [unit=xx]

このコマンドは、以前トレース・セッションで使用されたトレース・バイナリー・ファイルの内容を分析します。コンポーネントまたは関数を分析するように指定できますが、これは、ファイルがこのコンポーネントまたは関数に関連するデータを含む場合のみ有効です。たとえば、ファイルの生成時にトレース・マスクを指定するか、または特定のコンポーネントを選択した場合、バイナリー・ファイルにはこれら選択された項目のみが見つかり、したがってこの基準外のコンポーネントまたは関数は、分析では使用できません。

注: 全コマンドを使用する予定である場合、 **ANALYSE** のスペルに *S* を付けてください。この **MONMQ** コマンドを、**OpenVMS** コマンド **ANALYZE** と混同しないでください。2つのコマンドは互いに異なっています。

unit パラメーターは、分析用の時間の単位を指定するのに使用され、値 (xx) として、秒、ミリ秒、マイクロ秒、ナノ秒のうちどれかを指定することができます。デフォルトはミリ秒です。

たとえば、出力をマイクロ秒で表示するには、コマンド **analyse unit=micro:** を使用します。次に、このコマンドの出力例を示します。

```

=====
                        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_mapgb1
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
=====

```

各欄の意味は次のとおりです。

Calls トレース・セッション中の関数への入り口の数。
Minimum 関数内でかかる最速の時間。

ANALYSE

Average	呼び出しの回数で割った関数へのすべての呼び出し内にかかる合計時間。
Maximum	関数内でかかる最長の時間。
Total	すべての呼び出しに関してこの関数にかかる合計時間。
Function	関数名。

注: 分析の有効範囲は、バイナリー・トレース・ファイルの内容です。トレース・バイナリー・ファイルをオープンし、望みどおりの時間にトレースを使用可能、または使用不可にすることによって分析の境界を定義することは、ユーザーの責任です。

MQSeries スレッドの現行状態を表示する

FFST [chl=*range*]

このコマンドは、ターゲット・チャンネルに接続されるスレッドを強制して、FFST を強制します。このコマンドは、実行のターゲット・スレッド・パスには影響しません。このコマンドを使用すると、どの MQSeries スレッドの現行状態でもスナップショットを取ることができます。FFST カットには、スレッドのリソース使用法、特権、およびその他の役立つシステム情報が入っています。FFST は、MONMQ によって作成されたものとしてヘッダーで明確にマークされており (以下を参照)、これは失敗の結果ではありません。

次に出力例の一部を示します。

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.....

トレースのクローズと MONMQ の終了

EXIT

このコマンドは、CLOSE コマンドを実行して MONMQ を終了します。

トレースをクローズしないで MONMQ を終了する

QUIT

このコマンドは、CLOSE コマンドを実行しませんが MONMQ を終了します。このコマンドは、トレースを実行したままにして、MONMQ をシャットダウンしたい場合に役立ちます。次回 MONMQ がアクティブにされると、前のトレース・セッションが再開します。

共用メモリーを MONMQ で管理する

異例の状況、たとえばキュー・マネージャー障害または OpenVMS stop /id コマンドでの強制シャットダウンの場合、MQSeries 共用メモリー・セグメントが、キュー・マネージャーによって自動的に削除されないことがあります。この事態が発生した場合、**strmqm** がキュー・マネージャーがすでに実行中であることを報告するので、キュー・マネージャーを再開することはできません。

MONMQ は、現在存在する MQ 共用メモリー (グローバル・セクション) をリストし、これらの共用メモリー・セクションを削除できます。

注: MONMQ ユーティリティを使用して共用メモリー・セグメントを削除する前に、必ずすべてのキュー・マネージャーをシャットダウンしてください。実行中のキュー・マネージャーの共用メモリーを削除すると、実行キュー・マネージャーは失敗し、キュー・ファイルを破壊することがあります。

MONMQ SHOW PROCESS コマンドを使用して、実行中の MQ プロセスがないことを確認できます。endmqm コマンドで停止できない失敗したキュー・マネージャーで実行するプロセスがある場合、OpenVMS DCL コマンド stop /id=<pid> を使用できます。

次のコマンドを使用して、実行中の MQSeries プロセスがないことを検査します。

```
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_ffffff (00000000) WRT      DZRO TMP SYS Pgltcnt/Refcnt=144/63
MQS1_shm_00000000 (00000000) WRT      DZRO TMP SYS Pgltcnt/Refcnt=6304/394
```

次のコマンドを使用して、現在存在する共用メモリー・グローバル・セクションをリストします。

```
MQT> show globals
```

次のグローバル・セクションを削除します。

```
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
```

エラー削除セクション MQS_shm_00000000 が予期されます。このセクションが MONMQ によって使用されるためです。これで、次のコマンドを発行して MONMQ を終了できます。

```
MQT> exit
```

すべてのキュー・マネージャーが停止していることが確かであれば、スクリプト内から削除コマンドを使うことができます。

```
$ 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
```

MONMQ 中のスクリプトとマクロ

MONMQ コマンドのスクリプトは、MONMQ 内、またはコマンド・プロンプトのどちらからも実行できます。スクリプトは、データのセットを収集するか、または MONMQ 環境を構成するのに役立ちます。MONMQ の開始時に、SYS\$MANAGER:MQS_TRACE_STARTUP.MQT からスクリプトが実行され、MONMQ のトレース変数を構成します。

注: スクリプトが現行ディレクトリーにない場合、スクリプトへの絶対パス名は引用符で囲む必要があります。たとえば、次のとおりです。

```
MQT> ! "sys$manager:test.mqt"
```

また、マクロを定義して共通または反復タスクを短縮することもできます。マクロ宣言は、3 つの部分から構成されます。

1. 最初の部分はマクロ名で、固有のコマンド名であるべきです。

スクリプトとマクロ

2. 2 番目の部分はマクロ本体で、複数行になることができ、MQSeries コマンドのリストから構成されます。マクロ本体は、{ および } で区切られます。
複数行のマクロ本体が宣言されている場合、MONMQ プロンプトが ****MACRO>** に変わります。任意の \$n (n は 1 桁の数値) が、マクロ・コマンド行のパラメーター n に置き換わります。
3. マクロ定義の 3 番目の部分は、ヘルプ <macroname> の使用時に表示される、短いヘルプ・テキスト記述です。ヘルプ・テキストは引用符で囲む必要があります。

マクロの宣言時には、タイミングの問題を考慮する必要があります。マクロは非常にすばやく処理を行いますが、MONMQ コマンドによっては、タスクを実行するようにリモート・プロセスにシグナルを出すものがあり、このタスクは次のマクロ・コマンドの開始前に終了していなければなりません。

この理由で、短い遅延が時々必要です。これは、10 分の 1 秒で指定された遅延パラメーターを持つ、**sleep** コマンドを使用して行います。

次のコマンドは、チャンネルを切断するマクロを作成し、トレース・マスクをリセットし、チャンネルを解放するために入力できます。

注: “;” を区切り文字として使用することによって、複数の MONMQ コマンドを 1 行に入れることができます。

```
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
```

トレース・セッションのサンプル

このセクションは、各 MONMQ コマンドを順番に示す典型的なトレース・セッションを説明します。このサンプルは、実行中のキュー・マネージャーの実行コントローラー、および関連するエージェント・メイン・スレッドをトレースしています。

トレースの開始前に、次の条件を満たしている必要があります。

- トレースするキュー・マネージャーを開始する - STRMQM BKM3
- `sys$manager:mqs_trace_startup.mqt` に、事前インストールされたデフォルトから離れた追加コマンドがないことを検査する。
- 論理 `MQS_DEF_TRACE` が定義されていないことを検査する。定義されている場合、`MONMQ` で **ONSTARTUP END** を実行します。

`momq` を開始します。

```
>momq
```

`MONMQ` プロンプトが表示されます。

`MQT>`

ID ゼロ、10 個のチャネル、および 100 のメッセージのヒストリー・バッファを持つ、1 つの LU をオープンします。(注: 100 のメッセージは、通常のトレース目的では小さすぎます。通常 1000 が適当です。)

```
MQT> open lu=0 nochls=10 buffer=100
ok - LU:0 opened
```

LU1 定義を表示します。

```
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
=====
```

トレース・セッションのサンプル

MQSeries プロセスの表示

```
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	AMQHASM	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

実行コントローラーとエージェント・プロセスを識別し、それらをそれぞれチャンネル 1 と 2 に接続します。

```
MQT>connect pid=0x238 tid=1 lu=1 chl=1  
MQT>connect pid=0x23D tid=1 lu=1 chl=2
```

接続の詳細を検査します。

```
MQT>show channel full connected lu=1
```

```

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
=====

```

chl、lu および tid にデフォルトを設定し、後続のコマンドへの入力のたびに保管します。

```
MQT>default chl=1:2 lu=1 tid=1
```

チャンネル・カラーを、異なるトレース・メッセージ間で見分けられるように設定します。デフォルト (1:2) が使用される場合、両方のチャンネルが黄色、次いでシアンに設定されるので、これら 2 つのコマンドで chl パラメーターが指定されます。

```
MQT>set chl=1 color=yellow
MQT>set chl=2 color=cyan
```

トレース・セッションのサンプル

次にチャンネルを表示します。

```
MQT>show channels
```

Ch1	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

両方のプロセスがチャンネルに接続されたので、スタックを調べることができません。

```
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 -->| z1aMain
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
```

これらの 2 つのチャンネルへのタイム・スタンプを使用可能にすることによって、プロセスがハングしたかどうかを検査できます。

```
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 -->| z1aMain
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
```

これで、有効なタイム・スタンプを持ついくつかのメッセージが表示できます。これは、両方のプロセスがアクティブであることを示します。この場合、両方のプロセスが、10 秒のタイムアウト期間を持つイベント・ループにありま

す。このタイムアウトは、タイム・スタンプ・データに変更が生じるまで、**show stack** コマンドを連続して実行することにより、メッセージ・タイム・スタンプに対して検査できます。

ヒストリーを使用可能にすることによって、各プロセスが、トレース・メッセージを循環バッファに書き込むように強制できます。

```
MQT>enable history
MQT> show history
```

この時点で、すべてのトレース・メッセージをバッファに書き込んでいます。トレース・マスクとコンポーネント / 関数テーブルを表示することによってこの検査ができます。

```
MQT>show 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 -
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 -
```

トレース・セッションのサンプル

```
MQT> select
```

```
Ch1:1 - Cmp/fnc selection criteria  
ALL component/functions
```

```
-----  
Ch1:2 - Cmp/fnc selection criteria  
ALL component/functions
```

```
-----  
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 -
```

ここで、特定のタイプのメッセージに注目してみます。たとえば、共用メモリの診断メッセージだけに興味があるとします。

```
MQT>set mask=shm  
MQT>show mask
```

両方のプロセスが、診断メモリー・タイプ・メッセージだけをバッファーに書き込むようにします。バッファーを削除し、2、3 秒待機し、バッファーの内容を再度調べます。

```
MQT>clear history
(wait a few seconds)
MQT> show history
```

```
*** 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_fffffff 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_fffffff 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_fffffff 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_fffffff 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_fffffff mapped at 1510000-1511fff
*** End of buffer ***
```

```
*** Trace History Ch1: ***
*** End of buffer ***
```

次に、トレース出力にイベント・タイプ診断メッセージも表示します。マスクの設定後、数秒待機することが必要です。

```
MQT>set mask=evt | shm
(wait a few seconds)
MQT>show history
```

トレース・セッションのサンプル

```
*** Trace History Ch1:1 ***
0977 - 00:00:00.00 00 - 04 ..... | vms_mapgb1 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_mapgb1 key : ffffffff - addr : 0/0
0981 - 00:00:00.00 00 - 04 ..... | vms_mapgb1 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_mapgb1 key : ffffffff - addr : 0/0
0992 - 00:00:00.00 00 - 04 ..... | vms_mapgb1 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
```

次に、特定の関数のトレースに注目します。**SHOW COMPONENT**、および**SHOW FUNCTION** を使用して、トレースしたい特定のエリアを識別できます。この例では、共通サービス関数 `xcsRequestMutexSem` をトレースします。コンポーネントは 0x17 で、関数コードは 0x1b です。これは、次の 2 つのうちいずれかの方法で設定できます。


```
MQT>select comp=0x17 func=0x1b
```

または、

```
MQT>select fname="xcsRequestMutexSem"
```

ここでヒストリーを使用可能にする場合、バッファーに出力が表示されないことが分かります。これは、トレース・マスク・ビットをすべてにリセットする必要があるためです。

```
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
```

トレース・セッションのサンプル

```
*** Trace History Ch1:1 ***
0973 - 00:00:00.00 01 - 02 ---->| xcsRequestMutexSem
0974 - 00:00:00.00 01 - 03 ---->| xllSemReq
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 ----<| xllSemReq
0981 - 00:00:00.00 01 - 02 ----<| xcsRequestMutexSem
0982 - 00:00:00.00 01 - 03 ---->| xcsRequestMutexSem
0983 - 00:00:00.00 01 - 04 ---->| xllSemReq
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 ----<| xllSemReq
0990 - 00:00:00.00 01 - 03 ----<| xcsRequestMutexSem
0991 - 00:00:00.00 01 - 03 ---->| xcsRequestMutexSem
0992 - 00:00:00.00 01 - 04 ---->| xllSemReq
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 ----<| xllSemReq
0999 - 00:00:00.00 01 - 03 ----<| xcsRequestMutexSem
*** End of buffer ***
```

これで、指定された関数と子関数だけが両方のプロセス用にトレースされます。最高 8 つのコンポーネントおよび関数が、**select** コマンドを使って同時にトレースできます。リアルタイムで (つまり、発生と同時に) トレースを使用可能にするには、クライアント・プロセスを作成して、特定の LU へのメッセージを表示する必要があります。これは、**TRACE** コマンドを実行することによって行います。

これによって指定されたノードでクライアント・プロセスが立ち上げられ、着信するトレース・メッセージを待機します。それでもクライアント・ウィンドウ上のトレース・セッションは、**MONMQ** を使用して制御できます。

```
MQT>trace start node="mihell"
```

次に、クライアント・プロセスを使用可能にし、着信するメッセージをそのとおりに表示します。

```
MQT>enable trace
```

MQSeries スレッドは、望みどおりにトレース出力から追加したり除去したりできます。スレッドは接続されたままですが、そのトレース・データは使用不可にして、トレースがパフォーマンスに悪影響を及ぼさないようにできます。

トレースは、プロセスまたはスレッドが開始するときに開始できます。

ONSTARTUP コマンドはこれを実行するために使用され、その結果すべての新規 MQSeries プロセスが始動からトレースされます。

トレース・セッションをシャットダウンするには、すべてのアクティブ・チャネルを使用付加にし、クライアント・プロセスを終了する必要があります。

close コマンドがこれをすべて行います。

トレースを実行したままにしておきたい場合、MONMQ から **quit** を使用し、後日トレースを再開します。

トレース・マスク・ビットおよびコンポーネント / 関数の選択は非常に異なっていることに注意してください。トレース・マスク・ビットはトレース・メッセージ・タイプの出力を制御します。たとえば、トレース・エントリーおよびトレース出力はメッセージ・タイプです。これらを使用不可にすると、**select** コマンドを使用して設定したものはすべて効果がありません。これは、コンポーネント / 関数の選択が、これら設定されているマスク・ビットに依存しているためです。

付録I. ユーザー出口

MQSeries for Compaq OpenVMS は、チャンネル出口プログラムおよびデータ変換出口プログラムの両方をサポートします。チャンネル出口の詳細については、*MQSeries 相互通信* を参照してください。データ変換出口の詳細は、*MQSeries アプリケーション・プログラミング・ガイド*、および *MQSeries アプリケーション・プログラミング・リファレンス* を参照してください。

この付録では、MQSeries for Compaq OpenVMS での出口プログラムの使用に固有の情報を提供します。

チャンネルおよびワークロード出口

Exit の個別スレッド化バージョンへの要件は、MQSeries for Compaq OpenVMS では適用されません。

MQSeries クラスタ・ワークロード出口

OpenVMS でワークロード出口をリンクするとき、リンカー・オプション・ファイルで次のように指定する必要があります。

```
sys$share:mqm/share  
sys$share:mqutl/share  
SYMBOL_VECTOR=(c1wlFunction=PROCEDURE,MQStart=PROCEDURE)
```

出口イメージを参照するため、システム全体に渡る実行論理名が必要です。たとえば、出口名が SYSSHARE:AMQSWLM.EXE である場合、次の論理名を定義する必要があります。

```
$DEFINE/SYSTEM/EXEC AMQSWLM SYSSHARE:AMQSWLM
```

.EXE ファイル拡張子は、論理名定義で指定するべきではありません。

この論理名をシステム起動中に定義するには、SYSSMANAGER:MQS_SYSTARTUP.COM で定義します。

付録J. トラストッド・アプリケーション

ご使用の環境においてパフォーマンスが重要な考慮事項であり、環境が安定したものである場合、ユーザー・アプリケーション、チャンネル、およびリスナーは「トラストッド」として定義することができます。つまり、高速パス・バイインディングを使用します。(非持続性メッセージの MQPUT および MQGET 呼び出しを処理するのにかかる時間は、OpenVMS システムで、最高 400% 削減できます。)

トラストッド・アプリケーションでは、MQSeries アプリケーションとローカル・キュー・マネージャー・エージェントは、同一のプロセスとなります。アプリケーションはキュー・マネージャー・リソースに直接接続し、事実上キュー・マネージャーの拡張になります。このストレージの上書きからの保護はないので、このオプションは、キュー・マネージャーの整合性を損なう可能性があります。

また、トラストッド・アプリケーションでは、共用メモリーなど、特定のリソースを作成する必要がある場合があります。これらのリソースは、別のキュー・マネージャー・プロセスによってアクセスされる必要があるかもしれず、したがって同一の UIC に所有されなければなりません。キュー・マネージャーは、MQM アカウントの下のすべての実行を処理し、したがってトラストッド・アプリケーションもこのアカウントの下で実行する必要があります。

上記で詳述された問題は、トラストッド・アプリケーションを使用する前に考慮することが必要です。

ユーザー・アプリケーション

MQM アカウントから直接アプリケーションを実行する必要はありません。キュー・マネージャーに正常に接続した後、MQSeries は、自動的にアクティブなスレッドのセキュリティ・プロファイルを変更し、スレッドが MQM アカウントの識別を前提とするようにします。スレッドの本来の識別は、キュー・マネージャーからの切断の呼び出しに続いて再開されます。

トラストッド・アプリケーションがキュー・マネージャーに接続されている間、アプリケーションが効率的に MQM アカウントの下で実行していることに注意することは重要です。キュー・マネージャーへの接続中に、スレッドの識

ユーザー・アプリケーション

別を別の UIC に変更する必要がある場合、次の MQI 呼び出しを行う前に必ず MQM に戻しておくことが必要です。

トラステッド・アプリケーションのセットアップ

トラステッド・アプリケーションを MQSeries for OpenVMS で実行するには、MQCONNX の [Options] フィールドのバインディング・タイプを、MQCNO_FASTPATH_BINDING に指定することが必要です。(標準バインディングには、MQCNO_STANDARD_BINDING オプションを使用してください。) オプションが指定されない (MQCNO_NONE) 場合、デフォルトで STANDARD_BINDING を使用します。

さらに、論理名 MQ_CONNECT_TYPE を使用して、MQCONNX 呼び出しで指定されるバインディング・タイプをオーバーライドすることもできます。論理名が定義される場合、必要なバインディングのタイプを選択するため、値 FASTPATH または STANDARD がなければなりません。しかし、FASTPATH バインディングは、接続オプションが MQCONNX 呼び出しで適切に指定される場合のみ使用されます。この論理名によって、FASTPATH_BINDING で問題が発生した場合、アプリケーションを再構築しなくても、STANDARD_BINDING を使ってアプリケーションを実行できます。

要約すると、トラステッド・アプリケーションを実行するには、

- MQCONNX 呼び出しで MQCNO_FASTPATH_BINDING オプションを指定し、MQ_CONNECT_TYPE 論理名を FASTPATH として定義する。

または、

- MQCONNX 呼び出しで MQCNO_FASTPATH_BINDING オプションを指定し、MQ_CONNECT_TYPE 論理名を未定義のままにする。

トラステッド・アプリケーションの使用に関する詳細は、MQSeries 相互通信を参照してください。

トラステッド・アプリケーションとしてチャンネルとリスナーを実行する

runmqsc start channel コマンドを使用して開始されるチャンネル・プログラムは、MQM アカунツの下で実行します。着呼 TCP (または DECnet 接続) 要求により開始したチャンネル受信側プログラムも、MQM アカунツの下で実行します。

runmqchl および **runmqslr** コマンドは、MQM アカунツの下で実行する切り離されたプロセスを作成します。MQ_CONNECT_TYPE 論理名と、キュー

トラステッド・アプリケーションとしてチャンネルとリスナーを実行する

一・マネージャーの `qm.ini` ファイルのチャンネル・スタンプの `MQIBindType` の組み合わせは、チャンネルまたはリスナーのどちらかがトラステッドとして実行されるかを定義します。

トラステッド・チャンネルまたはリスナーをセットアップするには、

- `qm.ini` ファイルで `MQIBindType=FASTPATH` を指定し、論理名を `FASTPATH` に設定する

または、

- `qm.ini` ファイルで `MQIBindType=FASTPATH` を指定し、論理名を未定義のままにする

高速、非持続性のメッセージ

非持続メッセージ速度 (NPMSPEED) チャンネル属性は、非持続メッセージが送信される速度を指定するのに使用できます。標準か、高速のどちらかを指定できます。デフォルトは高速になっています。これは、チャンネル上の非持続メッセージが、検索用に使用可能になる前の同期点を待機する必要がないということです。このようなメッセージは非常に早く検索用に使用可能になりますが、伝送障害があった場合、またはメッセージの転送中にチャンネルが停止した場合には失われる可能性があります。トラステッド・アプリケーションとしてのチャンネルとリスナーの実行、および高速で非持続性のメッセージについての詳細は、*MQSeries 相互通信* を参照してください。

付録K. 補足情報

この付録は、MQSeries for Compaq OpenVMS のセットアップが必要な補足情報をリストしています。

この付録に含まれる情報は、次回このマニュアルが更新される際に挿入されます。

アプリケーション・プログラミング・ガイド

OpenVMS 上のプログラミングに関する情報は、Message Queue Interface 呼び出しが AST ルーチン内からは行えないことを注釈するように改訂されます。

この理由は、MQSeries が AST ルーチン自体を使用し、これらのルーチンは別の AST ルーチンがアクティブである場合に実行できないことによります。

アプリケーション・トリガー操作

コマンド・ファイル MQTRIGGER.COM が、MQSeries トリガー・モニター (RUNMQTRM) の提供するパラメーターを使用し、たMQTMC2 構造でフィールドを分離するために設計されたコマンド・ファイルの例として提供されます。

コマンド・ファイルでは、最初のパラメーターがイメージ、またはコマンド・ファイルとして予期し、MQTMC2 構造から選択されたフィールドで呼び出します。

MQTRIGGER は、次のフィールドを MQTMC2 構造から、呼び出されたイメージまたはコマンド・ファイルに渡します。

パラメーター	MQTMC2 フィールド
1	QName
2	ProcessName
3	TriggerData
4	ApplType
5	UserData
6	QMgrName

例

1. amqsech イメージを起動する方法は次のとおりです。
トリガー・プロセス定義の [Applicid] フィールドは次のように指定されます。

```
APPLICID('@mqs_examples:mqtrigger $mqbin:amqsech')
```

この例は、MQBIN 論理ディレクトリーが次のように定義されたという前提になっています。

```
SYS$SYSROOT:[SYSHLP.EXAMPLES.MQSERIES.BIN]
```

2. コマンド・ファイルを呼び出すには、dka200:[user]cmd.com:
トリガー・プロセス定義の [Applicid] フィールドは次のように指定されます。

```
APPLICID('@mqs_examples:mqtrigger @dka200:[user]cmd')
```

付録L. 特記事項

本書はアメリカ合衆国で提供されている製品およびサービス用に作成されたものであり、本書に記載の製品、サービス、またはフィーチャーが日本においては提供されていない場合があります。日本で利用可能な製品、サービス、およびフィーチャーについては、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の 知的所有権を侵害することのない、機能的に同等な製品、プログラム、またはサービスを使用することができます。ただし、IBM 製以外の製品と組み合わせた場合、その操作の評価と検証については、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31

AP 事業所

IBM World Trade Asia Corporation

Intellectual Property Law & Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書は定期的に見直され、必要な変更 (たとえば、技術的に不適確な表現や誤植など) は、本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するもので

特記事項

はありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。また、IBM 以外の製品に関するパフォーマンスの正確性、互換性、またはその他の要求は確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あ

らゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

商標

以下は、IBM Corporation の米国およびその他の国における商標です。

AIX	IBM
MQSeries	AS/400
MVS/ESA	NetView
CICS	OS/2
First Failure Support Technology	VSE/ESA
OS/390	BookManager

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

Microsoft、Windows、および Windows ロゴは、Microsoft Corporation の商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。

参照文献

この節では、現在の MQSeries® 全プロダクトに関する資料について説明します。

MQSeries 複数のプラットフォームに共通する資料

これらの資料のほとんど (MQSeries 「ファミリー」と呼ばれる場合もある) は、すべての MQSeries Level 2 プロダクトに適用されます。最新の MQSeries Level 2 プロダクトは次のとおりです。

- MQSeries for AIX V5.2
- MQSeries for AS/400 V5.2
- MQSeries for AT&T GIS UNIX V2.2
- MQSeries for Compaq OpenVMS Alpha, V5.1
- MQSeries (Compaq Tru64 UNIX 版) V5.1
- MQSeries for HP-UX V5.2
- MQSeries for Linux V5.2
- MQSeries (OS/2 Warp 版) V5.1
- MQSeries for OS/390 V5.2
- MQSeries for SINIX and DC/OSx V2.2
- MQSeries for Sun Solaris V5.2
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- MQSeries for Tandem NonStop Kernel V2.2.0.1
- MQSeries for VSE/ESA V2.1.1
- MQSeries for Windows V2.0
- MQSeries for Windows V2.1
- MQSeries for Windows NT and Windows 2000 V5.2

複数のプラットフォームにまたがる MQSeries 資料は次のとおりです。

- *MQSeries Brochure* (G511-1908)
- *An Introduction to Messaging and Queuing* (GC33-0805)
- *MQSeries 相互通信* (SC88-7775)
- *MQSeries キュー・マネージャー・クラスター* (SD88-7165)
- *MQSeries クライアント* (GC88-7495)
- *MQSeries システム管理の手引き* (SC88-7776)
- *MQSeries コマンド・リファレンス* (SC88-7315)
- *MQSeries イベント・モニター* (SC88-8705)
- *MQSeries プログラム式システム管理* (SC88-7562)
- *MQSeries 管理インターフェースのプログラミング・ガイドおよびリファレンス* (SD88-7145)
- *MQSeries メッセージ* (GC88-7777)
- *MQSeries アプリケーション・プログラミング・ガイド* (SC88-7253)
- *MQSeries アプリケーション・プログラミング・リファレンス* (SC88-7354)
- *MQSeries Programming Interfaces Reference Summary* (SX33-6095)
- *MQSeries C++ の使用* (SC88-7778)
- *MQSeries Java の使用* (SC88-7163)
- *MQSeries アプリケーション・メッセージング・インターフェース* (SC88-8704)

MQSeries プラットフォーム固有の資料

各 MQSeries プロダクトには、MQSeries ファミリーの資料以外に、プラットフォーム固有の資料が少なくとも 1 冊あります。

MQSeries for AIX V5.2

MQSeries for AIX V5.0 インストール・ガイド (GC88-7770)

MQSeries for AS/400 V5.2

MQSeries for AS/400 インストール・ガイド V5R2 (GD88-7309)

MQSeries (AS/400 版) システム管理の手引き V5R1 (SD88-7310)

MQSeries (AS/400 版) V5.1 アプリケーション・プログラミング・リファレンス (RPG) (SC88-7354)

MQSeries for AT&T GIS UNIX V2.2

MQSeries (AT&T GIS UNIX 版) システム管理の手引き (SC88-7646)

MQSeries for Compaq OpenVMS Alpha, V5.1

MQSeries for Compaq OpenVMS Alpha スタートアップ・ガイド (GC88-8943)

MQSeries for Compaq OpenVMS Alpha システム管理ガイド (SC88-8942)

MQSeries (Compaq Tru64 UNIX 版) V5.1

MQSeries (Compaq Tru64 UNIX 版) V5.1 インストールの手引き (GC88-8625)

MQSeries for HP-UX V5.2

MQSeries for HP-UX V5.0 インストール・ガイド (GC88-7772)

MQSeries for Linux V5.2

MQSeries for Linux インストール・ガイド (GD88-7354)

MQSeries (OS/2 Warp 版) V5.1

MQSeries for OS/2 Warp V5.0 インストールの手引き (GC88-7771)

MQSeries for OS/390 V5.2

MQSeries for OS/390 概説および計画ガイド (GC88-8615)

MQSeries for OS/390 システム・セットアップ・ガイド (SC88-8616)

MQSeries for OS/390 システム管理ガイド (SC88-8617)

MQSeries for OS/390 問題判別ガイド (GD88-7153)

MQSeries (OS/390 版) メッセージおよびコード (GD88-7153)

MQSeries for OS/390 Licensed Program Specifications (GC34-5893)

MQSeries for OS/390 Program Directory

MQSeries link for R/3, Version 1.2

MQSeries link for R/3 V1.2 使用者の手引き (GD88-7041)

MQSeries for SINIX and DC/OSx V2.2

MQSeries for SINIX and DC/OSx System Management Guide (GC33-1768)

MQSeries for Sun Solaris V5.2

MQSeries for Sun Solaris V5.0 インストール・ガイド (GC88-7773)

MQSeries for Sun Solaris, Intel Platform Edition, V5.1

MQSeries for Sun Solaris, Intel Platform Edition インストール・ガイド (GC88-8749)

MQSeries for Tandem NonStop Kernel V2.2.0.1

MQSeries for Tandem NonStop Kernel System Management Guide (GC33-1893)

MQSeries for VSE/ESA V2.1.1

MQSeries for VSE/ESA Licensed Program Specifications (GC34-5365)

MQSeries for VSE/ESA System Management Guide (GC34-5364)

MQSeries for Windows V2.0

MQSeries (Windows 拡張版) V2.0 使用者の手引き (GC88-7649)

MQSeries for Windows V2.1

MQSeries (Windows 拡張版) V2.1 使用者の手引き (GC88-7208)

MQSeries for Windows NT and Windows 2000 V5.2

MQSeries for Windows NT V5.0 インストール・ガイド (GD88-7162)

MQSeries (Windows NT 版) コンポーネント・オブジェクト・モデル・インターフェースの使用 (SD88-7161)

MQSeries LotusScript Extension (SD88-7146)

ソフトコピー資料

MQSeries に関する大部分の資料は、ハードコピー形式とソフトコピー形式の両方で提供されています。

HTML 形式

以下の MQSeries プロダクトには、HTML 形式の MQSeries 関連資料が付属しています。

- MQSeries for AIX V5.2
- MQSeries for AS/400 V5.2
- MQSeries for Compaq OpenVMS Alpha, V5.1
- MQSeries (Compaq Tru64 UNIX 版) V5.1
- MQSeries for HP-UX V5.2
- MQSeries for Linux V5.2
- MQSeries (OS/2 Warp 版) V5.1
- MQSeries for OS/390 V5.2
- MQSeries for Sun Solaris V5.2
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- MQSeries for Windows NT and Windows 2000 V5.2 (コンパイルされた HTML)
- MQSeries link for R/3, V1.2

HTML 形式の MQSeries 資料は、次の URL アドレスの MQSeries ソフトウェア・サーバーのホーム・ページにもあります。

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

PDF

Adobe Acrobat Reader を使用することによって、PDF ファイルの表示や印刷を行えます。

Adobe Acrobat Reader を入手したい場合、Acrobat Reader の使用が可能なプラットフォーム

参照文献

ームについての最新情報が必要な場合は、次の Adobe Systems Inc. Web サイトにアクセスしてください。

<http://www.adobe.com/>

以下の MQSeries プロダクトには、PDF バージョンの MQSeries 関連資料が付属しています。

- MQSeries for AIX V5.2
- MQSeries for AS/400 V5.2
- MQSeries for Compaq OpenVMS Alpha, V5.1
- MQSeries (Compaq Tru64 UNIX 版) V5.1
- MQSeries for HP-UX V5.2
- MQSeries for Linux V5.2
- MQSeries (OS/2 Warp 版) V5.1
- MQSeries for OS/390 V5.2
- MQSeries for Sun Solaris V5.2
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- MQSeries for Windows NT and Windows 2000 V5.2
- MQSeries link for R/3, V1.2

現在の MQSeries 全資料の PDF バージョンは、次の MQSeries プロダクト・ファミリー Web サイトからも入手できます。

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

インターネットで利用できる MQSeries 情報

MQSeries プロダクト・ファミリーのホーム・ページの URL は以下のとおりです。

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

この URL からのリンクを利用することにより、次のようなサービスを利用できます。

- MQSeries プロダクト・ファミリーの最新情報へのアクセス

- HTML 形式の MQSeries 資料の入手
- MQSeries SupportPac™ のダウンロード。

参考資料

- *Compaq OpenVMS Performance Management*, January 1999
このマニュアルは、OpenVMS システムでパフォーマンスを最適化するのに役立つ情報を提供します。
- *Compaq OpenVMS System Management Utilities 2 volumes*, January 1999
このマニュアルには、OpenVMS を使用したシステム管理ユーティリティの参照情報が含まれます。
- *Character Data Representation Library, Character Data Representation Architecture, Reference and Registry* (SC09-2190-00)
このマニュアルは、Character Data Representation Architecture (CDRA) を概説し、アーキテクチャーの要素を解説書の形式で定義します。
- *DecNet SNA Gateway for Synchronous Transport Installation (OpenVMS)*, November 1993
このガイドは、DecNet SNA Gateway のインストールおよび構成方法を説明します。
- *Digital SNA APPC/LU6.2 Programming Interface for OpenVMS*, May 1996
このガイドは、SNA APPC/LU6.2 のインストールおよび構成方法を説明します。
- *Digital TCP/IP Services for OpenVMS Installation and Configuration*, January 1999
このガイドは、Digital TCP/IP のインストールおよび構成方法を説明します。

- *Guidelines for OpenVMS Cluster Configurations*, January 1999

このガイドは、OpenVMS クラスターの可用性と拡張容易性を最大にする方法を説明します。

- *Introduction to Compaq Networking and Data Communications* (Compaq Part No. 093148)

このガイドは、Compaq ネットワーキングおよびデータ通信の概念、タスク、製品、およびマニュアルについての概要を提供します。

用語集および略語集

この用語集では、本書で使用する MQSeries の用語および省略語を定義しています。探している用語が見つからない場合は、索引または *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994 を参照してください。

この用語集には、*American National Dictionary for Information Systems*, ANSI X3.172-1990 (copyright 1990 by the American National Standards Institute (ANSI)) から引用した用語および定義が含まれています。この資料は American National Standards Institute, 11 West 42 Street, New York, New York 10036 から購入することができます。この資料からの定義には、定義の最後に (A) という記号を付けて区別してあります。

[ア行]

アーカイブ・ログ (archive log). 「リカバリー・ログ (recovery log)」を参照。

アクティブ・ログ (active log). 「リカバリー・ログ (recovery log)」を参照。

アダプター (adapter). MQSeries for OS/390 と TSO、IMS、CICS、またはバッチ・アドレス・スペース間のインターフェース。アダプターは、アプリケーションを MQSeries サービスにアクセスするための接続機能。

アドレス・スペース (address space). 特定のジョブに使用可能な、仮想記憶域。

アドレス・スペース ID (ASID) (address space identifier (ASID)). アドレス・スペースの、固有、かつシステムが割り当てた ID。

アプリケーション環境 (application environment). アプリケーション・プログラムによりアクセス可能なソフトウェア機能。OS/390 プラットフォームにおけるアプリケーション環境の例としては、CICS および IMS がある。

アプリケーション・キュー (application queue). アプリケーションが使用するキュー。

アプリケーション・ログ (application log). Windows NT では、有効なアプリケーション・イベントを記録するログ。

アラート (alert). ネットワーク内の管理サービス中心拠点へ送られるメッセージで、問題または緊急問題を識別するためのもの。

アラート・モニター (alert monitor). MQSeries for OS/390 では、スケジュールにないイベントを扱う CICS アダプターのコンポーネントで、MQSeries for OS/390 への接続要求の結果として発生する。

異常終了理由コード (abend reason code). MQSeries for OS/390 の問題を固有に識別する 4 バイトの 16 進コード。MQSeries for OS/390 異常終了理由コードの完全なリストとその説明は、*MQSeries (OS/390 版) メッセージおよびコード* にある。

一時動的キュー (temporary dynamic queue). クローズ時に削除される動的キュー。一時動的キューは、キュー・マネージャーで障害が発生した場合に回復されないため、非持続メッセージしか入れることができない。「永続動的キュー (permanent dynamic queue)」と対比。

イベント (event). 「チャンネル・イベント (channel event)」、「観測イベント

(*instrumentation event*)」、「パフォーマンス・イベント (*performance event*)」、および「キュー・マネージャー・イベント (*queue manager event*)」を参照。

イベント・キュー (event queue). キュー・マネージャーが、イベントを検出した後、イベント・メッセージを書き込むキュー。イベントの各カテゴリ (キュー・マネージャー、パフォーマンス、またはチャンネルのイベント) には、それぞれ独自のイベント・キューがある。

イベント・データ (event data). イベント・メッセージでは、イベントについての情報 (キュー・マネージャー名、イベントを引き起こしたアプリケーションなど) を含んでいるメッセージ・データ部分。「イベント・ヘッダー (*event header*)」も参照。

イベント・ビューアー (Event Viewer). ログ・ファイルを検査および管理するために Windows NT によって提供されるツール。

イベント・ヘッダー (event header). イベント・メッセージでは、イベントについての理由コードのイベント・タイプを識別するメッセージ・データ部分。

イベント・メッセージ (event message). MQSeries システムのネットワーク内での観測イベントの起点に関連した情報 (イベントのカテゴリ、イベントを引き起こしたアプリケーションの名前、キュー・マネージャー統計など) を内容とする。

イベント・ログ (event log). 「アプリケーション・ログ (*application log*)」を参照。

インストール可能サービス (installable services). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000において、独立したコンポーネントとして提供される追加機能。各コンポーネントをインストールするかどうかはオプションであり、お客様作成のコンポーネント、

または第三者のコンポーネントを代わりに使用できる。「許可サービス (*authorization service*)」、「名前サービス (*name service*)」、および「ユーザー識別子サービス (*user identifier service*)」も参照。

永続動的キュー (permanent dynamic queue). 削除が明示的に要求された場合に限って、クローズ時に削除される動的なキュー。永続動的キューはキュー・マネージャーが失敗したときに回復されるので、持続メッセージを入れることができる。「一時動的キュー (*temporary dynamic queue*)」と対比。

応答側 (responder). 分散キューイングにおいて、他のシステムからのネットワーク接続要求に応答するプログラム。

応答先キュー (reply-to queue). MQPUT 呼び出しを出したプログラムが応答メッセージまたはレポート・メッセージの送信を望む、キューの名前。

応答メッセージ (reply message). 要求メッセージへの応答に使用されるメッセージのタイプ。「要求メッセージ (*request message*)」および「レポート・メッセージ (*report message*)」と対比。

オブジェクト (object). MQSeries では、オブジェクトとは、キュー・マネージャー、キュー、プロセス定義、チャンネル、名前リスト、または保管クラス (OS/390 のみ) を指す。

オブジェクト記述子 (object descriptor). 特定の MQSeries オブジェクトを識別するデータ構造のこと。オブジェクト記述子には、オブジェクトの名前およびオブジェクト・タイプが含まれる。

オブジェクト権限マネージャー (object authority manager (OAM)). MQSeries (UNIX システム用)、MQSeries for AS/400 および MQSeries for Windows NT and Windows 2000では、コマンドおよびオブジェクトの管理のためのデフォルトの許可サービス。OAM は、顧客提供のセキュリティ・サービスで置き換えたり、顧

客提供のセキュリティー・サービスと組み合わせて実行したりすることができる。

オブジェクト・ハンドル (object handle). プログラムがその稼働対象である MQSeries オブジェクトにアクセスする際に用いる、ID またはトークン。

オフロード (off-loading). MQSeries for OS/390 では、キュー・マネージャーのアクティブ・ログがアーカイブ・ログに転送される際に使用する、自動プロセス。

[力行]

解決パス (resolution path). アプリケーションが、MQOPEN 呼び出しに対する入力において別名またはリモート・キューを指定する時にオープンされる一組のキュー。

開始キュー (initiation queue). キュー・マネージャーがトリガー・メッセージを入れるローカル・キュー。

回復機能委任 (percolation). エラー・リカバリーにおいて、事前に定めた経路に沿って、あるリカバリー・ルーチンから、さらに高いレベルのリカバリー・ルーチンに進むこと。

外部セキュリティー・マネージャー (external security manager (ESM)). OS/390 システム許可機能により起動されたセキュリティー・プロジェクト。RACF は ESM の一例。

環境 (environment). 「アプリケーション環境 (application environment)」を参照。

監視プログラム呼び出し (supervisor call (SVC)). 指示が示す特定のサービスが実行できるように、実行中のプログラムに割り込み、監督に制御を渡す OS/390 の指示。

観測イベント. MQSeries システムのネットワーク内でのキュー・マネージャーの操作を監視するために使用できる機能。MQSeries は、キュー・

マネージャーのリソース定義、パフォーマンス条件、およびチャネル条件をモニターするために、観測イベントを提供する。観測イベントは、イベントをシステム・オペレーターに表示する管理アプリケーション内の、ユーザー作成報告メカニズムで使用することができる。さらに、観測イベントを使用すれば、アプリケーションは、報告書をモニターし、適切なアラートを作成するために、他の管理ネットワークのエージェントとして動作することができる。

管理者コマンド (administrator commands).

MQSeries オブジェクト (たとえば、キュー、プロセス、名前リスト) を管理するために使用する MQSeries のコマンド。

完了コード (completion code). MQI 呼び出しがどのように終了したかを示す戻りコード。

機能回復ルーチン (FRR) (functional recovery routine (FRR)). リカバリー・ルーチンがプログラム割り込みで制御を取得するための、OS/390 リカバリー / 終了マネージャー機能。

基本マッピング・サポート (basic mapping support (BMS)). さまざまな端末に使用される制御文字に関係なく、入出力表示データをフォーマットし、複数ページの出力メッセージを発送する、CICS とアプリケーション・プログラム間のインターフェース。

キュー (queue). MQSeries オブジェクト。メッセージ・キューイング・アプリケーションは、キューにメッセージを置き、またキューからメッセージを得ることができる。キューは、キュー・マネージャーによって所有され、保守される。ローカル・キューには、処理を待っているメッセージのリストを含めることができる。その他のタイプのキューには、メッセージを含めることができない。それらはその他のキューを指す。または動的キューのモデルとして使用できる。

キューイング (queuing). 「メッセージ・キューイング (message queuing)」を参照。

キュー・マネージャー (queue manager). アプリケーションにキュー・サービスを提供するシステム・プログラム。このプログラムは、キュー・マネージャーが所有するキュー上のメッセージにプログラムがアクセスできるようにする、アプリケーション・プログラミング・インターフェースを提供する。「ローカル・キュー・マネージャー (*local queue manager*)」および「リモート・キュー・マネージャー (*remote queue manager*)」も参照。特定のキュー・マネージャーの属性を定義する MQSeries オブジェクト。

キュー・マネージャー・イベント (queue manager event). 次のことを示すイベント。

- キュー・マネージャーが使用するリソースに関連してエラー条件が発生した。たとえば、キューが利用できない。
- キュー・マネージャーで重大な変更が行われた。たとえば、キュー・マネージャーが停止または開始した。

強化メッセージ (hardened message). メッセージがシステム障害のイベントで脱落しないように、補助 (ディスク) ストレージに書き込まれるメッセージ。「持続メッセージ (*persistent message*)」も参照。

強制シャットダウン (forced shutdown). CICS アダプターのシャットダウンの 1 つのタイプで、現在の活動タスクの状態にかかわらず、アダプターがすぐに MQSeries for OS/390 から切断するというもの。「静止シャットダウン (*quiesced shutdown*)」と対比。

許可検査 (authorization checks). たとえば、キューをオープンしたりキュー・マネージャーに接続するためにユーザーがオブジェクトに対して管理コマンドを発行しようとするときに実行されるセキュリティ検査。

許可サービス (authorization service).

MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、MQSeries for Windows NT and Windows 2000 では、コマンドまたは MQI 呼び

出しに関連したユーザー識別子について、コマンドおよび MQI 呼び出しに関する許可検査を行うサービス。

許可ファイル (authorization file). MQSeries (UNIX システム用) において、あるオブジェクト、あるオブジェクト・クラス、またはすべてのオブジェクト・クラスに関するセキュリティ定義を提供するファイル。

クライアント (client). ローカル・ユーザー・アプリケーションに、サーバーのキューイング・サービスへのアクセスを提供する実行時コンポーネント。アプリケーションが使用するキューは、サーバーに置かれている。「MQSeries クライアント (*MQSeries client*)」も参照。

クライアント接続チャンネル・タイプ (client connection channel type). MQI クライアントに関連付けられた MQSeries チャンネル定義のタイプ。「サーバー接続チャンネル・タイプ (*server connection channel type*)」も参照。

クライアント・アプリケーション (client application). ワークステーションで実行されるアプリケーションであって、サーバーのキューイング・サービスにアクセスできるようにクライアントにリンクされているもの。

クラスター (cluster). 何らかの方法で論理的に関連付けられているキュー・マネージャーのネットワーク。

グローバル・トレース (global trace). トレース・データが MQSeries for OS/390 サブシステム全体から来る、MQSeries for OS/390 トレース・オプション。

コード化文字セット識別子 (coded character set identifier (CCSID)). 各文字とそれに割り当てられたコード・ポイントの集合、すなわちコード化セットの名前。

コールバック (call back). MQSeries において、要求側のメッセージ・チャンネルが、まず送信

側を呼び出し、次にクローズしてコールバックを待つことにより、送信側チャンネルからの転送を開始する。

構成ファイル (configuration file). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 において、たとえば、ログ、通信、インストール可能サービスなどに関する構成情報が格納されるファイル。「.ini ファイル (.ini file)」と同義。「スタンザ (stanza)」も参照。

コマンド (command). MQSeries では、キュー・マネージャーが実行できる管理命令。

コマンド処理プログラム (command processor). コマンドを処理する MQSeries のコンポーネント。

コマンド接頭部 (command prefix (CPF)). MQSeries for OS/390 では、MQSeries for OS/390 コマンドの送信先、および MQSeries for OS/390 オペレーター・メッセージの発信元である、キュー・マネージャーを識別する文字ストリング。

コマンド・サーバー (command server). MQSeries のコンポーネントの 1 つで、システム・コマンド入力キューからコマンドを読み取り、そのコマンドを検査して、コマンド・プロセッサに有効なコマンドを渡すもの。

コミット (commit). 現在のリカバリー単位または作業単位中に行われるすべての変更に応用する操作。この操作の完了後、新規のリカバリー単位または作業単位が開始される。「バックアウト (backout)」と対比。

コンテキスト (context). メッセージの起点についての情報。

コンテキスト・セキュリティー (context security). MQSeries において、メッセージ記述子内のメッセージの起点の詳細を、メッセージが伝えることを余儀なくされるような仕方、セキュリティーが処理されることを可能にする方式。

[サ行]

サーバー (server). (1) MQSeries では、リモート・ワークステーションで実行されるクライアント・アプリケーションにキュー・サービスを提供するキュー・マネージャー。(2) クライアント / サーバーという 2 つのプログラム間の独特な情報フロー・モデルにおいて、情報に関する要求に応答するプログラム。「クライアント (client)」も参照。

サーバー接続チャンネル・タイプ (server connection channel type). キュー・マネージャーを実行するサーバーに関連した MQI チャンネル定義のタイプ。「クライアント接続チャンネル・タイプ (client connection channel type)」も参照。

サーバー・チャンネル (server channel). メッセージ・キューイングにおいて、要求側チャンネルに応答し、伝送キューからメッセージを除去したり、通信リンクを介してメッセージを要求側チャンネルに移動したりするチャンネル。

サービス間隔 (service interval). サービス間隔イベントの条件が満たされたかどうかを判別するときに、キュー・マネージャーが書き込みまたは読み取りと、それに続く読み取りとの間の経過時間を比較する対象となる時間間隔。キューのサービス間隔はキュー属性によって指定される。

サービス間隔イベント (service interval event). サービス間隔に関連したイベント。

再同期 (resynch). MQSeries において、メッセージ転送は再始動せずに、チャンネルを開始し、未確定の状況メッセージがあれば解決するように指示するオプション。

先入れ先出し法 (first-in-first-out (FIFO)). キューに最も長い間入っていた項目が、次に検索される項目となるような仕方で行われる、キューイングの技法。(A)

作業単位 (unit of work). 2 つの整合点の間でアプリケーションにより実行される操作の回復可能な一連の操作。作業単位はトランザクションが開始する時、またはユーザーが要求した同期点の後で始まる。また、ユーザー要求の同期点、またはトランザクションの終了時に、作業単位は終わる。「リカバリー単位 (*unit of recovery*)」と対比。

作業論理単位 (logical unit of work (LUW)).
「作業単位 (*unit of work*)」を参照。

サブシステム (subsystem). OS/390 では、OS/390 に依存した関数を提供するモジュールのグループ。たとえば、MQSeries for OS/390 は OS/390 のサブシステムです。

時間非依存メッセージング (time-independent messaging). 「非同期メッセージング (*asynchronous messaging*)」を参照。

シグナル (signaling). MQSeries for OS/390 および MQSeries for Windows 2.1 では、予期されたメッセージがキューに着いたときに、オペレーティング・システムがプログラムに通知するための機能。

システム権限機能 (System Authorization Facility (SAF)). MQSeries for OS/390 が RACF などの外部セキュリティ・マネージャーと通信するときに使用する、OS/390 機能。

システム初期設定テーブル (system initialization table (SIT)). 開始時に CICS が使用するパラメーターを含むテーブル。

システム診断作業域 (system diagnostic work area (SDWA)). SYS1.LOGREC エントリーに記録されるデータで、プログラムまたはハードウェア・エラーを記述する。

システム制御コマンド (system control commands). バッファ・プール、保管クラス、ページ・セットなどの、プラットフォームに特定のエンティティの操作に使用されるコマンド。

システム・コマンド入力キュー (system-command input queue). アプリケーション・プログラムが MQSeries コマンドを入れることができるローカル・キュー。コマンドはコマンド・サーバーによってキューから検索される。コマンド・サーバーはコマンドの妥当性検査を行い、実行のため、それらをコマンド・プロセッサに渡す。

持続メッセージ (persistent message). キュー・マネージャーの再始動の後も残るメッセージ。「非持続メッセージ (*nonpersistent message*)」と対比。

シャットダウン (shutdown). 「即時シャットダウン (*immediate shutdown*)」、「優先シャットダウン (*preemptive shutdown*)」、および「静止シャットダウン (*quiesced shutdown*)」を参照。

終了通知 (termination notification). CICS サブシステムが正常に MQSeries for OS/390 に接続しようとするときに、アクティブにされる保留イベント。

受信側チャネル (receiver channel). メッセージ・キューイングにおいて、送信側チャネルに応答するチャネルで、通信リンクからメッセージを受け取り、それらをローカル・キューに入れる。

出力パラメーター (output parameter). MQI 呼び出しのパラメーターで、呼び出しが完了または失敗した場合にキュー・マネージャーがこのパラメーターに情報を戻す。

出力ログ・バッファ (output log-buffer). MQSeries for OS/390 では、アーカイブ・ログに書きこまれる前にリカバリー・ログ・レコードを保留するバッファ。

循環ロギング (circular logging). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 において、すべての再始動データをログ・ファイルの 1 つのリング内に保持するプロセス。ロギングでは、リングの最初のファイルに書き込みが行われ、それから次のファイルに進むという方法で、すべてのファイルが満杯になるまで書き込みが行われる。すべてのファイルが満杯になった時点で、スペースが解放されている場合やもう必要ない場合は、リング内の最初のファイルに戻って再びログを開始する。循環ロギングは、再始動リカバリー時に使用され、システムの停止時に進行中だったトランザクションのロールバックにログが使用される。「リニア・ロギング (linear logging)」と対比。

順次配布 (sequential delivery). MQSeries では、受信チャンネルがメッセージを保管する際、メッセージの順序を再設定できるように、順序番号を付けてメッセージを伝送する方式。これは、メッセージを一度だけ、正しい順序で送達しなければならない場合に必要ない方式である。

順序番号折り返し値 (sequential number wrap value). MQSeries では、通信リンクの両端が確実に、その現行メッセージの順序番号を同時にリセットするための方式。順序番号を付けてメッセージを伝送すれば、受信チャンネルは、メッセージを保管する際、メッセージの順序を確実に再設定することができる。

症状ストリング (symptom string). IBM ソフトウェア・サポート・データベースの検索用に設計された、構造化形式で表示される診断情報。

初期化入力データ・セット (initialization input data sets). 始動時に MQSeries for OS/390 が使用するデータ・セット。

スイッチ・プロファイル (switch profile). MQSeries for OS/390 では、MQSeries が始動したとき、または最新セキュリティー・コマンドが発行されたときに使用される RACF プロファイ

ル。MQSeries が検出する各スイッチ・プロファイルは、指定されたりソースの検査をオフにする。

据え置き接続 (deferred connection). CICS サブシステムが MQSeries for OS/390 の開始前に MQSeries for OS/390 に接続しようとするときに、アクティブにされる保留イベント。

スタンザ (stanza). 構成ファイルの中の 1 グループの行であって、キュー・マネージャー、クライアント、またはチャンネルの動作を修正するパラメーターに値を割り当てるもの。MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 では、構成 (.ini) ファイルに多くのスタンザを含めることができる。

ストレージ・クラス (storage class). MQSeries for OS/390 では、ストレージ・クラスは、特定のキューへのメッセージを保留するページ・セットを定義する。ストレージ・クラスはキューの定義時に指定される。

スレッド (thread). MQSeries では、1 つのオペレーティング・システム・プラットフォームで利用できる並行実行の最低レベル。

制御インターバル (CI) (control interval (CI)). VSAM がレコードを保管し、分散フリー・スペースを作成する直接アクセス記憶装置の固定長エリア。制御インターバルは、VSAM が直接アクセス記憶装置と送受信する情報の単位。

制御言語 (CL) (Control Language (CL)). MQSeries for AS/400 では、コマンド行で、または CL プログラムを作成することによって、コマンドを発行するのに使用する言語。

制御コマンド (control command). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 において、オペレーティング・システムのコマンド行から対話型で入力できるコマンド。これらのコマンドでは、MQSeries プロダクトがインストールされていることのみが必要であり、コ

マンドを実行するための特別なユーティリティーやプログラムは不要である。

制御されたシャットダウン (controlled shutdown). 「静的シャットダウン (*quiesced shutdown*)」を参照。

静止 (quiescing). MQSeries において、停止される前のキュー・マネージャーの状態。この状態では、プログラムは処理を終了することはできるが、新規プログラムを開始することはできない。

静的シャットダウン (quiesced shutdown). MQSeries において、接続しているすべてのアプリケーションが切断できるようにするキュー・マネージャーのシャットダウン。「即時シャットダウン (*immediate shutdown*)」および「優先シャットダウン (*preemptive shutdown*)」と対比。CICS アダプターのシャットダウンの 1 つのタイプで、現在の活動タスクのすべてが完了してはじめてアダプターが MQSeries から切断するというもの。「強制シャットダウン (*forced shutdown*)」と対比。

セキュリティ有効化インターフェース (security enabling interface (SEI)). 許可の検査、ユーザー ID の提供、または認証を行う顧客作成またはベンダー作成のプログラムが従う必要がある MQSeries インターフェース。MQSeries フレームワークの一部。

セッション ID (session ID). MQSeries for OS/390 では、メッセージを伝送キューからリンクに移動するとき、メッセージ・チャンネル・エージェントが使用する通信リンクを定義する CICS 固有の ID。

接続 (connect). キュー・マネージャーの接続ハンドルを提供することによって、その接続ハンドルはそれ以降の MQI 呼び出しでアプリケーションによって使用される。接続は、MQCONN 呼び出しによって行われる、または MQOPEN 呼び出しによって自動的に行われる。

接続アドレス・スペース (allied address space). 「同盟 (*ally*)」を参照。

接続ハンドル (connection handle). プログラムが、接続先のキュー・マネージャーにアクセスするための ID またはトークン。

送信側チャンネル (sender channel). メッセージ・キューイングにおいて、転送を開始し、伝送キューからメッセージを除去したり、通信リンクを介してメッセージを受信側または要求側チャンネルに移動したりするチャンネル。

相対バイト・アドレス (relative byte address (RBA)). 所属するデータ・セットに割り振られるストレージ・スペースの始めからの、保管レコードまたは制御インターバルのバイトでの変位。

送達不能キュー (dead-letter queue (DLQ)). キュー・マネージャーまたはアプリケーションがそれらの正しい宛先に送達できないようなメッセージの送信先となるキュー。

送達不能キュー・ハンドラー (dead-letter queue handler). MQSeries 提供のユーティリティーであって、ユーザー作成のルール・テーブルに従って送達不能キュー (DLQ) を監視し、キュー上のメッセージを処理する。

即時シャットダウン (immediate shutdown). MQSeries において、アプリケーションが切断するまで待つことのないキュー・マネージャーのシャットダウン。現行の MQI 呼び出しは処理を完了できるが、即時シャットダウンが要求された後の新規の MQI 呼び出しは失敗する。「静止シャットダウン (*quiesced shutdown*)」および「優先シャットダウン (*preemptive shutdown*)」と対比。

属性 (attribute). MQSeries オブジェクトの特性を定義する、一組の特性の 1 つ。

[タ行]

ターゲット・ライブラリー高水準修飾子 (target library high-level qualifier (thlqual)). OS/390 ターゲット・データ・セット名の高水準修飾子。

第 1 障害サポート・テクノロジー (First Failure Support Technology (FFST)). ソフトウェア問題を検出して報告するために、MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、MQSeries for Windows NT and Windows 2000、および MQSeries (AS/400 版) で使用される。

代替ユーザー・セキュリティー (alternate user security). MQSeries オブジェクトをオープンするなどの、1 つのユーザー ID の許可を別のユーザー ID が使用することを可能にするセキュリティー機能。

対話式システム生産性向上機能 (ISPF)(Interactive System Productivity Facility (ISPF)). フルスクリーン・エディターおよびダイアログ・マネージャーとしての役割を果たす IBM ライセンス・プログラム。アプリケーション・プログラムの作成に使用され、標準スクリーン・パネル、およびアプリケーション・プログラマーと端末ユーザー間の対話式ダイアログの生成方法を提供する。

対話式問題制御システム (IPCS)(Interactive Problem Control System (IPCS)). オンライン問題管理、対話式問題診断、ディスク常駐異常終了ダンプ用のオンライン・デバッグ、問題トラッキング、および問題報告を許可する OS/390 のコンポーネント。

タスク切り替え (task switching). 入出力操作の重複と、複数のタスク間の処理。MQSeries for OS/390 では、いくつかの MQI 呼び出しが、メイン CICS TCB ではなく、サブタスクの下で実行されるようにすることで、タスク切り替え機能がパフォーマンスを最適化する。

タスク制御ブロック (task control block (TCB)). MQSeries for OS/390 または CICS などの OS/390 サブシステムに接続される、アドレス・スペース内のタスクについての情報を通信するのに使用される OS/390 制御ブロック。

単一フェーズ・コミット (single-phase commit). 別のリソース・マネージャーによって制御されているリソースに対して行った更新との間での調整を行わなくても、プログラムがキューに対して更新をコミットできる方式。「2 フェーズ・コミット (two-phase commit)」と対比。

単一フェーズ・バックアウト (single-phase backout). まだ進行中のアクションが終了しないようにし、そのアクションの一部をなすすべての変更が全部取り消されるようにする方式。

単一ロギング (single logging). 各変更が 1 つのデータ・セットのみに記録される、MQSeries for OS/390 アクティビティーの記録方法。「重複ロギング (dual logging)」と対比。

ダンプ分析重複回避機能 (DAE)(dump analysis and elimination (DAE)). インストール先システムが、SVC 以前作成されたダンプと重複しているために必要ない SVC ダンプ、および ABEND SYSUDUMP ダンプを抑制できるようにする OS/390 サービス。

チェックポイント (checkpoint). 重要な情報がログに書き込まれる時点。「同期点 (syncpoint)」と対比。MQSeries (UNIX システム用) では、ログに記述されているデータ・レコードがキューのデータ・レコードと同じになる時点。チェックポイントは自動的に生成され、システム再始動処理時に使用される。

蓄積交換 (store and forward). パケット、メッセージ、またはフレームが、その宛先に再送される前に、データ・ネットワーク内に一時的に保管されること。

チャネル (channel). メッセージ・チャネル (message channel)も参照。

チャンネル制御機能 (channel control function (CCF)). MQSeries において、伝送キューから通信リンクへ、および通信リンクからローカル・キューへメッセージを移動するプログラムで、操作盤インターフェースと連携してチャンネルのセットアップならびに制御を可能にするもの。

チャンネル定義ファイル (channel definition file (CDF)). MQSeries では、伝送キューと通信リンクを関連付ける通信チャンネル定義を含んでいるファイル。

チャンネル・イベント (channel event). チャンネル・インスタンスが利用可能になったか利用不可になったかを示すイベント。チャンネル・イベントは、チャンネルの両端のキュー・マネージャーで生成される。

重複モード (dual mode). 「重複ロギング (*dual logging*)」と対比。

重複ロギング (dual logging). MQSeries for OS/390 アクティビティを記録する方法の 1 つ。再始動が必要で 1 つのデータ・セットが読み取り不可能な場合、他のデータ・セットを使用するため、各変更が 2 つのデータ・セットに記録される。「単一ロギング (*single logging*)」と対比。

データグラム (datagram). MQSeries がサポートする最も単純なメッセージ。このタイプのメッセージは応答を必要としない。

データ変換インターフェース (data conversion interface (DCI)). 異なる機械エンコードと CCSID の間でのアプリケーション・データの変換を行う顧客作成またはベンダー作成のプログラムが従う必要がある MQSeries インターフェース。MQSeries フレームワークの一部。

デフォルト・オブジェクト (default object). すべての属性が定義されているオブジェクト (たとえば、キュー) の定義。ユーザーがオブジェクトを定義したときにそのオブジェクトに指定可能な属性を全部は指定しなかった場合は、キュー・マ

ネージャーが、指定されていない属性の代わりにデフォルト属性を使用する。

伝送キュー (transmission queue). リモート・キュー・マネージャーを宛先とする準備済みメッセージが一時的に保管されるローカル・キュー。

伝送プログラム (transmission program). 「メッセージ・チャンネル・エージェント (*message channel agent*)」を参照。

同期点 (syncpoint). トランザクションの保護リソースが整合される、トランザクションの処理中の中間点また終点。同期点では、リソースへの変更内容を安全にコミットしたり、または直前の同期点にそれらをバックアウトしたりできる。

同期メッセージング (synchronous messaging). プログラムがメッセージをメッセージ・キューに書き込む、プログラム間の通信の方式。同期メッセージングでは、送信側プログラムは、そのメッセージに対する応答を待ってから、それ自身の処理を再開する。「非同期メッセージング (*asynchronous messaging*)」と対比。

動的キュー (dynamic queue). プログラムがモデル・キュー・オブジェクトをオープンするときには作成されるローカル・キュー。「永続動的キュー (*permanent dynamic queue*)」および「一時動的キュー (*temporary dynamic queue*)」も参照。

同盟 (ally). MQSeries for OS/390 に接続される OS/390 アドレス・スペース。

トランザクション識別子 (transaction identifier). CICS では、トランザクションの定義時に指定される名前で、トランザクションを呼び出すときに使用されるもの。

トリガー処理 (triggering). MQSeries では、キュー上の事前決定条件が満たされると、キュー・マネージャーが自動的にアプリケーションを開始できるようにする機能。

トリガー・イベント (trigger event). キュー・マネージャーに、開始キュー上にトリガー・メッセージを作成させるイベント (たとえば、キューに届いているメッセージなど)。

トリガー・メッセージ (trigger message). トリガー・モニターが始動するプログラムに関する情報が入られるメッセージ。

トリガー・モニター (trigger monitor). 連続して実行されるアプリケーションで、1 つまたは複数の開始キューのためにサービスを行うもの。トリガー・メッセージが開始キューに到達すると、トリガー・モニターがそのメッセージを検索する。トリガー・モニターは、トリガー・イベントが発生したキューをサービスするプロセスを開始するために、トリガー・メッセージの中の情報を使用する。

トリガー・モニター・インターフェース (trigger monitor interface (TMI)). 顧客作成またはベンダー作成のトリガー・モニター・プログラムが従う必要がある MQSeries インターフェース。MQSeries フレームワークの一部。

トレース (trace). MQSeries では、MQSeries アクティビティを記録するための機能。トレース入力の宛先には、GTF およびシステム管理機能 (SMF) を含めることができる。「グローバル・トレース (global trace)」および「パフォーマンス・トレース (performance trace)」も参照。

[ナ行]

名前サービス (name service). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp および MQSeries for Windows NT and Windows 2000 では、指定されたキューを所有しているキュー・マネージャーを判別する機能。

名前サービス・インターフェース (name service interface (NSI)). キュー名の所有権を解決する顧客作成またはベンダー作成のプログラムが従う

必要がある MQSeries インターフェース。MQSeries フレームワークの一部。

名前変換 (name transformation). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 では、使用中のシステムで固有であり有効であるように、キュー・マネージャー名を変更する内部プロセス。外部から見れば、キュー・マネージャー名は変更されないままである。

名前リスト (namelist). 名前、たとえば、キュー名のリストが入っている MQSeries オブジェクト。

入出力パラメーター (input/output parameter). MQI 呼び出しのパラメーターで、呼び出しを行う際ユーザーはこのパラメーターに情報を提供し、MQI 呼び出しが完了または失敗すると、キュー・マネージャーがこのパラメーター内の情報を変更する。

入力パラメーター (input parameter). MQI 呼び出しのパラメーターで、呼び出しを行う際ユーザーはこのパラメーターに情報を提供する。

ヌル文字 (null character). 'X'00' で表される文字。

[ハ行]

バックアウト (backout). 現行のリカバリー単位または作業単位中に行われたすべての変更内容を取り消す操作。この操作の完了後、新規のリカバリー単位または作業単位が開始される。「コミット (commit)」と対比。

バッファー・プール (buffer pool). MQSeries for OS/390 キュー、メッセージ、およびオブジェクト定義に使用される主記憶域。「ページ・セット (page set)」も参照。

パフォーマンス・イベント (performance event). 限界条件が起きたことを示すイベントのカテゴリ。

パフォーマンス・トレース (performance trace). トレース・データがパフォーマンス分析およびチューニングに使用される、MQSeries トレース・オプション。

ハンドル (handle). 「接続ハンドル (*connection handle*)」および「オブジェクト・ハンドル (*object handle*)」を参照。

汎用コマンド・プリプロセッサ (generalized command preprocessor (GCPC)). MQSeries コマンドを処理し、実行する MQSeries for OS/390 コンポーネント。

汎用トレース機能 (GTF) (Generalized Trace Facility (GTF)). 監視プログラム呼び出しおよび入出力開始操作など、重要なシステム・イベントを、問題判別の目的で記録する OS/390 サービス・プログラム。

非持続メッセージ (nonpersistent message). キュー・マネージャーを再始動すると消えるメッセージ。「持続メッセージ (*persistent message*)」と対比。

非同期メッセージング (asynchronous messaging). プログラムがメッセージをメッセージ・キューに書き込む、プログラム間の通信の方式。非同期メッセージングでは、送信側プログラムは、そのメッセージへの応答を待つことなく、送信側の処理を進行させる。「同期メッセージング (*synchronous messaging*)」と対比。

ブートストラップ・データ・セット (bootstrap data set (BSDS)). 次のものを含む VSAM データ・セット。

- MQSeries for OS/390 に通知されるすべてのアクティブおよびアーカイブ・ログのインベントリー。
- すべての最近の MQSeries for OS/390 アクティビティの循環インベントリー。

MQSeries for OS/390 サブシステムが再始動しなければならない場合は、BSDS が必要。

ブラウズ (browse). メッセージ・キューイングにおいて、MQGET 呼び出しを使用してキューからメッセージを除去せずにメッセージをコピーすること。「読み取り (*get*)」も参照。

ブラウズ・カーソル (browse cursor). メッセージ・キューイングにおいて、次の順序のメッセージを識別するためにキューをブラウズする際に使用される標識。

プラットフォーム (platform). MQSeries では、キュー・マネージャーがそのもとで実行されているオペレーティング・システムのこと。

プリンシパル (principal). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 において、ユーザー識別子を表すために使用する用語。システム・リソースについての許可を検査するためにオブジェクト権限マネージャーが使用する。

フレームワーク (Framework). MQSeries では、お客様またはベンダーが、MQSeries プロダクトの提供する一部の機能を拡張したり置き換えたりするプログラムを作成できるようにするプログラミング・インターフェースの集合。インターフェースは次のとおり。

- MQSeries データ変換インターフェース (DCI)
- MQSeries メッセージ・チャンネル・インターフェース (MCI)
- MQSeries 名前サービス・インターフェース (NSI)
- MQSeries セキュリティー有効化インターフェース (SEI)
- MQSeries トリガー・モニター・インターフェース (TMI)

プログラム一時修正 (program temporary fix (PTF)). プログラムの現行版の変更が加えられていないリリースの欠陥の結果であると、IBM 技術員が診断した問題の解決またはバイパス。

プログラム式コマンド形式 (programmable command format (PCF)). 次の場合に使用される MQSeries メッセージのタイプ。

- ユーザー管理アプリケーションが、指定されたキュー・マネージャーのシステム・コマンド入力キューに PCF コマンドを書き込むために使用する。
- ユーザー管理アプリケーションが、指定されたキュー・マネージャーから PCF コマンドの結果を読み取るために使用する。
- キュー・マネージャーが、イベント発生の通知として使用する。

「MQSC」と対比。

プログラム診断依頼書 (authorized program analysis report (APAR)). 未変更の現行リリースのプログラムに生じた障害のために引き起こされた問題のレポート。

プロセス定義オブジェクト (process definition object). MQSeries アプリケーションの定義を含む MQSeries オブジェクト。たとえば、キュー・マネージャーはトリガー・メッセージを用いた作業を行うときにこの定義を使用する。

分散アプリケーション (distributed application). メッセージ・キューイングにおいて、それぞれが異なるキュー・マネージャーに接続できる一方で、全体として単一のアプリケーションを構成するアプリケーション・プログラムの集合。

分散キュー管理 (distributed queue management). メッセージ・キューイングでは、他のシステムのキュー・マネージャーへのメッセージ・チャネルの設定と制御。

分散コンピューター環境 (Distributed Computing Environment (DCE)). いくつかの基本サービスを提供し、分散アプリケーションの開発を容易にするミドルウェア。DCE は Open Software Foundation (OSF) によって定義されている。

ページ・セット (page set). MQSeries for OS/390 が主記憶域のバッファーから永続バッキング・ストレージ (DASD) にデータ (たとえば、キューおよびメッセージなど) を移動するときに使用される VSAM データ・セット。

別名キュー・オブジェクト (alias queue object). MQSeries オブジェクトの 1 つで、ローカル・キュー・マネージャーに対して定義されている基本キューの別名を名前として持つもの。アプリケーションまたはキュー・マネージャーが別名キューを使用するときには、別名が解決され、要求した操作が関連する基本キューに対して実行される。

保留イベント (pending event). CICS アダプターからの接続要求の結果として発生した未スケジュールのイベント。

[マ行]

マシン・チェック割り込み (machine check interrupt). 装置の誤動作またはエラーの結果発生する割り込み。マシン・チェック割り込みは、ハードウェアで回復可能か、ソフトウェアで回復可能か、または回復不能。

未確定リカバリー単位 (unit of recovery). MQSeries で、同期点が要求されているがまだ確認されていないリカバリー単位の状況。

未配布メッセージ・キュー (undelivered-message queue). 「送達不能キュー (dead-letter queue)」を参照。

メッセージ (message). メッセージ・キューイング・アプリケーションにおいて、プログラム間で送信される連絡。「持続メッセージ (persistent message)」および「非持続メッセージ (nonpersistent message)」も参照。システム・プログラミングにおいて、端末オペレーターまたはシステム管理者に宛てられた情報。

メッセージ記述子 (message descriptor). メッセージの形式および表現を記述する制御情報で、MQSeries メッセージの一部として伝送されるもの。メッセージ記述子の形式は、MQMD 構造によって定義される。

メッセージ順序番号付け (message sequence numbering). 通信リンク上での伝送中に、メッセージに固有の番号を与えるプログラミング技法。これにより、すべてのメッセージが受信されたかどうかの検査、メッセージを元の順序でキュー内に配置すること、および重複メッセージの廃棄のための受信処理が可能になる。

メッセージ優先順位 (message priority). MQSeries において、キュー上のメッセージが検索される順序およびトリガー・イベントが生成されるかどうかに影響を与え得るメッセージの属性。

メッセージング (messaging). 「同期メッセージング (*synchronous messaging*)」および「非同期メッセージング (*asynchronous messaging*)」を参照。

メッセージ・キュー (message queue). キュー (*queue*) の同義語。

メッセージ・キューイング (message queuing). アプリケーション内の各プログラムが、他のプログラムとの通信を、キューにメッセージを入れることで行うプログラミング技法。

メッセージ・キュー・インターフェース (message queue interface (MQI)). MQSeries キュー・マネージャーにより提供されるプログラミング・インターフェース。このプログラミング・インターフェースによって、アプリケーション・プログラムはメッセージ・キュー・サービスにアクセスすることができる。

メッセージ・チャンネル (message channel). 分散メッセージ・キューイングにおいて、メッセージをあるキュー・マネージャーから別のキュー・マネージャーに移動するメカニズム。メッセー

ジ・チャンネルは、2つのメッセージ・チャンネル・エージェント (一方は送信側で他方は受信側) および1つの通信リンクで構成される。「MQI チャンネル (*MQI channel*)」と対比。

メッセージ・チャンネル・インターフェース (message channel interface (MCI)). MQSeries キュー・マネージャーと他のメッセージ・システムとの間でメッセージを送信する顧客作成またはベンダー作成のプログラムが従う必要がある MQSeries インターフェース。MQSeries フレームワークの一部。

メッセージ・チャンネル・エージェント (message channel agent (MCA)). 準備済みメッセージを、伝送キューから通信リンクに、あるいは通信リンクから宛先キューに伝送するプログラム。「メッセージ・キュー・インターフェース (*message queue interface*)」も参照。

メディア・イメージ (media image). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 において、あるオブジェクトのイメージが格納されている一連のログ・レコード。オブジェクトはこのイメージから再作成できる。

モデル・キュー・オブジェクト (model queue object). プログラムが動的キューを作成するときにテンプレートとして動作するキュー属性のセット。

戻りコード (return codes). 完了コードおよび理由コードの集成的な名前。

[ヤ行]

ユーザー識別子サービス (User identifier service (UIS)). MQSeries for OS/2 Warpでは、MQI アプリケーションが、デフォルト・ユーザー ID 以外のユーザー ID を MQSeries メッセージと関連付けることができるようにする機能。

ユーティリティ (utility). MQSeries では、システム・オペレーターまたはシステム管理者に、MQSeries コマンドによって提供される機能に加えて、さらに機能を提供する一連のプログラム。複数の機能呼び出すユーティリティもある。

優先シャットダウン (preemptive shutdown). MQSeries において、接続しているアプリケーションが切断するまで待つことも、現行 MQI 呼び出しが完了するまで待つこともないキュー・マネージャーのシャットダウン。「即時シャットダウン (*immediate shutdown*)」および「静止シャットダウン (*quiesced shutdown*)」と対比。

要求側チャンネル (requester channel). メッセージ・キューイングにおいて、送信側チャンネルがリモートから開始できるチャンネル。要求側チャンネルは、通信リンク全体の送信側チャンネルからメッセージを受け入れ、そのメッセージをメッセージ内の指定されたローカル・キューに入れる。「サーバー・チャンネル (*server channel*)」も参照。

要求メッセージ (request message). 別のプログラムからの応答を要求するために使用されるメッセージのタイプ。「応答メッセージ (*reply message*)」および「レポート・メッセージ (*report message*)」と対比。

[ラ行]

リカバリー終了マネージャー (recovery termination manager (RTM)). 終了する関数と関連したりカバリー・ルーチンに制御を渡すことによって、タスクのすべての通常および異常終了を処理するプログラム。

リカバリー単位 (unit of recovery). 単一のリソース・マネージャー内のリカバリー可能な一連の操作。「作業単位 (*unit of work*)」と対比。

リカバリー点 (point of recovery). MQSeries for OS/390 では、MQSeries for OS/390 ページ・セットのバックアップ・コピーのセット、およびこれらのページ・セットを回復するのに必要な対

応するログ・データ・セットを記述するのに使用される用語。これらのバックアップ・コピーは、ページ・セットの欠落 (たとえば、ページ・セット入出力エラー) のイベント時に、潜在的な再始動点を提供する。

リカバリー・ログ (recovery log). MQSeries for OS/390 では、メッセージ、キュー、および MQSeries サブシステムの回復に必要な情報を含むデータ・セット。MQSeries for OS/390 は各レコードを、アクティブ・ログと呼ばれるデータ・セットに書き込む。アクティブ・ログがいっぱいになると、その内容は DASD、またはアーカイブ・ログと呼ばれるテープ・データ・セットにオフロードされる。「ログ (*log*)」と同義。

リスナー (listener). MQSeries の分散キューイングにおいて、着信側ネットワーク接続を監視するプログラム。

リソース (resource). ジョブまたはタスクが必要とするコンピューター・システム、またはオペレーティング・システムの機能。MQSeries for OS/390 では、リソースの例としては、バッファーク・プール、ページ・セット、ログ・データ・セット、キュー、およびメッセージがある。

リソース・マネージャー (resource manager). メモリー・バッファークやデータ・セットなどの共用リソースへのアクセスを管理、制御するアプリケーション、プログラム、またはトランザクション。MQSeries、CICS、および IMS はリソース・マネージャーに相当する。

リソース・リカバリー・サービス (Resource Recovery Services (RRS)). 参加しているリソース・マネージャーすべてにわたって、2 フェーズ同期点サポートを提供する OS/390 機能。

リニア・ロギング (linear logging). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000において、再始動データを一連のファイルに保持するプロセス。新規ファイルが必要に応じて

一連のファイルに追加される。データが書き込まれるスペースは、キュー・マネージャーが再始動するまで利用されない。「循環ロギング (*circular logging*)」と対比。

リモート・キュー (remote queue). リモート・キュー・マネージャーに属するキュー。プログラムはリモート・キューにメッセージを入れることはできるが、リモート・キューからメッセージを得ることはできない。「ローカル・キュー (*local queue*)」と対比。

リモート・キューイング (remote queuing). メッセージ・キューイングにおいて、アプリケーションが、他のキュー・マネージャーに属しているキューにメッセージを入れることができるようにするサービスの提供。

リモート・キューのローカル定義 (local definition of a remote queue). ローカル・キュー・マネージャーに属する MQSeries オブジェクト。このオブジェクトは、別のキュー・マネージャーが所有するキューの属性を定義する。また、キュー・マネージャーの別名付けおよび応答先キューの別名付けに使用される。

リモート・キュー・オブジェクト (remote queue object). 「リモート・キューのローカル定義 (*local definition of a remote queue*)」を参照。

リモート・キュー・マネージャー (remote queue manager). プログラムにとって、そのプログラムの接続先のものでないキュー・マネージャー。

理由コード (reason code). MQI 呼び出しの失敗または部分的な成功の理由を記述する戻りコード。

ルール・テーブル (rules table). 送達不能キュー・ハンドラーが DLQ 上のメッセージに適用する 1 つまたは複数の規則が入った制御ファイル。

レジストリー (Registry). Windows NT では、システムとアプリケーション構成データのために単一のソースを提供する保護データベース。

レジストリー編集プログラム (Registry Editor). Windows NT では、ユーザーがレジストリーを編集できるプログラム項目。

レジストリー・ハイブ (Registry Hive). Windows NT では、レジストリーに保管されているデータの構造。

レポート・メッセージ (report message). 別のメッセージに関する情報を与えるメッセージのタイプ。メッセージが送達された、その宛先に到着した、その期限が切れた、何らかの理由でメッセージを処理できなかったなどということを、レポート・メッセージは示すことができる。「応答メッセージ (*reply message*)」および「要求メッセージ (*request message*)」と対比。

ローカル定義 (local definition). ローカル・キュー・マネージャーに属する MQSeries オブジェクト。

ローカル・キュー (local queue). ローカル・キュー・マネージャーに属するキュー。ローカル・キューには、処理を待っているメッセージのリストが含まれている。「リモート・キュー (*remote queue*)」と対比。

ローカル・キュー・マネージャー (local queue manager). プログラムが接続されるキュー・マネージャーであって、メッセージ・キューイング・サービスをプログラムに提供する。プログラムが接続されていないキュー・マネージャーは、それらがそのプログラムと同じシステム上で実行されているとしても、リモート・キュー・マネージャーと呼ばれる。

ロールバック (rollback). 「バックアウト (*back out*)」と同義。

ログ (log). MQSeries では、キュー・マネージャーがメッセージを受信、伝送、および送達する

ときに行う作業を記録するファイル。これにより、障害時にキュー・マネージャーは回復できる。

ログ制御ファイル (log control file). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 において、ログ・ファイルの使用状況 (ログ・ファイルのサイズ、格納場所、次に利用可能なファイルの名前など) を監視するのに必要な情報や、次に使用可能なファイルの名前が格納されるファイル。

ログ・ファイル (log file). MQSeries (UNIX システム用)、MQSeries for OS/2 Warp、および MQSeries for Windows NT and Windows 2000 において、キュー・マネージャーが制御しているデータに対する有効な変更がすべて記録されるファイル。1 次ログ・ファイルが満杯になると、MQSeries は 2 次ログ・ファイルを割り振る。

ロケール (locale). UNIX システムでは、特定の文化のための約束 (時刻、数値、通貨などの形式設定、および文字の分類、照合、変換など) を定義するユーザー環境のサブセットの 1 つ。キュー・マネージャー CCSID は、キュー・マネージャーを作成したユーザー ID のロケールに由来する。

[数字]

2 フェーズ・コミット (two-phase commit). 単一のトランザクションによって複数のリソース・マネージャーが使用されるときに、リカバリ可能リソースへの変更内容を調整するためのプロトコル。「単一フェーズ・コミット (single-phase commit)」と対比。

A

APAR. プログラム診断依頼書 (authorized program analysis report)。

ASID. アドレス・スペース ID。

B

BMS. 基本マッピング・サポート。

BSDS. ブートストラップ・データ・セット。

C

CCF. チャネル制御機能 (channel control function)。

CCSID. コード化文字セット識別子 (Coded character set identifier)。

CDF. チャネル定義ファイル (Channel definition file)。

CI. 制御インターバル。

CL. 制御言語。

CPF. コマンド接頭部。

D

DAE. ダンプ分析と除去。

DCE. 分散コンピューター環境 (Distributed Computing Environment)。

DCI. データ変換インターフェース (Data conversion interface)。

DLQ. 送達不能キュー (Dead-letter queue)。

DQM. 分散キュー管理 (Distributed queue management)。

E

ESM. 外部セキュリティ・マネージャー。

ESTAE. ESTAE 出口ルーチン。

ESTAE 出口ルーチン (extended specify task abnormal exit (ESTAE)). リカバリー機能を提供し、処理、異常終了の診断、または再試行アドレスの指定のため、指定された出口ルーチンに制御を与える OS/390 マクロ。

F

FFST. 第 1 障害サポート・テクノロジー (First Failure Support Technology)。

FIFO. 先入れ先出し法 (first-in-first-out)。

FRR. 機能回復ルーチン。

G

GCPC. 汎用コマンド・プリプロセッサ。

get. メッセージ・キューイングにおいて、MQGET 呼び出しを使用してキューからメッセージを除去すること。「ブラウズ (*browse*)」も参照。

GTF. 汎用トレース機能。

I

IPCS. 対話式問題制御システム。

ISPF. 対話式システム生産性向上機能。

M

MCA. メッセージ・チャンネル・エージェント (message channel agent)。

MCI. メッセージ・チャンネル・インターフェース (Message channel interface)。

MQAI. MQSeries 管理インターフェース (MQSeries Administration Interface)。

MQI. メッセージ・キュー・インターフェース (Message queue interface)。

MQI チャンネル (MQI channel). MQSeries クライアントをサーバー・システム上のキュー・マネージャーに接続して、MQI 呼び出しと応答のみを両方向に転送するもの。「メッセージ・チャンネル (*message channel*)」と対比。

MQSC. MQSeries コマンド (MQSeries command)。

MQSeries. メッセージ・キュー・サービスを提供する、IBM ライセンス・プログラムのファミリー。

MQSeries 管理インターフェース (MQSeries Administration Interface (MQAI)). MQSeries に対するプログラミング・インターフェース。

MQSeries クライアント (MQSeries client). キュー・マネージャー全体をインストールしなくてもシステムにインストールできる MQSeries プロダクトの一部。MQSeries クライアントは、アプリケーションから MQI 呼び出しを受け入れて、サーバー・システム上のキュー・マネージャーと通信する。

MQSeries コマンド (MQSeries commands (MQSC)). すべてのプラットフォームで同じ形を持ち、MQSeries オブジェクトの操作に使用される、人が読めるコマンド。「プログラム式コマンド形式 (*programmable command format (PCF)*)」と対比。

N

New Technology File System (NTFS). ファイルのセキュリティを提供する Windows NT 回復可能ファイル・システム。

NSI. 名前サービス・インターフェース (Name service interface)。

NTFS. New Technology File System の略。

O

OAM. オブジェクト権限マネージャー (object authority manager)。

P

PCF. プログラム式コマンド形式 (Programmable command format)。

PCF コマンド (PCF command). 「プログラム式コマンド形式 (*programmable command format*)」を参照。

PING (ping). 分散キューイングにおいて、テスト・メッセージの交換を使用することによりメッセージ・チャンネルまたは TCP/IP 接続が機能しているかどうかを確認する診断援助機能。

PTF. プログラム一時修正 (program temporary fix)。

R

RBA. 相対バイト・アドレス (relative byte address)。

RESLEVEL. MQSeries for OS/390 では、MQSeries for OS/390 で API リソース・セキュリティのために検査される、CICS ユーザー ID の数を制御するオプション。

RRS. リソース・リカバリー・サービス。

RTM. リカバリー終了マネージャー。

S

SAF. システム権限機能。

SDWA. システム診断作業域。

SEI. セキュリティー有効化インターフェース (Security enabling interface)。

SIT. システム初期設定テーブル。

SVC. 監視プログラム呼び出し。

SYS1.LOGREC. プログラムおよびハードウェア・エラーについての情報を含む保守援助機能。

T

TACL. タンデム拡張コマンド言語。

TCB. タスク制御ブロック。

thlqual. ターゲット・ライブラリー高水準修飾子。

TMI. トリガー・モニター・インターフェース (Trigger monitor interface)。

tranid. 「トランザクション識別名 (*transaction identifier*)」を参照。

U

UIS. ユーザー識別子サービス (User identifier service)。

undo/redo レコード (undo/redo record). リカバリーに使用されるログ・レコード。このレコードの redo の部分は、MQSeries オブジェクトに加えらるる予定の変更を記述する。undo の部分は、作業がコミットされない場合に、変更をどのようにバックアウトするかを記述する。

[特殊文字]

.ini ファイル (.ini file). 「構成ファイル (*configuration file*)」を参照。

索引

日本語、数字、英字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。

[ア行]

- アクション・キーワード、ルール・テーブル 137
- 宛先キュー
 - 所有キュー・マネージャーの名前 95
 - 定義 95
 - 未配布メッセージ 131
- 宛先キュー・マネージャー
 - クラスターに属さない 16
 - チャンネルの開始 90
 - チャンネルの作成 88
 - 定義 95
 - 伝送キューの作成 88
 - への別名の割り当て 98
- アプリケーション
 - クライアント / サーバー環境 21
 - 時間に依存しない 7
 - 設計の考慮事項 256
 - データ 8
 - トラステッド 475
 - プログラミング・エラーの例 252
 - ローカル・キュー・マネージャーに接続されている 94
 - MQI 管理サポート 45
- 一時キュー 10
- イベント
 - 観測
 - 解説 147
 - 使用する理由 149
 - 説明 147
 - タイプ 149
 - メッセージ 152

- イベント (続き)
 - 観測 (続き)
 - 有効化および無効化 151
 - キュー 150
 - タイプ 149
 - チャンネル 150
 - トリガー 150
 - show events (MONMQ) 441
 - イベントのタイプ 149
 - イベント・キュー 17
 - イベント・ドリブ処理 8
 - インストール可能サービス
 - オブジェクト権限マネージャー 105
 - オブジェクト権限マネージャーの使用不可
 - 使用不可 108
 - 名前サービス 219
 - インストール可能なコンポーネント
 - 権限マネージャー (OAM) 105
 - 名前サービス 219
 - 永続キュー 10
 - エスケープ PCF 28, 78
 - エラー・メッセージ 50
 - エラー・ログ
 - 概要 262
 - 確立前に生じるエラー 263
 - 例 264
 - 応答先キュー 17
 - 応答先キュー別名 98
 - 大文字小文字の区別 30
 - キュー・マネージャー名 29
 - 制御コマンド 30
 - MQSC コマンド 31
- オブジェクト
- アクセス 103
 - オブジェクトを表すファイル 314
 - 管理 11
 - キュー 14

- オブジェクト (続き)
 - キュー・マネージャー
 - MQI 呼び出し 13
 - 再作成コマンド 358
 - システム・デフォルト 19, 397
 - 属性 12
 - タイプ 10
 - デフォルト
 - 属性 60
 - トリガー操作のための 73
 - 名前 11, 48
 - 名前変換 29
 - 名前リスト 19
 - プロセス定義 18
 - 命名規則 313
 - メディア・イメージ 186
 - リカバリー 187
 - リモート管理 83
- オブジェクト権限マネージャー 105
 - 機能 106
 - 重要操作 112
 - 使用不可 108
 - デフォルト権限 ID 107
 - プリンシパル 106
 - dspmqaout コマンド 111
 - setmqaut コマンド 109, 110
- オブジェクト権限マネージャーの使用不可 108
- オペレーター・コマンド、応答なし 253
- オペレーター・メッセージ 263
- オペレーティング・システム論理名、セキュリティを使用不可にする 108

[カ行]

- 開始
 - キュー・マネージャー 41
 - チャンネル 89
 - フェールオーバー・セット内のキュー・マネージャー 292

- 開始キュー
 - 説明 15
 - 定義 75
- 開始コマンド・プロシージャ 289
- 確認、MQSC コマンドの 55
- 各国語サポート
 - 命名規則 313
 - EBCDIC NL 文字の ASCII への
変換 224
- 環境変数
 - MQSPREFIX 224
 - ORACLE_HOME、Oracle 159
 - ORACLE_SID、Oracle 159
- 監視プログラム・フェールオーバ
ー・モニター
 - 説明 284
 - 停止 297
- 関数
 - コンポーネント内の表示 444
- 幹線図、見方 314
- 観測イベント
 - 使用可能化 151
 - 使用する理由 149
 - 説明 147
 - タイプ 149
 - メッセージ 152
- 管理
 - 許可 122
 - コマンド・セット 25
 - 制御コマンド 26
 - プログラム式コマンド形式
(PCF) コマンド 27
 - MQSeries コマンド
(MQSC) 26
 - リモート 85
 - オブジェクト 83
 - チャンネル 87
 - 伝送キュー 87
 - ローカル 45
- 管理、アクセスの 106
- 管理、ログ・ファイルの 184
- キュー
 - アプリケーション、トリガー操作
のための定義 73
 - 一時 10
 - イベント 17
- キュー (続き)
 - イベント通知 150
 - 応答先 17, 97
 - オブジェクト
 - 別名 14
 - モデル 15
 - リモート 14
 - ローカル 14
 - 開始
 - 定義 75
 - トリガー・メッセージ 15
 - 共用、構成タスク 219
 - 許可 113
 - クラスター伝送 16
 - コマンド 17
 - 事前定義 10
 - 障害 254
 - 説明 9
 - 送達不能 17
 - 指定 37
 - 属性 13
 - 変更 62
 - 定義 13
 - 伝送 16
 - 作成 97
 - 定義 88
 - デフォルト 38, 97
 - リモート管理 87
 - 動的 10
 - 取り扱い 58
 - の共用、さまざまなキュー・マネ
ージャー上の 219
 - ブラウズ 63
 - 分散、不正出力 260
 - 別名 14
 - 別名、取り扱い 69
 - 未配布メッセージ 17
 - 指定 37
 - モデル 15
 - 定義 72
 - 取り扱い 71
 - リモート 13, 14
 - キュー・マネージャー別名
97
 - 作成 93
 - 取り扱い 97
- キュー (続き)
 - ローカル 12, 14
 - クリア 63
 - コピー 61
 - 削除 63
 - 定義 59
 - デフォルト 19
 - MQSeries アプリケーションのた
めの 45
 - キュー項目数
 - 現行 61
 - 判別 61
 - キューの共用 219
 - キューの属性を変更 62
 - キュー・マネージャー
 - イベント 149
 - オブジェクト
 - MQI 呼び出し 13
 - オブジェクト権限マネージャー
 - 使用不可 108
 - 説明 105
 - 開始 41
 - 数 36
 - 許可 112
 - 許可ディレクトリー 126
 - クラスター 16, 18
 - 形式化されたシステム・ログのダ
ンプ (dmpmqlog) コマンド 328
 - 構成ファイル
 - 指定 39
 - コマンド・サーバー 80
 - 固有の名前 36
 - 再始動 44
 - 削除 44
 - 作成 35, 40
 - シャットダウン
 - 制御された 42
 - 静止 42
 - 循環ロギング、リカバリーの再始
動 177
 - 説明 12
 - 即時シャットダウン 42
 - 停止 42
 - ディレクトリー 112
 - デフォルト 37
 - 誤った削除 322

- キュー・マネージャー (続き)
 - 誤った変更 41
 - 変更 41
 - 名前変換 29
 - フェールオーバー・セット内の移動 293
 - フェールオーバー・セット内の始動 292
 - フェールオーバー・セット内の終了 293
 - メッセージの変換 99
 - メディア・イメージの記録 187
 - モニター 149
 - 優先シャットダウン 43
 - リカバリー・ログの内容のダンプ 194
 - リニア・ロギング 175
 - リモート管理 83
 - リモート・キューを使用した別名 97
 - ローカル管理 45
 - ログ 174
 - CCSID への変更後の再始動 101
 - crtmqm コマンド 320
 - dltmqm コマンド 326
 - endmqm コマンド 346
 - MVS/ESA での 92
 - qm.ini ファイル 223
 - runmqsc に指定 52
- キュー・マネージャーの起動コマンド 385
- キュー・マネージャー・クラスター
 - クラスターも参照 18
- キュー・モード、runmqsc の 91
- 行の最大長、MQSC コマンドの 54
- 共用キュー
 - 構成タスク 219
- 共用メモリーの管理 458
- 許可
 - 管理 122
 - 権限 ID 108
 - リスト 109
 - dspmqaout コマンド 111
 - MQI 117
 - setmqaut コマンド 111
- 許可ファイル
 - 管理 129
 - 許可 130
 - クラス 129
 - 全クラス 129
 - ディレクトリー 126
 - 内容 127
 - パス 126
 - 理解 125
- クライアント 20
 - アプリケーションのリンク 21
 - そのためのチャネルの作成 21
 - トリガー・モニターの起動コマンド 375
- 問題判別 275
- DOS および Windows 上のエラー・メッセージ 276
- クラスター
 - キュー・マネージャー
 - 説明 18
 - 伝送キュー 16
 - 名前リストの使用 19
 - ワークロード出口 22, 473
 - OpenVMS クラスターとの違い 19
 - キュー・マネージャーの 11
 - 説明 85
 - リモート・キューイング 83
 - ExitProperties スタンザの属性 226
 - OpenVMS
 - キュー・マネージャー・クラスターとの相違 281
 - フェールオーバー・セット 283
 - MQSeries のインストール 282
- クラスター伝送キュー
 - 説明 16
- クラスター別名サービス 301
- クラスター・ワークロード出口 473
- クリア、ローカル・キューの 63
- グローバル作業単位
 - 定義 153
 - qm.ini への XAResourceanager スタンザの追加、Oracle 161
- グローバル・セクション 439
- グローバル・メモリー
 - システム・パラメーターを使用した制御 277
- 形式、ログの 174
- 権限
 - インストール可能サービス 111
 - コマンド 111
 - コンテキスト 114
 - 設定 / リセット・コマンド 377
 - 代替ユーザー 113
- 権限 ID
 - 権限に関するデフォルト 107
 - デフォルト、nobody 107
 - MQM 104
- 権限 ID、権限の 106
- 現行キュー項目数 61
- コード化文字セット 99
 - ID 427
- コード・セット 427
- 更新、キュー・マネージャーの属性 52
- 構成
 - キュー・マネージャー構成ファイル qm.ini 223
 - データベース、qm.ini 236
 - データベース・マネージャー 156
 - 変更のインプリメント 223
 - 編集 221, 222
 - 優先順位 222
 - ログ 233
 - AllQueueManagers スタンザ、mqsc.ini 224
 - Channels スタンザ、qm.ini 238
 - ClientExitPath スタンザ、mqsc.ini 226
 - DefaultQueueManager スタンザ、mqsc.ini 226
 - Exitpath スタンザ、qm.ini 242
 - ExitProperties スタンザ、mqsc.ini 226
 - failover.ini 288
 - Log スタンザ、qm.ini 233
 - LogDefaults スタンザ、mqsc.ini 227

構成 (続き)

LU62 スタンザ、qm.ini 240
mqs.ini、説明 223
OpenVMS クラスタ・フェール
オーバー・セット 288
Oracle 159
qm.ini ファイルの例 243
QueueManager スタンザ、
mqs.ini 230
Service スタンザ、qm.ini 231
ServiceComponent スタンザ、
qm.ini 232
TCP スタンザ、qm.ini 240
XAResourceManager スタンザ、
qm.ini 236

構成ファイル

キュー・マネージャ (qm.ini)
オブジェクト権限マネージャ
ーの使用不可 108
内容 223, 243
Channels スタンザ 238
ExitPath スタンザ 242
Log スタンザ 233
LU62 および TCP スタンザ
240
Service Component スタンザ
232
Service スタンザ 231
XARsourceManager スタンザ
236

編集 222

failover.template 417

MQSeries (mqs.ini)

内容 223
パス 56

AllQueueManagers スタンザ
224

ClientExitPath スタンザ 226

DefaultQueueManager スタンザ
226

ExitProperties スタンザ 226

LogDefaults スタンザ 227

QueueManager スタンザ 230

OpenVMS クラスタ・フェール
オーバー・セット 288

高速パス・バインディング 475

構文

エラー、MQSC コマンドの 50
図、見方 314
ヘルプ 316

構文に関するヘルプ 316

コマンド

オブジェクトの再作成

(rcrmqobj) 358

キュー・マネージャの開始

(strmqm) 385

キュー・マネージャの削除

(dlmqm) 326

キュー・マネージャの作成

(crtmqm) 320

キュー・マネージャの終了

(endmqm) 346

許可の表示 (dspmqaut) 330

クライアント・トリガー・モニタ

ーの起動 (runmqmtmc) 375

権限の設定 / リセット

(setmqaut) 110, 377

構文に関するヘルプ 316

コマンド・サーバーの開始

(strmqcsv) 384

コマンド・サーバーの終了

(endmqcsv) 343

コマンド・サーバーの表示

(dspmqcsv) 335

障害モニターの開始

(runmqfm) 367

制御 26

セキュリティ・コマンド

dspmqaut 111

setmqaut 109

セットの比較 409

送達不能キュー・ハンドラーの実
行 365

ダンプ・ログ (dmpmqlog) コマン
ド 328

チャンネル開始プログラムの実行

(runmqchi) 363

チャンネルの実行 (runmqchl) 364

データ変換 (crtmqcvx) 318

トリガー・モニターの開始

(runmqtrm) 376

コマンド (続き)

フェールオーバーの管理 (フェー
ルオーバー) 351

プログラム式コマンド形式
(PCF) 27

メディア・イメージの記録

(rcdmqimg) 355

リスナーの実行 (runmqslr) コマン
ド 369

リスナーの終了 (endmqslr) コマ
ンド 345

DLQ ハンドラーの実行

(runmqdlq) 131

MQSC

ALTER QLOCAL 62

ALTER QREMOTE 96

DEFINE CHANNEL 89

DEFINE QALIAS 69

DEFINE QLOCAL 61

DEFINE QLOCAL LIKE 61

DEFINE QLOCAL

REPLACE 62

DEFINE QMODEL 72

DEFINE QREMOTE 94

DELETE QLOCAL 63

DELETE QREMOTE 96

DISPLAY QREMOTE 96

MQSC コマンド・ファイル

出力レポート 54

入力 53

MQSeries (MQSC)

確認 55

使用 27

MQSeries コマンド (MQSC) 26

MQSeries コマンドの実行

(runmqsc) 371

MQSeries 定様式トレース出力の
表示 (dspmqtrc) 340

MQSeries トランザクションの解
決 (rsvmqtrn) 361

MQSeries トランザクションの表
示 (dspmqtrn) 341

MQSeries トレースの開始

(strmqtrc) 387

MQSeries トレースの終了

(endmqtrc) 349

- コマンド (続き)
 - MQSeries ファイルの表示 (dspmqfls) 337
 - runmqsc 49
 - コマンド、MQSeries 用の 26
 - コマンド・キュー 17
 - コマンド・サーバー
 - 開始コマンド 384
 - コマンド・サーバーの開始 80
 - コマンド・サーバーを停止する 81
 - 終了コマンド 343
 - 状況の表示 80
 - 表示コマンド 335
 - リモート管理 80
 - コマンド・セット
 - 管理 25
 - 比較 409
 - コマンド・ファイル 53
 - コマンド・プロシージャ 289
 - 変更 303
 - 例 303
 - コンテキスト権限 114
 - コンポーネント
 - show functions (MONMQ) 444
- ## [サ行]
- サーバー接続チャンネルの自動定義 90
 - 再始動、キュー・マネージャの 44
 - 再始動、リカバリーの、循環ログイン グ 177
 - 最大
 - キュー項目数 14
 - キューのサイズ 9
 - 名前長、MQSeries オブジェクトの 11
 - メッセージ数 10, 14
 - メッセージ長 9, 12
 - 作業単位
 - 混在する実行結果 168
 - 定義 153
 - 明示再同期 (rsvmqtrn コマンドの) 167
 - 作業負荷の平衡化
 - クラスターを使用した 11
 - MQI 呼び出しのリダイレクト 69
 - 削除
 - キュー・マネージャ 44
 - ローカル・キュー 63
 - dltmqm コマンド 326
 - 作成
 - オブジェクト 11
 - キュー 13
 - キュー・マネージャ 35, 40
 - プロセス定義 75
 - crtmqm コマンド 320
 - 参考資料 488
 - サンプル
 - トレース・データ 268
 - プログラムの使用 413
 - MQSC ファイル 413
 - 時間に依存しないアプリケーション 7
 - システム
 - デフォルト・オブジェクト 19, 397
 - システム・パラメーター 277
 - 事前定義キュー 10
 - シャットダウン
 - キュー・マネージャ 42
 - 制御された 42, 43
 - 静止 42
 - 即時 42
 - 優先 43
 - 終了、キュー・マネージャの 43
 - 終了、対話式 MQSC コマンドの 60
 - 終了コマンド・プロシージャ 289
 - 終了処理コマンド・プロシージャ 289
 - 受信側チャンネル定義 84
 - 受信側のチャンネルの自動定義 90
 - 出力
 - 宛先変更する 453
 - 循環ログイン 175
 - 使用可能化
 - イベント 151
 - セキュリティ 108
 - 所定稼働環境 393
 - 処理、イベント・ドリブン 8
 - スイッチ・ロード・ファイルの作成 157
 - スクリプト
 - MONMQ 459
 - スタンザ
 - AllQueueManagers、mqs.ini 224
 - Channels、qm.ini 238
 - ClientExitPath、mqs.ini 226
 - DefaultQueueManager、mqs.ini 226
 - ExitPath、qm.ini 242
 - ExitProperties、mqs.ini 226
 - LogDefaults、mqs.ini 227
 - Log、qm.ini 233
 - LU62、qm.ini 240
 - QueueManager、mqs.ini 230
 - ServiceComponent、qm.ini 232
 - Service、qm.ini 231
 - TCP、qm.ini 240
 - XAResourceManager、qm.ini 236
 - 制御コマンド 26
 - 大文字小文字の区別 30
 - 使用 33
 - runmqsc 49
 - 制御されたシャットダウン 42
 - 制限
 - オブジェクト名 313
 - データベース調整サポート 156
 - MQM オブジェクトへのアクセス 103
 - 静止シャットダウン、キュー・マネージャ 42
 - セキュリティ 103
 - コマンドの使用 109, 111
 - 使用可能化 108
 - フェールオーバー・セット・モニター 299
 - リモート 115
 - セル、DCE とキュー 219
 - 送信側チャンネル定義 84
 - 送達不能キュー (DLQ)
 - 指定 37
 - 説明 17
 - 送達不能キュー・ハンドラーの実行 (runmqdlq) 365

送達不能キュー (DLQ) (続き)

ハンドラー、「DLQ ハンドラー」も参照 131

送達不能ヘッダー、MQDLH 131

属性

キュー 13

キュー・マネージャー

更新 52

表示 51

キュー・マネージャーの表示 51

更新 52

デフォルト 60

変更 62

ALL 属性 60

MQSC と PCF の比較 28

ソフトコピー資料 487

[夕行]

代替ユーザー権限 113

タイプ、オブジェクトの 10

タイムアウト、MQSC コマンドの
答の 91

タイム・スタンプ

disable timestamp (MONMQ) 449

enable timestamp (MONMQ) 449

対話式 MQSC

からのフィードバック 50

終了 50

使用 49

ダンプ

形式化されたシステム・ログ

(dmpmqlog) コマンド 328

リカバリー・ログの内容のダンプ
194

ログ・レコードのダンプ

(dmpmqlog コマンド) 194

ダンプ・ログ

形式 328

パラメーター 328

目的 328

チャンネル

イベント 150

エスケープ・コマンドの許可

122

開始 89

チャンネル (続き)

開始プログラムの実行コマンド
363

キュー・マネージャー間での定義
14

コマンド 115

コマンド・セキュリティ要件
115

実行コマンド 364

自動定義 90

受信側チャンネル定義 84

セキュリティ 115, 116

説明 18, 83

送信側チャンネル定義 84

ターゲット・スレッドの接続先
446

ターゲット・スレッドを切断する
446

定義 88

トラステッド 477

メッセージの履歴の表示
438

リモート管理 87

リモート・キューイング 83

Channels スタンプ、qm.ini 238

fastpath 239

show channel details

(MONMQ) 436

show mask (MONMQ) 437

チャンネルの自動定義 90

チャンネルの動的定義 90

通知、イベントの 150

データベース・マネージャー

構成 156

構成情報の変更 169

スイッチ・ロード・ファイルの作
成 157

制約事項、データベース調整サポ
ート 156

接続 156

調整 154

データベース・マネージャーのイ
ンスタンス、削除 170

未確定の作業単位 165

dspmqtrn コマンド、未解決の

UOW の検査 165

データベース・マネージャー (続き)

qm.ini へのデータベース・マネ
ージャーの定義 157

rsvmqtrn コマンド、UOW の明示
再同期 167

データ変換 99

デフォルトのデータ変換 100

ユーザー定義のメッセージ形式
100

ConvEBCDICNewline 属性、
AllQueueManagers スタンプ
224

crtmqcvx コマンド 318

EBCDIC NL 文字の ASCII への
変換 224

ディレクトリー

キュー・マネージャー 112

許可 126

ディレクトリー構造 401

出口

クラスター・ワークロード 473

クラスター・ワークロード出口
22

チャンネル出口 22

ユーザー出口 22

デバッグ

共通プログラミング・エラー
252

予備的な検査 247

2 次検査 252

デフォルト

オブジェクト 19

オブジェクトの属性 60

キュー・マネージャー 37

誤った削除 322

誤った変更 41

処理されるコマンド 49

変更 41, 52

権限 ID、権限の 107

システム・オブジェクト 397

伝送キュー 38

デフォルト伝送キュー 97

デフォルトのデータ変換 100

デフォルト変数コマンド

(MONMQ) 434

伝送キュー
 キュー・マネージャー間での定義 14
 クラスタ 16
 作成 97
 説明 16, 84
 定義 88
 デフォルト 16, 38, 97
 名前の指定 95
 リモート管理 87
同期点、パフォーマンスの考慮事項 257
動的キュー 10
 許可 113
 説明 10
トラステッド・アプリケーション 475
 パインディング 476
トランザクション
 MQSeries コマンドの解決 361
 MQSeries の表示コマンド 341
トランザクション・サポート
 トランザクション・サポート 153
トリガー
 イベント、観測イベントとの比較 150
 イベント・キュー 151
障害モニター
 開始コマンド 367
メッセージ、開始キュー上の 15
モニター
 開始コマンド 376
 説明 16
 名前リストの使用 19
トリガー操作 479
 アプリケーション・キュー、定義 73
 定義 8
 のためのオブジェクトの管理 73
トリガー操作のためのオブジェクトの管理 73
取り出しアルゴリズム、メッセージの 10

トレース
 コンポーネントまたは関数の指定 447
 データ・サンプル 268
 パフォーマンスの考慮事項 268
 プロセスの開始時に開始 445
 disable trace (MONMQ) 450
 enable trace (MONMQ) 449
 exit trace (MONMQ) 457
 MONMQ セッションのサンプル 460
 MONMQ トレースの終了 457
 MONMQ の使用 431
 start trace (MONMQ) 446
 stop trace (MONMQ) 446

[ナ行]

名前
 オブジェクト 11
 オブジェクトに許可される 313
 最大文字数 11
名前サービス 219
名前リスト
 キュー・マネージャー・クラスタ
 ーが使用する 19
 説明 19
入出力のリダイレクト、MQSC コマ
ンドでの 48

[ハ行]

バイナリー
 トレース・バイナリー・ファイル
 のオープン 448
 トレース・バイナリー・ファイル
 のクローズ 448
ハイパーテキスト・マークアップ言
語 (HTML) 487
パインディング
 トラステッド・アプリケーション
 のための 476
パターン・マッチング・キーワー
ド、ルール・テーブル 136
発行、MQSeries コマンドの 47
パフォーマンス上の考慮事項
 同期点 257

パフォーマンス上の考慮事項 (続き)
 メッセージ長 256
 メッセージの持続性 256
 様々なメッセージ長 257
 CorrelId 257
 MQPUT1 の利点 258
 MsgId 257
パフォーマンスの考慮事項
 トレースを使用する場合 268
パフォーマンスのチューニング 277
パフォーマンス・イベント 150
非持続メッセージ 477
ヒストリー
 チャンネル中のメッセージの 438
 delete history (MONMQ) 450
 disable history (MONMQ) 450
 enable history (MONMQ) 450
 reset for channel (MONMQ) 451
 set history (MONMQ) 450
表示
 アクティブな MQSeries プロセス
 438
 キュー・マネージャーの属性 51
 許可コマンド 330
 コマンド・サーバーの状況 80
 コマンド・サーバー・コマンド
 335
 コンポーネントの 16 進 ID 443
 ターゲット・スレッド・スタック
 437
 プロセス定義 76
 メモリー・テーブル 442
MQSeries 定様式トレース出力コ
マンド 340
MQSeries トランザクション・コ
マンド 341
MQSeries ファイル・コマンド
337
プール割り当て量 279
ファイル
 キュー・マネージャー構成 223
 許可
 管理 129
 許可 130
 クラス 129
 全クラス 129

- ファイル (続き)
 - 内容 127
 - パス 126
 - 理解 125
- 構成
 - 問題判別での 267
 - 名前の理解 28
 - ログ制御 174
 - MQSeries 構成 223
- フィードバック、MQSC コマンドからの 50
- フェールオーバー
 - 説明 284
- フェールオーバー・コマンド 285, 290, 351
 - 関連コマンド 354
 - パラメーター 351
 - 戻りコード 353
 - 例 353
- フェールオーバー・セット
 - 管理 291
 - キュー・マネージャーの移動 293
 - キュー・マネージャーの始動 292
 - キュー・マネージャーの終了 293
 - 構成 286
 - 構成ファイル 285
 - 構成前のステップ 286
 - 構成例 301
 - コマンド・プロシージャ 289
 - 状態の表示 294
 - 状態の変更 298
 - 説明 283
 - トラブルシューティング 300
 - Intra Cluster Communication (ICC) の使用 299
 - MultiNet for OpenVMS の使用 301
- フェールオーバー・セットの管理 351
- フェールオーバー・セット・テンプレート 417
- フェールオーバー・モニター
 - 開始 292
- フェールオーバー・モニター (続き)
 - 監視プログラム 284
 - 説明 284
 - 停止 297
- フェイルバック
 - 説明 284
- 不正出力 258
- ブラウザ、キューの 63
- プリンシパル
 - アクセスの管理 106
 - 複数の権限 ID を保持している 106
- プログラミング・エラーの例 252
- プログラム、提供される例 413
- プログラム式コマンド形式 (PCF)
 - エスケープ PCF 28
 - 許可 117
 - コマンドの説明 27
 - セキュリティ要件 115
 - 属性 28
 - による管理 27
- プロセス
 - トレースの開始点 445
- プロセス権限の許可 106
- プロセス定義
 - 作成 75
 - 説明 18
 - 表示 76
- 分散キューイング
 - 送達不能キュー 17
 - 不正出力 260
 - 未配布メッセージ・キュー 17
- 別名
 - 応答先キュー 98
 - キュー・マネージャー 98
- 別名キュー 69
 - 許可 113
 - 説明 14
- 保護されるリソース 107
- 補足情報 479
- [マ行]**
- マクロ
 - MONMQ 459
- マスク
 - set mask (MONMQ) 451
- 未確定トランザクション
 - データベース・マネージャー 165
- 未配布メッセージ・キュー 131
- 無許可アクセスからの保護 103
- 無効化、イベントの 151
- 命名規則、各国語サポート 313
- メッセージ
 - オペレーター 263
 - 可変長 257
 - 観測イベントの 152
 - 記述子 8, 78
 - キューイング 7
 - キューに現れない 258
 - 最大長 9
 - 説明 8
 - 特定の検索 257
 - 取り出しアルゴリズム 10
 - パフォーマンスの考慮事項
 - 持続 256
 - 長さ 256
 - 非持続メッセージの速度 477
 - 未配布 267
 - 予期しない情報を含む 260
 - DOS および Windows クライアント上のエラー 276
- メッセージ長、短くする 63
- メッセージ・キューイング 7
- メッセージ・キュー・インターフェース 7
- メッセージ・チャンネル・エージェント (MCA)
 - 説明 84
 - AdoptNewMCA 属性 238
 - RETRY 状態のチャンネル 241
- メッセージ・ドリブ処理 7
- メディア・イメージ
 - 記録 187
 - 記録コマンド 355
 - 説明 186
 - リカバリー 187
- メディア・イメージのリカバリー 187
- メモリー・テーブル 442
- モデル・キュー
 - 説明 15

モデル・キュー (続き)
定義 72
取り扱い 71
戻りコード 248
値の解釈 317
フェールオーバー・コマンド
353
crtmqcvx コマンド 318
crtmqm コマンド 324
dlmqm コマンド 326
dspmqaut コマンド 333
dspmqcsv コマンド 335
dspmqfls コマンド 338
dspmqtrn コマンド 342
endmqcsv コマンド 343
endmqslr コマンド 345
endmqm コマンド 347
endmqtrc コマンド 349
rcdmqimg コマンド 356
rcrmqobj コマンド 359
rsvmqtrn コマンド 362
runmqchi コマンド 363
runmqchl コマンド 364
runmqfm コマンド 367
runmqslr コマンド 370
runmqsc コマンド 373
runmqtmc コマンド 375
runmqtrm コマンド 376
setmqaut コマンド 382
strmqcsv コマンド 384
strmqm コマンド 385
strmqtrc コマンド 389
モニター、キュー・マネージャーの
149
問題
そこからリカバリーする 185
リモート側での MQSC の使用
93
MQSC コマンドの実行 56
MQSC のローカルでの使用 56
問題判別 247
共通プログラミング・エラー
252
クライアント 275
構成ファイル 267
コマンドから応答がない 253

問題判別 247 (続き)
最初に検査すること 247, 251
さらに他の検査 252
トレース 268
不正出力
キューに現れないメッセージ
258
分散キューイング 260
予期しない情報を含むメッセ
ージ 260

[ヤ行]

ユーザー ID
許可 112
グループ nobody に属する 107
権限 104
プリンシパル 106
OpenVMS ログイン・ユーザー
112
ユーザー定義のメッセージ形式の変
換 100
ユーザー出口 473
クラスター・ワークロード 22,
473
説明 22
チャンネル出口 22
データ変換出口 22
ユーロ通貨記号サポート 395
優先キュー・マネージャー・シャッ
トダウン 43
要件
ソフトウェア 393
ハードウェア 393
用語、本書で使用する 491
用語集 491

[ラ行]

リカバリーの手順
キュー・マネージャーのオブジェ
クトの損傷 193
単独オブジェクトの損傷 193
ディスク・ドライブの障害 191
リスナー
トラステッド 477

リスナー (続き)
リスナーの実行 (runmqslr) コマン
ドの使用 369
リスナーの終了 (endmqslr) コマ
ンド 345
リソース
保護される 107
保護する理由 103
リニア・ロギング 175
リモート
管理 85
キュー
応答先キュー別名として 98
キュー・マネージャー別名と
しての 97
キューイング
勧告 92
キュー定義の作成 93
キュー・オブジェクトの取り扱い
97
セキュリティに関する考慮事項
115
MQSC コマンドの発行 91
リモート管理
オブジェクトの 83
コマンド・サーバー 80
初歩的な問題 93
リモート・キュー
許可 113
説明 13, 14
リモート・キューイング 83
理由コード 12, 131, 137
ルール・テーブル、DLQ ハンドラー
133
構文 139
処理 141
制御データ・エントリー 133
INPUTQ キーワード 134
INPUTQM キーワード 134
RETRYINT キーワード 134
WAIT キーワード 134
パターンと処理 (ルール) 135
ACTION キーワード 137
APPLIDAT キーワード 136
APPLNAME キーワード 136
APPLTYPE キーワード 136

ルール・テーブル、DLQ ハンドラー

133 (続き)

DESTQ キーワード 136
DESTQM キーワード 136
FEEDBACK キーワード 136
FORMAT キーワード 136
FWDQ キーワード 138
FWDQM キーワード 138
HEADER キーワード 139
MSGTYPE キーワード 136
PERSIST キーワード 137
PUTAUT キーワード 139
REASON キーワード 137
REPLYQ キーワード 137
REPLYQM キーワード 137
RETRY キーワード 139
USERID キーワード 137

例 144

例

エラー・ログ 264
共通プログラミング・エラー
252
伝送キューを作成する 97
フェールオーバー・コマンド
353
crtmqcvx コマンド 319
crtmqm コマンド 324
dlmqm コマンド 327
dspmqaut コマンド 334
dspmqcsv コマンド 335
dspmqfls コマンド 338
dspmqtrc コマンド 340
endmqcsv コマンド 344
endmqm コマンド 348
endmqtrc コマンド 349
mqs.ini ファイル、MQSeries for
Compaq OpenVMS 242
qm.ini ファイル 243
redmqimg コマンド 357
rcrmqobj コマンド 360
runmqfm コマンド 367
runmqfsr コマンド 370
runmqsc コマンド 373
setmqaut コマンド 383
strmqcsv コマンド 384
strmqm コマンド 386

例 (続き)

strmqtrc コマンド 389
ローカル管理 45
ローカル作業単位
定義 153
ローカル・キュー
開始 15
クリア 63
コマンド 17
削除 63
説明 12, 14
送達不能 17
定義のコピー 61
伝送 16
未配布メッセージ 17
ローカル・キューの定義 59

ログイン

循環 175
タイプ 175
チェックポイント 177
メディア・リカバリー 187
リニア 175

ログ

エラー 262
エラー、例 264
管理 182
キュー・マネージャー 174
形式 174
構成 233
内容のダンプ 194
パラメーター 39
ファイル
位置 185
再利用 177
制御 174
リカバリーのための使用 185
ログ・レコードのダンプ
(dmpmqlog コマンド) 194
dmpmqlog コマンドからの出力
195
Log スタンザ、qm.ini 233

ログ・ファイル 262

ロケール 427

論理装置

close (MONMQ) 436

論理装置 (続き)

show segment を使った表示
(MONMQ) 435

論理名、セキュリティの使用不可
108

[数字]

1 次グループの許可 106

1 次権限 ID、権限の 106

16 進 ID

コンポーネント用の表示 443

A

AdoptNewMCA 属性 238

AllQueueManagers スタンザ、

mqs.ini 224

amqsdlq、サンプル DLQ ハンドラー
132

analyse trace コマンド

(MONMQ) 454

APPLIDAT キーワード、ルール・テ
ーブル 136

APPLNAME キーワード、ルール・
テーブル 136

APPLTYPE キーワード、ルール・テ
ーブル 136

AUTHORIZE ユーティリティ 279

C

CCSID 427

キュー・マネージャーの再始動
101

データ変換 99

MQSeries for Compaq OpenVMS
によるサポート 395

ccsid.tbl 99

Channels スタンザ、qm.ini 238

ClientExitPath スタンザ、

mqs.ini 226

close binary コマンド

(MONMQ) 448

close LU コマンド (MONMQ) 436

close text コマンド (MONMQ) 449

connect コマンド (MONMQ) 446

CorrelId、使用時のパフォーマンスの
考慮事項 257
crtmqcvx コマンド
 パラメーター 318
 戻りコード 318
 例 319
crtmqm コマンド 320
 関連コマンド 325
 パラメーター 321
 戻りコード 324
 例 324

D

DCE
 キューの共用 219
 構成 220
 セキュリティー 23
 セル 219
DefaultQueueManager スタンザ、
 mq5.ini 226
delete history コマンド
 (MONMQ) 450
deselect index コマンド
 (MONMQ) 447
DESTQ キーワード、ルール・テーブ
 ル 136
DESTQM キーワード、ルール・テー
 ブル 136
Digital TCP/IP Services for
 OpenVMS 291
disable history コマンド
 (MONMQ) 450
disable timestamp コマンド
 (MONMQ) 449
disable tracing コマンド
 (MONMQ) 450
disconnect コマンド (MONMQ) 446
DLQ ハンドラー
 サンプル、amqsdlq 132
 呼び出し 131
 ルール・テーブル 133
dltmqm コマンド 326
 関連コマンド 327
 パラメーター 326
 戻りコード 326

dltmqm コマンド 326 (続き)
 例 327
DOS クライアントのエラー・メッセ
 ージ 276
dspmqaut コマンド 330
 関連コマンド 334
 使用 109, 111
 パラメーター 330
 戻りコード 333
 例 334
dspmqcsv コマンド 335
 関連コマンド 335
 パラメーター 335
 戻りコード 335
 例 335
dspmqfls コマンド 337
 パラメーター 337
 戻りコード 338
 例 338
dspmqtrc コマンド 340
 関連コマンド 340
 パラメーター 340
 例 340
dspmqtrn コマンド 341
 関連コマンド 342
 パラメーター 341
 戻りコード 342

E

EBCDIC NL 文字の ASCII への変換
 224
enable history コマンド
 (MONMQ) 450
enable timestamp コマンド
 (MONMQ) 449
enable tracing コマンド
 (MONMQ) 449
EndCommand プロシージャ
 テンプレート 421
endmqcsv コマンド 343
 関連コマンド 344
 パラメーター 343
 戻りコード 343
 例 344

endmqslr (リスナーの終了) コマンド
 形式 345
 パラメーター 345
 目的 345
 戻りコード 345
endmqm コマンド 42, 346
 関連コマンド 348
 パラメーター 346
 戻りコード 347
 例 348
endmqtrc コマンド 349
 関連コマンド 350
 パラメーター 349
 戻りコード 349
 例 349
exit コマンド (MONMQ) 457
Exitpath スタンザ、qm.ini 242
ExitProperties スタンザ、mq5.ini 226

F

failover.ini 構成ファイル 302
 編集 288
failover.template 302
fastpath チャンネル 239
FEEDBACK キーワード、ルール・テー
 ブル 136
FFST
 FFST (MONMQ) 456
FFST コマンド (MONMQ) 456
FFST、検査 269
FORMAT キーワード、ルール・テー
 ブル 136
FWDQ キーワード、ルール・テーブ
 ル 138
FWDQM キーワード、ルール・テー
 ブル 138

H

HEADER キーワード、ルール・テー
 ブル 139
HTML (ハイパーテキスト・マークア
 ップ言語) 487

I
INPUTQ キーワード、ルール・テーブル 134
INPUTQM キーワード、ルール・テーブル 134
Intra Cluster Communication (ICC) 299

L
LIKE 属性 61
ListenerBacklog 属性 241
Log スタンザ、qm.ini 233
LogDefaults スタンザ、mqs.ini 227
LU62 スタンザ、qm.ini 240

M
MAXMSGL 12
MaxMsgLength 属性 12
MODPARAMS.DAT ファイル 278
monmq ユーティリティー
概要 431
共用メモリーの管理 458
コマンド
カラーの設定 453
終了 457
出力の設定 453
選択 447
デフォルト変数 434
トレースの分析 454
ヒストリーの削除 450
close binary 448
close lu 436
close text 449
connect 446
deselect index 447
disable history 450
disable timestamp 449
disable trace 450
disconnect 446
enable history 450
enable timestamp 449
enable trace 449
FFST 456
onstartup start 445

monmq ユーティリティー (続き)
コマンド (続き)
onstartup stop 445
open 435
open binary 448
open text 448
set depth 450
set free 451
set mask 451
show channels 436
show components 443
show events 441
show functions 444
show globals 439
show history 438
show mask 437
show memory 442
show mutex 440
show process 438
show segment 435
show stack 437
trace start 432, 446
trace stop 446
MQSeries プロセスのトレース
トレース・セッションのサンプル 460
MONMQ 中のスクリプトとマクロ 459
MONMQ 内の変数 432
MQAI
説明 79
MQDATA 276
MQDLH、送達不能ヘッダー 131
MQI
キュー・マネージャー呼び出し 13
許可 117
説明 7
ローカル管理のサポート 45
MQM
権限 ID 104
ユーザー ID 112
MQOPEN 許可 117
MQPUT および MQPUT1、パフォーマンスの考慮事項 258
MQPUT 許可 117

MQSC
エスケープ PCF 28
大文字小文字の区別 31
行の最大長 54
コマンド 26
コマンド応答のタイムアウト 91
コマンドの確認 55
コマンドの使用 52
コマンドの発行方法 47
コマンド・ファイル
実行 55
出力レポート 54
入力 53
サンプル・ファイル 413
属性 28
対話式でのコマンドの発行 49
対話式での入力の終了 50
チャンネルでのセキュリティ要件 116
テキスト・ファイルからのコマンドの実行 52
入出力のリダイレクト 48
問題
リモート 93
ローカル 56
リモート側での発行 91
MQSC コマンド
大文字小文字の区別 31
行の最大長 54
使用 27
対話式での発行 49
ALTER QLOCAL 62
ALTER QREMOTE 96
DEFINE CHANNEL 89
DEFINE QALIAS 69
DEFINE QLOCAL 61
DEFINE QLOCAL LIKE 61
DEFINE QLOCAL REPLACE 62
DEFINE QMODEL 72
DEFINE QREMOTE 94
DELETE QLOCAL 63
DELETE QREMOTE 96
DISPLAY QREMOTE 96
MQSeries
権限 ID、MQM 104
OpenVMS の概要 393

MQSeries for Compaq OpenVMS の
概要 393
MQSeries 参考資料 485
MQSeries トレースの起動コマンド
387
MQSeries トレースの終了 349
MQSNOAUT 論理名 108
MQSPREFIX、環境変数 224
mq5.ini
定義 221
へのパス 56
編集 222
優先順位 222
AllQueueManagers スタンザ 224
ClientExitPath スタンザ 226
DefaultQueueManager スタンザ
226
ExitProperties スタンザ 226
LogDefaults スタンザ 227
QueueManager スタンザ 230
MQZAO 定数および許可 118
M5gId、使用時のパフォーマンスの考
慮事項 257
MSGTYPE キーワード、ルール・テ
ーブル 136
MultiNet for OpenVMS 301
テンプレート・ファイルの構成
291
mutex テーブル 440
MVS/ESA キュー・マネージャー 92

N

NL 文字、EBCDIC から ASCII への
変換 224
nobody、デフォルト権限 ID 107

O

OAM 105
onstartup start コマンド
(MONMQ) 445
onstartup stop コマンド
(MONMQ) 445
open binary コマンド
(MONMQ) 448
open text コマンド (MONMQ) 448

open コマンド (MONMQ) 435

OpenVMS

概要 393
必要なソフトウェア 393
必要なハードウェア 393
OpenVMS クラスタ 281
OpenVMS クラスタ・フェールオー
バー・セット 286
OpenVMS 権限 ID
デフォルト、nobody 107
MQM 104
OpenVMS ログイン・ユーザー
ID 112

Oracle

環境変数の設定、確認 159
構成 159
構成パラメーター、変更 163
最小サポート・レベル 159
スイッチ・ロード・ファイル、作
成 160
Oracle XA サポート、使用可能化
160
ORACLE_HOME、環境変数 159
ORACLE_SID、環境変数 159
XAResourceManager スタンザ、
qm.ini への追加 161

P

PCF コマンド

エスケープ PCF 78
MQAI、の使い方を単純化するた
めの使用 79
MQSC および PCF での属性 78
PCF を使用した管理用タスクを自
動化する 77

PDF (Portable Document Format) 487

PERSIST キーワード、ルール・テ ーブル 137

Portable Document Format (PDF) 487

PUTAUT キーワード、ルール・テ ーブル 139

Q

qm.ini 構成ファイル

定義 223
編集 222
優先順位 222
Channels スタンザ 238
Exitpath スタンザ 242
Log スタンザ 233
LU62 スタンザ 240
Service スタンザ 231
ServiceComponent スタンザ 232
TCP スタンザ 240
XAResourceManager スタンザ
236
QueueManager スタンザ、
mq5.ini 230
quit コマンド (MONMQ) 457

R

rcdmqimg コマンド 355

関連コマンド 357
パラメーター 355
戻りコード 356
例 357

rcrmqobj コマンド 358

関連コマンド 360
パラメーター 358
戻りコード 359
例 360

REASON キーワード、ルール・テ ーブル 137

REPLACE 属性、DEFINE コマンド 54

REPLYQ キーワード、ルール・テ ーブル 137

REPLYQM キーワード、ルール・テ ーブル 137

RETRY キーワード、ルール・テ ーブル 139

RETRYINT キーワード、ルール・テ ーブル 134

rsvmqtrn コマンド 361

関連コマンド 362
パラメーター 361
戻りコード 362

runmqchi コマンド 363
 パラメーター 363
 戻りコード 363
runmqchl コマンド 364
 パラメーター 364
 戻りコード 364
runmqdlq コマンド 131
runmqfm コマンド 284, 367
 関連コマンド 368
 パラメーター 351, 367
 戻りコード 367
 例 367
runmqlsr (リスナーの実行) コマンド
 形式 369
 パラメーター 369
 目的 369
 戻りコード 370
 例 370
runmqsc
 確認 55
 キュー・マネージャーの指定 52
 キュー・モード 91
 終了 50
 使用 52
 対話式での使用 49
 フィードバック 50
 問題 56
 MQSC コマンドの発行 47
runmqsc コマンド 371
 入出力のリダイレクト 48
 パラメーター 372
 戻りコード 373
 例 373
runmqtmc コマンド 375
 パラメーター 375
 戻りコード 375
runmqtrm コマンド 376
 パラメーター 376
 戻りコード 376

S

SAFEGUARD 394
select コマンド (MONMQ) 447
Service スタンザ、qm.ini 231

ServiceComponent スタンザ、
 qm.ini 232
set color コマンド (MONMQ) 453
set depth コマンド (MONMQ) 450
set free コマンド (MONMQ) 451
set mask コマンド (MONMQ) 451
set output コマンド (MONMQ) 453
setmqaut コマンド 377
 インストール可能サービス 111
 関連コマンド 383
 使用 109, 110
 パラメーター 379
 戻りコード 382
 例 383
show channels コマンド
 (MONMQ) 436
show components コマンド
 (MONMQ) 443
show events コマンド
 (MONMQ) 441
show functions コマンド
 (MONMQ) 444
show globals コマンド
 (MONMQ) 439
show history コマンド
 (MONMQ) 438
show mask コマンド (MONMQ) 437
show memory コマンド
 (MONMQ) 442
show mutex コマンド
 (MONMQ) 440
show processes コマンド
 (MONMQ) 438
show segment コマンド
 (MONMQ) 435
show stack コマンド (MONMQ) 437
StartCommand プロシージャー
 テンプレート 419
strmqcsv コマンド 384
 関連コマンド 384
 パラメーター 384
 戻りコード 384
 例 384
strmqm コマンド 385
 関連コマンド 386
 パラメーター 385

strmqm コマンド 385 (続き)
 戻りコード 385
 例 386
strmqtrc コマンド 387
 関連コマンド 389
 パラメーター 387
 戻りコード 389
 例 389
SupportPac 488
SYSS\$INPUT、runmqsc での 52
SYSS\$OUTPUT、runmqsc での 52

T

TCP スタンザ、qm.ini 240
TCP/IP
 構成 240
 サポートされるパッケージ 393
 接続が拒否された 241
 チャンネルの開始 89
 チャンネルのセットアップ 87
 リスナー・プロセス用の定義
 369
 リモート管理 20
Digital TCP/IP Services for
 OpenVMS 291
OpenVMS クラスタ操作の必須
 事項 285
OpenVMS クラスタのサポート
 283
StartCommand プロシージャーに
 より構成される 290
TidyCommand プロシージャー
 テンプレート 425
trace start コマンド (MONMQ) 446
trace stop コマンド (MONMQ) 446

U

USERID キーワード、ルール・テー
 ブル 137

W

WAIT キーワード、ルール・テーブ
 ル 134

Windows クライアントのエラー・メッセージ 276

X

XA 準拠リレーショナル・データベース 155

XAResourceManager スタンザ、qm.ini 236



部品番号: CT8YNJA

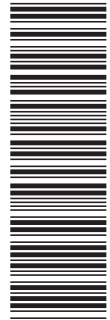
Printed in Japan

SC88-8942-00



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

(1P) P/N: CT8YNJA



Spine information:



MQSeries® for Compaq
OpenVMS Alpha®

MQSeries for Compaq OpenVMS
Alpha バージョン 5 リリース 1 システム
管理ガイド

バージョン 5
リリース 1