

OpenTP1 Version 7

OpenTP1 メッセージキューイング機能

TP1/Message Queue 使用の手引

解説・手引・文法・操作書

3000-3-D90-50

前書き

■ 対象製品

・適用 OS : Red Hat Enterprise Linux Server 6 (32-bit x86), Red Hat Enterprise Linux Server 6 (64-bit x86_64), Red Hat Enterprise Linux Server 7 (64-bit x86_64), Red Hat Enterprise Linux Server 8 (64-bit x86_64)

P-8164-C311 uCosminexus TP1/Message Queue 07-53

・適用 OS : Red Hat Enterprise Linux Server 6 (64-bit x86_64), Red Hat Enterprise Linux Server 7 (64-bit x86_64), Red Hat Enterprise Linux Server 8 (64-bit x86_64)

P-8264-C311 uCosminexus TP1/Message Queue (64) 07-53

・適用 OS : AIX V6.1, AIX V7.1, AIX V7.2, AIX 7.3

P-1M64-C381 uCosminexus TP1/Message Queue 07-53

P-1M64-D121 uCosminexus TP1/Message Queue (64) 07-53

・適用 OS : Windows 10, Windows 10 x64 Edition, Windows 11, Windows Server 2016, Windows Server 2019, Windows Server 2022

P-2464-C394 uCosminexus TP1/Message Queue 07-60

・適用 OS : Windows 10 x64 Edition, Windows 11, Windows Server 2016, Windows Server 2019, Windows Server 2022

P-2964-C334 uCosminexus TP1/Message Queue (64) 07-60

適用 OS : Red Hat Enterprise Linux Server 9 (64-bit x86_64)

P-8764-C311 uCosminexus TP1/Message Queue 07-60*

P-8864-C311 uCosminexus TP1/Message Queue (64) 07-60*

注※ この製品については、サポート時期をご確認ください。

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI, HiRDB, JP1, OpenTP1, OSAS, ServerConductor, TPBroker, uCosminexus, XDM, XMAP は、株式会社 日立製作所の商標または登録商標です。

AMD は、Advanced Micro Devices, Inc.の商標です。

IBM, AIX, MQSeries, WebSphere は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Intel は、Intel Corporation またはその子会社の商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標です。

Microsoft, Visual C++, Visual Studio, Windows, Windows Server は、マイクロソフト 企業グループの商標です。

Oracle(R), Java, MySQL 及び NetSuite は、Oracle, その子会社及び関連会社の米国及びその他の国における登録商標です。

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, and JBoss are registered trademarks of Red Hat, Inc. in the United States and other countries. Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.

UNIX は、The Open Group の登録商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ 発行

2024 年 1 月 3000-3-D90-50

■ 著作権

All Rights Reserved. Copyright (C) 2006, 2024, Hitachi, Ltd.

(C) Copyright International Business Machines Corporation 1999, 2002. All rights reserved.

変更内容

変更内容 (3000-3-D90-50) TP1/Message Queue 07-60

追加・変更内容	変更箇所
適用 OS に次の OS を追加した。 <ul style="list-style-type: none">• AIX 7.3• Red Hat Enterprise Linux Server 9 (64-bit x86_64)• Windows 11• Windows Server 2022	—
適用 OS から次の OS を削除した。 <ul style="list-style-type: none">• Windows 7• Windows 8• Windows 8.1• Windows Server 2008 R2• Windows Server 2012• Windows Server 2012 R2	—

単なる誤字・脱字などはお断りなく訂正しました。

変更内容 (3000-3-D90-40) TP1/Message Queue 07-53

追加・変更内容
次の適用 OS を追加した。 <ul style="list-style-type: none">• Red Hat Enterprise Linux Server 8 (64-bit x86_64)• Windows Server 2019
キューファイル二重化構成の説明を変更した。
クラスタ機能のリターンコードの説明を削除した。
ワークロード管理アルゴリズムの説明を追加した。
リポジトリ情報の保持期間の説明を変更した。
環境不正によるリポジトリ情報の有効期限切れの通知の説明を追加した。
リポジトリ情報送信契機のお知らせと定義の説明を追加した。
リポジトリ管理サーバの環境変数の形式の説明を追加した。
リポジトリ管理サーバの環境変数のオペランドの説明を追加した。
クラスタ運用時の注意事項を追加した。
mqaqueatl (モデルキューの属性定義) の機能の説明を変更した。
mqachgque コマンドの注意事項を追加した。

追加・変更内容
mqainit コマンドの注意事項を追加した。
mqamkque コマンドの注意事項を追加した。
mqrls (クラスタ情報の表示) コマンドに、問い合わせ情報の表示に関する説明を追加した。

変更内容 (3000-3-D90-31) TP1/Message Queue 07-52

追加・変更内容
適用 OS に HP-UX 11i V3 (IPF) を追加した。
キューファイル作成時の説明を変更した。
オブジェクトの属性の設定についての説明を変更した。
MQT サーバのトリガ起動によるチャンネルの開始の図について、次の説明を追加および変更した。 <ul style="list-style-type: none"> mqaprcdef (プロセス定義) mqaqueatl (モデルキューの属性定義)
mqrssp のサービス定義に関する注意事項について、次の説明を変更した。 <ul style="list-style-type: none"> message_buflen の計算式 message_store_buflen の計算式 また、message_buflen および message_store_buflen の計算式の記号の説明を追加した。
リポジトリ管理サーバに関連する定義の説明を追加した。
HP-UX の場合、MQC サーバ機能のインストール時に作成されるファイルを変更した。
チャンネル出口パラメタブロックの説明を変更した。
MQT 実行形式プログラムを作成するコマンドに -e オプションを追加した。
MQT 実行形式プログラムを C 言語で作成する例について、注記を変更した。
MQT 実行形式プログラムを C または C++ で作成するコマンドについての注※3 の説明を変更した。
次のオペランドの説明を変更した。 set 形式の MQA サービス定義 <ul style="list-style-type: none"> mqa_mqo_queuefile_alarm_level mqa_mqo_queue_alarm_level mqa_getwait_timeout
mqaquegrp (キューファイルグループの構成定義) の -u オプションの説明を変更した。
mqalsmsg コマンドの注意事項を変更した。
mqtstatrc コマンドの機能の説明を変更した。
mqtstptrc コマンドの機能の説明を変更した。
HP-UX 版の 32 ビット版の OpenTP1 を使用する場合の、共用メモリの見積もりに関する注意事項を変更した。

追加・変更内容

チャンネル経由でメッセージを送信する場合の図を変更した。

変更内容 (3000-3-D90-30) TP1/Message Queue 07-52

追加・変更内容

FAP レベル 4 について製品バージョンを変更した。

TCP/IP ソケット関数発行シーケンスに、ソケット属性に TCP_NODELAY を設定する場合の説明を追加した。

Windows の場合の MQC ゲートウェイサーバの作成ファイルを変更した。

MQT 実行形式プログラムの作成 (Windows) について、コマンドのオプションを変更した。また、Visual Studio 2013 を使用する場合の注意事項を追加した。

次のオプションの説明を変更した。

mqamqtnam (MQT デーモン構成定義)

- -b

MQT サービス定義の種類に mqt_tcpnodelay を追加した。

次のオペランドの説明を追加した。

set 形式の MQT サービス定義

- mqt_tcpnodelay

アプリケーション mqget のコーディング例 (C 言語) を変更した。

mqrls (クラスタ情報の表示) コマンドに、有効期限情報を表示するオプション (-u) を追加した。

MQA サーバ用の共用メモリの計算式を変更した。

MQC サーバ機能の共用メモリの見積もりの計算式、および凡例に動的共用メモリを追加した。

変更内容 (3000-3-D90-20) TP1/Message Queue 07-50

追加・変更内容

Message Queue - Operation に関する記載を削除した。

FAP レベル 4 について製品バージョンを変更した。

イニシエーションキューで必要な mqaqueatl 定義の説明を変更した。

フルリポジトリを保持するキューマネージャの選択の説明を追加した。

TP1/Message Queue の製品バージョンを変更した。

事前定義クラスタセンダチャンネルの接続先変更の説明を追加した。

MQT 実行形式プログラムの作成 (Windows) について説明を変更した。

定義の規則で、次の定義の説明を変更した。

- ロックサービス定義

追加・変更内容

- ネームサービス定義
- 旧製品の MQA サービス定義を使用する場合の注意事項

定義の規則で、次の定義の説明を追加した。

- システム環境定義

1 メッセージの最大レコード数の計算式を変更した。

- mqa_max_msg_recnum

MQA サービス定義の種類の説明を変更した。

- mqa_prf_trace_level

次のオペランド、またはオプションの説明を変更した。

set 形式の MQA サービス定義

- max_socket_descriptors
- mqa_mqo_conf
- mqa_mqo_queuefile_alarm_level
- mqa_mqo_queuefile_recover_level
- mqa_mqo_queue_alarm_level
- mqa_mqo_queue_recover_level
- mqa_msg_stay_time
- mqa_prf_trace_level

mqaquegrp (キューファイルグループの構成定義)

- -s

mqaquemgr (キューマネージャ定義)

- -t

mqtalccha (チャンネル定義)

- -b の bretryint

set 形式の MQT サービス定義

- max_socket_descriptors

MQT 通信構成定義の種類の説明を変更した。

mqttcp (TCP 構成定義)

- -t の disk と bufcnt

mqttcpcs (クラスタセンダ TCP 構成定義)

- -t の disk と bufcnt

mqttcpcr (クラスタレシーバ TCP 構成定義)

- -t の disk と bufcnt

次のオペランド、またはオプションの説明を変更した。

- -t の disk と bufcnt

次のオペランド、またはオプションの説明を変更した。

- -t の disk と bufcnt

追加・変更内容
次のオペランド，またはオプションの説明を変更した。 <ul style="list-style-type: none"> • -t の disk と bufcnt
旧製品の MQT 定義を使用する場合の注意事項に， TP1/Message Queue 07-50 以降の説明を追加した。
次のオペランド，またはオプションの説明を変更した。 <ul style="list-style-type: none"> • atomic_update
次のコマンドの説明を変更した。 <ul style="list-style-type: none"> • mqainit (キューファイルの割り当て)
共用メモリを見積もる際の注意事項の説明を変更した。
MQA サーバ用の共用メモリの計算式について変更した。
リポジトリ管理機能の共用メモリの見積もりの静的共用メモリ， および動的共用メモリの計算式を変更した。
MQC サーバ機能の共用メモリの見積もりの計算式， および凡例の説明を変更した。
MQT マネジャサーバの共用メモリの見積もりの計算式， および注記の説明を変更した。
ステータスファイルの使用容量とキー数 (MQA サーバ)， およびステータスファイルの使用容量とキー数 (MQC リスナサーバ) の計算式を変更した。
ジャーナルレコード長 (TP1/Message Queue) の計算式， および凡例の説明を変更した。
チェックポイントダンプファイルの使用容量 (TP1/Message Queue)， および MQC チェックポイントダンプファイルの使用容量 (TP1/Message Queue Access) の計算式を変更した。
キューファイルの使用容量の計算式を変更した。
次の用語解説について説明を変更した。 <ul style="list-style-type: none"> • FAP

変更内容 (3000-3-D90-10) TP1/Message Queue 07-01

追加・変更内容
メッセージシーケンス番号の遷移の図を変更した。
メッセージシーケンス番号のリセットコマンドを実行したあとのチャンネルの動作について，説明を追加した。
送信側 MCA の転送キューのメッセージについて説明を追加した。
メッセージシーケンス番号不一致が発生した場合について，説明を追加した。
時間監視機能で，時間監視の範囲が「TCP/IP コネクション確立時～開始要求受信時」と「MQT の終了処理開始～動作中チャンネルのメッセージ処理完了」の場合に，指定する定義を変更した。 また，時間監視の範囲が「TCP/IP コネクション確立時～開始要求受信時」の場合に，ネゴシエーションが続いている間は，時間監視の対象外となるという説明を追加した。
mqrsp, mqrsp で設定できる値についての注意事項を変更した。

追加・変更内容
64ビット版 TP1/Message Queue の場合について、説明を追加した。
MQC サーバ機能のインストール時に作成されるファイルで、Windows の場合のコマンドのファイル名を変更した。
MQC ゲートウェイサーバの環境設定での注意事項で、MQC ゲートウェイサーバのユーザサービス定義について、説明を追加した。
MQT 実行形式プログラムを作成するコマンドの説明を変更した。
MQT メイン関数オブジェクトファイルを作成するコマンドの説明を変更した。
<p>次の定義の説明を追加した。</p> <ul style="list-style-type: none"> • ネームサービス定義 <p>また、定義の規則で、次の定義の説明を変更した。</p> <ul style="list-style-type: none"> • ロックサービス定義 • トランザクションサービス定義 • チェックポイントダンプサービス定義
TP1/Message Queue の定義情報の作成手順で、MQT 通信構成定義のオブジェクトファイルでのバージョン互換性について、注意事項を追加した。
<p>次のオペランド、またはオプションの説明を変更した。</p> <p>set 形式の MQA サービス定義</p> <ul style="list-style-type: none"> • mqa_ioproc_num • mqa_expiry • mqa_maxque_expiry • mqa_maxmsg_expiry <p>mqtalccha (チャンネル定義)</p> <ul style="list-style-type: none"> • -v の vretry • -t の tcpsndbuf • -t の tcprcvbuf • -d の cnvccsid • -jn の sndjnl <p>mqcenv (MQC リスナサーバ通信環境定義)</p> <ul style="list-style-type: none"> • -r <p>MQC ゲートウェイサーバユーザサービス定義</p> <ul style="list-style-type: none"> • atomic_update <p>トランザクションサービス定義</p> <ul style="list-style-type: none"> • trn_crm_use
mqttbuf (バッファグループ定義) で、注意事項を追加した。
MQT 定義オブジェクトファイルの作成手順で、結合した MQT 通信構成定義のオブジェクトファイルの登録について、説明を追加した。
MQC サービス定義で、mqcgwpsnam (MQC ゲートウェイサーバ名定義) の指定数についての説明を変更した。

追加・変更内容
トランザクションサービス定義で、注意事項を変更した。
mqaadd コマンドで、出力メッセージに KFCA31050-E を追加した。
mqafilinf コマンドで、-q オプション、および-x オプションの説明を変更した。
mqarstr コマンドで、キューファイルの上書きについての注意事項を変更した。
mqcls コマンドで、出力形式の bb....bb (RESERVE, CONNECT) に説明を追加した。
mqclstrn コマンドで、出力形式の ee....ee に説明を追加した。
mqted コマンドで、出力形式の b1...b1 と d1...d1 に説明を追加した。
通信障害時のメッセージ送受信の流れについて、説明を追加した。
共用メモリを見積もる際の注意事項を変更した。
TP1/Message Queue のジャーナルの取得契機について、説明を追加した。
チェックポイントダンプファイルのサイズの見積もり式について、注意事項を追加した。
MQT トレースファイルのサイズの見積もり式に、UOC の情報を追加した。
MQ 管理情報ファイルのサイズの見積もり式を追加した。
TP1/Message Queue が出力するファイルの一覧を追加した。
適用 OS から Solaris を削除した。

はじめに

このマニュアルは、TP1/Message Queue の機能、操作、および運用について説明したものです。

TP1/Message Queue およびマニュアルは、米国 International Business Machines Corporation とのライセンス契約に基づき、IBM MQ (旧称: WebSphere MQ または MQSeries) の MQI, MQFAP, MQ クラスタの仕様をベースに実装しています。

■ 対象読者

TP1/Message Queue を使用するシステム管理者およびシステム設計者で、OpenTP1 システム (TP1/Server Base) の知識がある方を対象としています。また、次のマニュアルを理解されていることを前提としています。

- OpenTP1 解説 (3000-3-D50)
- OpenTP1 プログラム作成の手引 (3000-3-D51)
- OpenTP1 システム定義 (3000-3-D52)
- OpenTP1 運用と操作 (3000-3-D53)
- OpenTP1 プログラム作成リファレンス C 言語編 (3000-3-D54)
- OpenTP1 プログラム作成リファレンス COBOL 言語編 (3000-3-D55)
- OpenTP1 メッセージ (3000-3-D56)
- OpenTP1 使用の手引 Windows(R)編 (3000-3-D64)

なお、アプリケーションの設計方法と作成方法については、次のマニュアルに記載してありますので、あわせてお読みください。

- TP1/Message Queue プログラム作成の手引 (3000-3-D92)

アプリケーションのプログラミングインタフェースについては、次のマニュアルに記載してありますので、あわせてお読みください。

- TP1/Message Queue プログラム作成リファレンス (3000-3-D93)

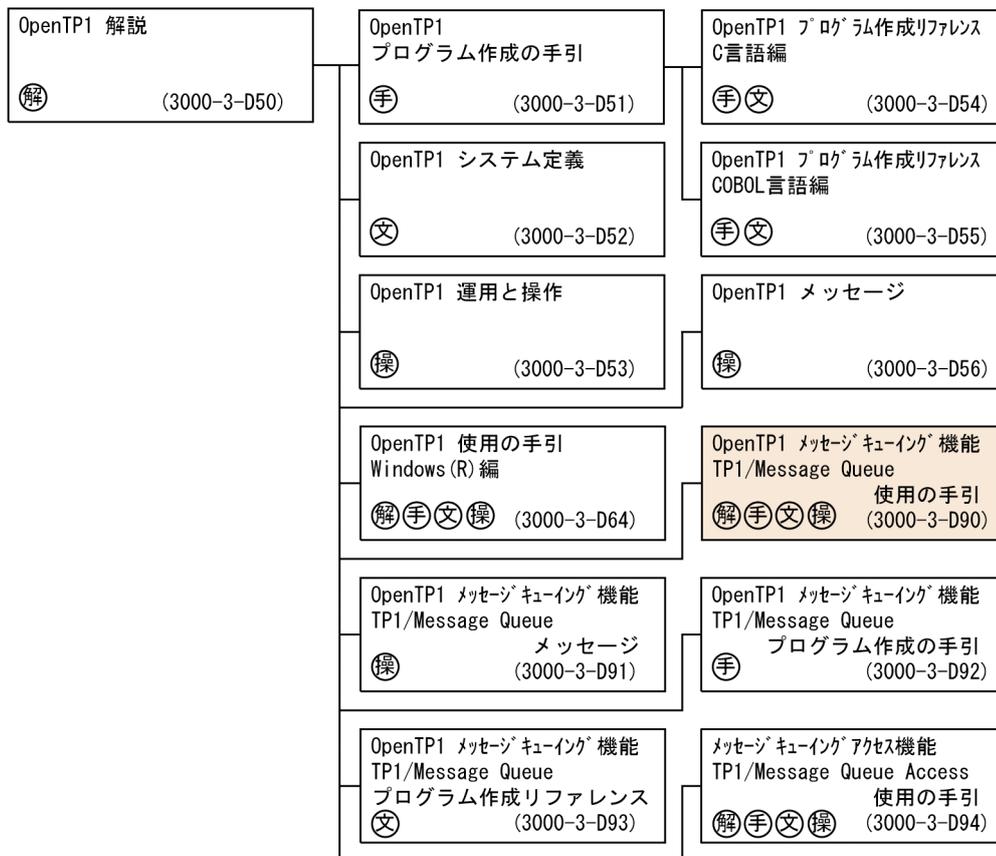
TP1/Message Queue および関連製品が出力するメッセージについては、次のマニュアルに記載してありますので、あわせてお読みください。

- TP1/Message Queue メッセージ (3000-3-D91)

その他、参照するマニュアルについては、関連マニュアルの欄を参照してください。

■ 関連マニュアル

●OpenTP1 Version 7



●その他の関連製品



<記号>

- ③ : 解説書
- ① : 手引書
- ② : 文法書
- ④ : 操作書

■ このマニュアルでの表記

(1) 製品名

このマニュアルで使用する製品名称の略称を次に示します。

名称	略称
AIX V6.1	AIX
AIX V7.1	
AIX V7.2	

名称	略称		
AIX 7.3	AIX		
COBOL2002 Net Developer	COBOL		
COBOL2002 Net Server Runtime			
COBOL2002 Net Server Runtime(64)			
COBOL2002 Net Server Suite			
COBOL2002 Net Server Suite(64)			
COBOL85			
HiRDB Server Version 9			
HiRDB Server Version 10			
HP-UX 11i V3 (IPF)	HP-UX(IPF)	HP-UX	
Itanium Processor Family	IPF		
J2EE™	J2EE		
JP1/Automatic Job Management System 2 - Agent	JP1/AJS - Agent	JP1/AJS	JP1
JP1/Automatic Job Management System 3 - Agent			
JP1/Automatic Job Management System 2 - Manager	JP1/AJS - Manager		
JP1/Automatic Job Management System 3 - Manager			
JP1/Automatic Job Management System 2 - View	JP1/AJS - View		
JP1/Automatic Job Management System 3 - View			
JP1/Automatic Job Management System 2 - Scenario Operation Manager	JP1/AJS2 - Scenario Operation Manager	JP1/AJS2 - Scenario Operation	
JP1/Automatic Job Management System 2 - Scenario Operation View	JP1/AJS2 - Scenario Operation View		
JP1/NETM/Audit - Manager	JP1/NETM/Audit		
JP1/NETM/DM Client	NETM/DM		
JP1/NETM/DM Manager			
JP1/NETM/DM SubManager			
JP1/ServerConductor/Deployment Manager	DPM		
Red Hat Enterprise Linux Server 6 (32-bit x86)	Linux (AMD64/ Intel EM64T/x86)	Linux	
Red Hat Enterprise Linux Server 6 (64-bit x86_64)			
Red Hat Enterprise Linux Server 7 (64-bit x86_64)			

名称	略称	
Red Hat Enterprise Linux Server 8 (64-bit x86_64)	Linux (AMD64/ Intel EM64T/x86)	Linux
Red Hat Enterprise Linux Server 9 (64-bit x86_64)		
uCosminexus TP1/Client/J	TP1/Client/J	TP1/Client
uCosminexus TP1/Client/P	TP1/Client/P	
uCosminexus TP1/Client/W	TP1/Client/W	
uCosminexus TP1/Client/W(64)		
uCosminexus TP1/Client for .NET Framework	TP1/Client for .NET Framework	Client .NET
uCosminexus TP1/Connector for .NET Framework	TP1/Connector for .NET Framework	Connector .NET
uCosminexus TP1/Server Base Enterprise Option	TP1/EE	
uCosminexus TP1/Server Base Enterprise Option(64)		
uCosminexus TP1/Extension 1	TP1/Extension 1	
uCosminexus TP1/Extension 1(64)		
uCosminexus TP1/FS/Direct Access	TP1/FS/Direct Access	
uCosminexus TP1/FS/Direct Access(64)		
uCosminexus TP1/FS/Table Access	TP1/FS/Table Access	
uCosminexus TP1/FS/Table Access(64)		
uCosminexus TP1/High Availability	TP1/High Availability	
uCosminexus TP1/High Availability(64)		
uCosminexus TP1/LiNK	TP1/LiNK	
uCosminexus TP1/Message Control	TP1/Message Control	
uCosminexus TP1/Message Control(64)		
uCosminexus TP1/Message Control/Tester	TP1/Message Control/Tester	
uCosminexus TP1/Message Control - Extension 1	TP1/Message Control - Extension 1	
uCosminexus TP1/Message Queue	TP1/Message Queue	
uCosminexus TP1/Message Queue(64)		
uCosminexus TP1/Message Queue Access	TP1/Message Queue Access	
uCosminexus TP1/Message Queue Access(64)		

名称	略称
uCosminexus TP1/Messaging	TP1/Messaging
uCosminexus TP1/Multi	TP1/Multi
uCosminexus TP1/NET/HDLC	TP1/NET/HDLC
uCosminexus TP1/NET/High Availability	TP1/NET/High Availability
uCosminexus TP1/NET/High Availability(64)	
uCosminexus TP1/NET/HSC	TP1/NET/HSC
uCosminexus TP1/NET/Library	TP1/NET/Library
uCosminexus TP1/NET/Library(64)	
uCosminexus TP1/NET/NCSB	TP1/NET/NCSB
uCosminexus TP1/NET/OSAS-NIF	TP1/NET/OSAS-NIF
uCosminexus TP1/NET/OSI-TP	TP1/NET/OSI-TP
uCosminexus TP1/NET/Secondary Logical Unit - TypeP2	TP1/NET/Secondary Logical Unit - TypeP2
	TP1/NET/SLU - TypeP2
uCosminexus TP1/NET/TCP/IP	TP1/NET/TCP/IP
uCosminexus TP1/NET/TCP/IP(64)	
uCosminexus TP1/NET/User Datagram Protocol	TP1/NET/UDP
uCosminexus TP1/NET/User Agent	TP1/NET/User Agent
uCosminexus TP1/NET/X25	TP1/NET/X25
uCosminexus TP1/NET/X25-Extended	TP1/NET/X25-Extended
uCosminexus TP1/NET/XMAP3	TP1/NET/XMAP3
uCosminexus TP1/Offline Tester	TP1/Offline Tester
uCosminexus TP1/Online Tester	TP1/Online Tester
uCosminexus TP1/Resource Manager Monitor	TP1/Resource Manager Monitor
uCosminexus TP1/Server Base	TP1/Server Base
uCosminexus TP1/Server Base(64)	
uCosminexus TP1/Shared Table Access	TP1/Shared Table Access
uCosminexus TP1/Web	TP1/Web
MS-DOS(R)	MS-DOS

名称	略称	
Windows(R) 10 Enterprise (x86)	Windows 10	Windows 10
Windows(R) 10 Pro (x86)		
Windows(R) 10 Enterprise (x64)	Windows 10 x64 Edition	
Windows(R) 10 Pro (x64)		
Windows(R) 11 Enterprise	Windows 11	
Windows(R) 11 Pro		
Windows Server(R) 2016 Datacenter	Windows Server 2016	
Windows Server(R) 2016 Standard		
Windows Server(R) 2019 Datacenter	Windows Server 2019	
Windows Server(R) 2019 Standard		
Windows Server(R) 2022 Datacenter	Windows Server 2022	
Windows Server(R) 2022 Standard		
Visual Studio(R) Professional 2013	Visual Studio 2013	Visual Studio
Visual Studio(R) Premium 2013		
Visual Studio(R) Ultimate 2013		
Visual Studio(R) Professional 2015	Visual Studio 2015	
Visual Studio(R) Enterprise 2015		
Visual Studio(R) Community 2015		
Visual Studio(R) Professional 2017	Visual Studio 2017	
Visual Studio(R) Enterprise 2017		
Visual Studio(R) Community 2017		
Visual Studio(R) Professional 2019	Visual Studio 2019	
Visual Studio(R) Enterprise 2019		
Visual Studio(R) Community 2019		
Visual Studio(R) Professional 2022	Visual Studio 2022	
Visual Studio(R) Enterprise 2022		
Visual Studio(R) Community 2022		

- AIX, HP-UX, および Linux を合わせて UNIX と表記することがあります。

- Windows 10, Windows 11, Windows Server 2016, Windows Server 2019 および Windows Server 2022 で機能差がない場合、Windows と表記することがあります。

(2) 適用 OS による違いについて

Windows 版の製品をご使用になる場合、マニュアルの記述を次のように読み替えてください。

項目	マニュアルの表記	読み替え
環境変数の表記	\$aaaaaa 例 \$DCDIR	%aaaaaa% 例 %DCDIR%
複数のパス名を列挙するときの区切り文字	:	;
ディレクトリの区切り文字	/	¥
完全パス名	ルートディレクトリから指定します。 例 /tmp	先頭にドライブ文字を付加して、ルートディレクトリから指定します。 例 C:¥tmp
実行形式ファイル名	ファイル名だけを指定します。 例 mcfmngrd	ファイル名に拡張子を付加して指定します。 例 mcfmngrd.exe
make コマンド	make	nmake

(3) インストールディレクトリのパスの違いについて

このマニュアルでは、OpenTP1 のインストールディレクトリを「/BeTRAN」と表記しています。インストールディレクトリは OS によって異なります。ご利用の OS に応じて、次の表のとおり読み替えてください。

このマニュアルでの表記	適用 OS ごとの読み替え		
	AIX, または HP-UX	Linux	Windows
/BeTRAN	/BeTRAN	/opt/OpenTP1	OpenTP1 をインストールしたディレクトリ

(4) JIS コード配列のキーボードと ASCII コード配列のキーボードとの違いについて

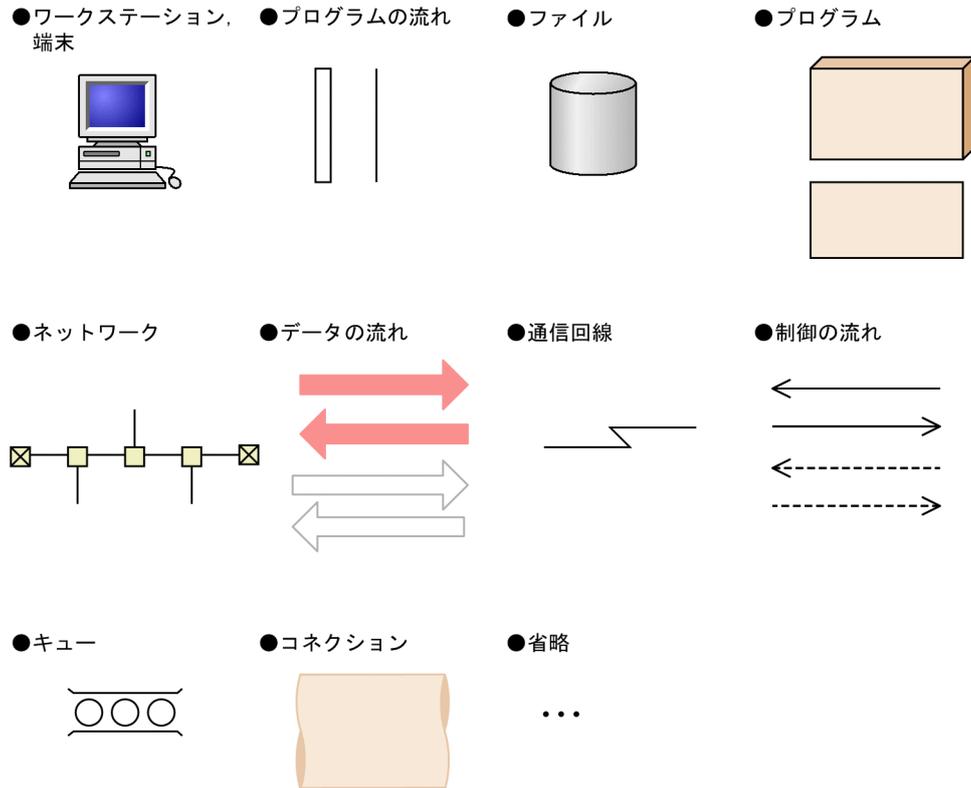
JIS コード配列と ASCII コード配列では、次に示すコードで入力文字の違いがあります。このマニュアルの文字入力例（コーディング例）の表記は、JIS コード配列（日本語のキーボード）に従った文字に統一しています。

コード	JIS コード配列	ASCII コード配列
(5c) ₁₆	'¥' (円記号)	'\' ' (バックスラッシュ)

コード	JIS コード配列	ASCII コード配列
(7e) ₁₆	' <u> </u> ' (オーバライン)	' <u> </u> ' (チルド)

■ 図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。



■ 文法の記号

このマニュアルで使用する各種の記号を説明します。

属性表示記号

ユーザ指定値の範囲などを説明する記号です。

属性表示記号	意味
~	この記号のあとにユーザ指定値の属性を示します。
《 》	ユーザが指定を省略したときの解釈値を示します。
< >	ユーザ指定値の構文要素を示します。
(())	ユーザ指定値の指定範囲を示します。

構文要素記号

ユーザ指定値の内容を説明する記号です。

構文要素記号	意味
英字	アルファベット (A~Z, a~z) と_ (アンダスコア)
英字記号	アルファベット (A~Z, a~z) と#, @, ¥
英数字	英字と数字 (0~9)
英数字記号	英字記号と数字 (0~9)
符号なし整数	数字列 (0~9)
16進数字	数字 (0~9) とアルファベット (A~F, a~f)
識別子	先頭がアルファベット (A~F, a~f) の英数字列
記号名称	先頭が英字記号の英数字記号列
文字列	任意の文字の配列
パス名	記号名称, / (スラント), および. (ピリオド) (ただし, パス名は使用する OS に依存)
OpenTP1 ファイル名	アルファベット (A~Z, a~z), 数字 (0~9), . (ピリオド), _ (アンダスコア), および@で構成される文字列 (最大 14 文字)
MQ 文字列	アルファベット (A~Z, a~z), 数字 (0~9), . (ピリオド), / (スラント), _ (アンダスコア), および% (パーセント) で構成される文字列

文法記述記号

記述形式を説明する記号です。

文法記述記号	意味
[]	この記号で囲まれている項目は省略してもよいことを示します。 (例) [-s MQT 通信プロセス識別子] -s オプションとそのオペランドを指定するか, 何も指定しないことを示します。
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。 (例) {server receiver} server と receiver のうち, どちらかを指定することを示します。
{{ }}	この記号で囲まれた複数の項目が一つの繰り返し項目の単位であることを示します。 (例) {{ mqttbuf }}

文法記述記号	意味
{ }	mqttbuf mqttbuf と指定できることを示します。
 (ストローク)	この記号で区切られた項目は選択できることを示します。 (例) disk = yes no yes または no を指定できることを示します。
...	この記号で示す直前の項目を繰り返し指定できることを示します。
△ (白三角)	空白を示します。 (例) コネクション ID1 △ コネクション ID2 コネクション ID1 とコネクション ID2 の間に、空白を 1 個入力することを示します。

■ 略語一覧

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字での表記
API	Application Programming Interface
CCSID	Coded Character Set Identifier
ESA	JP1/Cm2/Extensible SNMP Agent
FAP	Format and Protocols
FIFO	First-in-First-out
I/O	Input/Output
IPF	Itanium Processor Family
JIS	Japan Industrial Standard
JMS	Java Message Service
LAN	Local Area Network
LUW	Logical Unit of Work
MCA	Message Channel Agent
MCP	Message Channel Protocol
MDB	Message-driven Bean
MHP	Message Handling Program

英略語	英字での表記
MQA	Message Queue Access
MQC	TP1/Message Queue Access
MQI	Message Queue Interface
MQT	Message Queue Transfer
MTU	Maximum Transmission Unit
NTP	Network Time Protocol
OS	Operating System
PCF	Programmable Command Format
RPC	Remote Procedure Call
SNMP	Simple Network Management Protocol
SPP	Service Providing Program
SUP	Service Using Program
TCP/IP	Transmission Control Protocol/Internet Protocol
UAP	User Application Program
UOC	User Own Coding
VSAM	Virtual Storage Access Method

■ KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ 1,024 バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

■ その他の前提条件

このマニュアルをお読みになる際のその他の前提情報については、マニュアル「OpenTP1 解説」を参照してください。

目次

前書き	2
変更内容	4
はじめに	11

1 概要 29

1.1	TP1/Message Queue の特長	30
1.1.1	メッセージキューイングによる非同期通信を実現	30
1.1.2	幅広いプラットフォーム間での通信を実現	30
1.2	TP1/Message Queue の構成	31
1.3	TP1/Message Queue を使用したメッセージの送受信	33
1.4	MQA サービスで使用する共用メモリ	35
1.4.1	共用メモリの確保と解放	35
1.4.2	MQA サービス専用共用メモリの使用状況の表示	35
1.5	ソフトウェアの構成	36
1.5.1	TP1/Message Queue を組み込んだソフトウェア構成	36
1.5.2	Windows を使用する場合の注意事項	36
1.5.3	TP1/Message Queue の通信層の機能範囲	37
1.6	関連製品	39

2 機能 41

2.1	メッセージキューイング機能	42
2.1.1	メッセージキューイング機能の特長	42
2.1.2	メッセージキューイングの基本構成要素	42
2.1.3	アプリケーションからのメッセージ操作とキュー操作	46
2.1.4	MQI 命令使用時の OpenTP1 アプリケーションとの関係	46
2.1.5	トリガ機能	49
2.2	キューファイルの使用方法	51
2.2.1	キューファイルの構造	51
2.2.2	運用コマンドによるキューファイルの運用	53
2.3	チャンネルの機能	71
2.3.1	コーラとレスポンド	72
2.3.2	MCA のチャンネルタイプ	72
2.3.3	チャンネル状態	74
2.3.4	MQT サーバのトリガ起動によるチャンネル開始	81
2.3.5	チャンネルの開始	83

2.3.6	チャンネルの確立	86
2.3.7	チャンネル確立再試行	86
2.3.8	チャンネル確立時の TCP/IP ソケット関数発行シーケンス	96
2.3.9	チャンネルの解放	99
2.4	メッセージ送受信機能	101
2.4.1	メッセージの送信	101
2.4.2	メッセージの受信	103
2.4.3	メッセージとセグメントの関係	104
2.4.4	メッセージシーケンス番号でのメッセージの管理	105
2.4.5	メッセージのバッチ転送	113
2.5	時間監視機能	114
2.5.1	送受信監視	116
2.5.2	終了処理監視	116
2.5.3	チャンネル要求監視	117
2.5.4	イニシエーションキュー監視	117
2.5.5	転送キュー監視と切断時間間隔	117
2.5.6	バッチ終了待ちタイマ (バッチインターバル)	119
2.5.7	ハートビート機能	119
2.6	マルチキャスト機能	122
2.7	ファーストメッセージ機能	124
2.8	キューマネージャクラスタの機能	126
2.8.1	クラスタの概要	126
2.8.2	クラスタの使用によるシステム管理の軽減	130
2.8.3	クラスタのセットアップ	134
2.8.4	クラスタへの新しいキューマネージャの追加	138
2.8.5	クラスタの機能	140
2.8.6	ワークロード管理へのクラスタの使用	154
2.8.7	経路の選択	163
2.8.8	クラスタで使用する定義コマンド	164
2.8.9	クラスタ環境の通信構成	166
2.8.10	クラスタの管理	172
2.8.11	キューマネージャクラスタの準備	189
2.8.12	クラスタ運用時の注意事項	196
2.8.13	高度な作業	198
2.9	メッセージ編集出口 UOC を使用したファイル転送	211
2.9.1	処理の概要	211
2.9.2	使用方法	212
2.10	MQC サーバ機能	214
2.10.1	MQC サーバ機能のセットアップ	214

2.10.2 MQC サーバ機能の環境設定 214

2.10.3 MQC サーバ機能の運用 216

3 ユーザOWNコーディング 218

3.1 ユーザOWNコーディング 219

3.2 メッセージ編集出口 UOC 220

3.2.1 UOC の開始条件 220

3.2.2 UOC に渡されるデータ形式 220

3.3 UOC とのインタフェース 222

3.3.1 UOC の呼び出し形式 222

3.3.2 パラメタ 222

3.3.3 UOC パラメタ指定についての注意事項 244

3.4 UOC 呼び出しの流れ 245

3.5 UOC の組み込み 246

3.5.1 MQT メイン関数の作成 246

3.5.2 MQT 実行形式プログラムの作成 (UNIX) 247

3.5.3 MQT 実行形式プログラムの作成 (Windows) 249

3.5.4 システムサービス情報定義の作成 250

3.6 UOC の作成についての注意事項 252

3.6.1 UOC の異常処理 252

3.6.2 UOC が動作するプロセス 252

3.6.3 UOC のスタック領域 252

3.6.4 使用ライブラリの制限 252

3.6.5 バージョンアップ時の注意 252

3.6.6 UOC で使用できる関数 253

3.7 UOC の作成例 254

3.7.1 サンプル UOC の機能 254

3.7.2 サンプル UOC の所在 254

3.7.3 サンプル UOC のコーディング例 254

4 準備作業 265

TP1/Message Queue の環境作成手順 266

定義情報の作成手順 268

定義の記述形式 274

MQA サービス定義 276

set 形式の MQA サービス定義 282

mqaquegrp (キューファイルグループの構成定義) 297

mqaqueatl (モデルキューの属性定義) 303

mqaremque (リモートキューのローカル定義) 312

mqaalsque (別名キューの属性定義) 315

mqaprcdef (プロセス定義) 318

mqaquemgr (キューマネージャ定義) 321
mqamqtnam (MQT デーモン構成定義) 325
MQT サービス定義 328
set 形式の MQT サービス定義 329
MQT 通信構成定義 331
mqttenv (MQT 環境定義) 341
mqtttim (タイマ定義) 342
mqttbuf (バッファグループ定義) 343
mqttcp (TCP 構成定義) 346
mqttcps (クラスタセンダ TCP 構成定義) 351
mqttcpcr (クラスタレシーバ TCP 構成定義) 354
mqtalcccha (チャンネル定義) 359
デフォルトチャンネル定義 381
各チャンネルタイプに指定できるオプション 383
チャンネルのネゴシエーション 388
mqtalced (チャンネル終了定義) 390
MQT 定義オブジェクトファイルの作成手順 391
MQT 定義オブジェクト生成ユーティリティ 396
MQT 定義結合ユーティリティ 399
MQC サービス定義 401
mqcenv (MQC リスナサーバ通信環境定義) 403
mqcgwpsnam (MQC ゲートウェイサーバ名定義) 406
MQC ゲートウェイサーバユーザサービス定義 408
トランザクションサービス定義 410
システム構成例 412

5 開始と終了 438

- 5.1 TP1/Message Queue の開始 439
 - 5.1.1 正常開始 439
 - 5.1.2 再開始 439
- 5.2 TP1/Message Queue の終了 441
- 5.3 開始時のチャンネル情報引き継ぎ 443
 - 5.3.1 チャンネル管理情報格納ファイル 443
 - 5.3.2 チャンネル情報を引き継がない条件 444
 - 5.3.3 注意事項 445

6 システムの運用 447

運用コマンドの概要 448
TP1/Message Queue の運用コマンド 450
mqaadd (キューファイルの組み込み) 453
mqabkup (キューファイルのバックアップ) 458
mqachgque (キューの属性の変更) 459
mqaconvert (キューファイルのコンバート) 462
mqadel (キューファイルの削除) 464

mqadelmsg (メッセージの削除) 465
 mqadelque (キュー, メッセージの削除) 467
 mqadf (キューファイルグループの使用状況表示) 469
 mqafilinf (キューファイル内の情報表示) 472
 mqafills (キューファイルの情報表示) 478
 mqafrc (キューファイルの回復) 481
 mqagrpobs (キューファイルグループ内のキューファイル状態変更) 485
 mqainit (キューファイルの割り当て) 487
 mqainq (オブジェクトの属性表示) 490
 mqalsmsg (メッセージの表示) 501
 mqamkque (キューの作成) 507
 mqarcvr (ファイル二重化の回復) 509
 mqarles (キューファイルの閉塞解除) 511
 mqarm (キューファイルの切り離し) 513
 mqarstr (キューファイルのリストア) 516
 mqaset (オブジェクト属性の設定) 518
 mqccontrn (MQC トランザクション状態の変更) 520
 mqcls (MQC ゲートウェイサーバの状態表示) 523
 mqclstrn (MQC トランザクション状態の表示) 526
 mqcswptrc (MQC トレースの強制スワップ) 529
 mqrls (クラスタ情報の表示) 530
 mqrrefresh (リポジトリ情報の再作成) 546
 mqrremove (クラスタからの脱退) 550
 mqrswptrc (MQR トレースの強制スワップ) 552
 mqaltcha (チャンネル属性の変更) 553
 mqtdmped (MQT 通信プロセス情報の表示) 557
 mqted (MQT トレースファイルの編集出力) 567
 mqtlscha (チャンネルの状態表示) 576
 mqtpngcha (チャンネルのテスト接続) 587
 mqtrlvcha (メッセージ送達未確認の解決) 590
 mqtrstcha (メッセージシーケンス番号のリセット) 593
 mqststacha (チャンネルの開始) 597
 mqststatrc (MQT トレースのディスク出力機能の開始) 601
 mqststpcha (チャンネルの終了) 603
 mqststptrc (MQT トレースのディスク出力機能の終了) 606
 mqtswptrc (MQT トレースの強制スワップ) 608

7 障害対策 610

- 7.1 ファイル障害 611
 - 7.1.1 キューおよびキューファイルの障害とユーザの処理 611
 - 7.1.2 メッセージ送受信中の障害 615
 - 7.1.3 イニシエーションキューの障害 616
 - 7.1.4 キューファイルの入出力エラー 616
 - 7.1.5 クラスタ環境のシステムキューの障害 625

- 7.2 通信障害 636
 - 7.2.1 コネクション障害 (コーラ側) 636
 - 7.2.2 コネクション障害 (レスポンド側) 637
 - 7.2.3 MCP 障害 (チャンネル確立時の障害) 638
 - 7.2.4 MCP 障害 (チャンネル確立後, 自システムで検出した MQ プロトコル障害) 639
 - 7.2.5 MCP 障害 (チャンネル確立後, 相手システムで検出した MQ プロトコル障害) 639
 - 7.2.6 メモリ不足 640
 - 7.2.7 MCP 障害 (メッセージ送信時の転送障害) 641
 - 7.2.8 MCP 障害 (メッセージ受信時の転送障害) 642
 - 7.2.9 MCP 障害 (デッドレターキュー障害) 644
 - 7.2.10 イニシエーションキュー障害 645
 - 7.2.11 時間監視障害 645
 - 7.2.12 出口障害 647
 - 7.2.13 OpenTP1 システム, および MQ システム障害 647
 - 7.2.14 チャンネル管理情報格納ファイル障害 648
 - 7.2.15 トリガ起動処理障害 649
 - 7.2.16 MCA 数オーバ 650
 - 7.2.17 通信障害時のメッセージ送受信の流れ 650
- 7.3 ファーストメッセージ障害 652
 - 7.3.1 コネクション障害 (コーラ側・ファーストメッセージ) 652
 - 7.3.2 コネクション障害 (レスポンド側・ファーストメッセージ) 652
 - 7.3.3 MCP 障害 (チャンネル確立後, 自システムで検出した MQ プロトコル障害・ファーストメッセージ) 653
 - 7.3.4 MCP 障害 (チャンネル確立後, 相手システムで検出した MQ プロトコル障害・ファーストメッセージ) 654
 - 7.3.5 MCP 障害 (メッセージ送信時の転送障害・ファーストメッセージ) 654
 - 7.3.6 MCP 障害 (メッセージ受信時の転送障害・ファーストメッセージ) 655
 - 7.3.7 MCP 障害 (デッドレターキュー障害・ファーストメッセージ) 656
 - 7.3.8 時間監視障害 (ファーストメッセージ) 657
 - 7.3.9 出口障害 (ファーストメッセージ) 658
 - 7.3.10 OpenTP1 システム, および MQ システム障害 (ファーストメッセージ) 659
- 7.4 障害時に取得する情報 660
 - 7.4.1 MQT トレースファイルの出力 661
 - 7.4.2 MQR トレースファイルの出力 663

付録 665

- 付録 A 共用メモリの見積もり式 666
 - 付録 A.1 共用メモリを見積もる際の Σ の計算例および注意事項 666
 - 付録 A.2 MQA サーバ用の共用メモリの見積もり式 667
 - 付録 A.3 リポジトリ管理機能の共用メモリの見積もり式 670

付録 A.4	MQC サーバ機能の共用メモリの見積もり式	672
付録 A.5	MQT マネージャサーバの共用メモリの見積もり式	673
付録 B	ファイルの見積もり式	675
付録 B.1	ステータスファイルのサイズの見積もり式	675
付録 B.2	ジャーナルファイルのサイズの見積もり式	676
付録 B.3	チェックポイントダンプファイルのサイズの見積もり式	682
付録 B.4	キューファイルのサイズの見積もり式	684
付録 B.5	チャンネル管理情報格納ファイルのサイズの見積もり式	684
付録 B.6	スナップダンプファイルのサイズの見積もり式 (チャンネルの状態表示コマンド)	685
付録 B.7	MQC トレースファイルのサイズの見積もり式	686
付録 B.8	MQR トレースファイルのサイズの見積もり式	686
付録 B.9	MQT トレースファイルのサイズの見積もり式	686
付録 B.10	ソケット用ファイル記述子のサイズの見積もり式 (ユーザサーバで使用)	689
付録 C	系切り替え機能使用時の注意	691
付録 D	TP1/Message Queue が出力するファイル一覧	692
付録 E	用語解説	697

索引 703

1

概要

TP1/Message Queue は、OpenTP1 システムに組み込まれてメッセージキューイング機能を実現するプログラムです。この章では、TP1/Message Queue の概要を説明します。

1.1 TP1/Message Queue の特長

TP1/Message Queue は、OpenTP1 システムに組み込まれて動作するプログラムです。TP1/Message Queue を使用すると、IBM MQ に代表される MQ システムのアプリケーションとメッセージの送受信ができます。MQ システムは、メッセージキューイング機能のキューマネージャがあるシステムのことです。

このマニュアルでは、TP1/Message Queue の概要、機能、および使用方法について説明します。メッセージキューイング機能の詳細およびアプリケーションの作成方法については、マニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

次に TP1/Message Queue の特長を説明します。

1.1.1 メッセージキューイングによる非同期通信を実現

メッセージの送受信は、メッセージキューイング機能によって実現されます。メッセージキューイング機能とは、メッセージキューと呼ばれるキューを使用してメッセージを送受信する機能です。

メッセージを一時的に蓄積するメッセージキューを介してメッセージの送受信を行うので、非同期で通信できます。すなわち、送信する側は、通信相手の状況を意識しないで、メッセージを送信できます。また、受信する側は、メッセージを受信したい時だけアプリケーションを起動すればよいので、コンピュータの資源を有効に利用できます。

1.1.2 幅広いプラットフォーム間での通信を実現

TP1/Message Queue を使うと、メインフレーム、サーバ、ワークステーション、およびパーソナルコンピュータとの間で通信できます。

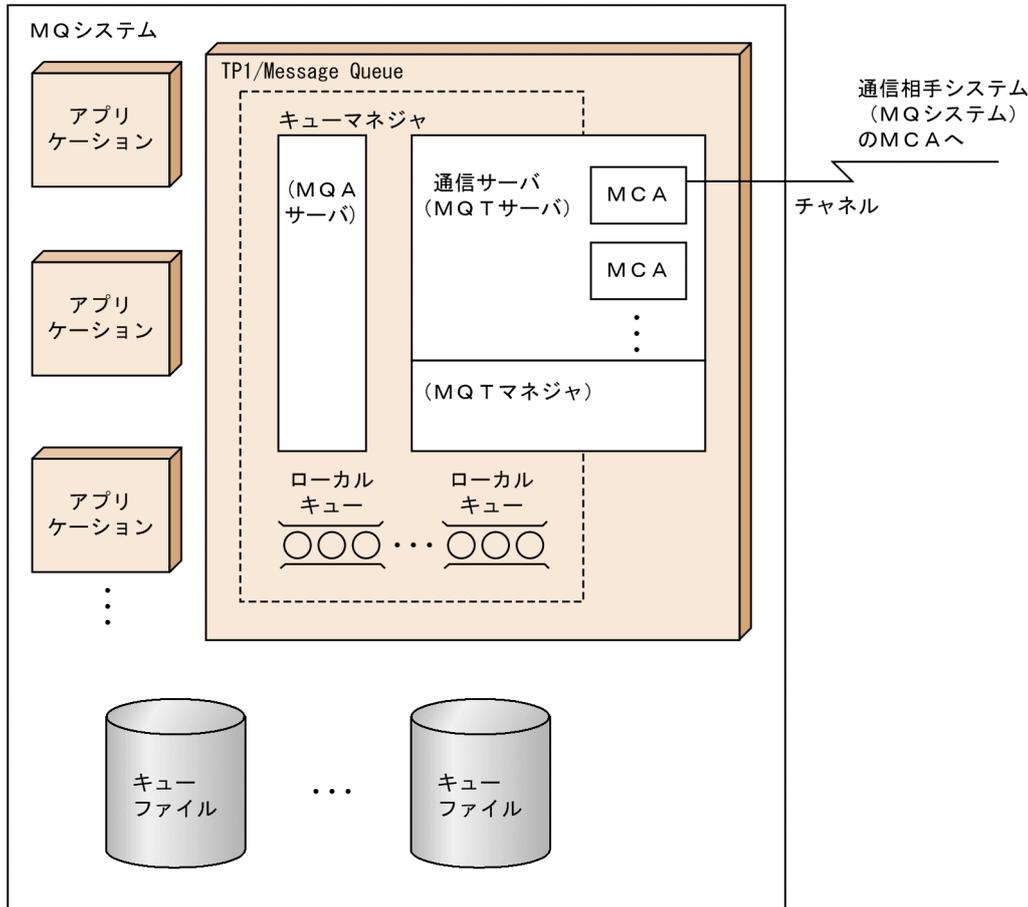
TP1/Message Queue によって、メインフレームを中心とした基幹業務システムと、ワークステーション・パーソナルコンピュータを中心としたオープンシステムとがメッセージベースで連携できます。

1.2 TP1/Message Queue の構成

TP1/Message Queue を組み込んだシステムを MQ システムといます。

MQ システムの構成を、次の図に示します。

図 1-1 MQ システムの構成



MQ システムは、キューを管理する部分とメッセージを送受信する部分に分けられます。メッセージを送受信する部分を通信サーバといい、一つの MQ システムに複数指定できます。また、通信サーバには、通信相手システムとの通信路を制御するメッセージチャンネルエージェント (MCA) が含まれます。自システムの MCA と通信相手システムの MCA をつなぐ通信経路をチャンネルといいます。

メッセージキューを管理してメッセージを送受信するプログラムをキューマネージャ[※]といいます。TP1/Message Queue は、OpenTP1 システム上でキューマネージャの役割を持ちます。

キューマネージャの中で、特に、キューを管理する部分を MQA サーバ (MQA)、通信サーバを MQT サーバ (MQT) と呼びます。MQT サーバの開始と終了を制御する部分を MQT マネージャと呼びます。

アプリケーションは、目的の業務に応じてユーザが作成するプログラムです。ユーザはアプリケーションを作成して、メッセージキューイング機能を利用します。

送受信するメッセージは、キューファイルという物理的なファイルに格納されます。

注※

マニュアル「TP1/Message Queue プログラム作成の手引」および「TP1/Message Queue プログラム作成リファレンス」で使用されているキューマネージャは、次に示す 2 とおりの意味で使われています。

- TP1/Message Queue の全体 (MQA サーバ, MQT サーバ, および MQT マネージャ)
- MCA を除く TP1/Message Queue

1.3 TP1/Message Queue を使用したメッセージの送受信

TP1/Message Queue を使用すると、OpenTP1 システム内のアプリケーション同士および他システムのキューマネージャとの間でメッセージの送受信ができます。送受信ができるキューマネージャについては、「1.5.1 TP1/Message Queue を組み込んだソフトウェア構成」の通信相手システムを参照してください。これらのキューマネージャと通信する場合、TCP/IP プロトコルを使用します。

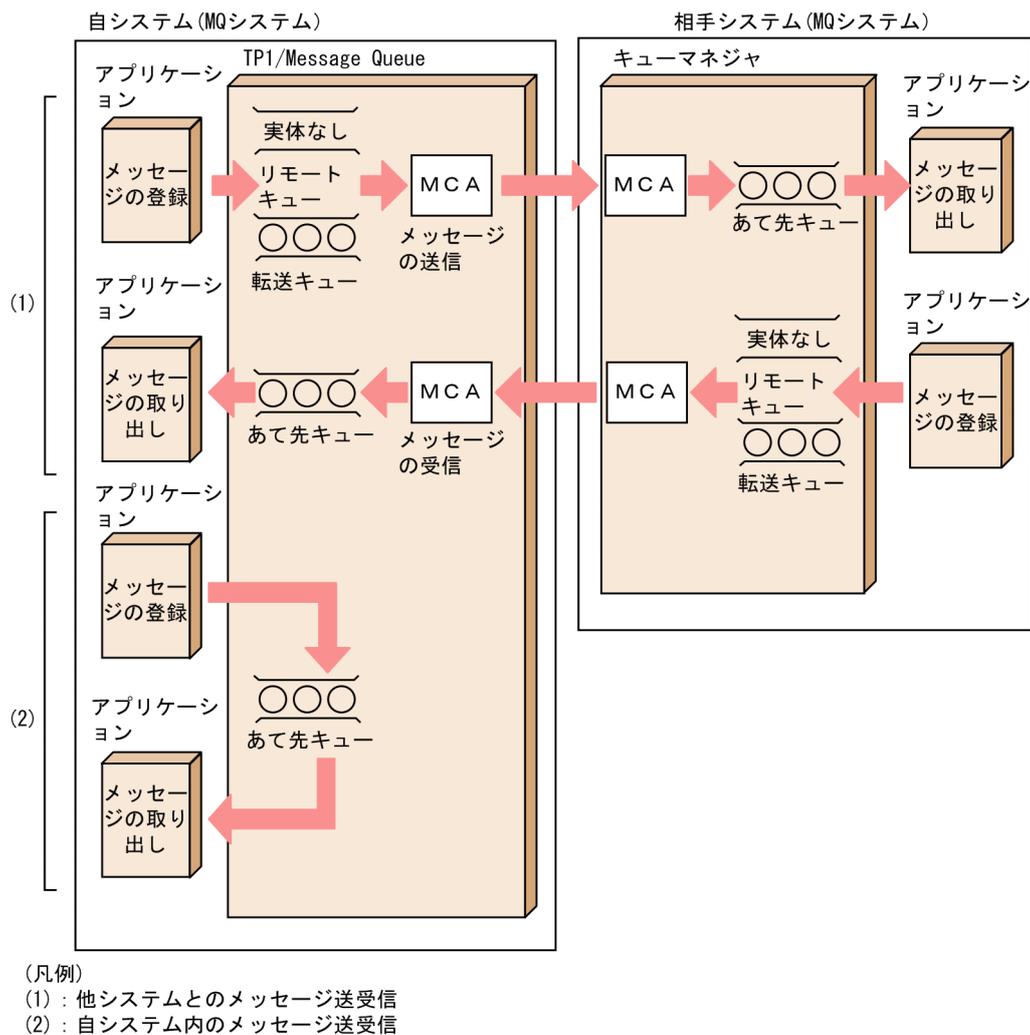
ユーザは、アプリケーションを作成して、キューにメッセージを登録したりキューからメッセージを取り出したりします。キューの間のメッセージ転送は、キューマネージャによって行われます。

TP1/Message Queue を使用したメッセージ送受信の特長を次に示します。

- 一度に大量のデータを送受信できます。
- メッセージはいったんキューに登録されるので、メッセージの送受信中にアプリケーションが動作している必要はありません。

メッセージ送受信の概要について、次の図に示します。

図 1-2 メッセージ送受信の概要



自システムのアプリケーションが、自システムのリモートキューにメッセージを登録すると、キューマネージャは転送キューヘッダ（MQXQH 構造体）をメッセージに付加し転送キューに格納します。

MCA は転送キューに登録されたメッセージを、TCP/IP プロトコルを使用して相手システムの MCA に送信します。受信側の MCA は、転送キューヘッダ（MQXQH 構造体）に指定されたあて先キューにメッセージを登録します。

1.4 MQA サービスで使用する共用メモリ

MQA サービスで使用する共用メモリについて説明します。

1.4.1 共用メモリの確保と解放

TP1/Message Queue では、OpenTP1 のシステム環境定義で指定する静的共用メモリと動的共用メモリの一部を使用します。さらに、システム環境定義で指定した共用メモリとは別に、MQA サービス専用共用メモリ（MQA サービス専用共用メモリの固定領域）を TP1/Message Queue の開始時に確保します。MQA サービス専用共用メモリの固定領域は、MQA サービス定義の `mqaquegrp` 定義コマンドで指定されたキューファイルで使用する共用メモリのサイズ分を確保します。

また、TP1/Message Queue の開始時には、`mqaadd` コマンドでオンライン中にキューファイルを組み込むための、組み込むキューファイルが使用する MQA サービス専用共用メモリを、MQA サービス専用共用メモリの固定領域とは別の領域（MQA サービス専用共用メモリの拡張領域）に確保します。

確保した MQA サービス専用共用メモリは、TP1/Message Queue の終了時に解放されます。

1.4.2 MQA サービス専用共用メモリの使用状況の表示

OpenTP1 の稼働中に使用している MQA サービス専用共用メモリの状態を知りたい場合は、`dcshmls` コマンド（`-r` オプションを指定）を実行します。`dcshmls` コマンドを実行すると、現在使用している MQA サービス専用共用メモリの合計値や MQA サービス専用共用メモリの使用率などを表示できます。`dcshmls` コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

1.5 ソフトウェアの構成

TP1/Message Queue のソフトウェア構成を示します。

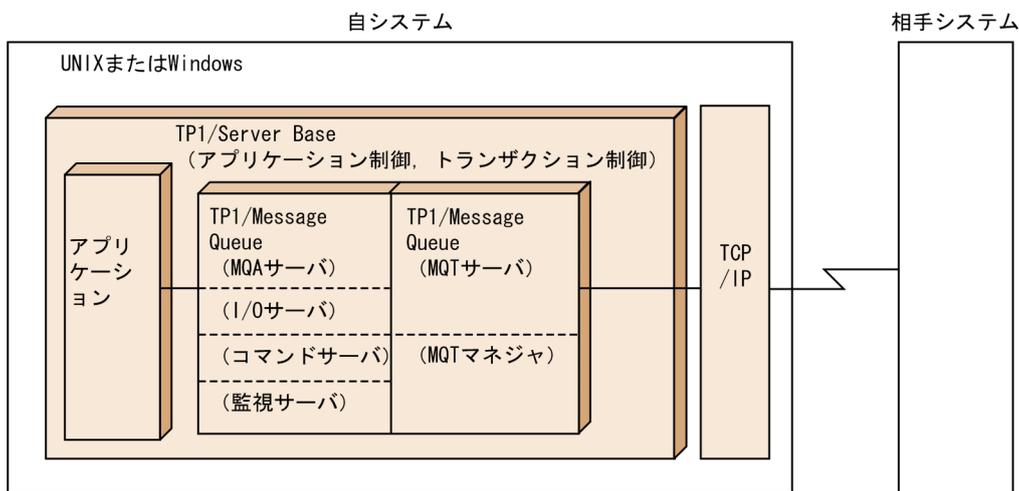
1.5.1 TP1/Message Queue を組み込んだソフトウェア構成

TP1/Message Queue は、OpenTP1 システムに組み込まれて動作するプログラムです。TP1/Message Queue は、TP1/Server Base 上で動作します。

自システムのアプリケーションが、相手システムのキューに関連づけられたキューにメッセージを登録すると、TP1/Message Queue は、TCP/IP プロトコルを使用して相手システムにメッセージを送信します。相手システムは指定されたキューにメッセージを登録します

TP1/Message Queue を組み込んだソフトウェア構成の例を次の図に示します。

図 1-3 TP1/Message Queue を組み込んだソフトウェア構成の例



TP1/Message Queue の通信相手システムを次に示します。

- IBM MQ
- VOS3 XDM/Q を組み込んだ XDM E2 系のシステム
- VOS3 TMS-4V/SP/Message Queue
- TP1/Message Queue を組み込んだ OpenTP1 システム

1.5.2 Windows を使用する場合の注意事項

Windows の TP1/Message Queue を使用する場合、次に示す注意事項があります。

- Windows の環境変数の表記は、\$DCDIR ではなく%DCDIR%になります。

- UNIX 版ではコロン (:) が複数のパス名の区切り文字ですが、Windows ではコロンはドライブ名とディレクトリ名との区切り文字になります。Windows ではパス名の区切り文字にセミコロン (;) を使用してください。
- パス名を完全パスで指定する際には、必ずドライブ名を記述してください。
- COBOL の API を使用する場合など、OpenTP1, Windows, または Visual C++ が提供していない DLL を使用する際には、prcpath コマンド、またはプロセスサービス定義の prcsvpath 定義コマンドを使用して DLL のあるディレクトリを指定してください。
- OpenTP1 は大文字と小文字を区別します。コマンドのオプションや、定義ファイルに記述した文字列をコマンド引数で使用するような場合は、注意してください。
例えば、-a オプションと-A オプションでは異なるオプションを表します。
- 環境変数 DCDIR および環境変数 DCCONFPATH を設定する場合、そのコンピュータ内ですべて同じ文字列になるよう設定してください。また、大文字と小文字を区別してください。
- ファイル名を指定するときは、ドライブ名以外にコロン (:) を含んだファイル名を指定しないでください。
- OpenTP1 ホームディレクトリ配下のファイル (%DCDIR%で指定したディレクトリ) を圧縮しないでください。このディレクトリを圧縮した場合、OpenTP1 は正常に動作しません。

1.5.3 TP1/Message Queue の通信層の機能範囲

TP1/Message Queue は、TCP/IP プロトコル上のメッセージチャネルプロトコル (MCP) を使用しています。通信層での TP1/Message Queue の機能の範囲について、次の図に示します。

図 1-4 通信層の機能範囲

TP1/Message Queueの層構造 OSI 7層構造

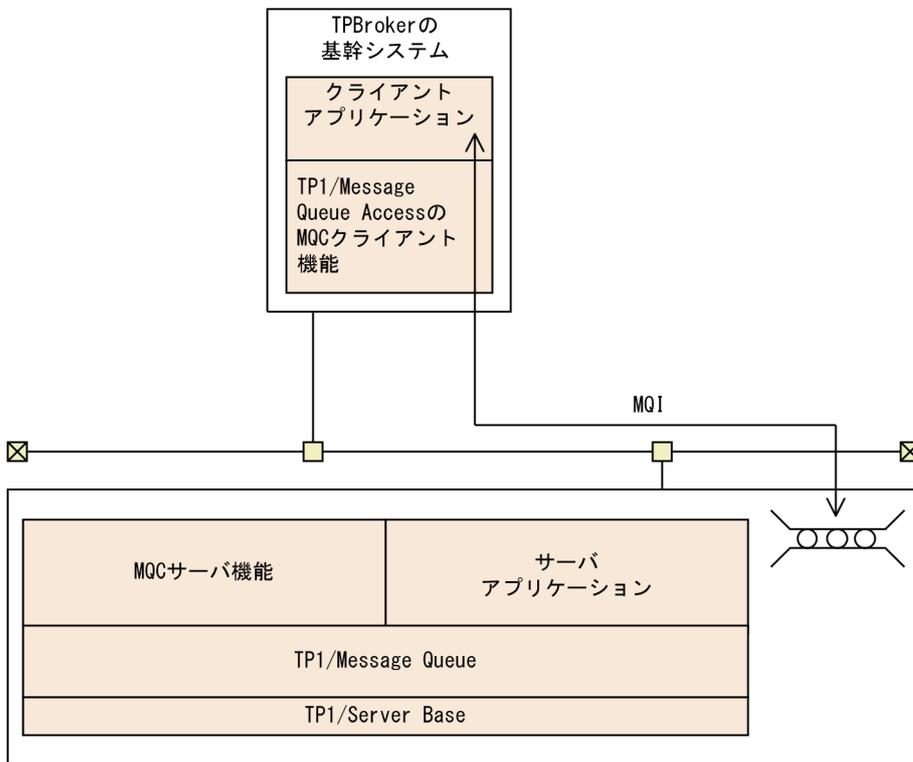
TP1/Message Queue (MQA) MQサービス機能	アプリケーション層
TP1/Message Queue (MQT) メッセージチャネル プロトコル	プレゼンテーション層
	セッション層
TCPプロトコル	トランスポート層
IPプロトコル	ネットワーク層
データリンク層	
物理層	

1.6 関連製品

TP1/Message Queue には TP1/Message Queue を前提とする関連製品があります。関連製品を使用するとメッセージキューイングを使用した柔軟で拡張性のあるシステムを構成できます。

TP1/Message Queue と関連製品について、次の図に示します。

図 1-5 TP1/Message Queue と関連製品



図中のプログラムについて説明します。

- TP1/Message Queue Access

クライアントアプリケーションから TP1/Message Queue のメッセージキューにメッセージを登録したり、取り出したりする機能を提供するプログラムです。TPBroker などのトランザクションマネージャと XA 接続すると、トランザクショナルな処理を作成することもできます。クライアントアプリケーションのプログラミングインタフェースとして、C、COBOL、C++、および Java 言語を使用できます。

クライアント側に MQC クライアント機能を提供し、サーバ側に MQC サーバ機能を提供します。

MQC クライアント機能については、マニュアル「TP1/Message Queue Access 使用の手引」を参照してください。

MQC サーバ機能は TP1/Message Queue 上で動作し、次に示す要素で構成されます。

- MQC リスナサーバ

MQC リスナサーバは、TP1/Message Queue に組み込まれて動作します。MQC クライアント機能からの通信を基に MQC ゲートウェイサーバを予約し、トランザクションを処理します。

- MQC ゲートウェイサーバ

MQC ゲートウェイサーバは、OpenTP1 システムのユーザサーバとして動作します。MQC ゲートウェイサーバは、MQC クライアント機能からのキューアクセスを処理します。また、送受信するメッセージを格納するために TP1/Message Queue のキューを使用します。

MQC サーバ機能を構成するファイルは、TP1/Message Queue に添付されています。MQC サーバ機能の定義、運用、および障害対策については、「[2.10 MQC サーバ機能](#)」を参照してください。

これらの製品には TP1/Message Queue 上で動作する部分（MQC サーバ機能など）があり、OpenTP1 のメッセージログファイルや syslog などにメッセージが出力されることがあります。これらのメッセージについては、マニュアル「TP1/Message Queue メッセージ」を参照してください。相手システム側で出力されるメッセージについては、該当する製品のマニュアルを参照してください。

2

機能

TP1/Message Queue を使用すると、メッセージキューイング機能を実現できます。この章では、メッセージキューイング機能の概要、メッセージ送受信機能などについて説明します。

この章で使用している定義コマンドについては、「[4. 準備作業](#)」を参照してください。また、運用コマンドについては、「[6. システムの運用](#)」を参照してください。

2.1 メッセージキューイング機能

メッセージキューイングとは、キューを使用してメッセージを送受信する方法です。アプリケーションがキューに登録したメッセージは、キューマネージャによって目的のキューまで送信されます。

TP1/Message Queue を使用すると、メッセージキューイングのためのアプリケーションを作成できます。メッセージキューイングの詳細およびアプリケーションの作成については、マニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

2.1.1 メッセージキューイング機能の特長

メッセージキューイング機能を使用したメッセージの送受信には、次に示す特長があります。

- キューとキューの間の通信はキューマネージャによって行われるため、メッセージの送受信時に、アプリケーションが動作している必要はありません。
- アプリケーションはキューに対する操作をするだけでよく、プログラム間の通信路を意識する必要はありません。

2.1.2 メッセージキューイングの基本構成要素

メッセージキューイングの基本構成要素はメッセージとキューです。

(1) メッセージの概要

メッセージはアプリケーションにとって意味のあるバイト文字列です。メッセージはアプリケーション間または同じアプリケーション内の異なる部分の間で情報を転送するために使用されます。アプリケーションは同じ OS 上または異なる OS 上で動作します。

メッセージは次に示す二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、それを使用するアプリケーション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メッセージを識別するとともに追加の制御情報を含みます。例えば、メッセージタイプや送信側アプリケーションによって割り当てられた優先度などを含みます。

メッセージ記述子の形式は IBM MQ によって規定されます。詳細については、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

(2) キューの概要

キューはメッセージを保存するために使用されるデータ構造体です。

各キューはキューマネージャに属します。キューマネージャは所属するキューを管理し、受信したメッセージを適切なキューへ保存します。メッセージは通常操作の一部として、アプリケーションプログラムやキューマネージャによってキューに登録されます。

キューマネージャが管理するキューの種類について、次の表に示します。

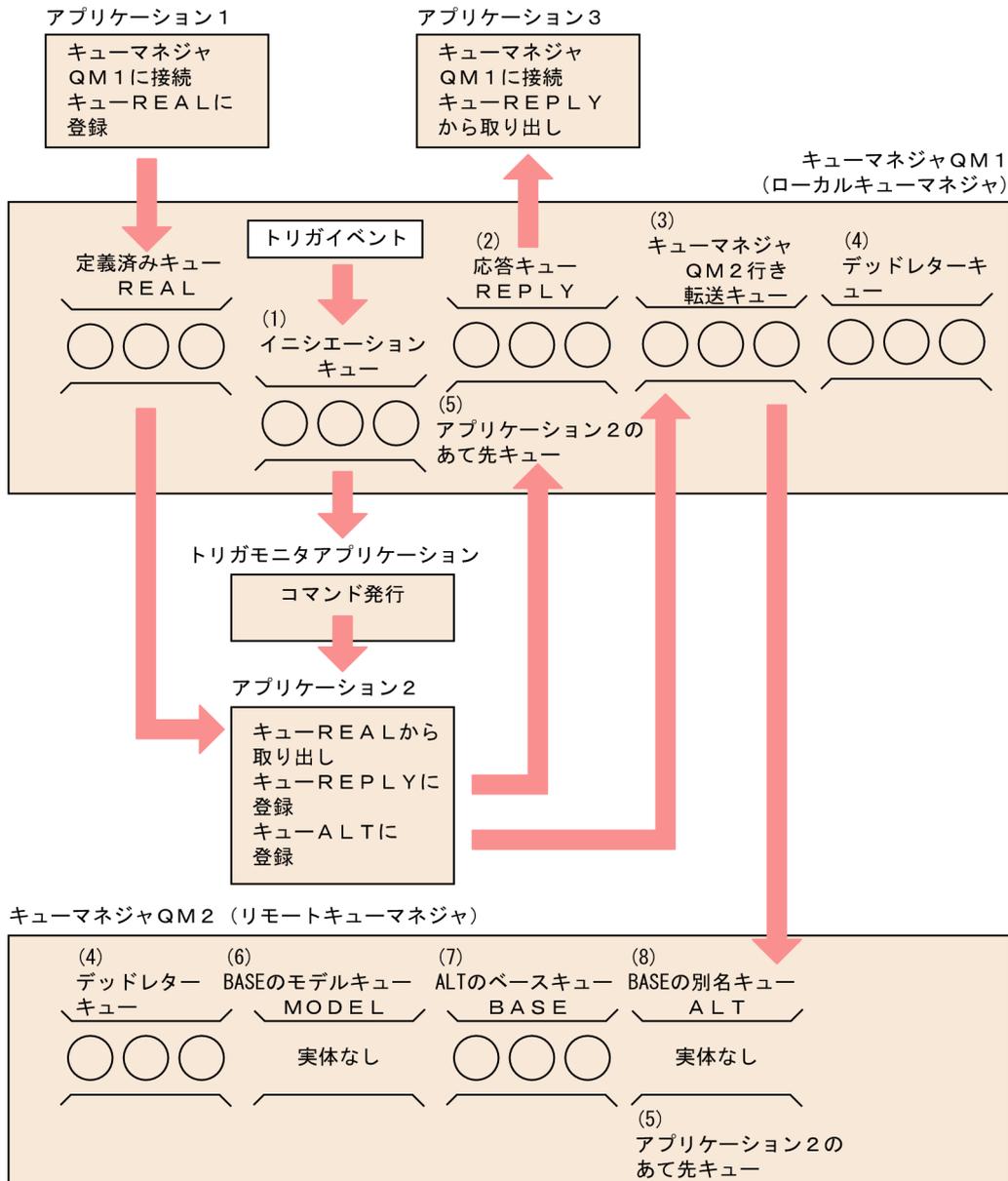
表 2-1 キューの種類

キューの種類	説明	作成方法	使用方法
ローカルキュー	アプリケーションが MQCONN 命令を利用して接続するキューマネージャに属しているキューです。	次に示す 2 とおりの作成方法があります。 <ul style="list-style-type: none">mqamkque コマンドで作成 (定義済みキュー)アプリケーションから MQOPEN 命令で作成 (動的キュー)	あて先キュー イニシエーションキュー 応答キュー デッドレターキュー 転送キュー ベースキュー
リモートキュー (リモートキューのローカル定義による)	アプリケーションが MQCONN 命令を利用して接続するキューマネージャとは別のキューマネージャに属しているキューです。	リモートキューのローカル定義で指定します。また、特に定義はなく、アプリケーションで指定する場合もあります。	あて先キュー 応答キュー ベースキュー
別名キュー	ローカルキューまたはリモートキューのローカル定義の別名です。	別名キューの属性定義で指定します。	別名キュー
モデルキュー	運用中に作成するキューの基になる属性を持つキューです。	モデルキューの属性定義で指定します。	モデルキュー

(3) キューの説明

TP1/Message Queue のキューの構造について、次の図に示します。

図 2-1 TP1/Message Queue のキューの構造



(a) キューの使用方法による分類

キューの使用方法およびそのキューがローカルキューからリモートキューになるかについて、次の表に示します。表の項番は、図 2-1 の括弧の数字に対応します。

表 2-2 キューの使用方法による分類

項番	使用方法	意味	ローカルキュー	リモートキュー
1	イニシエーションキュー	トリガメッセージを登録するキュー	○	×
2	応答キュー	アプリケーションが応答メッセージを受け取るキュー	○	○

項番	使用方法	意味	ローカルキュー	リモートキュー
3	転送キュー	特定のリモートキューマネージャにメッセージを転送するキュー	○	×
4	デッドレターキュー	障害などでキューに登録できなかったメッセージを登録するキュー	○	×
5	あて先キュー	メッセージのあて先になるキュー	○	○
6	モデルキュー	属性が一時的動的キュー、永続的動的キューの基になるキュー	—	—
7	ベースキュー	別名キューの基になるキュー	○	○
8	別名キュー	キューに付けられた別名だけのキュー	—	—

(凡例)

○：該当キューになります。

×：該当キューになりません。

—：実体のないキューのため該当しません。

(b) ローカルキューの作成方法による分類

ローカルキューには、2とおりの作成方法があります。作成方法によって定義済みキューと動的キューに分けられます。

定義済みキュー

mqamkque コマンドで作成します。作成方法については、「[2.2.2\(2\) キューの作成と削除](#)」を参照してください。

動的キュー

アプリケーションから MQOPEN 命令で作成します。定義タイプ (DefinitionType 属性) によって永続的動的キューと一時的動的キューに分けられます。永続的動的キューは、キューマネージャの障害発生時に回復されます。一時的動的キューは、キューマネージャの障害発生時に回復されません。

(4) キューの属性

キューには、属性と呼ばれる特性があります。キューの属性には、キュー名、キュータイプ、取り出し許可などがあります。属性については、マニュアル「[TP1/Message Queue プログラム作成リファレンス](#)」を参照してください。

キューの属性の決定・変更方法を次に示します。

- 次に示す MQA サービス定義の定義コマンドで指定します。
 - mqaqueatl 定義コマンド (モデルキューの属性定義)
 - mqaremque 定義コマンド (リモートキューのローカル定義)

- mqaalsque 定義コマンド (別名キューの属性定義)
- 一度指定した属性を MQSET 命令で変更できます。ただし、変更できる属性と変更できない属性があります。MQSET 命令については、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。
- 直接設定はできませんが、キューマネージャの動作に応じて値が変動する属性もあります。例えば、キューのメッセージ登録数などです。

2.1.3 アプリケーションからのメッセージ操作とキュー操作

メッセージキューイングを実行するには、命令を使用してアプリケーションを作成します。TP1/Message Queue には、次に示す命令があります。

- キューマネージャへの接続 (MQCONN 命令)
- キューマネージャからの切り離し (MQDISC 命令)
- オブジェクト (メッセージキュー) のオープン (MQOPEN 命令)
- オブジェクト (メッセージキュー) のクローズ (MQCLOSE 命令)
- メッセージの登録 (MQPUT 命令)
- メッセージの取り出し (MQGET 命令)
- 1 メッセージの登録 (MQPUT1 命令)
- オブジェクトの属性の照会 (MQINQ 命令)
- オブジェクトの属性の設定 (MQSET 命令)

ユーザは、これらの命令を組み合わせたアプリケーションを作成して、メッセージキューイング機能を実現し、メッセージの送受信をします。命令の詳細については、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

2.1.4 MQI 命令使用時の OpenTP1 アプリケーションとの関係

TP1/Message Queue のアプリケーションは、OpenTP1 の SUP (サービス利用プログラム)、SPP (サービス提供プログラム)、または MHP (メッセージ処理プログラム) として作成します。アプリケーションからのメッセージの登録および取り出しは、OpenTP1 のトランザクションと同期させることができます。

TP1/Message Queue のアプリケーションは、C 言語または COBOL 言語で作成します。データ操作言語では作成できません。

(1) TP1/Message Queue で使用できる命令

TP1/Message Queue で使用できる命令を次の表に示します。なお、命令の詳細については、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

表 2-3 TP1/Message Queue で使用できる命令

命令	SUP		SPP			MHP	オフラインの業務 をするアプリケーション
	トランザクション 範囲でない	トランザク ション範囲 (ルート)	トランザクション 範囲でない	トランザクション 範囲			
				ルート	ルート 以外		
MQCONN	○	○	○	○	○	○	—
MQDISC	○	○	○	○	○	○	—
MQOPEN	○	○	○	○	○	○	—
MQCLOSE	○	○	○	○	○	○	—
MQPUT	○	○	○	○	○	○	—
MQGET	○	○	○	○	○	○	—
MQPUT1	○	○	○	○	○	○	—
MQINQ	○	○	○	○	○	○	—
MQSET	○	○	○	○	○	○	—

(凡例)

- ：該当する条件で呼び出せます。
- ：該当する条件で呼び出せません。

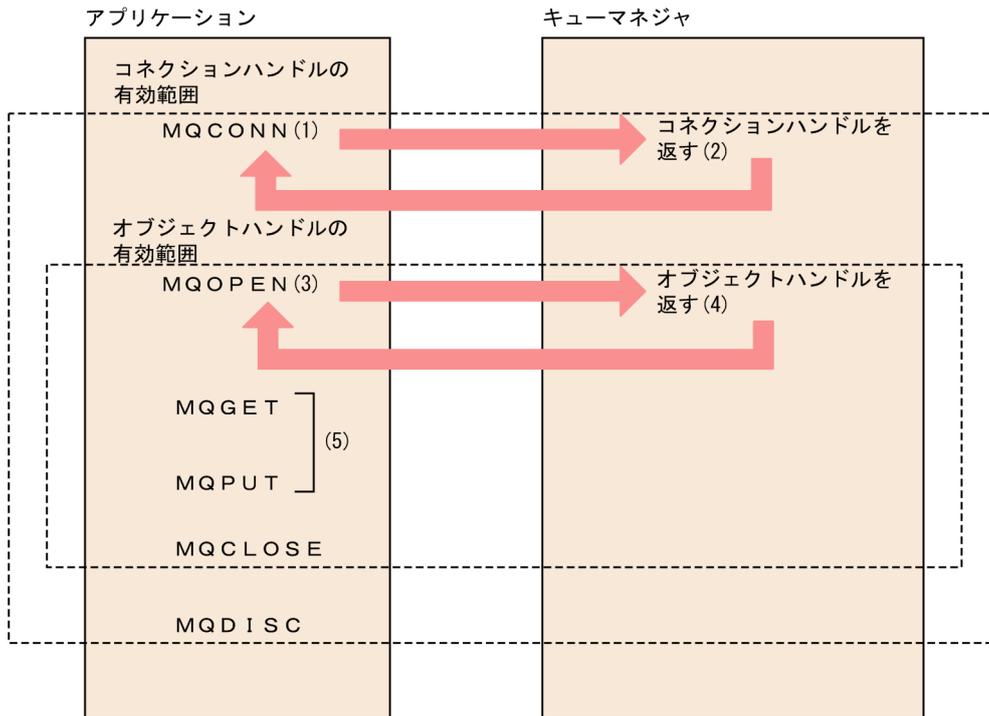
(2) アプリケーションとキューマネージャの接続

アプリケーションがメッセージキューイング機能を使用する場合、必ず MQCONN 命令を呼び出してキューマネージャに接続してください。キューマネージャに接続すると、コネクションハンドルという値がアプリケーションに返されます。

また、キューなどのオブジェクトを使用する場合は、MQOPEN 命令を呼び出してオブジェクトをオープンする必要があります（ただし、MQPUT1 命令を使用する場合は不要です）。オブジェクトをオープンすると、オブジェクトハンドルという値がアプリケーションに返されます。

アプリケーションとキューマネージャの接続の手順およびハンドルの有効範囲について、次の図に示します。

図 2-2 アプリケーションとキューマネージャの接続手順およびハンドルの有効範囲



図中に示したアプリケーションとキューマネージャの接続の手順を次に示します。図中の番号と、手順の番号は対応しています。

1. MQCONN 命令を呼び出してキューマネージャへ接続を要求します。
2. キューマネージャは、コネクションハンドルを MQCONN 命令に返します。
3. 手順 2. で返されたコネクションハンドルを指定して MQOPEN 命令を呼び出します。
4. キューマネージャは、オブジェクトハンドルを MQOPEN 命令に返します。
5. MQOPEN 命令以降に呼び出す命令で、該当するオブジェクトを利用する場合には、手順 4. で返されたオブジェクトハンドルを指定します。

ハンドルの有効範囲を次に示します。

コネクションハンドル

MQCONN 命令を呼び出したアプリケーション（アプリケーションとキューマネージャを MQCONN 命令で接続し、MQDISC 命令で切り離すまで）

オブジェクトハンドル

MQOPEN 命令を呼び出したアプリケーション（オブジェクトを MQOPEN 命令でオープンし、MQCLOSE 命令でクローズするまで）

(3) トランザクションとメッセージキューアクセスとの関係

メッセージの登録および取り出しを OpenTP1 のトランザクションと同期させることができます。OpenTP1 のトランザクション内で、MQPUT 命令、MQPUT1 命令、または MQGET 命令を呼び出すと、その命令による登録および取り出しは、トランザクションの一部として処理されます。ただし、OpenTP1 のトランザクション内であっても、同期点を取らないことを指定すると、その命令はトランザクションの一部にはなりません。

(4) アプリケーションからメッセージキューにアクセスする場合の条件

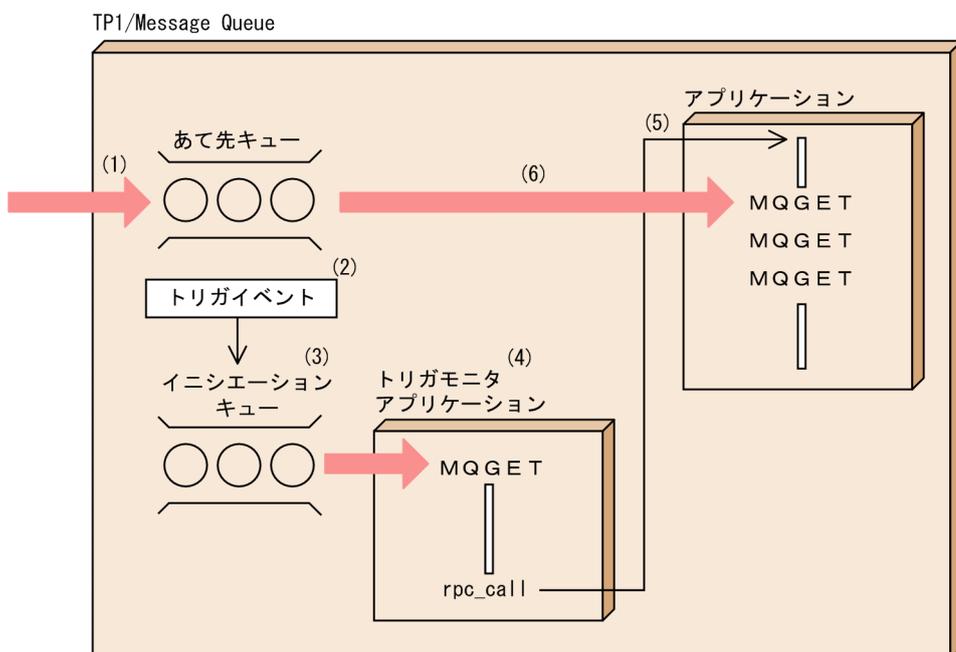
メッセージキューのアクセスでは、グローバルトランザクションの RPC の形態をすべて同期応答型 RPC にする必要があります。非同期応答型 RPC、非応答型 RPC からメッセージキューへアクセスした場合、動作は保証されません。

2.1.5 トリガ機能

メッセージがキューに到着した時、そのメッセージを処理するアプリケーションを起動するために、トリガイベントを使用できます。mqaqueatl 定義コマンドでトリガイベントを設定したキューにメッセージが到着すると、トリガメッセージがイニシエーションキューに登録されます。ユーザはイニシエーションキューを監視することで、メッセージが到着したかどうかわかります。トリガイベントの種類（トリガタイプ）などは、ローカルキューの属性として指定します。

メッセージがキューに登録されてからのトリガの処理の流れについて、次の図に示します。

図 2-3 トリガの処理の流れ



図中に示したトリガ処理の流れを次に示します。図中の番号と、流れの番号は対応しています。

1. メッセージ送信元のアて先キューとなるキューにメッセージが登録されます。
2. アて先キューの状態が設定した条件を満たすと、トリガイイベントが発生します。
3. トリガイイベントが発生したことを意味するトリガメッセージをイニシエーションキューに登録します。
4. トリガモニタアプリケーションは、常にイニシエーションキューを監視しています。
5. トリガモニタアプリケーションは、トリガメッセージを取り出して、メッセージを受け取るアプリケーションを起動します。
6. 流れ 5. で起動されたアプリケーションは、MQGET 命令を呼び出してメッセージを受け取ります。

トリガモニタアプリケーションは、イニシエーションキューを監視するユーザ作成のアプリケーションです。イニシエーションキューの状態を常に監視するように作成してください。

2.2 キューファイルの使用方法

TP1/Message Queue はメッセージキューイングの基本構成要素であるキューを、キューファイルという物理ファイル内で管理しています。この節では、キューファイルの構造および運用方法を説明します。

この節で使用している定義コマンドについては、「4. 準備作業」を参照してください。また、運用コマンドについては、「6. システムの運用」を参照してください。

2.2.1 キューファイルの構造

キューファイルの構造について説明します。

(1) キューファイル

キューファイルは、キューに出力したメッセージが格納される物理的なファイルです。キューファイルは、オンライン開始前に mqainit コマンドで OpenTP1 ファイルシステム上に作成しておきます。TP1/Message Queue の開始時にキューファイルをメモリ上に読み込み、キューに対する準備をします。

一つのキューファイルには、複数の論理的なファイルの情報が格納されます。

キューファイルには、次に示す 2 種類の物理ファイルが対応づけられます。mqainit コマンドによって、一つのキューファイルに対しそれぞれの物理ファイルが作成されます。

メッセージ情報管理ファイル

複数の論理的なキューの情報のうち、キューおよびメッセージの管理情報が格納されます。ファイル名には、後ろに「.cnt」が付けられます。

メッセージファイル

複数の論理的なキューの情報のうち、メッセージ自身が格納されます。ファイル名には、後ろに「.msg」が付けられます。

なお、MQA サービス定義の mqa_quefil_inf オペランドに Y を指定すると、キューファイル単位での容量監視ができます。

(2) キューファイルグループ

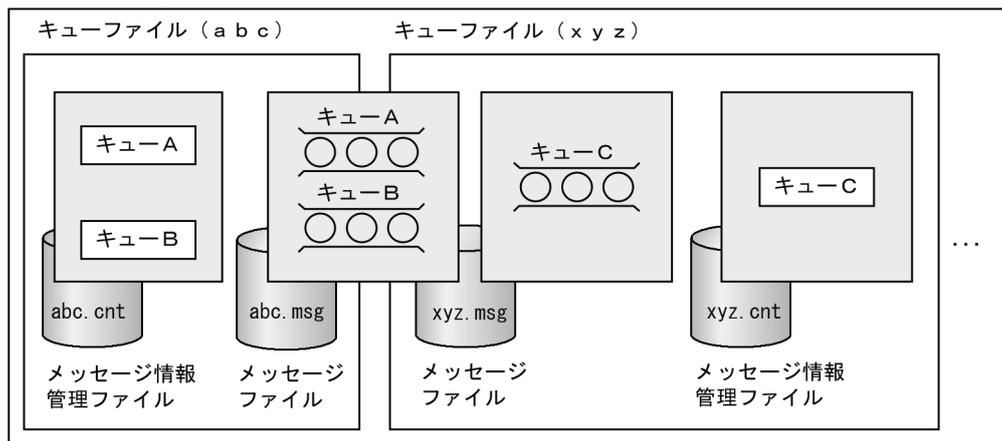
複数のキューファイルを、一つの論理的なキューファイルとして扱うことができます。複数のキューファイルのまとまりをキューファイルグループといいます。MQA サービス定義の mqaquegrp 定義コマンドで、キューファイルグループを構成するキューファイル名を指定します。

一つのキューファイルグループに複数のキューファイルが対応づけられている場合、どのキューファイルにメッセージが格納されるかは、TP1/Message Queue によって決定されます。

キューファイルグループの構造について、次の図に示します。

図 2-4 キューファイルグループの構造

キューファイルグループ



複数のキューファイルを一つのキューファイルグループにする場合の長所について、次に示します。

- キューの大容量化

キューファイルの最大格納容量は2ギガバイトです。一つのキューファイルでは一つのキューに格納できる最大格納容量がキューファイルの最大格納容量になります。しかし、キューファイルグループにすることによって、最大格納容量の上限がキューファイル数×キューファイルの最大格納容量になります。

- キューファイル入出力処理の負荷分散

一つのキューに複数のアプリケーションからアクセスする場合、キューファイルに対しての入出力処理が分散されます。

なお、MQA サービス定義の `mqaquegrp` 定義コマンドの `-k` オプションに `Y` を指定すると、キューファイルグループ単位での容量監視ができます。

注意事項

アプリケーションまたはチャネルによる複数のキュー操作が同時に一つのキューファイルグループに集中すると、キュー操作が遅くなることがあります。同時に使用するキューファイルは異なるキューファイルグループに分散してください。

(3) キューとキューファイルの関係

ユーザは、メッセージキューイング機能を使用するために、キューとキューファイルを対応づける必要があります。

キューとキューファイルの対応は、`mqamkque` コマンドで指定するキュー属性定義ファイル（UNIX テキストファイル、またはMQA サービス定義ファイル）に指定された `mqaqueatl` 定義コマンドで行います。`mqamkque` コマンドまたは `MQOPEN` 命令で作成したキューは、モデルキューの属性定義で関連づけられたキューファイルに対応づけられます。

2.2.2 運用コマンドによるキューファイルの運用

ユーザは、キューを使用してメッセージキューイング機能を実現します。キューの情報は、最終的には物理的なキューファイルに格納されます。ここでは、運用コマンドによるキューファイルの運用について説明します。

(1) キューファイルの作成

ユーザは、キューの情報を格納するためのキューファイルを、mqainit コマンドを使用して作成します。このとき、キャラクタ型スペシャルファイルまたは UNIX ファイルのアクセス権は、所有者およびグループにリード・ライト権を与えてください。また、OpenTP1 ファイルシステムのアクセス権はユーザ用に設定してください。アクセス権を正しく設定していない場合、KFCA31184-E が出力されて OpenTP1 の起動に失敗します。

mqainit コマンドを使用すると、二つの物理ファイルが作成されます。一つは、メッセージの管理情報が格納されるメッセージ情報管理ファイルです。もう一つは、メッセージ自身が格納されるメッセージファイルです。メッセージ情報管理ファイルとメッセージファイルの合計の最大格納容量は 2 ギガバイトです。

注意事項

キューファイルパスおよびキューファイル名にスペースを指定しないでください。指定した場合、動作を保証できません。

(2) キューの作成と削除

ユーザは、メッセージキューイングに使用するキューを作成する必要があります。

キューは、作成の方法によって、定義済みキューと動的キューに分けられます。

(a) 定義済みキューの作成

定義済みキューは、オンライン開始前にあらかじめ mqamkque コマンドを使用して作成しておきます。いったん作成すると、キューを削除するまでシステム内に存在します。

定義済みキューの作成方法について、次の図に示します。

図 2-5 定義済みキューの作成

(1) UNIXテキストファイルの作成 (2) キューの作成コマンドの実行 (3) キューの作成



- (1) UNIXテキストファイルにMQAサービス定義のmqaqueatl（モデルキューの属性定義）と同じ形式で定義済みキューの属性を指定します。
- (2) mqamkque（キューの作成）コマンドで(1)のファイルのパス名を指定して実行します。
- (3) 定義済みキューが作成されます。

(b) 動的キューの作成

動的キューは、オンライン中にアプリケーションから MQOPEN 命令で作成するキューです。

MQOPEN 命令では、モデルキューの属性定義で指定したモデルキューの定義名を指定します。MQOPEN 命令の詳細については、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

動的キューには、一時的動的キューと永続的動的キューがあります。一時的か永続的かは、モデルキューの属性定義（MQA サービス定義の mqaqueatl 定義コマンド）で指定します。

(c) キューの削除

キューを削除する場合は、mqadelque コマンドを使用します。mqadelque コマンドで、定義済みキューと動的キューの両方を削除できます。また、動的キューは、動的キューを作成した MQOPEN 命令のアプリケーションが終了すると、自動的に削除されます。

(3) キューの属性の変更

キューの属性を変更する場合は、mqachgque コマンドを使用します。mqachgque コマンドで変更できるのは、ローカルキューの属性です。なお、キューの属性のうち、定義タイプは変更できません。

(4) キューファイルの状態

キューファイルには、次に示す状態があります。

正常

正常に使用できる状態です。

論理閉塞

論理的に閉塞して、メッセージの登録および取り出しができない状態です。

障害閉塞

キューファイルに障害が発生した場合の状態です。

一時的障害閉塞

キューファイルに一時的な障害が発生した場合の状態です。次に示す状態があります。

- 一時的障害閉塞 (C)
メッセージ情報管理ファイルに一時的な障害が発生した状態
- 一時的障害閉塞 (CR)
メッセージ情報管理ファイルに一時的な障害が発生したあと回復済みの状態
- 一時的障害閉塞 (M)
メッセージファイルに一時的な障害が発生した状態
- 一時的障害閉塞 (C/M)
メッセージ情報管理ファイルおよびメッセージファイルに一時的な障害が発生した状態
- 一時的障害閉塞 (CR/M)
メッセージ情報管理ファイルに一時的な障害が発生したあと回復済みであるが、メッセージファイルに一時的な障害が発生した状態

キューファイル障害発生時に、障害要因が一時的である可能性のある場合は、一時的障害閉塞に移行します。キューファイル格納媒体損傷の場合も、一時的障害閉塞に移行します。

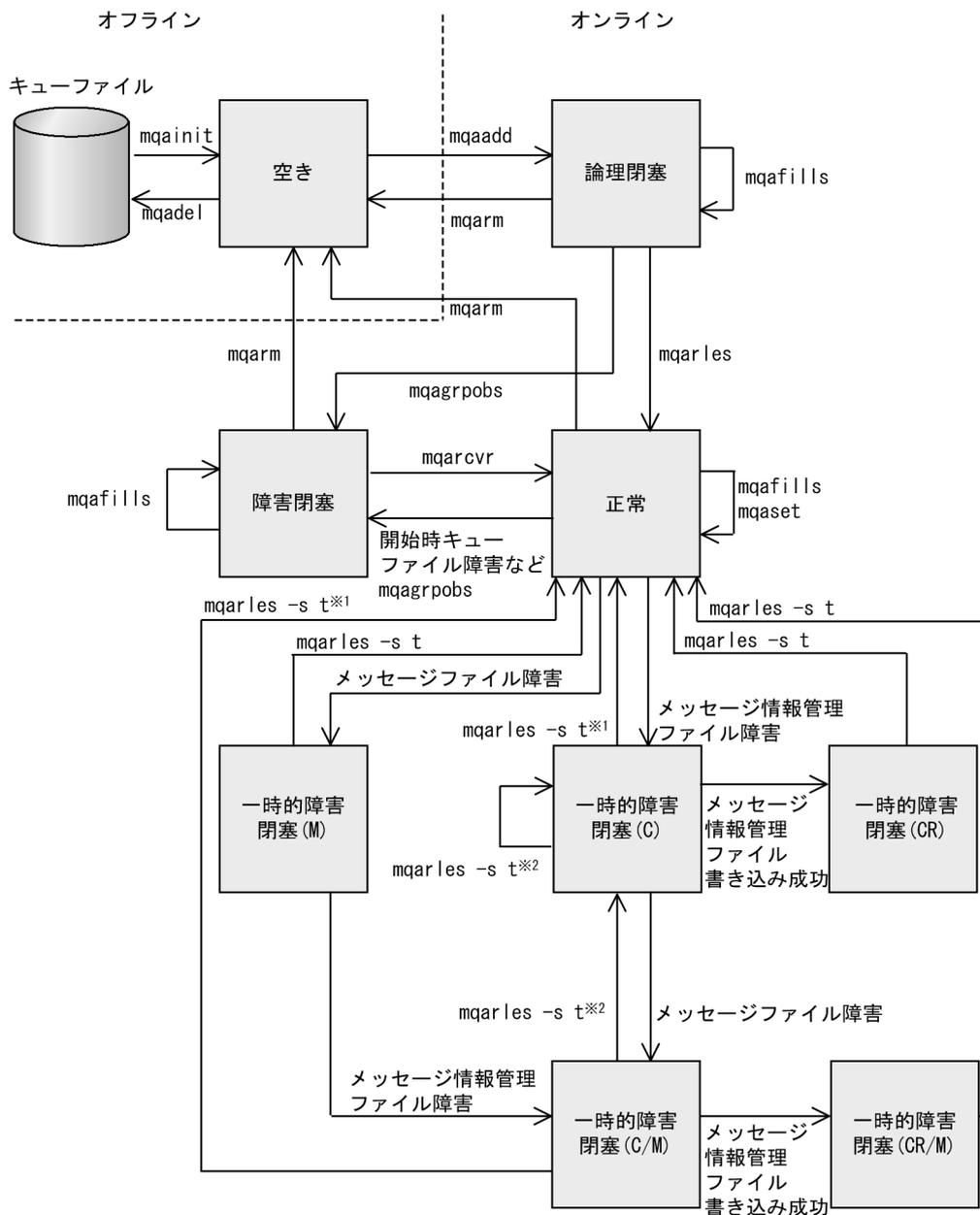
一時的障害閉塞に移行すると、障害発生時点でキューファイルに未反映である更新情報はメモリ上に保持されます（障害閉塞に移行する場合、更新情報は破棄されます）。そのため、障害要因を除去した後に、未反映である更新情報をキューファイルに書き込むことによって回復することができます。mqafrc コマンドによって、ジャーナルファイルを使用してキューファイルを回復する必要はありません。

空き

キューファイルは存在していますが、キューマネージャに認識されていない状態です。オンラインで使用するためには、キューファイルを組み込む必要があります。

キューファイルの状態とコマンドによる状態遷移について、次の図に示します。

図 2-6 キューファイルの状態遷移



注 1

図中の矢印以外に次に示す状態遷移があります。

- すべての「一時的障害閉塞」から `mqarm` コマンドによって「空き」に移行します。
- すべての「一時的障害閉塞」から `mqagrprobs` コマンドによって「障害閉塞」に移行します。
- 「一時的障害閉塞 (C)」および「一時的障害閉塞 (C/M)」から計画停止および再開始によって「障害閉塞」に移行します。

注 2

TP1/Message Queue が一定の間隔 (チェックポイントダンプ取得間隔または 3 分間隔) で回復処理を実行しているため、自動的にキューファイルの状態が移行することがあります。

「正常」から「障害閉塞」に移行するのは、次に示す場合です。

- 開始時および終了時にキューファイル障害を検知した場合
 - キューファイル二重化構成の片系のキューファイル障害（片系障害）を検知した場合
- これら以外でキューファイル障害を検知すると「一時的障害閉塞」に移行します。

注※1

mqarles コマンド入力時に、メッセージ情報管理ファイルへの書き込みが成功した場合

注※2

mqarles コマンド入力時に、メッセージ情報管理ファイルへの書き込みが失敗した場合

注意事項

すべての「一時的障害閉塞」から OpenTP1 を正常終了した場合、そのキューファイルは「障害閉塞」として扱われるため、キューファイルの再作成が必要です。

(5) オブジェクトの属性の設定（キューに登録の許可／禁止および取り出しの許可／禁止を設定）

キューに、登録の許可／禁止および取り出しの許可／禁止を設定する場合は、mqaset コマンドを使用します。mqaset コマンドのコマンド引数によって、次に示すキューに設定できます。

- ローカルキュー
- リモートキューのローカル定義で指定されたキュー※
- 別名キュー

注※

取り出しの許可／禁止は設定できません。

(6) オブジェクトの属性表示

オブジェクトの属性を表示する場合は、mqainq コマンドを使用します。mqainq コマンドのオプションによって、次に示すオブジェクトの属性を表示できます。

-m オプション

- キューマネージャ

-q オプション

- ローカルキュー
- モデルキュー
- リモートキューのローカル定義で指定されたキュー
- 別名キュー

(7) キューファイル内の情報表示

オフラインの状態では、指定したキューファイルの管理部の情報を表示するには、mqafilinf コマンドを使用します。mqafilinf コマンドのオプションによって、次に示す情報を表示できます。

-q オプション

- ローカルキュー情報

-x オプション

- キュー名一覧

(8) キューファイルの組み込みと切り離し

キューファイルを組み込む場合は、mqaadd コマンドを使用します。キューファイルを組み込むと、オンラインで使用できる状態になります。このあと、mqarles コマンドを使用して閉塞状態を解除すると、キューファイルはアプリケーションから使用できる状態になります。

キューファイルを切り離す場合、mqarm コマンドを使用します。キューファイルを切り離すと、キューファイルの中のキューおよびメッセージは破棄されます。

-r オプションを指定すると、キューファイルを強制的に切り離せます。

(9) キューファイルの論理閉塞と閉塞解除

mqaadd コマンドでキューファイルを組み込んだ直後は、キューファイルは論理閉塞状態です。閉塞状態を解除するために、mqarles コマンドを使用します。

また、一時的障害閉塞を解除するには、mqarles コマンドの-s オプションに t を指定します。

(10) キューファイルのバックアップとリストア

キューファイルをバックアップする場合は、mqabkup コマンドを使用します。mqabkup コマンドで指定したキューファイルがオンラインで使用中の場合、この操作はエラーになります。

バックアップしたファイルをリストアするには、mqarstr コマンドを使用します。

(11) キューファイルの回復

キューファイルの回復を MQA FRC といいます。MQA FRC は、mqafrc コマンドで実行します。MQA FRC はキューファイルの障害要因により回復方法が異なります。

なお、mqafrc コマンドの実行を含むキューファイルの回復手順については、「[7.1.4 キューファイルの入出力エラー](#)」を参照してください。

(a) 障害要因がキューファイル格納媒体損傷の場合

キューファイル格納媒体損傷の場合、次に示す資源が必要です。

- MQA サービス定義の `mqa_jnl_conf` オペランドに Y を指定して取得したアンロードジャーナルファイル
- キューファイルのバックアップファイル

回復方法については、「2.2.2(12) `mqafrfc` コマンドの使用例 (キューファイル格納媒体損傷の場合)」を参照してください。

(b) 障害要因がキューファイル格納媒体損傷以外の場合

キューファイル格納媒体損傷以外の場合、次に示す資源が必要です。

- アンロードジャーナルファイル
MQA サービス定義の `mqa_jnl_conf` オペランドの指定は Y でも N でもかまいません。

回復方法については、「2.2.2(13) `mqafrfc` コマンドの使用例 (キューファイル格納媒体損傷以外の場合)」を参照してください。

(12) `mqafrfc` コマンドの使用例 (キューファイル格納媒体損傷の場合)

障害要因がキューファイル格納媒体損傷の場合の、`mqafrfc` コマンドの使用例について説明します。使用例で指定する回復対象キューファイル定義ファイル (ファイル名 `mappingfile`) は、次に示すとおり記述します。

```
mqafilemap /dev/rdisk/rhd031/rcvf1 /dev/rdisk/rhd032/rcvf1
```

この定義例では、回復対象キューファイル名は `/dev/rdisk/rhd031/rcvf1`、回復先キューファイル名は `/dev/rdisk/rhd032/rcvf1` です。回復先キューファイル名には、実際に回復させるキューファイル名を指定します。回復対象キューファイル名は、回復先キューファイルを回復するために必要なアンロードジャーナル内の情報を特定するために使用します。回復対象キューファイル名が示すファイルの情報を基に回復するわけではありません。`mqafrfc` コマンドを実行する場合には回復先キューファイルが存在している必要があります。キューファイル `/dev/rdisk/rhd032/rcvf1` は、キューファイル `/dev/rdisk/rhd031/rcvf1` のバックアップを基にリストア済みで、`mqafrfc` コマンドの入力によってキューファイル `/dev/rdisk/rhd032/rcvf1` は障害発生直前の状態に回復されます。

- 回復対象キューファイルのバックアップ方法
回復対象キューファイルをバックアップするには、次に示すどちらかの方法を実行します。
 - `mqaabkup` コマンドで OpenTP1 ファイルシステム上のキューファイルを OS のファイルにバックアップ
 - OpenTP1 ファイルシステムを `filbkup` コマンドでバックアップ
- 回復先キューファイルのリストア方法
バックアップを回復先キューファイルにリストアするには、次に示すどちらかの方法を実行します。

- mqabkup コマンドでバックアップしたキューファイルを mqarstr コマンドで OpenTP1 ファイルシステムにリストア
- filbkup コマンドでバックアップした OpenTP1 ファイルシステムを filrstr コマンドに -r オプションを指定して OpenTP1 ファイルシステムにリストア

mqafrc コマンドに指定するアンロードジャーナルファイル名は次に示すとおりとします。

jnlf001, jnlf002, …, jnlf00n… (n: 世代番号)

(a) 使用例 1

ファイル分割されていないアンロードジャーナルファイルからキューファイルを回復する場合について説明します。

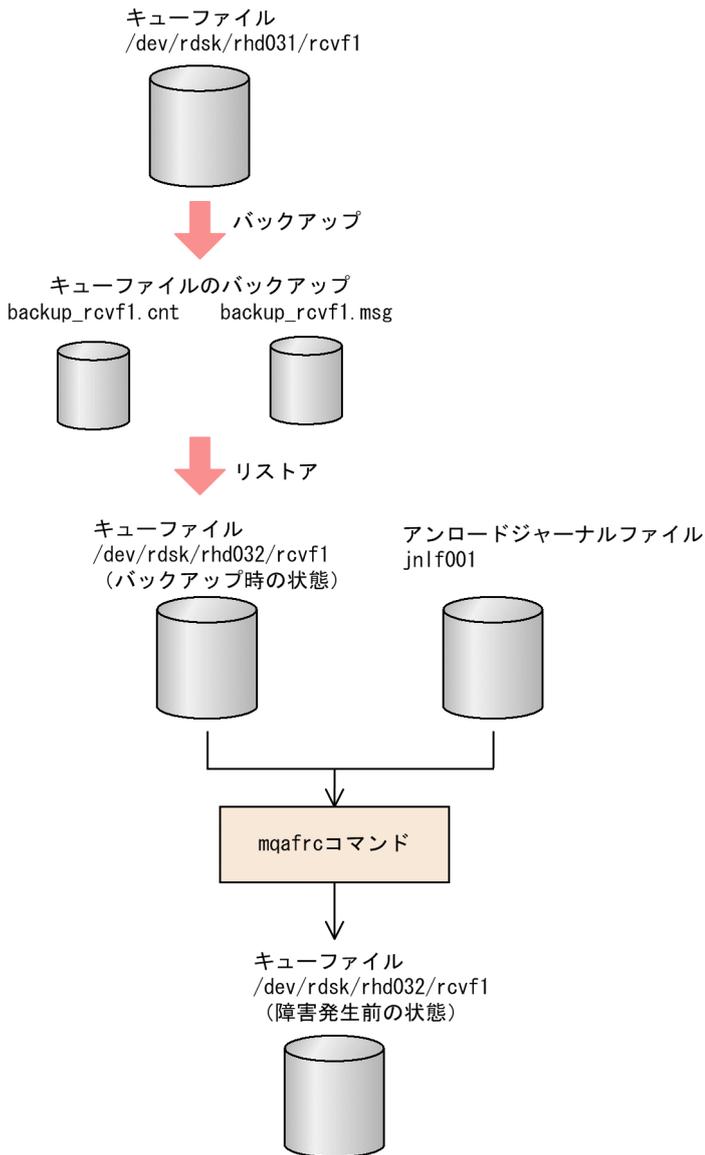
次に示すとおりコマンドを入力します。

```
mqafrc -s -e -f mappingfile /jnlf001
```

jnlf001 を入力として、回復先キューファイルが回復されます。

キューファイル回復の流れについて、次の図に示します。

図 2-7 キューファイル回復の流れ (キューファイル格納媒体損傷の場合)



(b) 使用例 2

三つのファイルに分割されているアンロードジャーナルファイルから、キューファイルを 1 回のコマンド入力で回復する場合について説明します。

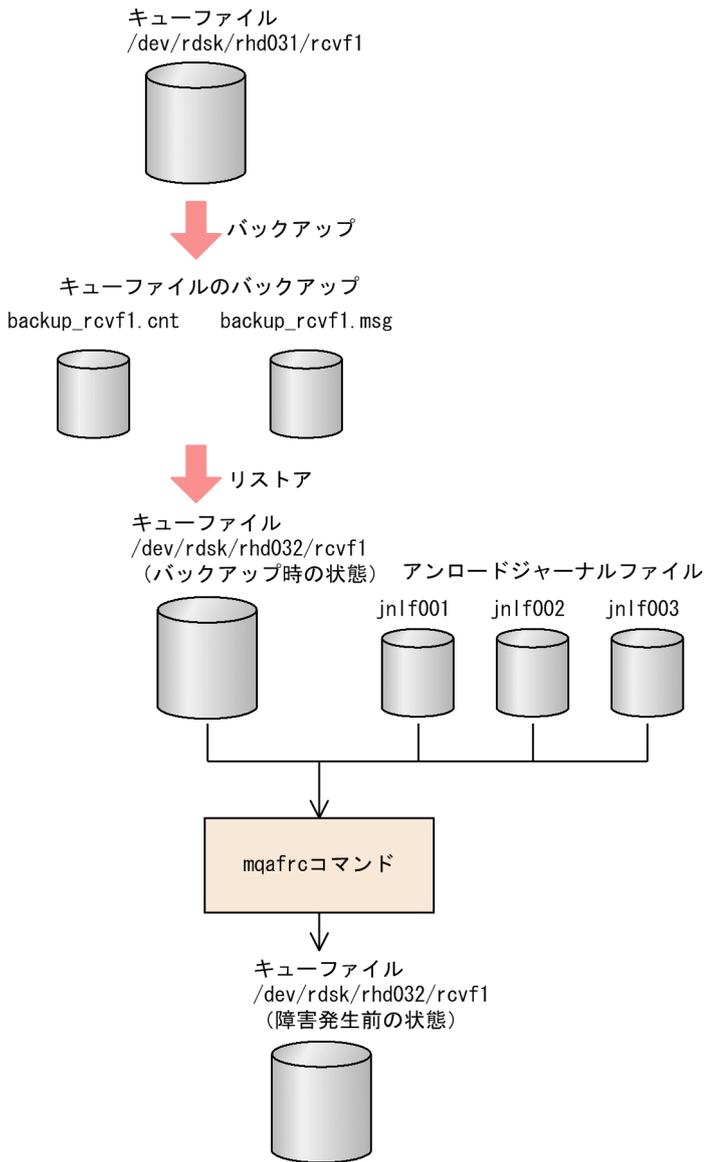
次に示すとおりコマンドを入力します。

```
mqafrc -s -e -f mappingfile /jnlf001 /jnlf002 /jnlf003
```

`jnlf001`, `jnlf002`, および `jnlf003` を入力として、回復先キューファイルが回復されます。

複数のアンロードジャーナルファイルとキューファイル回復の流れについて、次の図に示します。

図 2-8 複数のアンロードジャーナルファイルとキューファイル回復の流れ



(c) 使用例 3

四つのファイルに分割されているアンロードジャーナルファイルから、キューファイルを3回のコマンド入力で回復する場合について説明します。

(1 回目)

次に示すとおりコマンドを入力します。

```
mqafrc -s -f mappingfile /jnl f001 /jnl f002
```

jnl f001 および jnl f002 を入力として、回復先キューファイルが回復されます。また、引き継ぎファイルとして mqafrc001 および jnl colc001 が作成されます。

(2 回目)

次に示すとおりコマンドを入力します。

```
mqafrc -f mappingfile /jnlf003
```

jnlf003, mqafrc001, および jnlcolc001 を入力として、回復先キューファイルが回復されます。その後、mqafrc001 および jnlcolc001 の内容は更新されます。

(3回目)

次に示すとおりコマンドを入力します。

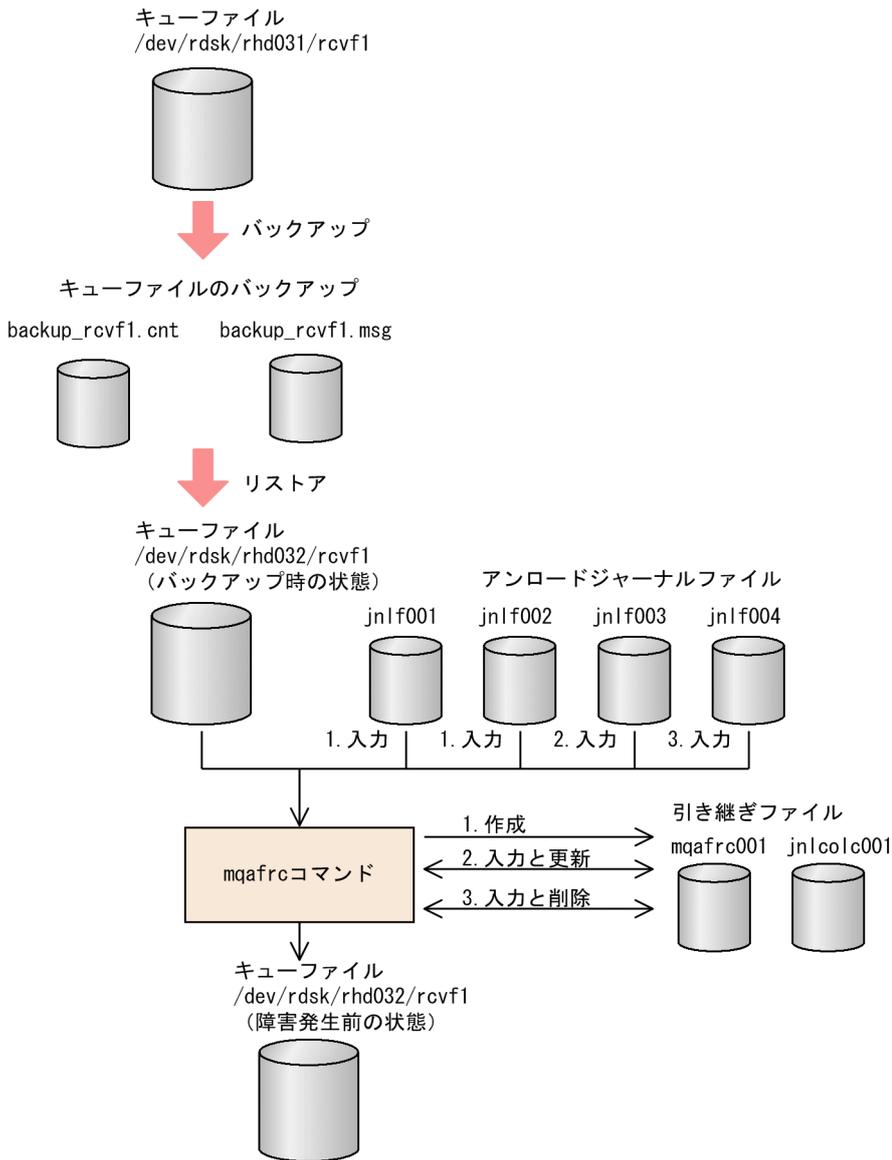
```
mqafrc -e -f mappingfile /jnlf004
```

jnlf004, mqafrc001, および jnlcolc001 を入力として、回復先キューファイルが回復されます。-e オプションを指定しているため、その後、mqafrc001 および jnlcolc001 は削除されます。

この場合、3回目の mqafrc コマンドが完了するまで、回復対象キューファイル定義ファイル mappingfile の内容を変更しないでください。

引き継ぎファイルを使用するキューファイル回復の流れについて、次の図に示します。図中の番号は何回目のコマンド入力による動作であるかを表します。

図 2-9 引き継ぎファイルを使用するキューファイル回復の流れ



(d) 使用例 4

世代番号が 1 のアンロードジャーナルファイルは使用しないで、世代番号 2 以降のアンロードジャーナルファイルを使用してキューファイルを回復する場合について説明します。

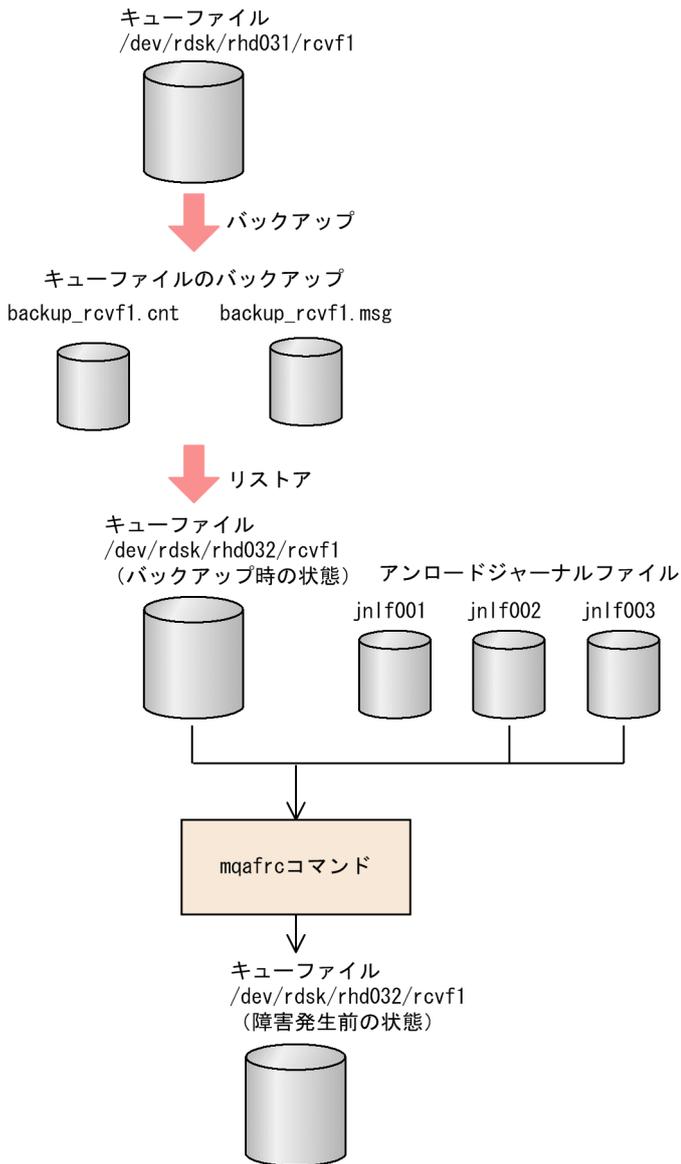
次に示すとおりコマンドを入力します。

```
mqafrc -s -e -g -f mappingfile /jnlf002 /jnlf003
```

`jnlf002` および `jnlf003` を入力として、回復先キューファイルが回復されます。

アンロードジャーナルファイルの世代番号とキューファイル回復の流れについて、次の図に示します。

図 2-10 アンロードジャーナルファイルの世代番号とキューファイル回復の流れ



(e) 使用例 5

キーを指定してキューファイルを回復する場合について説明します。

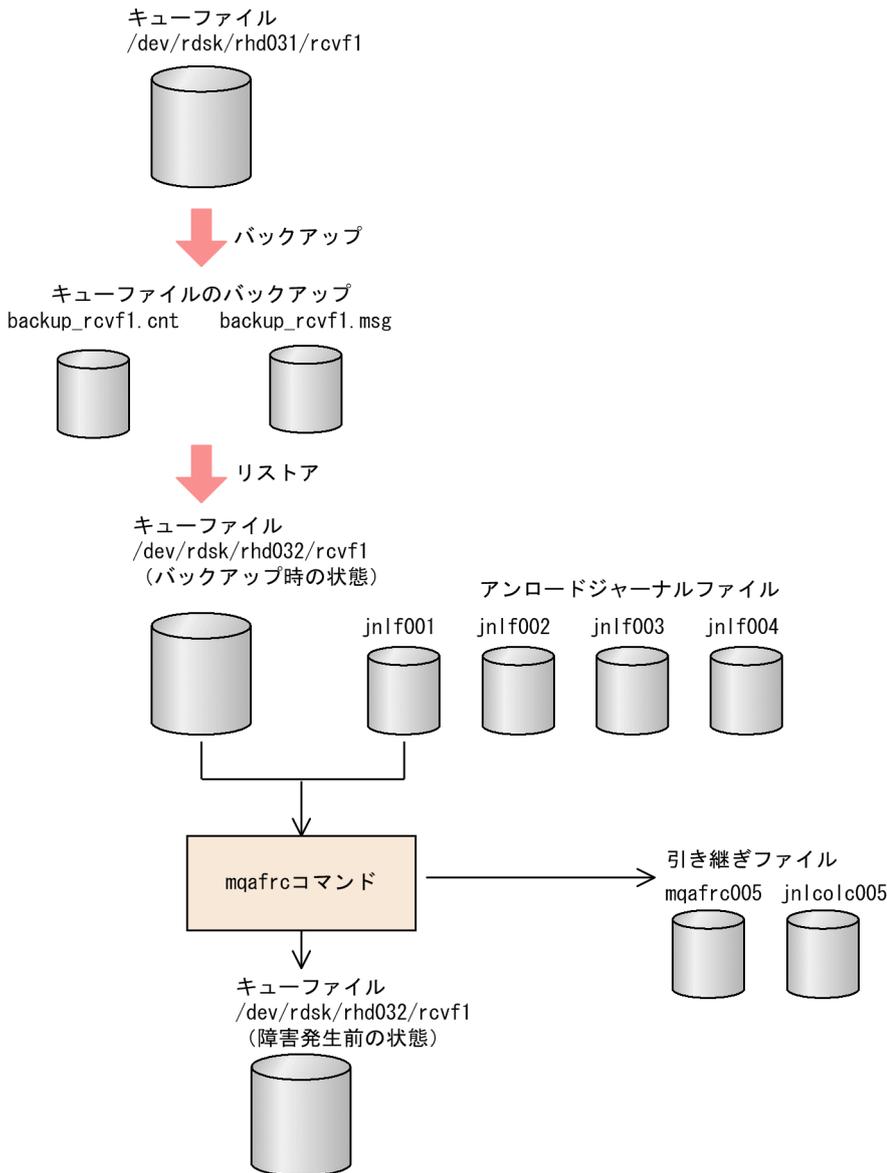
次に示すとおりコマンドを入力します。

```
mqafrc -s -k 005 -f mappingfile /jnlf001
```

mqafrc コマンドに-k オプションを指定すると、mqafrc コマンドが作成する引き継ぎファイルの名称の一部を指定できます。2 回目以降の MQA FRC でこの引き継ぎファイルを使用する場合は、前回と同じキーを指定してください。

引き継ぎファイルのキーの指定とキューファイル回復の流れについて、次の図に示します。

図 2-11 引き継ぎファイルのキーの指定とキューファイル回復の流れ



(13) mqafrc コマンドの使用例 (キューファイル格納媒体損傷以外の場合)

障害要因がキューファイル格納媒体損傷以外の場合の、mqafrc コマンドの使用例について説明します。使用例で指定する回復対象キューファイル定義ファイル (ファイル名 mappingfile) は、次に示すとおり記述します。

```
mqafilmap /dev/rds/rhd031/rcvf1
```

この定義例では、回復対象キューファイル名および回復先キューファイル名はともに /dev/rds/rhd031/rcvf1 です。回復先キューファイル名には、実際に回復させるキューファイル名を指定します。mqafrc コマンドを実行する場合には回復先キューファイルが存在している必要があります。キューファイル /dev/rds/rhd031/rcvf1 は障害発生直前の状態に回復されます。

mqafrc コマンドに指定するアンロードジャーナルファイル名は次に示すとおりとします。

jnlf001, jnlf002, …, jnlf00n… (n:世代番号)

(a) 使用例

ファイル分割されていないアンロードジャーナルファイルからキューファイルを回復する場合について説明します。

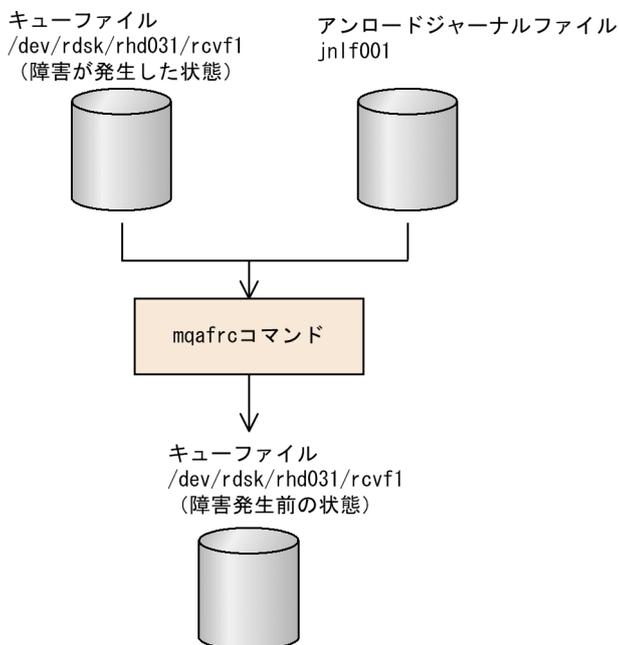
次に示すとおりコマンドを入力します。

```
mqafrc -s -e -f mappingfile /jnlf001
```

jnlf001 を入力として、回復先キューファイルが回復されます。

キューファイル回復の流れについて、次の図に示します。

図 2-12 キューファイル回復の流れ (キューファイル格納媒体損傷以外の場合)



(14) キューファイルの削除

キューファイルを削除する場合は、mqadel コマンドを使用します。mqadel コマンドを実行すると、指定したキューファイルに対応するメッセージ情報管理ファイルとキューファイルの二つが削除されます。

オンラインで使用中のキューファイルは削除できません。オンラインで使用中のキューファイルを指定した場合、エラーになります。

(15) キューファイル二重化構成

MQA サービス定義の mqaquegrp 定義コマンドで、指定したキューファイル名 (A 系ファイル名) に対するバックアップキューファイル名 (B 系ファイル名) を指定できます。この指定によって、A 系または B 系のどちらかのキューファイルに障害が発生すると、もう一方の系統のキューファイルを使用して運用

を行います。キューファイルの二重化は、ユーザアプリケーションが意識することなく行われます。また、キューファイルは、同期を取って更新しているので、A系またはB系だけに、メッセージが残ったり消えたりすることはありません。

障害が発生したファイルを回復する場合は `mquarcvr` コマンドを使用します。これによって、現在使用中のシステムキューファイルと情報が同じになります。

ファイル障害の詳細については、「[7.1.4 キューファイルの入出力エラー](#)」を参照してください。

注意事項

- A系またはB系の物理ファイルは複数のディスクに分散させて作成してください。一つのディスク上に同じシステムのすべての物理ファイルを作成すると、ディスクの全面障害が発生した場合、そのシステムは使用できません。
- 動的キューを作成するには、A系ファイル用の `mqaqueatl` 定義を MQA サービス定義ファイル (`$DCCONFPATH/mqa`) に作成してください。

キューファイル二重化構成は、次に示すどちらかの手順で定義および作成します。

手順 1

1. `mquabkup` コマンドでキューファイルをバックアップします。
2. 1.でバックアップしたキューファイルを `mquarstr` コマンドでリストアします。これがバックアップキューファイル (B系ファイル) になります。
3. 1.および2.で作成したキューファイルの名前を MQA サービス定義の `mqaquegrp` 定義コマンドに A系ファイル名およびB系ファイル名として指定します。

手順 2

1. `mquainit` コマンドで A系ファイルおよびB系ファイルを作成します。
同じオプション指定値でキューファイルを作成してください。
2. A系ファイルおよびB系ファイルについて、`mqaqueatl` 定義を作成します。
モデルキューの定義名およびキューファイル名以外のオプション指定値は同じにしてください。
A系ファイル用の `mqaqueatl` 定義およびB系ファイル用の `mqaqueatl` 定義は、キュー属性定義ファイル (UNIX テキストファイル) に作成してください。
3. `mquamkque` コマンドで同じ名前のキューを作成します。
4. 1.で作成したキューファイルの名前を MQA サービス定義の `mqaquegrp` 定義コマンドに A系ファイル名およびB系ファイル名として指定します。

なお、キューファイル二重化構成は、キューファイル二重化構成しない場合に比べて性能が低下します。

(16) キューファイルグループの使用状況表示

キューファイルグループの使用状況を表示する場合は、mqadf コマンドを使用します。表示されるキューファイルグループの使用状況の項目について、次に示します。

- キューファイルグループの全キューの個数
- 使用中のキューの個数
- 未使用キューの個数
- メッセージ格納領域の容量
- メッセージ格納領域の使用容量
- メッセージ格納領域の空き容量

(17) キューファイルの情報表示

キューファイル名の一覧およびキューファイルの情報を表示する場合は、mqafills コマンドを使用します。

mqafills コマンドで-x オプションを指定すると、MQA サービスで使用中のすべてのキューファイル名が表示されます。mqafills コマンドの-f オプションにキューファイル名を指定すると、指定したキューファイルの情報が表示されます。次に示す項目などが表示されます。

- キューファイル名
- キューファイルの状態
- 全キューの個数
- 使用中キューの個数
- 未使用キューの個数
- メッセージ長
- メッセージ情報管理ファイルのレコード長
- メッセージ情報管理ファイルのレコード数
- ファイル形態

(18) キューファイルのコンバート

キューファイルをコンバートする場合は、mqaconvert コマンドを使用します。過去のバージョンで使用していた TP1/Message Queue のキューファイルを、現在のバージョンで使用する場合は、mqaconvert コマンドでキューファイルをコンバートする必要があります。

mqaconvert コマンドを使用すると、mqabkup コマンドでバックアップした旧バージョンのキューファイル（メッセージ情報管理ファイル（キューファイル名.cnt）、およびメッセージファイル（キューファイル名.msg））を、最新バージョンのキューファイルへコンバートできます。なお、mqaconvert コマンドを使用する際は、次の点に注意してください。

- 古いバージョンへのコンバートはできません。
- このコマンドでコンバートできるのは、TP1/Message Queue 05-00 以降で作成したバックアップファイルです。

キューファイルのコンバートは、次に示す手順で行います。

1. 過去のバージョン環境で、mqabkup コマンドで変換元となるキューファイルをバックアップします。
2. 使用したいバージョン環境に変更します。
3. mqaconvert コマンドで、使用するキューファイルを変換します。

(19) メッセージの削除

キュー名に登録されているメッセージを削除する場合は、mqadelmsg コマンドを使用します。

(20) メッセージの表示

キュー名に登録されているメッセージ情報を表示する場合は、mqalsmsg コマンドを使用します。表示される情報の項目について、次に示します。

- MQMD 構造体の内容
- メッセージ内容

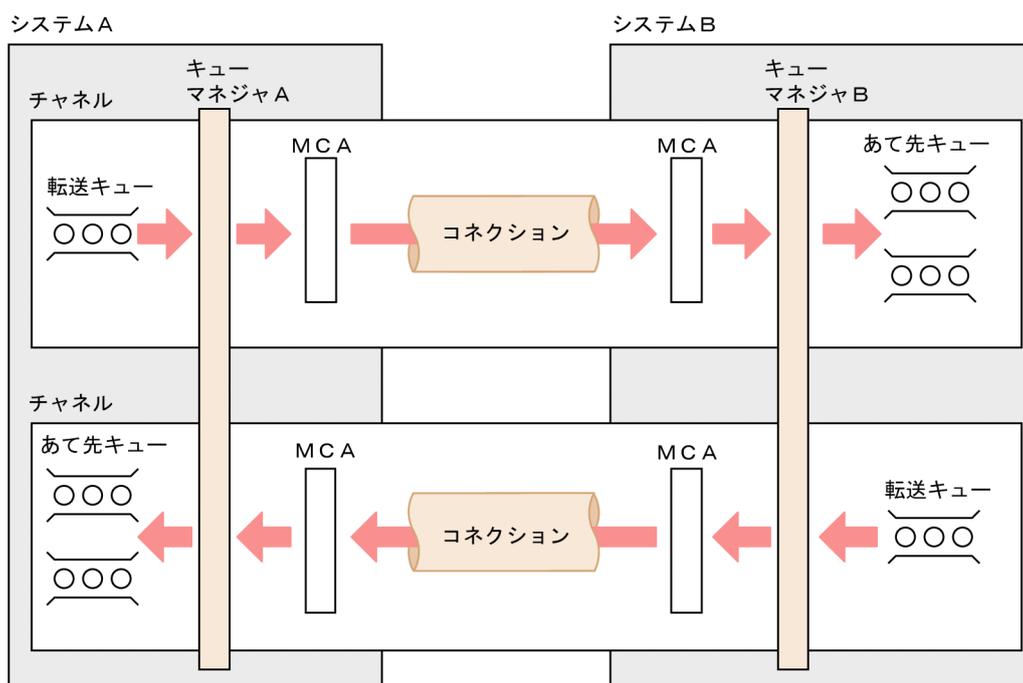
2.3 チャンネルの機能

TP1/Message Queue によるメッセージの送受信は、自システムと相手システムの間でチャンネルを確立することによって開始されます。チャンネルは、自システムと相手システムをつなぐ通信経路です。チャンネルの両端でチャンネルを管理および制御するプログラムを **MCA** といいます。

チャンネルを確立するには相手システムに対してチャンネル開始要求を送るか、または相手システムからチャンネル開始要求を受ける必要があります。チャンネル開始要求に対してチャンネル開始応答が返され、自システムと相手システムとの間で矛盾がなければ、チャンネルが確立されます。チャンネルが確立されるとメッセージの送受信が可能になります。

チャンネルの概念について、次の図に示します。

図 2-13 チャンネルの概念



一つのチャンネルには、通信相手システムと転送キューが一つずつ対応づけられます。一つのキューマネージャに対して複数のチャンネルを定義してもかまいません。

チャンネルにはあて先キューを対応づけないため、一つの転送キューに対して複数のリモートキューのローカル定義を対応づけると、一つのチャンネルで複数のあて先キューにメッセージを送信できます。

チャンネルの通信プロトコルは、チャンネルの機能に応じて FAP レベルが規定されています。自システムと相手システムとで、FAP レベルが異なる場合は、下位の FAP レベルの機能を使用して接続します。

なお、動作する FAP レベルと製品のバージョンとの対応を次に示します。

- FAP レベル 1 : TP1/Message Queue 01-xx^{*1}
- FAP レベル 4 : TP1/Message Queue 05-xx^{*1}, 06-xx^{*1}, 07-00, 07-01, および 07-5x^{*2}

注※1

xx には、任意の符号なし整数が入ります。

注※2

5x の x には、任意の符号なし整数が入ります。

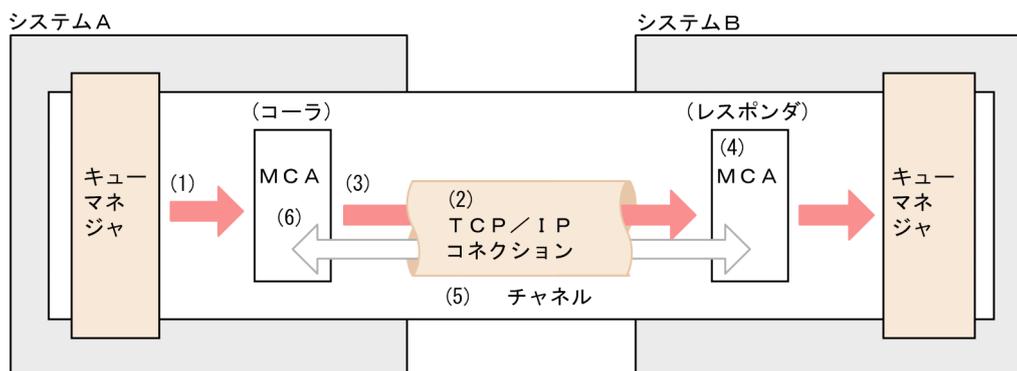
その他の MQ システムが動作する FAP レベルについては、使用する MQ システムが提供するマニュアルを参照してください。

2.3.1 コーラとレスポンド

チャンネル開始要求を送る側を**コーラ**、チャンネル開始要求を受ける側を**レスポンド**といいます。コーラ側の MCA がレスポンド側の MCA にチャンネル開始要求を送り、レスポンド側の MCA がコーラ側の MCA にチャンネル開始応答を返すことによって、チャンネルが確立されます。

チャンネルの開始処理の手順について、次の図に示します。

図 2-14 チャンネルの開始処理の手順



- (1) チャンネルを開始する要因が発生します。
- (2) TCP/IPコネクションが確立されます。
- (3) コーラはレスポンドにチャンネルの開始要求を出します。レスポンドはこれを受けます。
- (4) レスポンドはコーラからのチャンネルの属性と自分のチャンネルの属性をネゴシエーションします。
- (5) ネゴシエーションの結果、矛盾がなければチャンネルが確立されます。
- (6) メッセージを送受信します。

チャンネルの開始方法については、「[2.3.5 チャンネルの開始](#)」を参照してください。

2.3.2 MCA のチャンネルタイプ

チャンネルを制御する MCA には幾つかの**チャンネルタイプ**があります。チャンネルタイプが異なるとチャンネルを制御する MCA の動作も異なります。

TP1/Message Queue がサポートするチャンネルタイプと MCA の動作について、次に示します。チャンネルタイプは TCP 定義の mqtalccha 定義コマンドで指定します。

センド (sender)

通信相手システムのレシーバまたはリクエストをチャンネル開始要求で起動し、メッセージを転送キューから取り出してレシーバまたはリクエストに送信します。通信相手システムのリクエストから起動された場合は、センドから起動をし直します。

レシーバ (receiver)

通信相手システムのセンドまたはサーバからのチャンネル開始要求で起動され、センドまたはサーバからのメッセージを受信します。受信後、メッセージに指定されたあて先キューへ登録します。

リクエスト (requester)

通信相手システムのセンドまたはサーバをチャンネル開始要求で起動し、センドまたはサーバのチャンネル定義で指定された転送キューからのメッセージを受信します。受信後、メッセージに指定されたあて先キューへ登録します。センドまたはサーバからのチャンネル開始要求によって起動することもできます。

サーバ (server)

通信相手システムのリクエストからチャンネル開始要求で起動され、メッセージを転送キューから取り出して、レシーバまたはリクエストに送信します。通信相手システムを設定すると、レシーバまたはリクエストをチャンネル開始要求で起動できます。

クラスタセンド (clusdr)

クラスタ機能で使用する送信側のチャンネルタイプです。TCP 定義の mqtalcccha 定義コマンドで指定する事前定義クラスタセンドと、接続相手のクラスタレシーバの属性に応じて TP1/Message Queue が自動的に作成する自動定義クラスタセンドがあります。詳細については、「[2.8 キューマネージャクラスタの機能](#)」を参照してください。

クラスタレシーバ (clusrcvr)

クラスタ機能で使用する受信側のチャンネルタイプです。TCP 定義の mqtalcccha 定義コマンドで指定する事前定義クラスタレシーバと、事前定義クラスタレシーバの属性で TP1/Message Queue が自動的に作成する自動定義クラスタレシーバ (チャンネルインスタンス) があります。なお、事前定義クラスタレシーバには、実体はありません。詳細については、「[2.8 キューマネージャクラスタの機能](#)」を参照してください。

コーラ側 MCA になることができるチャンネルタイプは、センド、サーバ、リクエスト、またはクラスタセンドです。レスポンド側 MCA になることができるチャンネルタイプは、センド、サーバ、レシーバ、リクエスト、またはクラスタレシーバです。チャンネルは、コーラ側 MCA とレスポンド側 MCA を 1 対 1 で組み合わせます。一つの MCA に複数の MCA を組み合わせると、メッセージが二重に送信されたり、欠落したりすることがあります。

チャンネルタイプの組み合わせについて、次の表に示します。

チャンネルタイプの組み合わせは、システム設計時の最も早期に決定する必要がある検討事項です。

表 2-4 チャンネルタイプの組み合わせ

コーラ側 MCA のチャンネルタイプ	メッセージの送受信方向	レスポンド側 MCA のチャンネルタイプ
センド	→	レシーバ

コーラ側 MCA のチャンネルタイプ	メッセージの送受信方向	レスポンド側 MCA のチャンネルタイプ
センダ	→	リクエスト
サーバ	→	リクエスト
サーバ	→	レシーバ
リクエスト	←	センダ※
リクエスト	←	サーバ
クラスタセンダ	→	クラスタレシーバ

(凡例)

- ：コーラから送信します。
- ←：レスポンドから送信します。

注※

センダは、リクエストからのネットワークリクエストに対して、いったんチャンネルの終了を要求して、TCP/IP コネクションを解放します。その後、センダから再度コーラとしてチャンネルを確立します。

クラスタチャンネルでは一つのチャンネル定義に対して複数の MCA を動作させるマルチインスタンス機能を持っています。詳細については、「[2.8.9 クラスタ環境の通信構成](#)」を参照してください。

2.3.3 チャンネル状態

チャンネルはチャンネル確立中、チャンネル解放中など幾つかの異なる状態を持ちます。相手システムとの通信状態を確認するには、チャンネル状態を確認してください。チャンネル状態を確認するには `mqtlscha` コマンドを使用します。

チャンネルには次に示す状態があり、コマンドの入力や通信障害などのイベントによって遷移します。

チャンネル停止

MCA が停止している状態

- チャンネルタイプがセンダ、レシーバ、リクエスト、またはサーバ
`mqtalccha` 定義コマンドの `-i` オプションでチャンネルの自動確立をするよう指定した場合、TP1/Message Queue の開始直後に、この状態になります。
- 事前定義チャンネル（チャンネルタイプがクラスタセンダ、またはクラスタレシーバ）
`mqtalccha` 定義コマンドで事前定義されたクラスタチャンネルは、`mqtalccha` 定義コマンドの `-i` オプションでチャンネルの自動確立をするよう指定した場合、TP1/Message Queue の開始直後に、この状態になります。
- 自動定義チャンネル（自動定義クラスタセンダ、またはクラスタレシーバインスタンス）
自動的にクラスタチャンネルが作成される時、デフォルトチャンネル定義の `mqtalccha` 定義コマンドの `-i` オプションでチャンネルの自動確立をするよう指定した場合、またはデフォルトチャンネル定義を

指定しない場合、自動作成直後およびそれ以降の TP1/Message Queue の開始直後に、この状態になります。

MCA 起動中

開始を要求されたが、MCA がまだ起動していない状態

チャンネル確立中

TCP/IP コネクションの確立中、およびネゴシエーション中の状態

チャンネル動作中

メッセージ転送中、または転送キューへのメッセージの登録を待っている状態

チャンネル解放中

チャンネルの解放処理中、またはチャンネルの終了要求を受け付けた状態

チャンネル確立リトライ中

コネクション障害、時間監視障害、または MCP 障害が発生したため、チャンネル確立時再試行間隔で確立の再試行契機を待っている状態

チャンネル使用不可

mqtstпча コマンドの入力、チャンネル確立の再試行回数超過、または障害発生でチャンネルを解放して使用できない状態

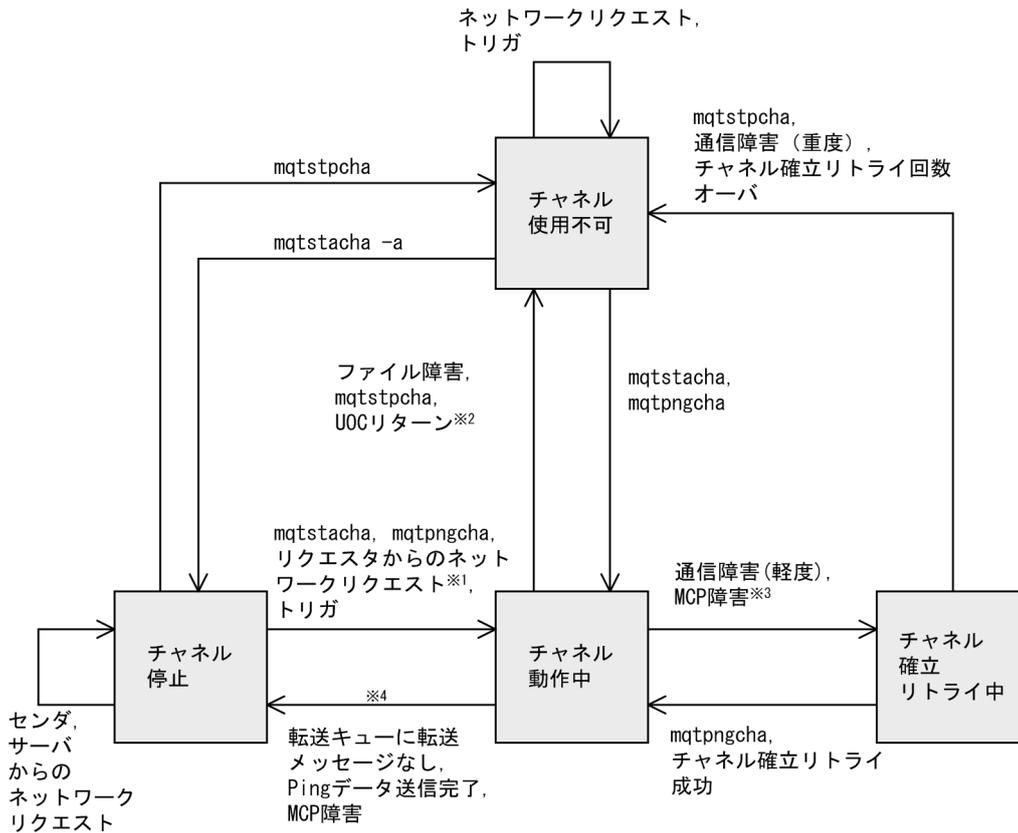
- チャンネルタイプがセンダ、レシーバ、リクエスタ、またはサーバ
mqtalच्cha 定義コマンドの-i オプションでチャンネルの自動確立をしないよう指定した場合、TP1/Message Queue の開始直後に、この状態になります。
- 事前定義チャンネル（チャンネルタイプがクラスタセンダ、またはクラスタレシーバ）
mqtalच्cha 定義コマンドで事前定義されたクラスタチャンネルは、mqtalच्cha 定義コマンドの-i オプションでチャンネルの自動確立をしないよう指定した場合、TP1/Message Queue の開始直後に、この状態になります。
- 自動定義チャンネル（自動定義クラスタセンダ、またはクラスタレシーバインスタンス）
自動的にクラスタチャンネルが作成される時、デフォルトチャンネル定義の mqtalच्cha 定義コマンドの-i オプションでチャンネルの自動確立をしないよう指定した場合、自動作成直後およびそれ以降の TP1/Message Queue の開始直後に、この状態になります。

チャンネル確立リクエスト中

チャンネルタイプがリクエスタの場合で、MCA がリモート MCA からのチャンネルの開始要求を待っている状態

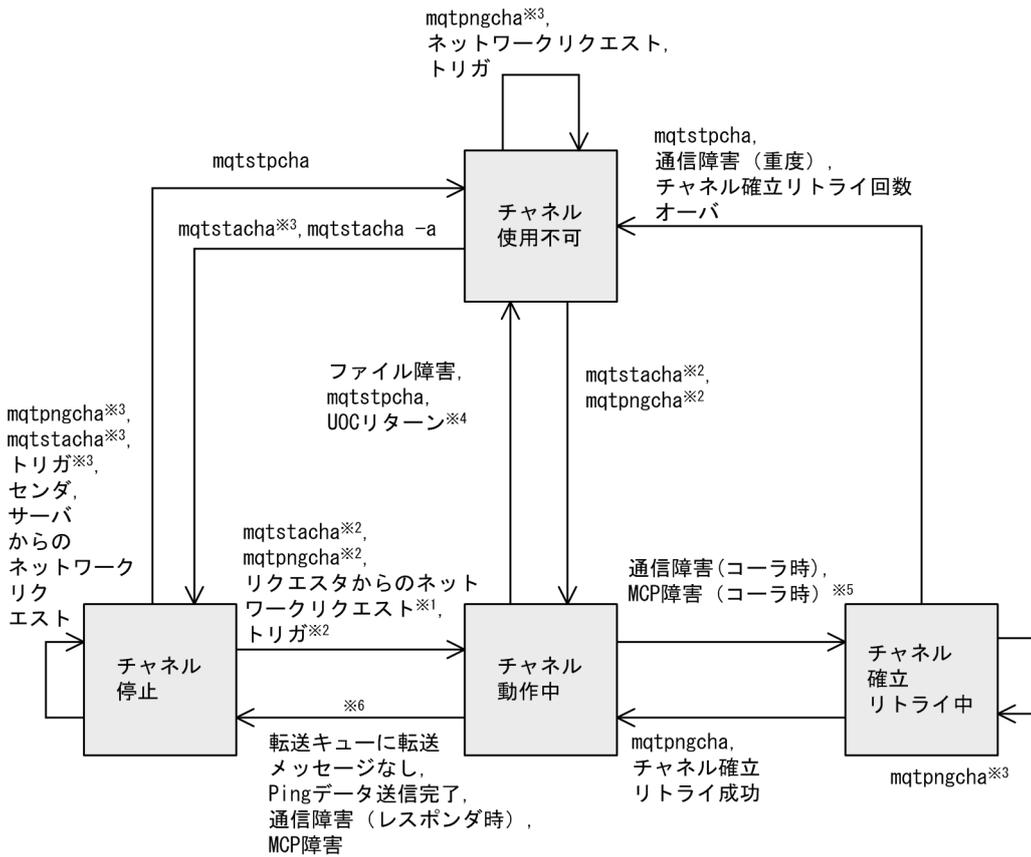
チャンネルタイプ別の状態遷移について、以降の図に示します。

図 2-15 チャンネルの状態遷移 (センダ)



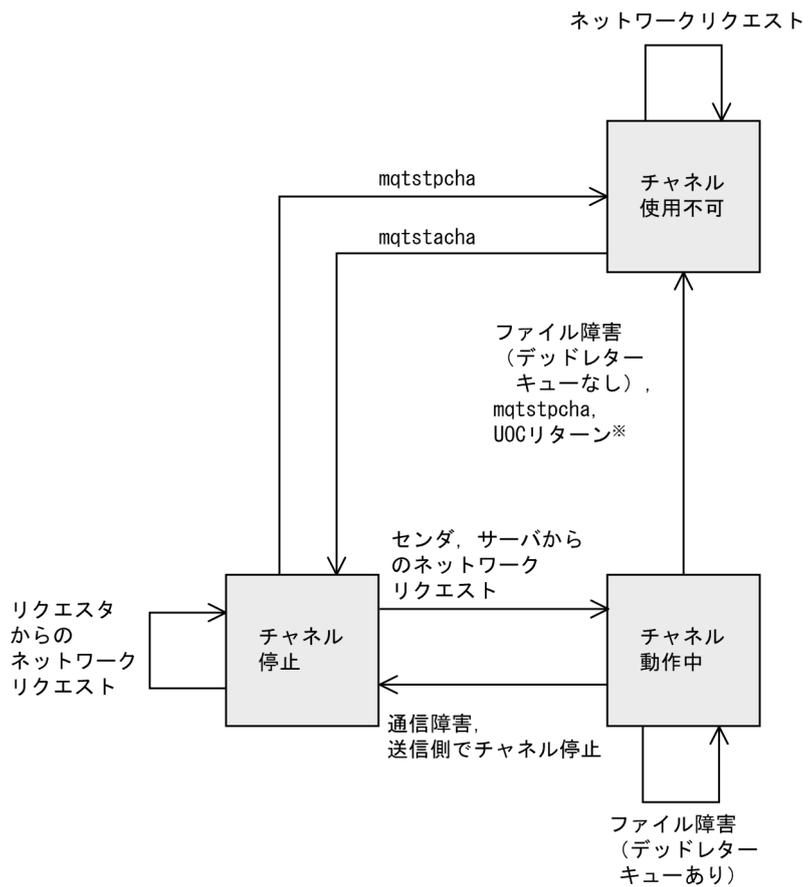
- 注※1 確立されたコネクションを使用しないで、自システムのチャンネル定義に指定された相手システムとコネクションを再確立します。
- 注※2 UOC/パラメタ (ExitResponse) にDCMTCQ_MQXCC_CLOSE_CHANNELが返された場合
- 注※3 mqtalccha定義コマンドの-bオプションにbretrymcp=yesを指定する場合
- 注※4 受信側チャンネルに対してメッセージ転送中にチャンネルを終了させるコマンドを実行した場合、またはハートビート機能を使用している受信側チャンネルに対してメッセージ転送を行っていないときにチャンネルを終了させるコマンドを実行した場合

図 2-16 チャンネルの状態遷移 (サーバ)



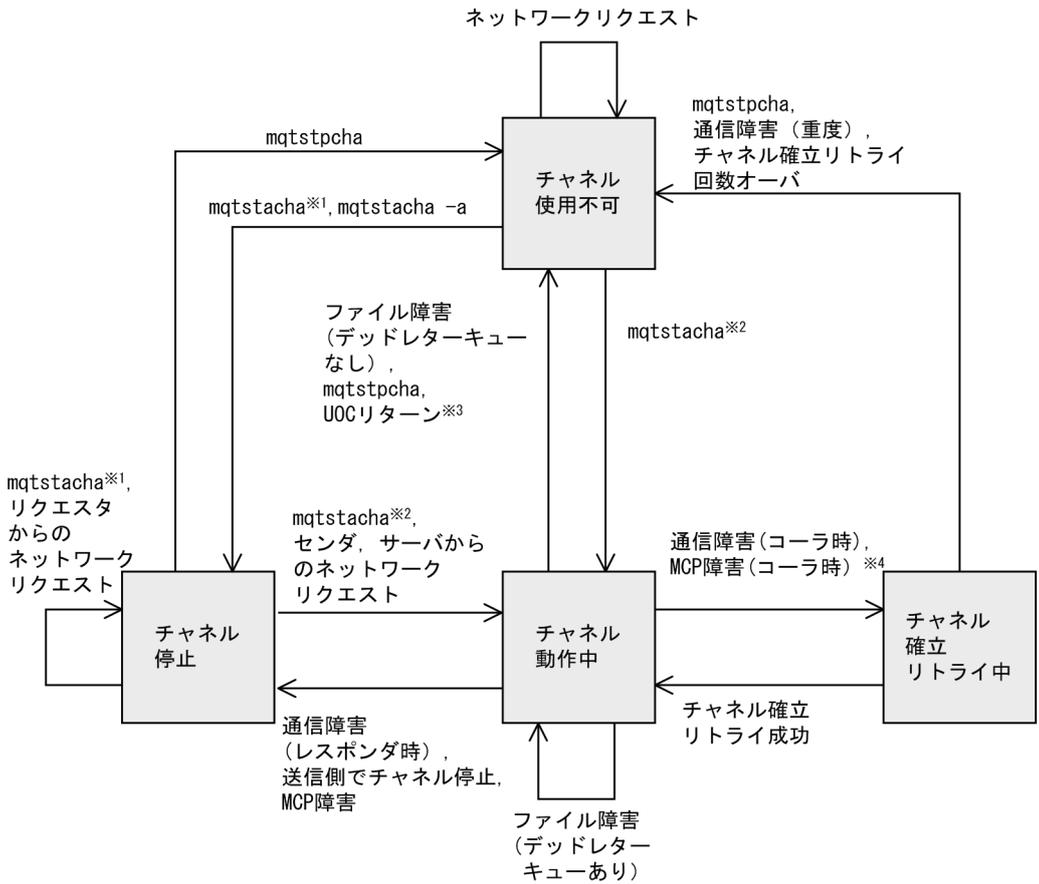
- 注※1 確立されたコネクションを使用します。
- 注※2 チャンネル定義に相手IPアドレス, ポート番号の指定がある場合
- 注※3 チャンネル定義に相手IPアドレス, ポート番号の指定がない場合
- 注※4 UOC/パラメタ (ExitResponse) にDCMTCQ_MQXCC_CLOSE_CHANNELが返された場合
- 注※5 mqtalocha定義コマンドの-bオプションにbretrymcp=yesを指定する場合
- 注※6 受信側チャンネルに対してメッセージ転送中にチャンネルを終了させるコマンドを実行した場合, またはハートビート機能を使用している受信側チャンネルに対してメッセージ転送を行っていないときにチャンネルを終了させるコマンドを実行した場合

図 2-17 チャンネルの状態遷移 (レシーバ)



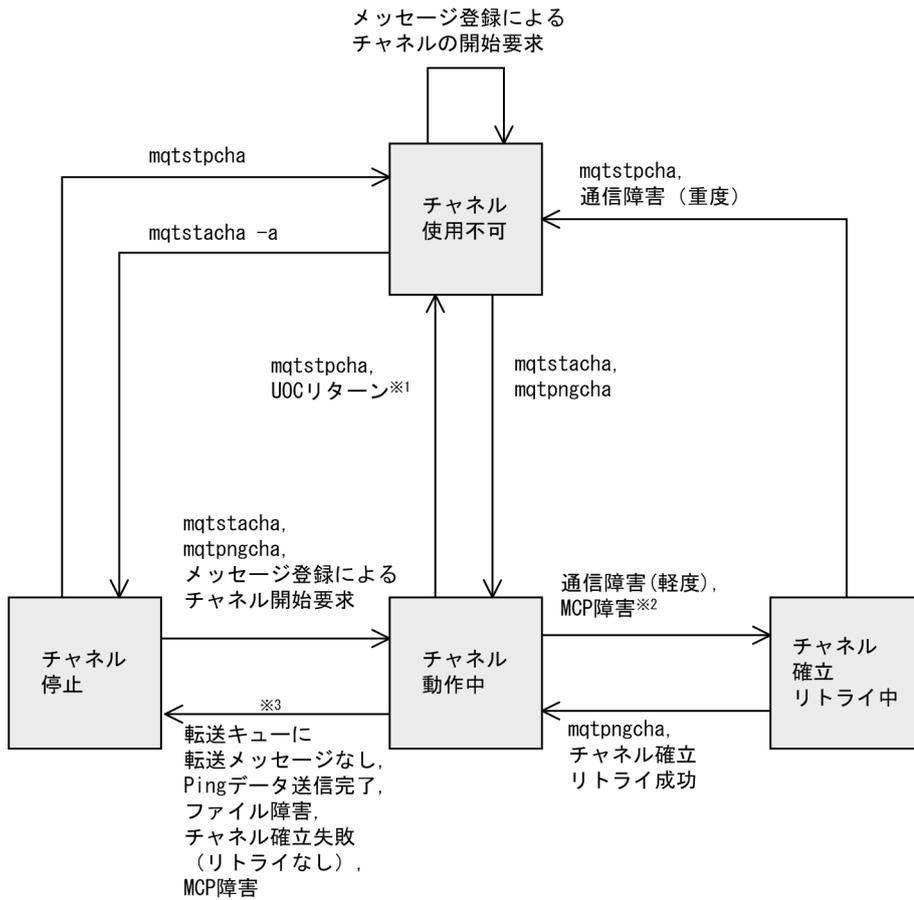
注※ UOCパラメタ (ExitResponse) にDCMTCQ_MQXGC_CLOSE_CHANNELが返された場合

図 2-18 チャンネルの状態遷移 (リクエスタ)



- 注※1 チャンネル定義に相手IPアドレス, ポート番号の指定がない場合
- 注※2 チャンネル定義に相手IPアドレス, ポート番号の指定がある場合
- 注※3 UOC/パラメタ (ExitResponse) にDCMTCQ_MQXCC_CLOSE_CHANNELが返された場合
- 注※4 mqtalocha定義コマンドの-bオプションにbretrymcp=yesを指定する場合

図 2-19 チャンネルの状態遷移 (クラスタセンダ)

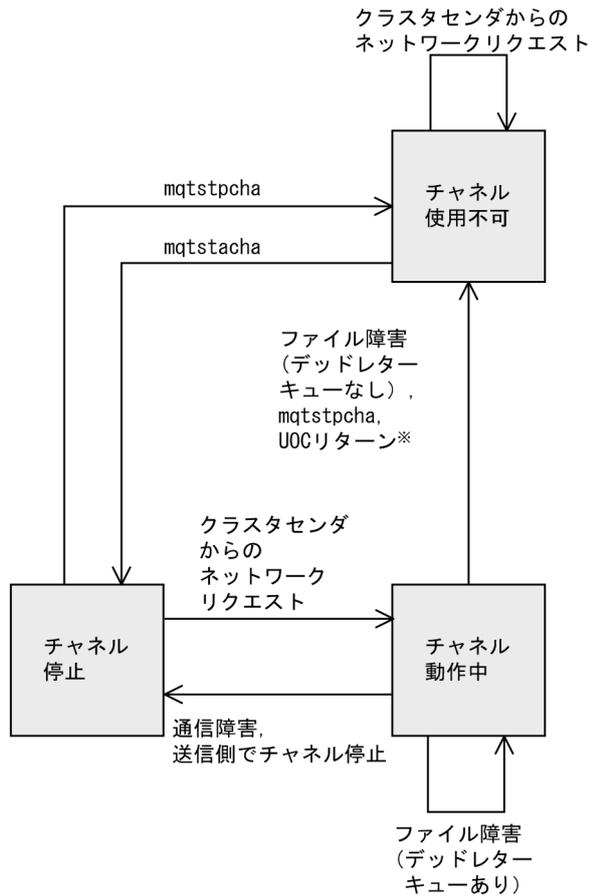


注※1 UOCパラメタ (ExitResponse) にDCMTCQ_MQXCC_CLOSE_CHANNELが返された場合

注※2 mqatalocha定義コマンドの-bオプションにbretrymcp=yesを指定する場合

注※3 受信側チャンネルに対してメッセージ転送中にチャンネルを終了させるコマンドを実行した場合、またはハートビート機能を使用している受信側チャンネルに対してメッセージ転送を行っていないときにチャンネルを終了させるコマンドを実行した場合

図 2-20 チャンネルの状態遷移 (クラスタレシーバ)



注※ UOCパラメタ (ExitResponse) にDCMTCQ_MQXCC_CLOSE_CHANNELが返された場合

2.3.4 MQT サーバのトリガ起動によるチャンネル開始

転送キューにメッセージが登録されたことを MQT サーバが検出し、チャンネルを開始する機能です。

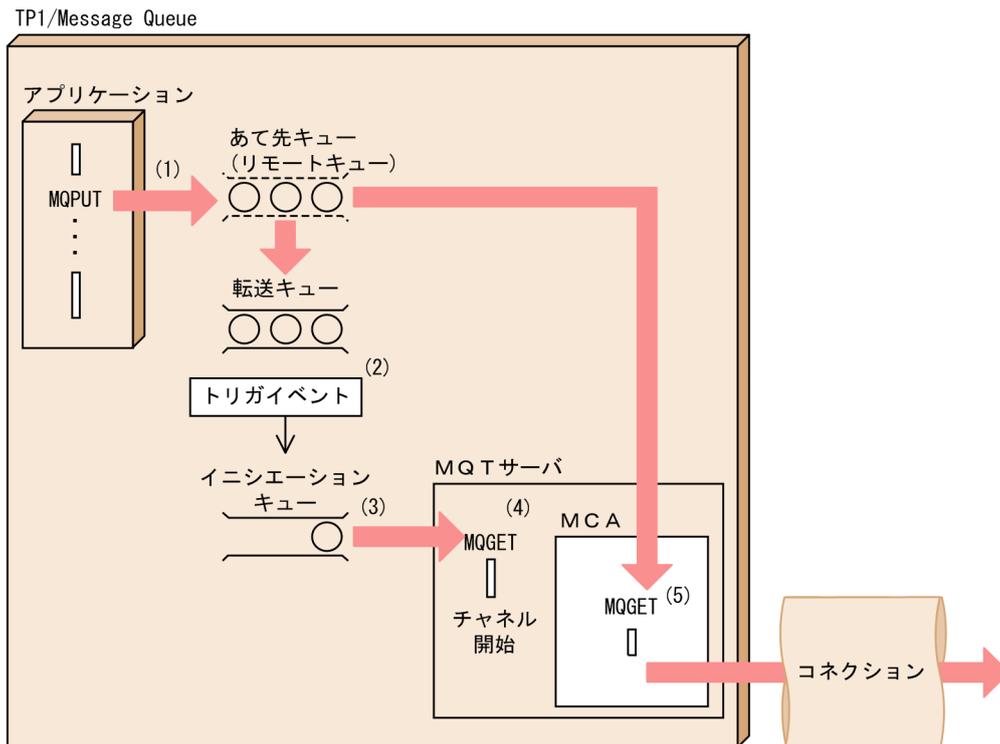
チャンネルタイプがセンダまたはサーバの場合、転送キューにメッセージが登録されると、イニシエーションキューにトリガメッセージが登録されます。MQT サーバはイニシエーションキューを監視することによって、転送キューへのメッセージ登録を検出して、MCA を起動します。イニシエーションキューは一つの MQT サーバに対して一つだけ指定できます。トリガ起動を使用するチャンネルは、使用するすべての転送キューを一つのイニシエーションキューに割り当ててください。一つのイニシエーションキューを複数の MQT サーバに割り当てることはできません。

チャンネルタイプがクラスタセンダの場合、イニシエーションキューではなく、チャンネル要求監視間隔 (mqtstpcs 定義コマンド-v stim オペランド指定値) でクラスタ転送キューを監視し、メッセージ登録時に決定した最適な MCA を起動します。そのため、メッセージを登録しても、-v stim オペランドで指定する次の監視時まで、チャンネルが開始されないことがあります。

転送キューと対応づけられたチャンネルで「チャンネル停止」の状態のコラ側 MCA が起動され、チャンネル開始要求を相手システムに送ってチャンネルを確立します。

トリガ起動によるチャンネルの開始について、次の図に示します。

図 2-21 MQT サーバのトリガ起動によるチャンネルの開始



- (1) アプリケーションがあて先キュー (リモートキュー) あてにメッセージの登録を要求すると、転送キューにメッセージが登録されます。
- (2) 転送キューの状態が設定した条件 (トリガタイプ: first) を満たすと、トリガイイベントが発生します。
転送キューのトリガ起動に必要な値および推奨値を次に示します。
mqaprcdef (プロセス定義)
プロセス名: 指定必須
mqaqueatl (モデルキューの属性定義)
-a (プロセス名): 指定必須
mqaprcdefのプロセス名と同じ名前を指定してください。
-i (イニシエーションキュー名): 指定必須
mqttcp (TCP構成定義) の-iオプションと同じ名前を指定してください。
-c on (トリガ制御): 指定必須
-t first (トリガタイプ): 推奨値
- (3) トリガイイベントが発生したことを意味するトリガメッセージをイニシエーションキューに登録します。
イニシエーションキューに必要な値および推奨値を次に示します。
mqaqueatl (モデルキューの属性定義)
-o shared (省略時の取り出しオプション): 推奨値
-m fifo (メッセージ配布順序): 指定必須
- (4) MQTサーバは、MQGET 命令を呼び出してイニシエーションキューからトリガメッセージを取り出します。すると、MCA (チャンネルタイプ: センダまたはサーバ) によってチャンネルが開始されます。
- (5) MCAは、MQGET 命令を呼び出して転送キューからメッセージを取り出して転送します。

イニシエーションキューの監視方法については、「[2.5.4 イニシエーションキュー監視](#)」を参照してください。

2.3.5 チャネルの開始

チャネルを開始する方法を次に示します。チャネルの開始後、チャネルの確立に成功するとメッセージの送受信が開始されます。

- 運用コマンド

mqtstacha コマンドを入力する場合

相手システム（TCP 定義の mqtalcccha 定義コマンドの-o オプション指定値）が指定されているとき、チャネル開始要求を相手システムに送ってチャネルを確立します。相手システムが指定されていないとき、「チャネル使用不可」の状態を「チャネル停止」の状態に遷移させることができます。

mqtstacha コマンドに-a オプションを指定して入力する場合

「チャネル使用不可」状態になっている MCA を、トリガ起動やネットワークリクエストによってチャネル開始のできる「チャネル停止」の状態に遷移させることができます。

- トリガ起動（自システムがコーラの場合）

転送キューにメッセージが登録されたことを MQT サーバが検出し、MCA がチャネル開始要求を相手システムに送ってチャネルを確立します。

詳細については、「[2.3.4 MQT サーバのトリガ起動によるチャネル開始](#)」を参照してください。

- ネットワークリクエスト（自システムがレスポンドの場合）

相手システムからチャネル開始要求を受けることをネットワークリクエストといいます。

「チャネル停止」の状態の MCA は、相手システムからのチャネル開始要求を受けると、チャネル開始応答を返してチャネルを確立します。

クラスタセンダ以外のすべてのチャネルタイプがレスポンド側 MCA になることができます。

- システム起動

mqtalcccha 定義コマンドの-i オプションに auto2 を指定した場合、MQT サーバのサービスが開始された直後に、自動的にチャネルが開始されます。ただし、相手システムが指定されていない場合は開始されません。

チャネルの開始方法とチャネル定義の指定値による動作の違いについて、次の表に示します。

表 2-5 チャネルの開始方法とチャネル定義の指定値による動作

チャネル開始方法	チャネルタイプ	コーラ/ レスポンド	通信相手の指定※1	動作
運用コマンド (mqtstacha)	センダ (sender)	コーラ	必ず指定	<ul style="list-style-type: none">• チャネルの確立（チャネル開始要求の送信）※2• メッセージ送信
	サーバ (server)	コーラ	指定あり	<ul style="list-style-type: none">• チャネルの確立（チャネル開始要求の送信）※2• メッセージ送信
		レスポンド	指定なし	<ul style="list-style-type: none">• ネットワークリクエストを待ちます。

チャンネル開始方法	チャンネルタイプ	コーラ/ レスポнда	通信相手の指定※1	動作
運用コマンド (mqststacha)	レシーバ (receiver)	レスポнда	—	<ul style="list-style-type: none"> ネットワークリクエストを待ちます。
	リクエスタ (requester)	コーラ	指定あり	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始要求の送信) ※2 メッセージ受信
		レスポнда	指定なし	<ul style="list-style-type: none"> ネットワークリクエストを待ちます。
	クラスタセンダ (cluscdr)	コーラ	必ず指定 (自動定義される)	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始要求の送信) ※2 メッセージ送信
	クラスタレシーバ (clusrcvr)	レスポнда	—	<ul style="list-style-type: none"> ネットワークリクエストを待ちます。
トリガ 起動	センダ (sender)	コーラ	必ず指定	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始要求の送信) メッセージ送信
	サーバ (server)	コーラ	指定あり	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始要求の送信) メッセージ送信
			指定なし	— (コネクションを持たないため起動されません)
	レシーバ (receiver)	—	—	— (転送キューを持たないため対象外)
	リクエスタ (requester)	—	—	— (転送キューを持たないため対象外)
	クラスタセンダ (cluscdr)	コーラ	必ず指定 (自動定義される)	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始要求の送信) メッセージ送信
	クラスタレシーバ (clusrcvr)	—	—	— (転送キューを持たないため対象外)
ネット ワーク リクエ スト	センダ (sender)	レスポнда	必ず指定	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始応答の送信) 後解放※3
		コーラ		<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始要求の送信) メッセージ送信
	サーバ (server)	レスポнда	—	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始応答の送信) メッセージ送信
	レシーバ (receiver)	レスポнда	—	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始応答の送信) メッセージ受信
	リクエスタ (requester)	レスポнда	—	<ul style="list-style-type: none"> チャンネルの確立 (チャンネル開始応答の送信) メッセージ受信

チャンネル開始方法	チャンネルタイプ	コーラ/ レスポнда	通信相手の指定※1	動作
ネットワーク ワーク リクエ スト	クラスタセンダ (cluscdr)	—	—	— (クラスタセンダは必ずコーラ側になるため対象外)
	クラスタレシー バ (clusrcvr)	レスポнда	—	<ul style="list-style-type: none"> • チャンネルの確立 (チャンネル開始応答の送信) • メッセージ受信
システム 起動	センダ (sender)	コーラ	必ず指定	<ul style="list-style-type: none"> • チャンネルの確立 (チャンネル開始要求の送信) • メッセージ送信
	サーバ (server)	コーラ	指定あり	<ul style="list-style-type: none"> • チャンネルの確立 (チャンネル開始要求の送信) • メッセージ送信
		レスポнда	指定なし	— (システム起動しないため対象外)
	レシーバ (receiver)	—	—	— (システム起動しないため対象外)
	リクエスタ (requester)	コーラ	指定あり	<ul style="list-style-type: none"> • チャンネルの確立 (チャンネル開始要求の送信) • メッセージ受信
		レスポнда	指定なし	— (システム起動しないため対象外)
	クラスタセンダ (cluscdr)	コーラ	必ず指定	<ul style="list-style-type: none"> • チャンネルの確立 (チャンネル開始要求の送信) • メッセージ送信
	クラスタレシー バ (clusrcvr)	—	—	— (システム起動しないため対象外)

(凡例)

— : 該当しません。

注※1

TCP 定義の mqtalcha 定義コマンドの-o オプションの指定です。

注※2

チャンネル状態が「チャンネル使用不可」の場合に mqtstacha コマンドに-a オプションを入力して実行すると、チャンネルの確立とメッセージ送受信の処理はしないで、チャンネル状態は「チャンネル停止」になります。

注※3

チャンネルタイプがセンダの MCA がネットワークリクエストによって起動されると、ネットワークリクエストを受信したコネクションをいったん解放して、再度センダ MCA のチャンネル開始要求によって、センダ MCA のチャンネル定義に指定された通信相手とチャンネルを確立します。

2.3.6 チャネルの確立

コーラ側 MCA からレスポンド側 MCA にチャネル開始要求が送られたあと、両方の MCA の間でチャネルの構成がチェックされます。チェックの結果、矛盾がなければ、レスポンド側 MCA からコーラ側 MCA にチャネル開始応答が返されチャネルが確立されます。

チャネル確立時に、相手システムとチャネルの構成をチェックして合わせることを**ネゴシエーション**といいます。自システムのチャネルの構成は、TCP 定義の `mqtalccha` 定義コマンドで指定した属性です。ネゴシエーションの結果、定義で指定した値が変更されることがあります。ネゴシエーションを行うチャネル属性については、4章の「[チャネルのネゴシエーション](#)」を参照してください。

- コーラの場合

コーラは、コーラのチャネルの構成をレスポンドに提案して了解させます。提案したチャネルの構成が、レスポンドの MCA から受け入れを拒否された場合、コーラ側でレスポンドに合わせることであれば修正し、再度確立を要求します。変更できない属性について受け入れを拒否された場合、またはレスポンド側で確立不可能と判断されて**チャネル確立拒否応答**が返された場合、KFCA16341-E または KFCA16342-E が出力され、チャネルの確立を中止します。

- レスポンドの場合

レスポンドは、コーラからのチャネルの構成とレスポンド側のチャネルの構成を比較します。比較の結果、矛盾がなければチャネルを確立します。矛盾がある場合、チャネルの確立要求を受け入れないで、受け入れられる値をコーラに連絡します。また、チャネル確立不可能と判断した場合、チャネル確立拒否応答を送信し、コーラにチャネルの確立を中止させます。

`mqtpngcha` コマンドを使用すると、チャネルの確立ができるかどうかを確認できます。送信するデータ (Ping データ) の長さを指定して通信相手システムに送信すると、指定した Ping データの長さを 1 セグメントとして転送できるかどうかを確認できます。

チャネルのテスト接続は次に示す場合に使用できます。

- チャネルタイプがセンダ、サーバ、またはクラスタセンダ
- チャネル状態が「チャネル停止」、「チャネル使用不可」、または「チャネル確立リトライ中」

2.3.7 チャネル確立再試行

チャネル確立再試行は、チャネルに通信障害が発生した時にチャネルを自動的に再確立する機能です。再試行には短期確立再試行と長期確立再試行があります。短期確立再試行および長期確立再試行の機能は、お互いに独立した機能です。それぞれを、単独で指定したり同時に指定したりできます。なお、同時に指定した場合は、短期確立再試行が終わってから長期確立再試行を実行します。

短期確立再試行の機能を使用する場合、TCP 定義の `mqtalccha` 定義コマンドの `-b` オプションに次に示す指定をしてください。

- 短期確立再試行の要否 (`bretry=yes`)

- 短期確立再試行回数 (bretrycnt=任意の値)
- 短期確立再試行間隔 (bretryint=任意の値)

長期確立再試行の機能を使用する場合、TCP 定義の mqtalcccha 定義コマンドの-b オプションに次に示す指定をしてください。

- 長期確立再試行の要否 (bretrylg=yes)
- 長期確立再試行回数 (bretrylgcnt=任意の値)
- 長期確立再試行間隔 (bretrylgint=任意の値)

ただし、確立再試行回数×確立再試行間隔(秒)の時間は、実際に確立再試行を実行する時間と必ずしも一致しません。これは、実際に確立再試行を実行する時間にチャンネルの確立処理時間を含むためです。特に、クラスタセンダチャンネルでは、実際に確立再試行を実行する時間はさらに長くなります。チャンネル確立再試行を実行する前に、メッセージ送信経路の再設定処理が完了するのを mqttcps 定義コマンド-v stim オペランドに指定した間隔で確認するためです。また、確立再試行間隔は MQTT 共通定義の mqtttim 定義コマンドの-t btim オペランドおよび bmtim オペランドで指定する基本タイマ値の影響を受けるため、動作間隔に誤差が発生することがあります。

チャンネル確立の再試行が動作するのは、次に示す通信障害が発生した場合です。

- **コネクション障害**

チャンネル状態が「チャンネル確立中」、「チャンネル動作中」、「チャンネル確立リトライ中」、または「チャンネル確立リクエスト中」の場合に、TCP/IP ソケット関数が軽度障害でエラーリターンしたときです。軽度障害とは、エラーナンバ ENOBUFS (Windows の場合は、WSAENOBUFS) 以外のエラーのことです。なお、重度障害とは、ENOBUFS (Windows の場合は、WSAENOBUFS) エラーのことであり、メモリ不足によって発生します。

- **時間監視障害**

チャンネル状態が「チャンネル確立中」、「チャンネル動作中」、または「チャンネル確立リクエスト中」の場合に、各種メッセージ受信監視タイマ (TCP 定義の mqtalcccha 定義コマンドの-v オプション指定値) がタイムアウトしたときです。TCP 定義の mqtalcccha 定義コマンドの-v オプションに vretry=yes を指定している場合、確立再試行が実行されます。vretry=no を指定している場合、チャンネルは終了します。

- **MCP 障害**

チャンネル状態が「チャンネル確立中」、「チャンネル動作中」、または「チャンネル確立リクエスト中」の場合に、MCP 障害が発生したときです。TCP 定義の mqtalcccha 定義コマンドの-b オプションに bretrymcp=yes を指定している場合、確立再試行が実行されます。bretrymcp=no を指定している場合、チャンネルは終了します。

(1) 確立再試行の詳細

通信障害が発生すると、KFCA16318-I (コネクション確立再試行開始) が出力されて確立再試行を実行します。

mqtalccha 定義コマンドの-b オプションに bretrymcp=no を指定した場合、次に示す処理のどれかが発生した時点で確立再試行処理は完了します。

- TCP/IP コネクションが確立した
- 指定回数分繰り返しても TCP/IP コネクションを確立できなかった
- メモリ不足が発生した
- mqtstpcha コマンドまたは dcstop コマンドが実行された

mqtalccha 定義コマンドの-b オプションに bretrymcp=yes を指定した場合、次に示す処理のどれかが発生した時点で確立再試行処理は完了します。

- チャンネルが確立した
- 指定回数分繰り返してもチャンネルを確立できなかった
- メモリ不足が発生した
- mqtstpcha コマンドまたは dcstop コマンドが実行された

チャンネルを確立するまでに発生する通信障害、出力されるログメッセージ、および再試行処理内容を次の表に示します。

表 2-6 再試行処理（チャンネル確立前）

通信障害の内容	出力するログメッセージ	mqtalccha 定義コマンドの-b オプションまたは-v オプション			
		bretry=yes または bretrylg=yes vretry=yes bretrymcp=yes	bretry=yes または bretrylg=yes vretry=no bretrymcp=yes	bretry=yes または bretrylg=yes vretry=yes bretrymcp=no	bretry=yes または bretrylg=yes vretry=no bretrymcp=no
コネクション障害 (TCP/IP エラー)	KFCA16331-E (TCP/IP インタフェースエラー)	◎	◎	○	○
時間監視障害 (開始応答受信監視タイムアウト)	KFCA16329-E (受信監視タイムアウト) 監視対象=init	◎	×	○	×
MCP 障害 (チャンネル属性不一致)	KFCA16341-E (開始要求応答受け入れ不可) 理由コード=00, または理由コード 15	◎	◎	×	×
MCP 障害	KFCA16341-E	◎	◎	×	×

通信障害の内容	出力するログメッセージ	mqtalcccha 定義コマンドの-b オプションまたは-v オプション			
		bretry=yes または bretrylg=yes vretry=yes bretrymcp=yes	bretry=yes または bretrylg=yes vretry=no bretrymcp=yes	bretry=yes または bretrylg=yes vretry=yes bretrymcp=no	bretry=yes または bretrylg=yes vretry=no bretrymcp=no
(該当するチャンネルがない)	(開始要求応答受け入れ不可) 理由コード=01	◎	◎	×	×
MCP 障害 (チャンネルタイプの組み合わせが不正)	KFCA16341-E (開始要求応答受け入れ不可) 理由コード=02	◎	◎	×	×
MCP 障害 (相手キューマネージャが動作していない)	KFCA16341-E (開始要求応答受け入れ不可) 理由コード=03	◎	◎	×	×
MCP 障害 (相手キューマネージャが終了中)	KFCA16341-E (開始要求応答受け入れ不可) 理由コード=05	◎	◎	×	×
MCP 障害 (相手チャンネルの状態が「チャンネル停止」でない)	KFCA16341-E (開始要求応答受け入れ不可) 理由コード=16	◎	◎	×	×
MCP 障害 (上記以外のエラー応答受信)	KFCA16341-E (開始要求応答受け入れ不可) 上記以外の理由コード	◎	◎	×	×

(凡例)

- ◎：チャンネルが確立するまで再試行を実行します。
- ：TCP/IP コネクション確立まで再試行を実行します。
- ×：再試行を実行しません。

チャンネルの確立後に発生する通信障害と、出力されるログメッセージ、および再試行処理内容を次の表に示します。

表 2-7 再試行処理 (チャンネル確立後)

通信障害の内容	出力するログメッセージ	mqtalcccha 定義コマンドの-b オプションまたは-v オプション			
		bretry=yes または bretrylg=yes vretry=yes bretrymcp=yes	bretry=yes または bretrylg=yes vretry=no bretrymcp=yes	bretry=yes または bretrylg=yes vretry=yes bretrymcp=no	bretry=yes または bretrylg=yes vretry=no bretrymcp=no
コネクション障害 (TCP/IP エラー)	KFCA16331-E (TCP/IP インタフェースエラー)	◎	◎	○	○
時間監視障害 (確認メッセージ受信監視タイムアウト)	KFCA16329-E (受信監視タイムアウト) 監視対象=resp	◎	×	○	×
時間監視障害 (接続セグメント受信監視タイムアウトまたは継続メッセージ受信監視タイムアウト)	KFCA16329-E (受信監視タイムアウト) 監視対象=next	◎	×	○	×
時間監視障害 (ハートビート間隔監視タイムアウト)	KFCA16329-E (受信監視タイムアウト) 監視対象=htbt	◎	×	○	×
MCP 障害 (受信メッセージをあて先キューまたはデッドレターキューに登録失敗)	KFCA16345-E (エラーデータ受信) 理由コード=06	◎	◎	×	×
MCP 障害	KFCA16343-E (メッセージシーケンス番号不一致)	◎	◎	×	×

通信障害の内容	出力するログメッセージ	mqtalcccha 定義コマンドの-b オプションまたは-v オプション			
		bretry=yes または bretrylg=yes vretry=yes bretrymcp=yes	bretry=yes または bretrylg=yes vretry=no bretrymcp=yes	bretry=yes または bretrylg=yes vretry=yes bretrymcp=no	bretry=yes または bretrylg=yes vretry=no bretrymcp=no
(メッセージシーケンス番号不一致)	KFCA16343-E (メッセージシーケンス番号不一致)	◎	◎	×	×
MCP 障害 (キューファイル障害)	KFCA16333-E (キューファイル障害) または KFCA16315-E (キューが他サーバで使用)	◎	◎	×	×
MCP 障害 (上記以外のMCP 障害)	KFCA16345-E (エラーデータ受信)	◎	◎	×	×

(凡例)

- ◎：チャンネルが確立するまで再試行を実行します。
- ：TCP/IP コネクション確立まで再試行を実行します。
- ×：再試行を実行しません。

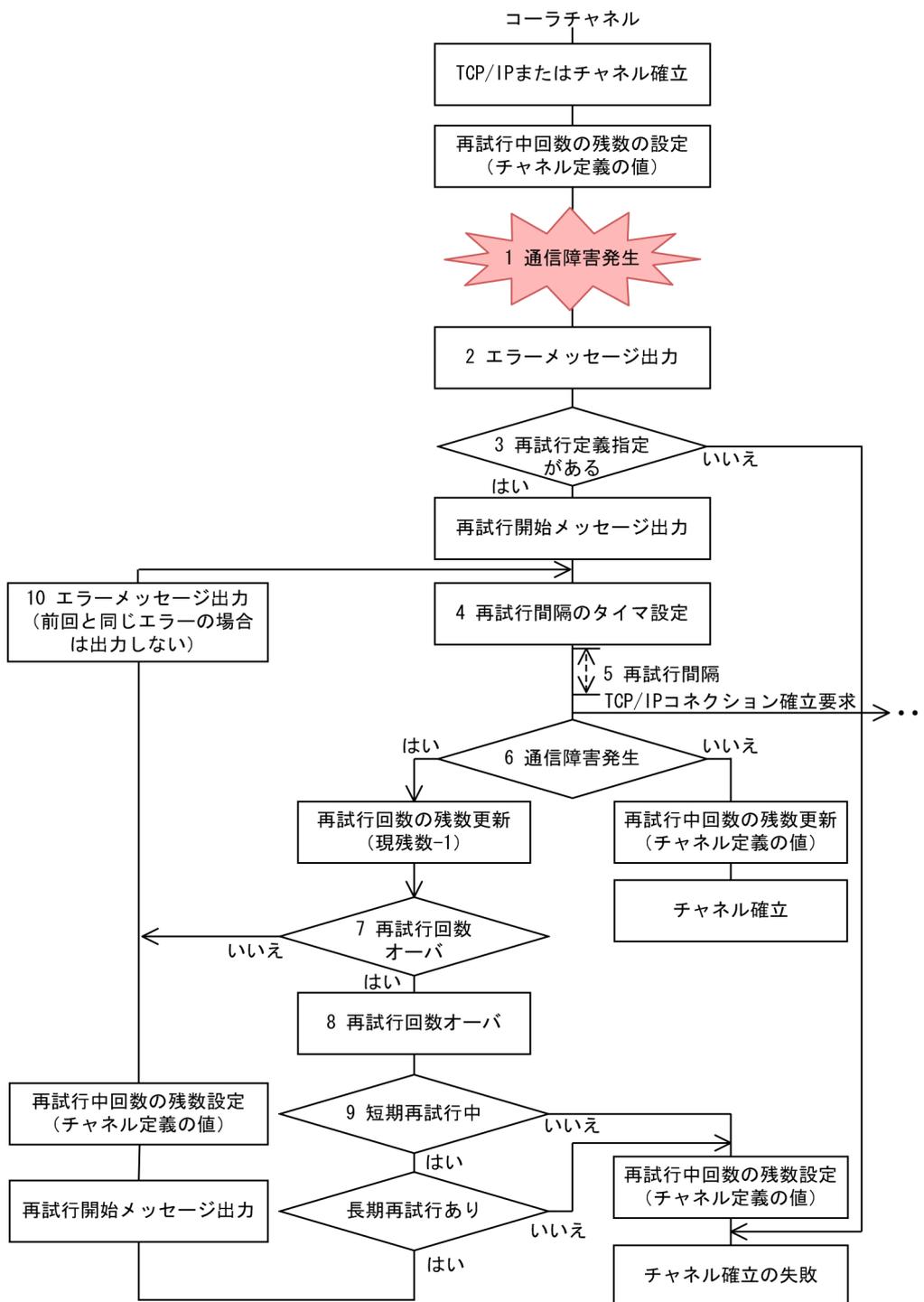
チャンネルの確立後に発生した通信障害によって確立再試行を実行する場合、再試行回数の残数は元に戻ります。このため、通信障害が解決されない場合、次に示す順番で処理が繰り返し実行されます。

1. チャンネル (TCP/IP) 確立 (確立再試行回数の残数=チャンネル定義の値)
2. 通信障害発生
3. チャンネル解放
4. 確立再試行処理 (確立再試行回数の残数=現在の残数-1)
5. チャンネル (TCP/IP) 確立 (確立再試行回数の残数=チャンネル定義の値)
6. 通信障害発生

確立再試行回数の残数は0にならないため、無限回の確立再試行を実行することがありますので、注意してください。

確立再試行処理の全体の流れを以降の図に示します。

図 2-22 短期および長期確立再試行処理の概要 (TCP/IP コネクション確立までの再試行)



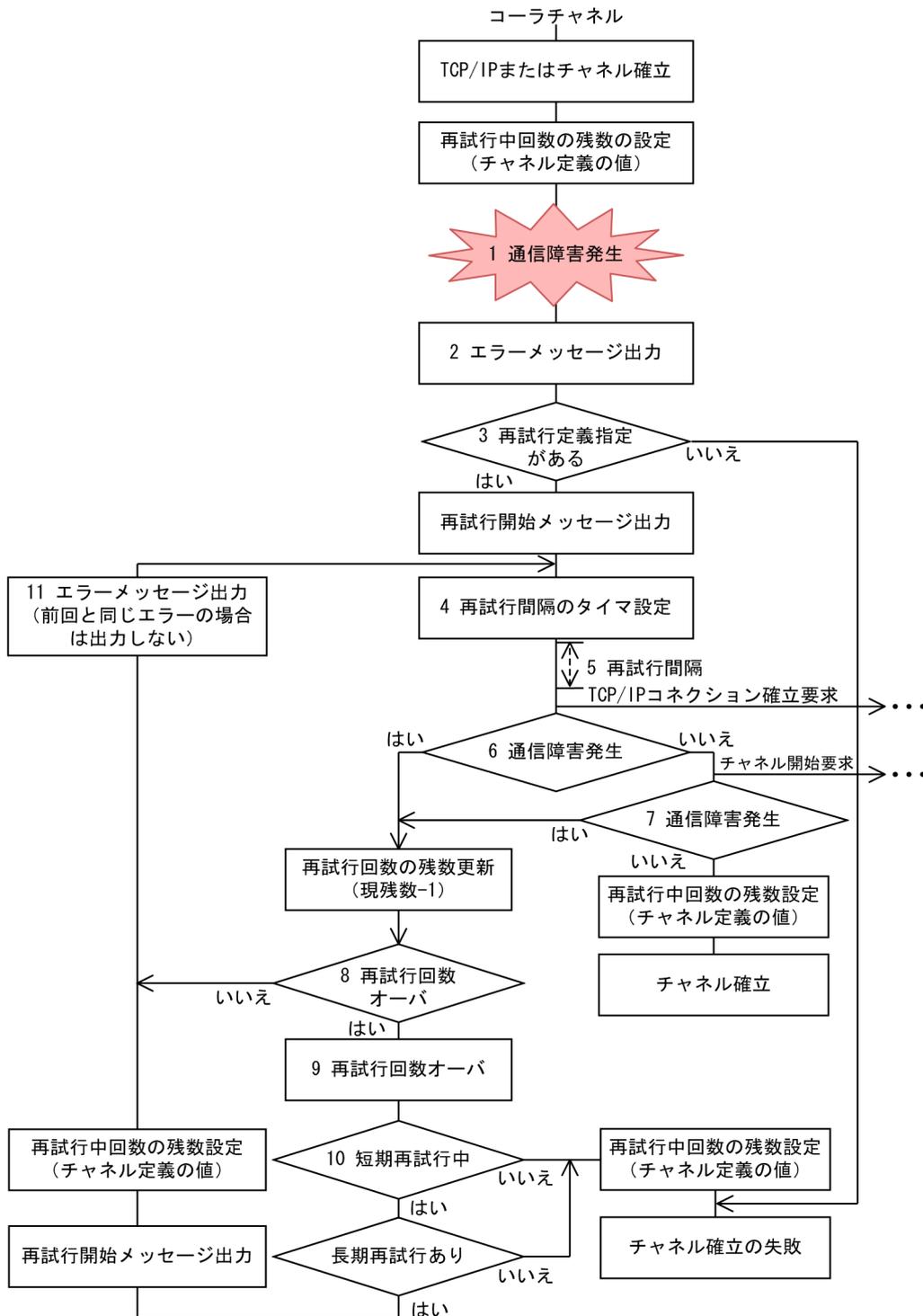
図中の 1~10 について説明します。

1. コネクション障害または時間監視障害が発生
2. 障害ログメッセージの出力 (表 2-6 および表 2-7 を参照)
3. 短期確立再試行を実行する場合 (mqtalcca 定義コマンドの-b オプションに bretry=yes を指定), または長期確立再試行を実行する場合 (mqtalcca 定義コマンドの-b オプションに bretrylg=yes を指

定)は、再試行開始ログメッセージ (KFCA16318-I 種別=short または KFCA16318-I 種別=long) の出力

4. 再試行間隔 (短期確立再試行または長期確立再試行) のタイマの設定
5. 指定した再試行間隔 (短期確立再試行または長期確立再試行) の経過を待つ
6. コネクション障害の発生の確認
 - コネクション障害が発生した場合、再試行回数の残数を更新して次の確立再試行を実行します。
 - コネクション障害が発生していない場合、確立再試行は完了します (再試行回数の残数はチャンネル定義の値に戻ります)。
7. 確立再試行回数の残数が 0 になっていないかの確認
8. 短期再試行中の場合、回数オーバーログメッセージ (KFCA16340-I 種別=short) の出力。長期再試行中の場合、回数オーバーログメッセージ (KFCA16340-I 種別=long) の出力。
9. 長期確立再試行を実行する場合、確立再試行回数の残数を設定して長期再試行処理開始ログメッセージ (KFCA16318-I 種別=long) の出力
10. 前回の障害と同じ障害かどうか確認し、同じ障害でなければ該当障害のログメッセージの出力 (表 2-6 および表 2-7 を参照)

図 2-23 短期および長期確立再試行処理の概要 (チャンネル確立までの再試行)



図中の 1~11 について説明します。

1. MCP 障害が発生
2. 障害ログメッセージの出力 (表 2-6 および表 2-7 を参照)
3. 短期確立再試行を実行する場合 (mqtalccha 定義コマンドの-b オプションに bretry=yes を指定), または長期確立再試行を実行する場合 (mqtalccha 定義コマンドの-b オプションに bretrylg=yes を指

定)、再試行開始ログメッセージ (KFCA16318-I 種別=short または KFCA16318-I 種別=long) の出力

ただし、MCP 障害の確立再試行が設定されている (mqtalcccha 定義コマンドの-b オプションに bretrymcp=yes を指定) 場合に限り、

4. 再試行間隔 (短期確立再試行または長期確立再試行) のタイマの設定
5. 指定した再試行間隔 (短期確立再試行または長期確立再試行) の経過を待つ
6. コネクション障害の発生の確認
 - コネクション障害が発生した場合、確立再試行回数の残数を更新して次の確立再試行を実行します。
 - コネクション障害が発生していない場合、チャンネル開始要求を送信します。
7. コネクション障害、時間監視障害、または MCP 障害のうちどれかについて発生の確認
 - コネクション障害、時間監視障害、または MCP 障害のどれかが発生した場合、確立再試行回数の残数を更新して確立再試行を実行します。
 - コネクション障害、時間監視障害、または MCP 障害のどれにも発生していない場合、確立再試行は完了します (再試行回数の残数は、チャンネル定義の値に戻ります)。
8. 確立再試行回数の残数 (0 になっていないか) の確認
9. 短期再試行中の場合、回数オーバーログメッセージ (KFCA16340-I 種別=short) の出力。また、長期再試行中の場合、回数オーバーログメッセージ (KFCA16340-I 種別=long) の出力
10. 長期確立再試行を実行する場合、確立再試行回数の残数を設定して長期再試行処理開始ログメッセージ (KFCA16318-I 種別=long) の出力
11. 前回の障害と同じ障害かどうか確認し、同じ障害でなければ該当障害のログメッセージの出力 (表 2-6 および表 2-7 を参照)

(2) MCP 障害による再試行 (bretrymcp=yes) の効果

TCP 定義の mqtalcccha 定義コマンドの-b オプションに bretrymcp=yes を指定している場合、mcp 障害時に確立再試行が実行されます。bretrymcp=no を指定している場合、mcp 障害時にチャンネルは終了します。チャンネル確立前、およびチャンネル確立後の通信障害の種類と TCP 定義の mqtalcccha 定義コマンドの-b bretrymcp オペランドの指定値 (yes または no) との関係については表 2-6 および表 2-7 を参照してください。

チャンネル確立再試行は、従来の TCP/IP レベルの再試行だけでなく、MQ プロトコルレベルでの再試行が実行できるため、特に次に示す場合に効果があります。

KFCA16341-E (開始要求の応答受け入れ不可) 出力時の再試行

チャンネル確立時に、次に示す状況で KFCA16341-E (開始要求の応答受け入れ不可) が出力され、MCA が確立処理を中断することがあります。このような場合、コーラチャンネルに対して再度チャンネル開始コマンドを入力して、チャンネルを開始する必要がありました。

チャンネル確立再試行を使用すると、これらのエラーに対して自動的にチャンネル確立再試行を実行し、相手システムの回復を待って、再確立を実行できます。

KFCA16341-E 出力時の状況

回線障害によるチャンネル状態の不一致

チャンネルが使用している TCP/IP プロトコルでは、回線に障害が発生しても一定時間障害を検出できないことがあります。このため、コーラチャンネルが先に障害を検出してレスポンドチャンネルの障害検出が遅れた場合に、コーラチャンネルからのチャンネル確立処理によってチャンネル状態不一致が発生したとき、レスポンドには KFCA16317-E (状態不一致) が出力され、コーラには KFCA16341-E (開始要求の応答受け入れ不可。理由コード=16) が出力されます。なお、理由コード=16 は、チャンネルは使用できない状態であることを意味します。

IBM MQ でのリスナとキューマネージャの開始状態の不一致

IBM MQ では、リスナとキューマネージャの開始状態が一致しない場合があります。このため、IBM MQ のキューマネージャの終了中に、TP1/Message Queue のコーラチャンネルが確立要求すると KFCA16341-E (開始要求の応答受け入れ不可。理由コード=03 または 05) が出力されます。理由コード=03 は、IBM MQ のキューマネージャが動作していないことを表します。なお、理由コード=05 は、IBM MQ のキューマネージャが終了処理中であることを表します。

チャンネル状態の不一致 (相手チャンネルの状態が「チャンネル停止」でない)

チャンネルの終了直後に、MCA を再起動しようとする時、相手システムの MCA が終了中で、チャンネルが確立できないため、KFCA16341-E (開始要求の応答受け入れ不可。理由コード=16) が出力され、チャンネルが確立できないことがあります。bretrymcp=yes を指定していない場合は、相手システムの状態を確認したあと、もう一度 mqstacha コマンドでチャンネルを開始してください。ただし、トリガ機能を使用している場合は、自動的にトリガメッセージが再作成され再度 MCA が起動されます。

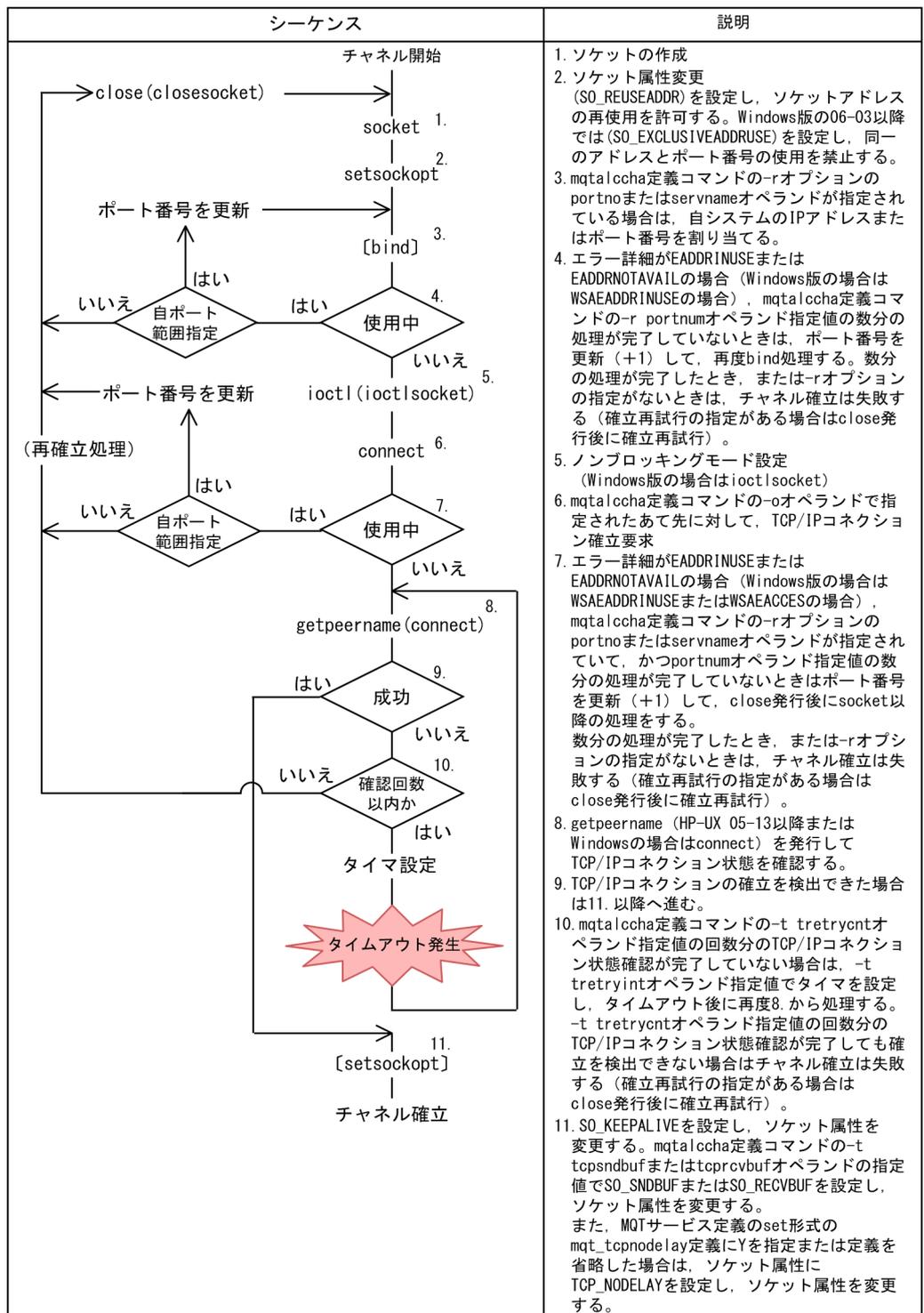
2.3.8 チャンネル確立時の TCP/IP ソケット関数発行シーケンス

チャンネル確立時の TCP/IP ソケット関数発行シーケンスを、コーラ側およびレスポンド側から説明します。

(1) コーラ側チャンネル確立時の TCP/IP ソケット関数発行シーケンス

コーラ側チャンネル確立時の TCP/IP ソケット関数発行シーケンスを次の図に示します。

図 2-24 コーラ側チャンネル確立時の TCP/IP ソケット関数発行シーケンス



(凡例)

[] : mqta1ccha定義コマンドで指定しない場合は発行しません。

図中に示す関数 (socket~setsockopt~ [bind] ~ioctl) でエラーになった場合、KFCA16331-E (TCP/IP インタフェースエラー) が出力されます。この時、確立再試行が指定されている場合は、確立再試行が実行されます (チャンネル状態は「チャンネル確立リトライ中」になります)。確立再試行が指定されていない場合は、チャンネル確立に失敗します (チャンネル状態は「チャンネル使用不可」になります)。

なお、自システムのポート番号が指定されている場合は、(mqtalcccha 定義コマンドの-r servname オペランド指定値の名称から変換したポート番号、または-r portno オペランド指定値のポート番号) ~ (該当ポート番号+r portnum オペランド指定値-1) の範囲内で bind 処理します。bind 処理するポート番号が 65535 を超える場合は 65535 で処理を中止します。すべてのポート番号が使用中の場合は KFCA16331-E (TCP/IP インタフェースエラー) が出力されます。

1 回目の TCP/IP コネクション確立要求 (connect) で接続を確認できない場合、mqtalcccha 定義コマンドの-t オプションで指定された回数または間隔 (tretrycnt, tretryint) で TCP/IP コネクションの状態が確認されます。指定された回数・間隔内に接続を確認できない場合は、KFCA16331-E (TCP/IP インタフェースエラー) が出力されて、確立再試行が実行されます (確立再試行の指定がある場合)。

TCP/IP コネクション確立後、[setsockopt] 関数でエラーになった場合も、KFCA16331-E (TCP/IP インタフェースエラー) が出力されて、確立再試行が実行されます (確立再試行の指定がある場合)。

(2) レスポンド側チャンネル確立時の TCP/IP ソケット関数発行シーケンス

レスポンド側チャンネル確立時の TCP/IP ソケット関数発行シーケンスを次の図に示します。

図 2-25 レスポンド側チャンネル確立時の TCP/IP ソケット関数発行シーケンス

シーケンス	説明	関連する定義	備考
TP1/Message Queue開始			
↓ socket	ソケットの作成	なし	
↓ setsockopt	ソケット属性変更 (SO_REUSEADDR) を設定	なし	ソケットアドレスの再利用を許可する。
↓ bind	自システムのポート番号の割り当て (IPアドレスには INADDR_ANYを指定する)	mqttcp定義コマンド およびmqttcpcr定義コマンド-r全オペランド	
↓ ioctl	ノンブロッキングモード設定	なし	Windowsの場合は ioctlsocketを発行する。
↓ listen	TCP/IPコネクション接続要求待ち	なし	
← 相手システムからの接続要求 → ↓ accept	TCP/IPコネクション接続要求受け入れ	なし	
↓ [setsockopt]	ソケット属性変更 (SO_KEEPALIVE, SO_SNDBUF, SO_RCVBUF) を設定	mqtalcccha定義コマンド -t tcpsndbuf -t tcpvcvbuf	mqtalcccha定義コマンドの-tオプションの指定がない場合は発行しない。
↓ チャンネル確立	また、MQTサービス定義の set形式のmqt_tcpnodelay 定義にYを指定または定義を省略した場合は、ソケット属性にTCP_NODELAYを設定		

(凡例)

[] : mqtalcccha定義コマンドで指定しない場合は発行しません。

図中に示す関数 (socket~setsockopt~bind~ioctl~listen) でエラーになった場合、KFCA16331-E (TCP/IP インタフェースエラー) が出力されて、ネットワークリクエストによるチャンネルの起動機能は縮退します (TP1/Message Queue を再度開始するまで縮退します)。

関数 (accept ~ [setsockopt]) でエラーになった場合は、KFCA16332-W (TCP/IP インタフェースエラー。処理続行) が出力されて、TCP/IP コネクションが解放されます。ただし、新たな接続要求は、受け付け処理できません。

(3) 注意事項

MCA は TCP/IP ソケットの回線監視 (keepalive) 機能を使用して回線障害を検出することがあります。keepalive による障害検出まで一定時間が必要なため、通信相手の MCA との間でチャンネルの状態が不一致になります。このため、コーラ側チャンネルで先に回線障害を検出し、チャンネルの確立再試行によってチャンネルを再確立すると、相手システムのチャンネルとの状態不一致が発生 (障害を検出していない MCA で KFCA16317-E メッセージを出力) します。

この場合、keepalive による障害検出を待つか、障害検出できていない MCA のチャンネルを、mqtstpcha コマンドで一度チャンネルを終了し、mqtstacha コマンドで再度チャンネルを開始してください。また、mqtalccha 定義コマンドの -f オプションの adoptchk オペランドで off 以外の値を指定すると、チャンネルの開始要求に対して動作中の MCA を強制停止して新たにチャンネルを開始できます。

2.3.9 チャンネルの解放

チャンネルを終了すると、チャンネルが解放され MCA は停止します。

次に示す場合に、チャンネルは終了します。

- 転送キューのメッセージの送信完了
転送キューに登録されているメッセージがすべて送信された場合
転送キュー監視機能を使用している場合は、監視処理完了後に終了します。転送キュー監視については、「2.5.5 転送キュー監視と切断時間間隔」を参照してください。
- 運用コマンド
 - ユーザが mqtstpcha または dcstop コマンドを入力した場合
受信側でチャンネルを終了する場合は、コマンドの入力後にメッセージを受信しないと、チャンネルは終了しません。正常終了させるにはメッセージの送信側から、アプリケーションが登録したメッセージを転送するか、ハートビートメッセージを転送してください。
 - ユーザが mqtstpcha コマンドに -f オプションを指定して入力した場合
処理を中断して即座に終了します。相手システムの MCA は通信障害になります。
次回にチャンネルを開始するには、mqtstacha コマンドを使用してください。
- チャンネル確立の再試行回数超過
TCP 定義の mqtalccha 定義コマンドの -b bretrycnt または bretrylgcnt オペランドで指定したチャンネルの確立再試行回数を超えた場合
次回にチャンネルを開始するには、mqtstacha コマンドを使用してください。
- 障害発生

TCP/IP コネクションでコネクションの重度障害またはキュー障害が発生し、チャンネルを使用できなくなった場合

メッセージ送受信中に運用コマンドまたは障害発生によってチャンネルが解放された場合の MQT の処理について、次の表に示します。

表 2-8 メッセージ送受信中にチャンネルが解放された場合の MQT の処理

解放の方法	種別	MQT の処理
mqststpcha コマンド	正常解放	送受信途中のメッセージ送受信完了後、TCP/IP コネクションを解放します。
	強制解放	メッセージ送信を中断し、TCP/IP コネクションを解放します。 送信途中のメッセージは転送キューに保留されます。受信途中のメッセージは破棄されます。
dcstop コマンド	正常終了	送受信途中のメッセージ送受信を完了し、TCP/IP コネクションを解放します。
	計画停止 A	
	計画停止 B	
	強制終了	メッセージ送受信を中断し、TCP/IP コネクションを解放します。 送信途中のメッセージは転送キューに保留されます。受信途中のメッセージは破棄されます。
障害発生	—	メッセージ送受信を中断し、TCP/IP コネクションを解放します。 送信途中のメッセージは転送キューに保留されます。受信途中のメッセージは破棄されます。

(凡例)

—：該当しません。

2.4 メッセージ送受信機能

TP1/Message Queue が送受信時のメッセージをどのように処理するか説明します。

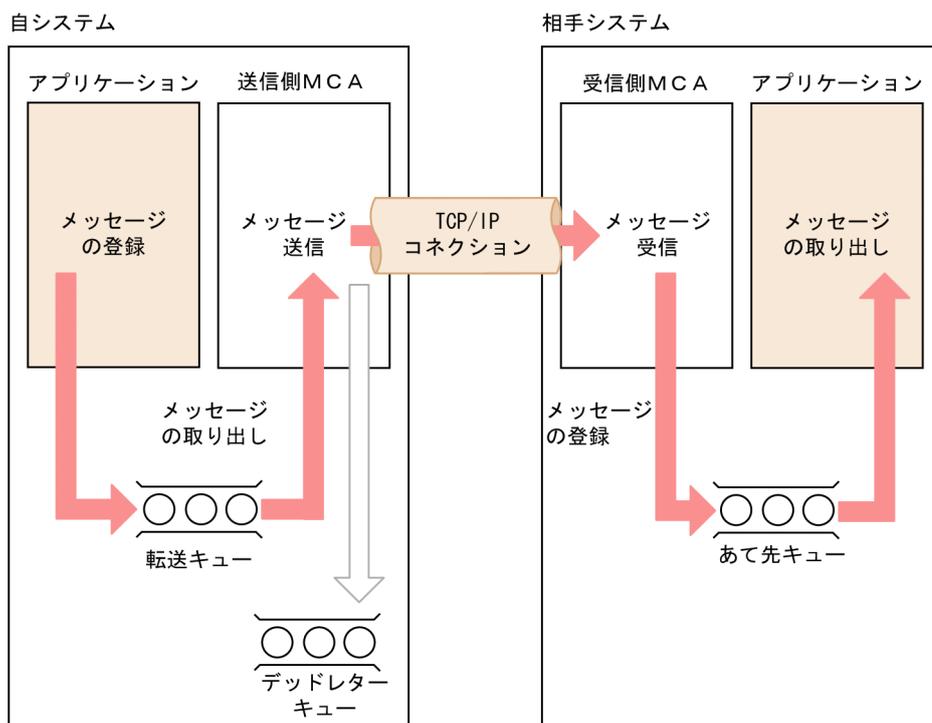
2.4.1 メッセージの送信

TP1/Message Queue は、TCP/IP プロトコルを使用してほかのシステムへメッセージを送信します。この項では、メッセージ送信の処理の流れについて説明します。また、メッセージがセグメント分割された場合のメッセージ送信についても説明します。

(1) メッセージ送信の流れ

自システムの送信側 MCA は、チャンネル確立後、TCP 定義の mqtalcccha 定義コマンドの-q オプションで指定した転送キューに登録されたメッセージを相手システムの受信側 MCA に送信します。メッセージ送信の流れについて、次の図に示します。

図 2-26 メッセージ送信の流れ



(凡例)

- ➡ : 適切な長さのメッセージの流れ
- ➡ : 長過ぎる場合のメッセージの流れ

メッセージが転送キューに登録されていれば、送信側 MCA は、転送キューに登録されているすべてのメッセージの取り出しを MQA サーバに要求し、受信側 MCA に送信します。取り出したメッセージがチャンネルの最大メッセージ長より大きい場合、メッセージはデッドレターキューへ登録されます。メッセージが登録されていなければ、チャンネルを解放します。

(2) 送信バッファ方式

MQT サーバが送信するメッセージの分割単位をセグメントといいます。

バッファ方式がセグメント方式のチャネルの場合、送信側 MCA は、転送キューに登録されているメッセージをセグメントごとに分割して取り出し送信します。セグメントサイズよりも大きいサイズの送信バッファが必要になります。

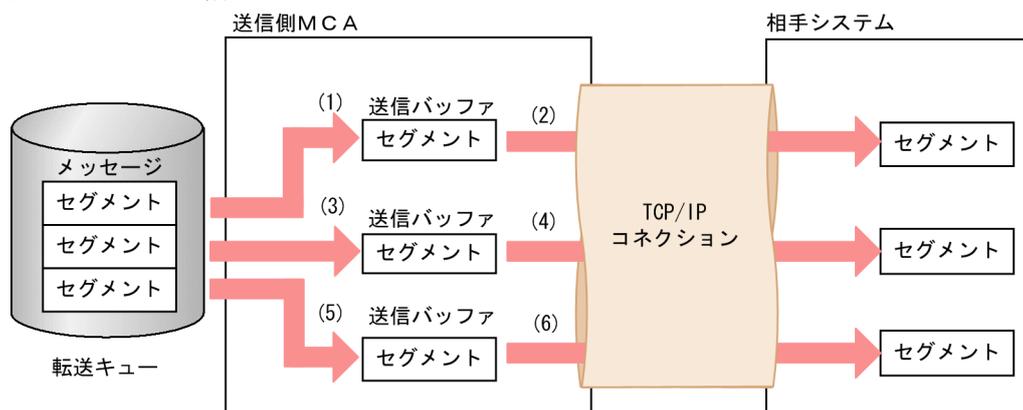
最大セグメントサイズを大きくすると、転送処理を削減でき、性能が向上します。また、受信側 MCA が分割受信機能を持たない場合、一つのセグメントに入りきれないメッセージはデッドレターキューに登録されます。

バッファ方式がメッセージ方式のチャネルの場合、送信側 MCA は、転送キューに登録されているメッセージを一度に取り出しセグメントごとに送信します。メッセージサイズよりも大きいサイズの送信バッファが必要になります。

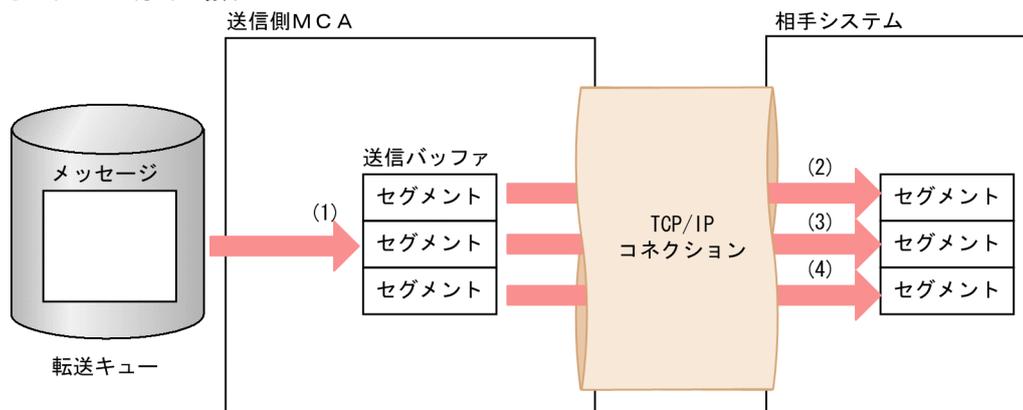
送信バッファ方式の概要について、次の図に示します。図中の括弧付き数字は、処理の順序を表します。

図 2-27 送信バッファ方式の概要

●セグメント方式の場合



●メッセージ方式の場合



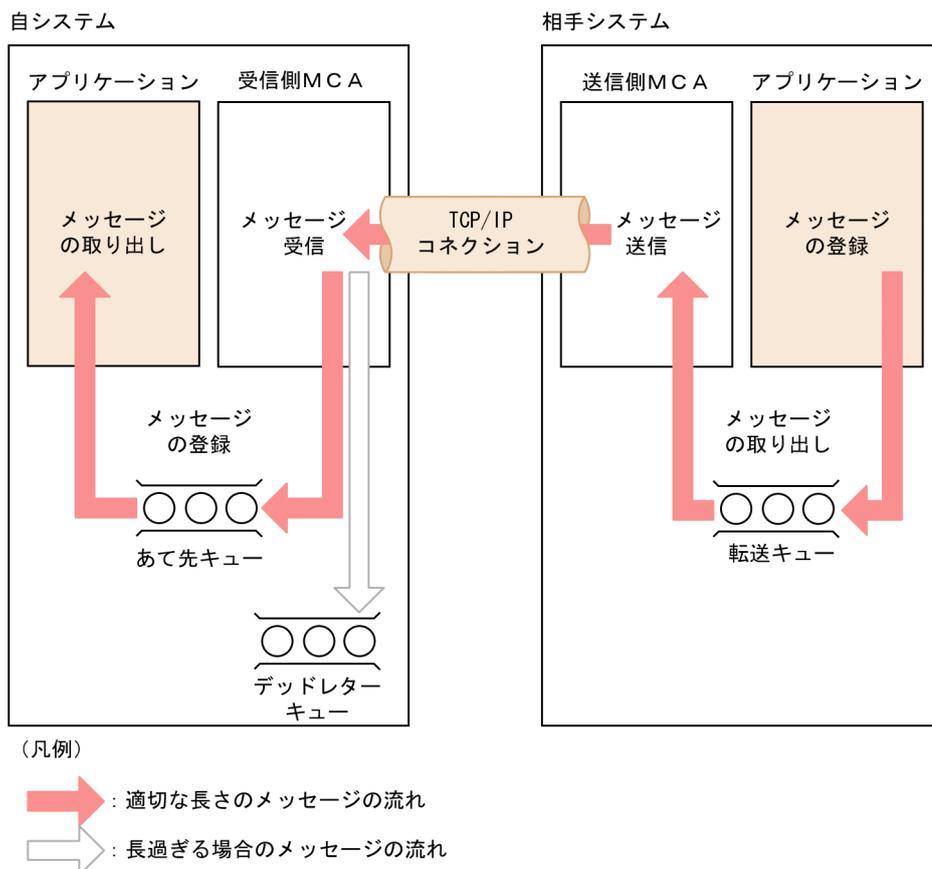
2.4.2 メッセージの受信

TP1/Message Queue は、TCP/IP プロトコルを使用してほかのシステムからメッセージを受信します。この項では、メッセージ受信の処理の流れについて説明します。また、メッセージがセグメントに分割されて送信された場合のメッセージ受信についても説明します。

(1) メッセージ受信の流れ

受信側 MCA は、チャンネル確立後、相手システムから受信したメッセージをあて先キューに登録します。メッセージ受信の流れについて、次の図に示します。

図 2-28 メッセージ受信の流れ



(2) 受信バッファ方式

MQT サーバが受信するメッセージの分割単位をセグメントといいます。

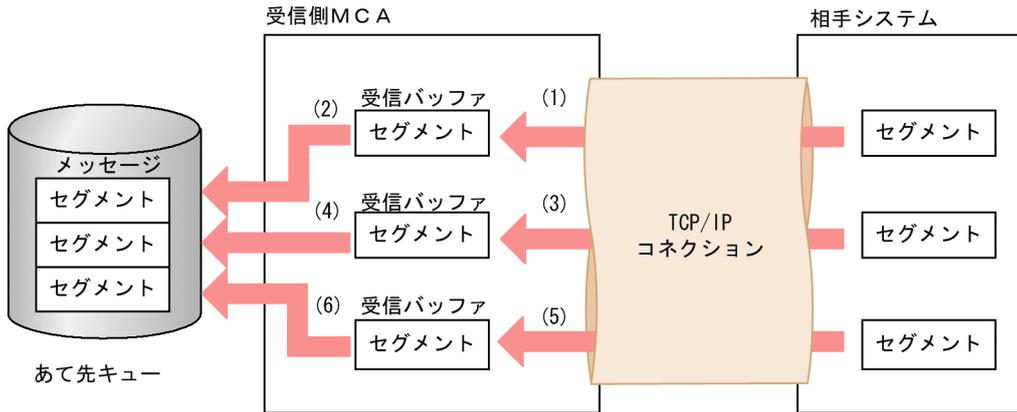
バッファ方式がセグメント方式のチャンネルの場合、受信側 MCA は送信側 MCA から送られてきたセグメントを受信し、セグメントごとにあて先キューに登録します。セグメントサイズよりも大きいサイズの受信バッファが必要になります。

バッファ方式がメッセージ方式のチャンネルの場合、受信側 MCA は送信側 MCA から送られてきたセグメントをすべて受信し一度にあて先キューへ登録します。メッセージサイズよりも大きいサイズの受信バッファが必要になります。

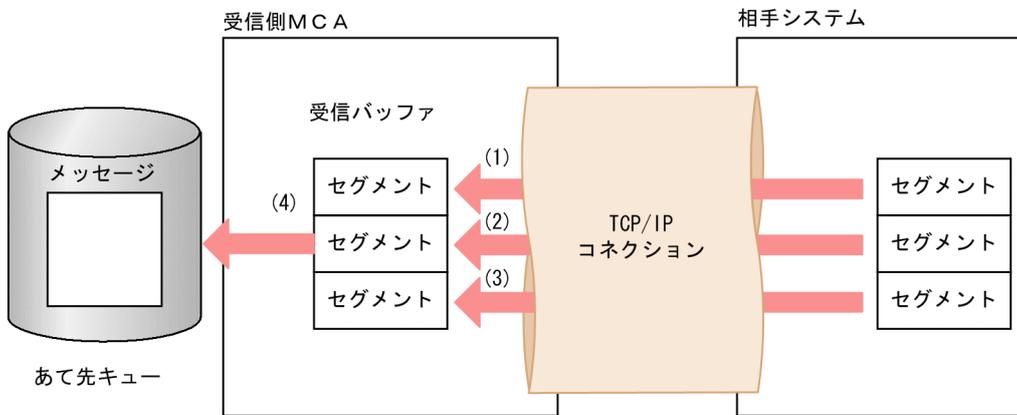
受信バッファ方式の概要について、次の図に示します。図中の括弧付き数字は、処理の順序を表します。

図 2-29 受信バッファ方式の概要

●セグメント方式の場合



●メッセージ方式の場合



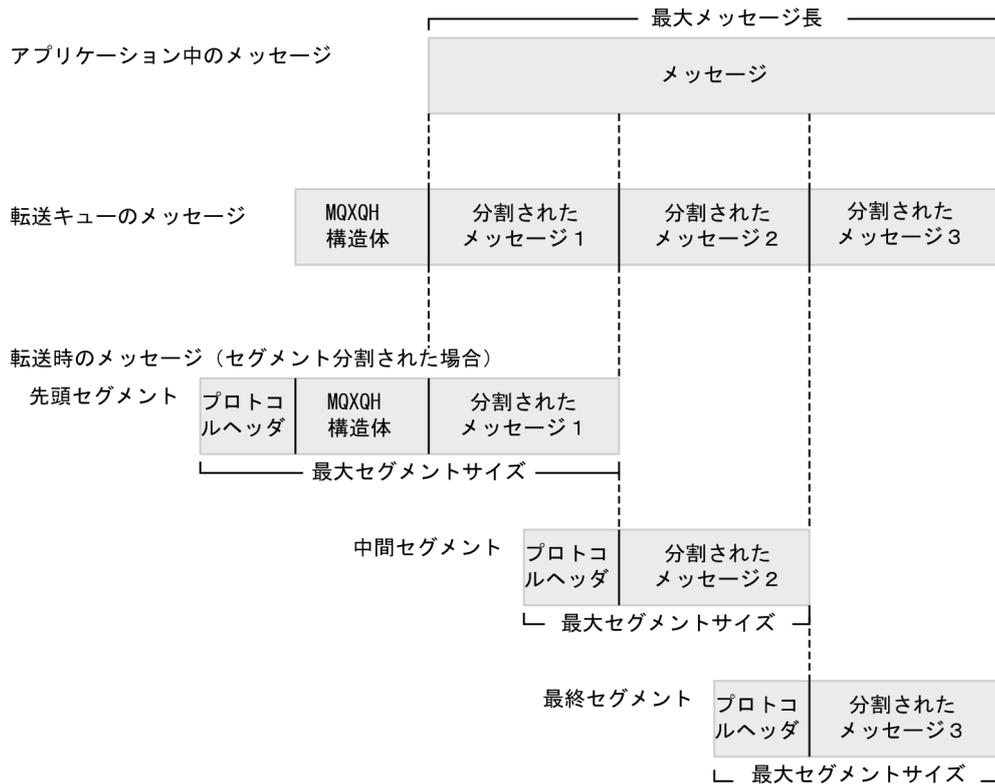
2.4.3 メッセージとセグメントの関係

MQT サーバが送受信するメッセージの分割単位をセグメントといいます。

送信側 MCA が分割するセグメントのサイズ、および MQT サーバが受信するセグメントのサイズは、TCP 定義の `mqtalcca` 定義コマンドの `-m maxseg` オペランドで指定した最大セグメントサイズを基に、チャネル確立時に相手システムとの間で決定された値です。

メッセージとセグメントの関係について、次の図に示します。

図 2-30 メッセージとセグメントの関係



(凡例)

- MQXQH構造体 : 転送キューヘッダ
- 最大メッセージ長 : TCP定義 (mqtalcccha定義コマンド)で指定します。
- 最大セグメントサイズ : TCP定義 (mqtalcccha定義コマンド)で指定します。
プロトコルヘッダの48バイトを含みます。

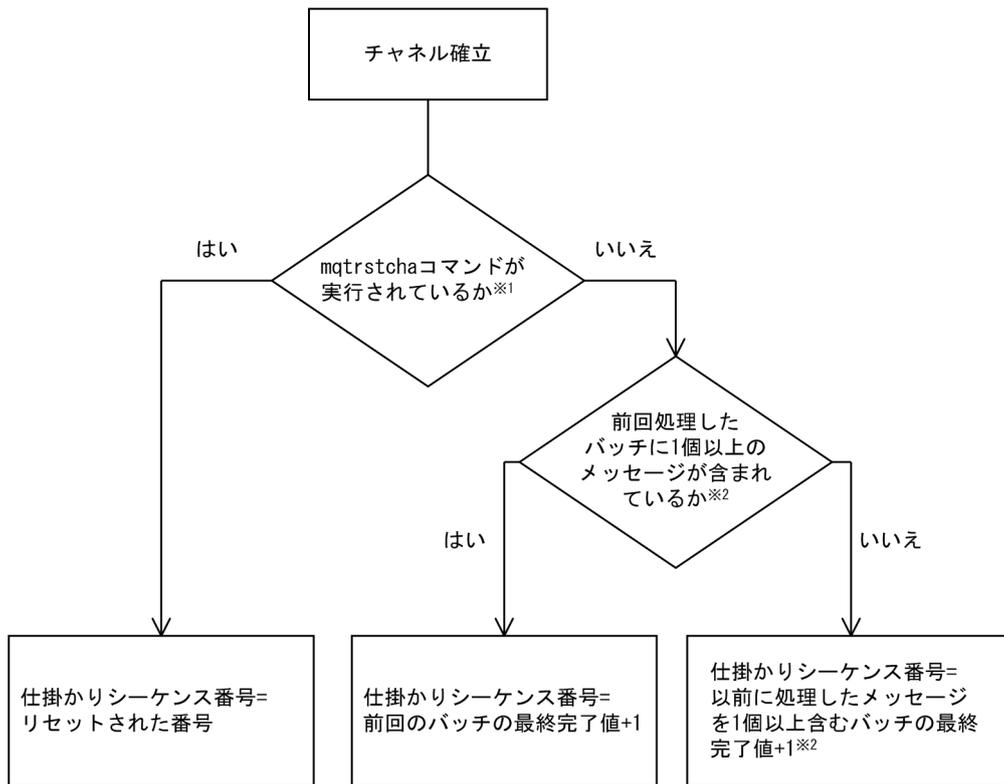
2.4.4 メッセージシーケンス番号でのメッセージの管理

MCA は、メッセージの二重送信を抑止するため、メッセージにメッセージシーケンス番号を付けます。各チャンネルの送信側 MCA は、メッセージにメッセージシーケンス番号を付けて送信します。

(1) メッセージシーケンス番号の遷移

メッセージシーケンス番号は、送信側 MCA がメッセージに 1 から一つずつ昇順に付けるチャンネル単位のメッセージ転送番号です。TCP 定義の mqtalcccha 定義コマンドの -w オプションで指定したメッセージシーケンス番号の最大値を超えると、1 に戻ります。この最大値は、相手チャンネルと同じ値を定義する必要があります。チャンネル確立直後に送受信する 1 番目のメッセージに与えられるメッセージシーケンス番号の決定の順序を次の図に示します。

図 2-31 メッセージシーケンス番号の遷移



注※1
受信側チャンネルの場合は、リセットデータを受信しているときも含まれます。

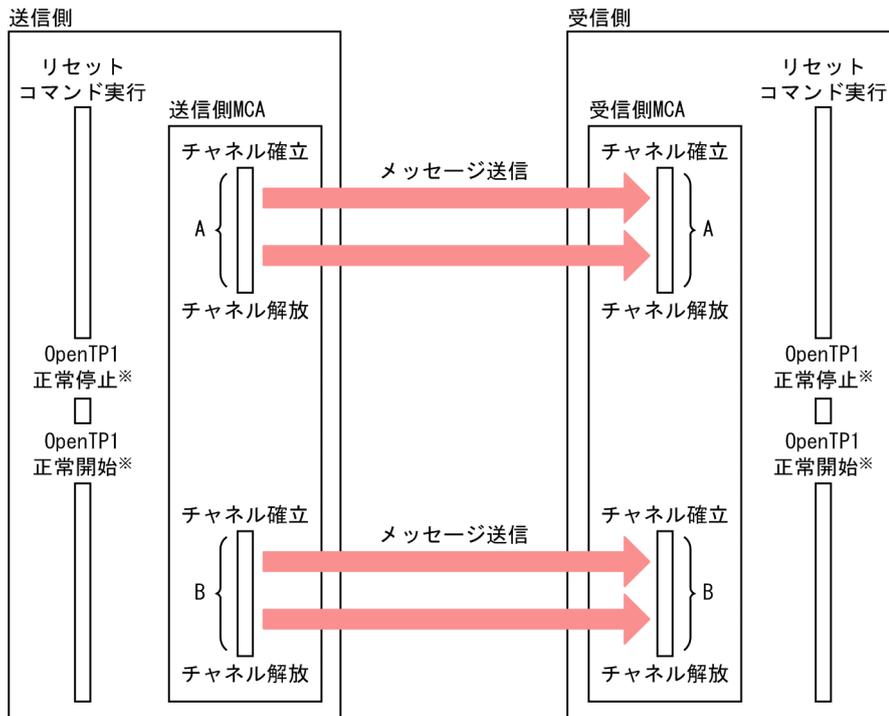
注※2
チャンネルがファーストメッセージ機能を使用している場合、通常メッセージだけが該当します。

チャンネル確立直後に送受信する 2 番目のメッセージからチャンネル解放までのメッセージのメッセージシーケンス番号は、直前のメッセージのメッセージシーケンス番号から一つずつ昇順に付けられます。

(2) リセットコマンド実行後のチャンネルの動作

送信側 MCA と受信側 MCA で、メッセージシーケンス番号のリセットコマンド (TP1/Message Queue の場合は mqtrstcha コマンド) を実行して、メッセージシーケンス番号をリセットしたあとのチャンネルの動作を次の図に示します。

図 2-32 リセットコマンド実行後のチャネルの動作



注※
チャネル管理情報格納ファイルで、チャネル情報を引き継ぎます。

なお、図中の送信側 MCA は、センダ、サーバ、またはクラスタセンダです。また、受信側 MCA は、レシーバ、リクエスタ、またはクラスタレシーバです。

この図中の A の場合、および B の場合の送信側 MCA と受信側 MCA の動作を次の表に示します。

表 2-9 リセットコマンド実行後のチャネルの動作 (図中の A の場合)

リセット実行の有無※ 1		チャネル確立後の送受信メッセージの種別	図中の A の場合	
送信側 MCA	受信側 MCA		送信側 MCA の動作	受信側 MCA の動作
○	×	すべてファーストメッセージ※2 ファーストメッセージ以外を含む※3	<ul style="list-style-type: none"> リセットの扱い 実行されたリセット値でリセット 開始するシーケンス番号 実行されたリセット値 	<ul style="list-style-type: none"> 開始するシーケンス番号 送信側で実行されたリセット値
○	○	すべてファーストメッセージ※2 ファーストメッセージ以外を含む※3	<ul style="list-style-type: none"> リセットの扱い 実行されたリセット値でリセット 開始するシーケンス番号 実行されたリセット値 	<ul style="list-style-type: none"> リセットの扱い 送信側で実行されたリセット値でリセット 開始するシーケンス番号 送信側で実行されたリセット値

リセット実行の有無※ 1		チャンネル確立後の送受信メッセージの種別	図中の A の場合	
送信側 MCA	受信側 MCA		送信側 MCA の動作	受信側 MCA の動作
×	×	すべてファーストメッセージ※2	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4 を含むバッチの最終シーケンス番号+1 	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4 を含むバッチの最終シーケンス番号+1
		ファーストメッセージ以外を含む※3		
×	○	すべてファーストメッセージ※2	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4 を含むバッチの最終シーケンス番号+1 	<ul style="list-style-type: none"> リセットの扱い 実行されたリセット値でリセット 開始するシーケンス番号 実行されたリセット値
		ファーストメッセージ以外を含む※3		

(凡例)

○：リセットを実行している。

×：リセットを実行していない。

注※1

リセットコマンドは、チャンネル確立前（図中の A の前）に実行します。

注※2

チャンネルがファーストメッセージ機能を使用し、かつすべてファーストメッセージである場合に該当します。

注※3

次のどちらかの場合に該当します。

- ・チャンネルがファーストメッセージ機能を使用していない場合。
- ・チャンネルがファーストメッセージ機能を使用し、かつ一つ以上の通常メッセージが含まれている場合。

注※4

チャンネルがファーストメッセージ機能を使用している場合、通常メッセージだけが該当します。

表 2-10 リセットコマンド実行後のチャンネルの動作（図中の B の場合）

リセット実行の有無※ 1		チャンネル確立後の送受信メッセージの種別	図中の B の場合	
送信側 MCA	受信側 MCA		送信側 MCA の動作	受信側 MCA の動作
○	×	すべてファーストメッセージ※2	<ul style="list-style-type: none"> リセットの扱い 実行されたリセット値でリセット 開始するシーケンス番号 実行されたリセット値 	<ul style="list-style-type: none"> 開始するシーケンス番号 送信側で実行されたリセット値
		ファーストメッセージ以外を含む※3	<ul style="list-style-type: none"> リセットの扱い リセットしない 開始するシーケンス番号 	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4 を含むバッチの最終シーケンス番号+1

リセット実行の有無※ 1		チャンネル確立後の送受信メッセージの種別	図中の B の場合	
送信側 MCA	受信側 MCA		送信側 MCA の動作	受信側 MCA の動作
○	×	ファーストメッセージ以外を含む※3	送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1
○	○	すべてファーストメッセージ※2	<ul style="list-style-type: none"> リセットの扱い 実行されたリセット値でリセット 開始するシーケンス番号 実行されたリセット値 	<ul style="list-style-type: none"> リセットの扱い リセットしない 開始するシーケンス番号 送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1
		ファーストメッセージ以外を含む※3	<ul style="list-style-type: none"> リセットの扱い リセットしない 開始するシーケンス番号 送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1 	
×	×	すべてファーストメッセージ※2	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1 	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1
		ファーストメッセージ以外を含む※3		
×	○	すべてファーストメッセージ※2	<ul style="list-style-type: none"> 開始するシーケンス番号 送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1 	<ul style="list-style-type: none"> リセットの扱い 実行されたリセット値でリセット 開始するシーケンス番号 実行されたリセット値
		ファーストメッセージ以外を含む※3		<ul style="list-style-type: none"> リセットの扱い リセットしない 開始するシーケンス番号 送受信が完了したメッセージ※4を含むバッチの最終シーケンス番号+1

(凡例)

○：リセットを実行している。

×：リセットを実行していない。

注※1

リセットコマンドは、チャンネル確立前（図中の A の前）に実行します。

注※2

チャンネルがファーストメッセージ機能を使用し、かつすべてファーストメッセージである場合に該当します。

注※3

次のどちらかの場合に該当します。

- ・チャンネルがファーストメッセージ機能を使用していない場合。
- ・チャンネルがファーストメッセージ機能を使用し、かつ一つ以上の通常メッセージが含まれている場合。

注※4

チャンネルがファーストメッセージ機能を使用している場合、通常メッセージだけが該当します。

(3) 送信側 MCA と受信側 MCA のメッセージシーケンス番号の扱い

メッセージシーケンス番号の扱いについて、送信側 MCA と受信側 MCA に分けて説明します。

(a) 送信側 MCA の場合

転送キューのメッセージ

コネクション障害またはオンラインシステムの停止などでメッセージの送信が中断された場合、転送キュー内のメッセージは次のように処理されます。

- 障害回復後の状態が「メッセージ送達未確認=YES」の場合
LUWID とメッセージシーケンス番号を基に、転送キュー内のメッセージを再送します。ただし、受信側のシステムでメッセージが受信済みになっているときは、送信側で転送キュー内のメッセージを削除します。
- 障害回復後の状態が「メッセージ送達未確認=NO」の場合
転送キュー内のメッセージを再送します。

メッセージ送達未確認については、mqtlscha コマンドを実行して出力されるチャンネルの状態の「メッセージ送達未確認=iii」で確認できます。

(b) 受信側 MCA の場合

直前のメッセージと同じメッセージシーケンス番号のメッセージが発生した場合

あとから発生したメッセージを破棄し、処理を続けます。

メッセージシーケンス番号が不正の場合（番号が飛ばされた場合など）

メッセージシーケンス番号不一致として、チャンネルを解放します。

(4) メッセージシーケンス番号不一致の原因と対処方法

メッセージシーケンス番号不一致が発生した場合、KFCA16343-E メッセージが出力されます。この原因と対処方法について次の表に示します。

表 2-11 メッセージシーケンス番号不一致の原因と対処方法

メッセージシーケンス番号不一致の発生箇所	原因	対処方法
送信側 MCA	<ul style="list-style-type: none">• メッセージ※¹を送信したあと、自システムで OpenTP1 の正常停止や正常開始を行った場合、自システムの正常開始時にチャンネル情報が引き継がれませんでした。詳細については、「5.3 開始時のチャンネル情報引き継ぎ」を参照してください。	mqtrstcha コマンドでリセットしてください。受信側も同じリセット値でリセットされます。

メッセージシーケンス番号不一致の発生箇所	原因	対処方法
送信側 MCA	<ul style="list-style-type: none"> メッセージ※1を送信したあと、相手システム（受信側）の開始時にチャンネル情報が引き継がれませんでした。 送信側のメッセージシーケンス番号と一致しないリセット値で、受信側メッセージシーケンス番号がリセットされました。 	mqtrstcha コマンドでリセットしてください。受信側も同じリセット値でリセットされます。
受信側 MCA	<ul style="list-style-type: none"> メッセージ※1を受信したあと、自システムで OpenTP1 の正常停止や正常開始を行った場合、自システムの正常開始時にチャンネル情報が引き継がれませんでした。※2 詳細については、「5.3 開始時のチャンネル情報引き継ぎ」を参照してください。 メッセージ※1を受信したあと、相手システム（送信側）の開始時にチャンネル情報が引き継がれませんでした。※2 送信側のメッセージシーケンス番号と一致しないリセット値で、受信側メッセージシーケンス番号がリセットされました。 	送信側のシーケンス番号に合わせて、mqtrstcha コマンドでリセットしてください。 ただし、送信側でもリセットされていた場合、送信側のリセット値が受信側でも有効になります。

注※1

チャンネルがファーストメッセージ機能を使用している場合、通常メッセージだけが該当します。

注※2

送信側の開始後、チャンネル確立前にメッセージシーケンス番号のリセットが行われている場合は除きます。詳細については、「(2) リセットコマンド実行後のチャンネルの動作」を参照してください。

(5) メッセージの送達未確認状態

メッセージの送受信中に通信障害、チャンネル停止、または TP1/Message Queue が異常終了すると、処理中のメッセージが送信先に届いたかどうかを確認できなくなります。そういう状態を、**メッセージの送達未確認状態**といいます。

メッセージの送達未確認状態には、次に示す場合があります。

あて先キューへ登録済み

転送中だったメッセージが送信側の転送キューと受信側のあて先キューの両方に登録されている状態

あて先キューへ未登録

転送中だったメッセージが送信側の転送キューに残り、あて先キューには登録されていない状態

チャンネルの再開を要求すると、メッセージの送達未確認状態は解決され、処理を続行できます。ただし、正常開始後にチャンネルの再開を要求する場合は、「(6) 正常開始する場合の準備」を参照してください。

(6) 正常開始する場合の準備

OpenTP1 システムを正常開始する場合で、チャンネル管理情報格納ファイルを使用しないとき、メッセージの送達未確認状態およびメッセージシーケンス番号の以前の状態は引き継がれません。そこで、メッセージシーケンス番号を管理するために、正常開始を要求する前に次に示す手順で準備をしてください。

(a) メッセージの送達未確認状態の解決

チャンネル情報引き継ぎ機能を使用しない場合、チャンネル情報を引き継がない条件※に該当する場合、または送信側 MCA が送達未確認状態の場合、メッセージを二重送信することがあります。次に示す手順をオンライン中に実行して、この問題点を解決してください。メッセージの送達未確認状態での問題点を解決できます。

1. mqtlscha コマンドを使用して、送信側 MCA でメッセージが送達未確認状態であるかどうかを確認します。送達未確認状態の場合、2.以下の処理を続けてください。
2. mqtlscha コマンドを使用して、送信側 MCA の送信仕掛かりメッセージシーケンス番号を確認します。
3. mqtlscha コマンドを使用して、受信完了したメッセージシーケンス番号を確認します。
4. mqtrlvcha コマンドを使用して、メッセージ送達未確認状態の問題点を解決します。

< 2.と 3.のシーケンス番号が一致していた場合 >

mqtrlvcha コマンドの-a オプションに commit を指定してください。仕掛かり中のメッセージをコミットして有効にします。

< 2.と 3.のシーケンス番号が不一致だった場合 >

mqtrlvcha コマンドの-a オプションに backout を指定してください。仕掛かり中のメッセージをロールバックして無効にします。

注※

チャンネル情報を引き継がない条件については、「5.3.2 チャンネル情報を引き継がない条件」を参照してください。

(b) メッセージシーケンス番号のリセット

OpenTP1 システムを正常開始すると、メッセージシーケンス番号は 1 に戻されます。自システムと相手システムの両方を正常開始する場合、問題ありません。自システムだけ正常開始する場合は、相手システムのメッセージシーケンス番号をリセットしてください。

(7) 二重受信メッセージの破棄

TP1/Message Queue では、受信側でメッセージの二重受信を検出して、メッセージを破棄する機能を持っています。

次に示す四つの条件を同時に満たす場合、KFCA16348-W のメッセージを出力して、受信したメッセージを破棄します。

- 受信側チャンネルで覚えている前回受信したメッセージのシーケンス番号と受信メッセージのシーケンス番号が一致する。
- 自システムまたは相手システムのどちらかのバッチサイズが 1 である。
- メッセージシーケンス番号がリセットされていない。
- 自システムまたは相手システムのどちらかが、FAP レベル 1 で動作している。

なお、動作する FAP レベルと製品のバージョンとの対応を次に示します。

- FAP レベル 1 : TP1/Message Queue 01-xx^{※1}
- FAP レベル 4 : TP1/Message Queue 05-xx^{※1}, 06-xx^{※1}, 07-00, 07-01, および 07-5x^{※2}

注※1

xx には、任意の符号なし整数が入ります。

注※2

5x の x には、任意の符号なし整数が入ります。

その他の MQ システムが動作する FAP レベルについては、使用する MQ システムが提供するマニュアルを参照してください。

2.4.5 メッセージのバッチ転送

転送キューに格納された幾つかのメッセージを**バッチ**といいます。バッチ単位でメッセージの送達確認をすることを**バッチ転送**といいます。バッチ転送をすると、メッセージ単位で送達確認をするよりも確認待ちの回数が少なくなります。このため、連続してメッセージを送信する時に効率が向上します。

一つのバッチで送信するメッセージの数を**バッチサイズ**といいます。バッチサイズは、TCP 定義の `mqtlccha` 定義コマンドの `-j` オプションで指定したバッチサイズを基に、チャンネル確立時に通信相手システムとの間で決定された値です。

送信側 MCA は、バッチサイズの個数のメッセージを送信するか、転送キューにメッセージがなくなるまで、送達確認しないで送信をします。バッチの送信の最後に、バッチの各メッセージが受信側のキューに格納されたかどうか、確認を要求します。受信側 MCA は、バッチのメッセージをすべて登録できたかどうかで、送信側 MCA に応答します。

すべてのメッセージの登録に成功した場合、受信側でメッセージの登録を完了し、送信側でメッセージを削除します。

一つでもメッセージの登録に失敗すると、受信側でメッセージの登録を取り消し、送信側でメッセージを転送キューに戻します。

2.5 時間監視機能

TP1/Message Queue は、チャンネルに関連する動作を各種タイマによって時間監視します。時間監視をするには、あらかじめ定義で監視タイマ値を指定しておく必要があります。TP1/Message Queue の時間監視の種類、およびそれを指定する定義について、次の表に示します。

すべてのタイマはタイマ定義に指定した基本タイマ（MQT 共通定義の mqtstim 定義コマンドの -t btim および bmtim オペランド指定値）によって、タイマ精度に影響を受けます。例えば、基本タイマが 5 秒の場合、指定したタイマが最大 5 秒単位で繰り上げられることがありますので注意してください。

NTP など各マシンのシステム時刻を合わせている場合、時刻の補正によって自システムや相手システムで監視時間タイムアウト（KFCA16329-E メッセージ）になる場合がありますので注意してください。

表 2-12 時間監視の種類

時間監視の種類		時間監視の範囲	タイムアウト時の処理	適用範囲	指定する定義
送受信監視	チャンネル確立完了監視	TCP/IP コネクション確立時 ～開始要求受信時 ^{*1}	TCP/IP コネクション解放	レスポンス側 MCA	mqttcp -v rtim mqttcpcr -v rtim
	要求応答受信監視	開始要求送信時 ～開始応答受信時	チャンネル終了 ^{*2}	コーラ側 MCA	mqtalccha -v tim1
		バッチ内最終セグメント送信時 ～送達確認応答受信時	<ul style="list-style-type: none"> 転送キューへのメッセージの戻し チャンネル終了^{*2} 	送信側 MCA	mqtalccha -v tim2
		ハートビートメッセージ要求送信時 ～ハートビートメッセージ応答受信時	<ul style="list-style-type: none"> チャンネル終了^{*2} 		
		Ping 要求送信時～Ping 応答受信時	<ul style="list-style-type: none"> チャンネル終了 		
		送達確認要求送信時～送達確認応答受信時	<ul style="list-style-type: none"> 転送キューへのメッセージの戻し チャンネル終了^{*2} 		
	継続セグメント受信監視	セグメント受信時 ～継続セグメント受信時	<ul style="list-style-type: none"> 受信済みセグメント破棄 チャンネル終了^{*2} 	受信側 MCA	
継続メッセージ受信監視	メッセージの最終セグメント受信時 ～継続メッセージ受信時	<ul style="list-style-type: none"> 受信済みメッセージ破棄 チャンネル終了^{*2} 	受信側 MCA	mqtalccha -v mtim	

時間監視の種類	時間監視の範囲	タイムアウト時の処理	適用範囲	指定する定義
終了処理監視	MQT の終了処理開始 ～動作中チャンネルの メッセージ処理完了	<ul style="list-style-type: none"> メッセージ送受信中断 チャンネル終了 	全チャンネル	mqttcp -v etim mqttcps -v etim mqttcpcr -v etim
チャンネル要求監視	MQT 開始 ～MQT 終了で繰り返し	MQR からのクラスタチャンネルの開始または削除連絡確認	クラスタプロセス	mqttcps -v stim mqttcpcr -v stim
イニシエーションキュー監視	MQT 開始 ～MQT 終了で繰り返し	イニシエーションキューへのトリガメッセージ登録を確認し該当するチャンネルを開始	タイマ方式のイニシエーションキュー監視	mqttcp -v itim
転送キュー監視	メッセージ未登録検出時 ～メッセージ登録の有無確認時	転送キュー監視回数分メッセージ登録を確認しチャンネル終了	タイマ方式の転送キュー監視をする送信側 MCA	mqtalccha -v tim4
切断時間間隔	バッチ終了時 ～次バッチ送信監視時	チャンネル終了	イベント方式の転送キュー監視をする送信側 MCA	mqtalccha -v dtim
バッチ終了待ちタイマ (バッチインターバル)	バッチサイズ内でメッセージ未登録検出時 ～メッセージ登録	バッチ終了	イベント方式の転送キュー監視をする送信側 MCA	mqtalccha -v btim mqtalccha -v bmtim
ハートビート機能	バッチ終了時またはハートビートメッセージ要求送信時 ～次バッチ要求送信時	<ul style="list-style-type: none"> 受信側 MCA は htim×2 の時間経過後チャンネル終了*2 送信側 MCA はハートビートメッセージ要求送信 	全チャンネル	mqtalccha -v htim

注※1

1 回目の開始要求受信時にタイマ監視を終了します。そのため、ネゴシエーションが続いている間は、時間監視の対象外です。

注※2

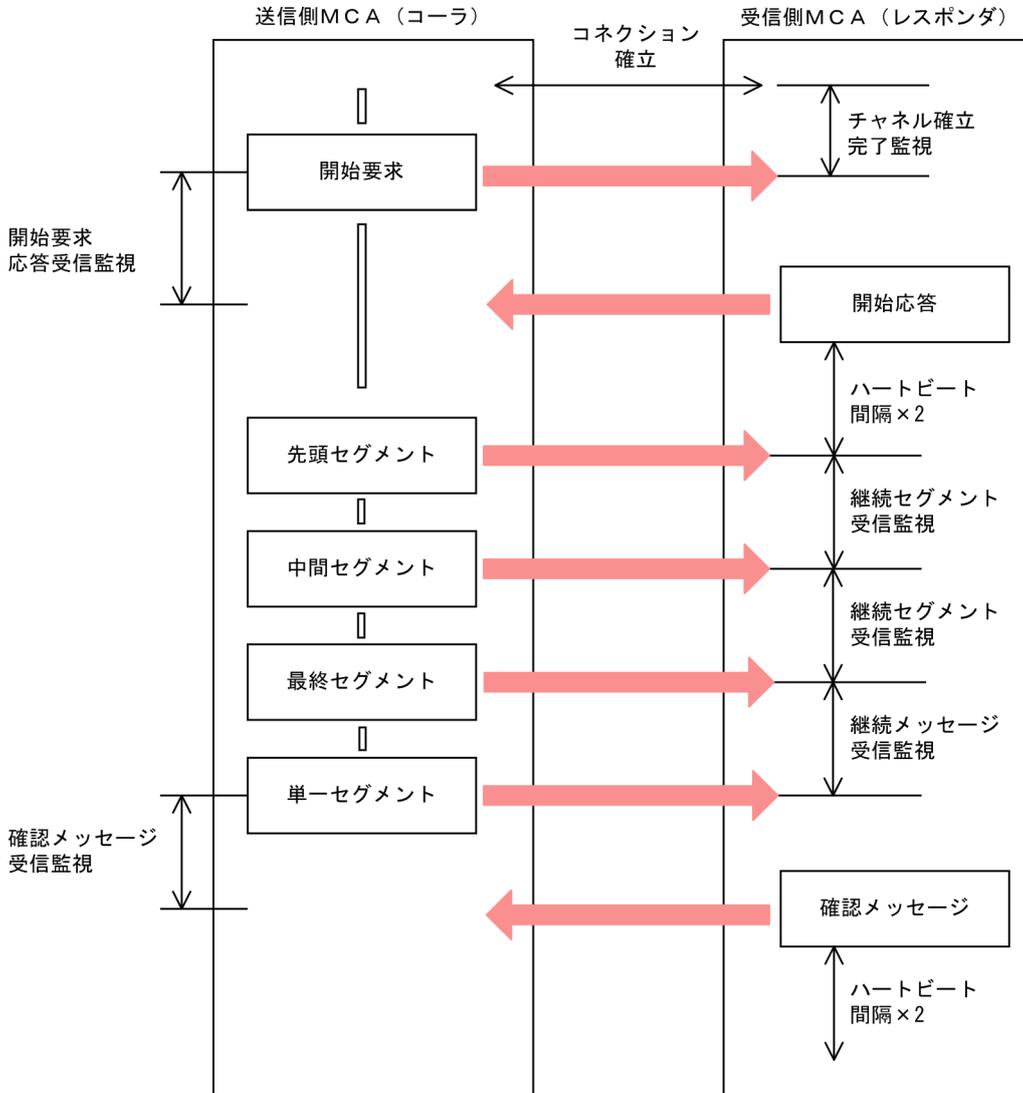
タイムアウトしたチャンネルがコーラ側 MCA の場合、タイムアウト時のチャンネル確立再試行を指定 (TCP 定義の mqtalccha 定義コマンドの -v vretry オペランドに yes を指定) すると、チャンネル確立再試行の指定 (TCP 定義の mqtalccha 定義コマンドの -b オプション) に従ってチャンネルを再確立できます。

2.5.1 送受信監視

MCA は、メッセージ送受信中に相手システムが無応答になるのを防ぐため、時間監視をします。

送受信監視とハートビート間隔による、MCA の送受信監視の例について、次の図に示します。

図 2-33 送受信監視の例



2.5.2 終了処理監視

MQT サーバが正常終了、計画停止 A または B をする場合、終了処理監視タイマ値で時間監視をします。終了処理監視タイマ値が経過してもチャネル状態が「チャネル停止」または「チャネル使用不可」ではない場合、該当のチャネルはコネクションを強制解放して終了します。

2.5.3 チャネル要求監視

MQR からクラスタプロセスへの要求（チャネル定義要求，チャネル開始要求，メッセージの送信経路再設定完了連絡，およびチャネル削除要求）の有無をこのタイマの間隔で確認します。

2.5.4 イニシエーションキュー監視

イニシエーションキューの監視方法には，次に示す方式があります。

<タイマ方式>

一定間隔でイニシエーションキューへのトリガメッセージの登録を確認し，チャネルを開始する方式です。

センダとサーバに，この方式を使用するには TCP 定義の `mqttcp` 定義コマンドで次に示す項目を指定します。

- イニシエーションキュー名 (-i オプション)
- イニシエーションキュー監視方式 (-z オプションに `time` を指定)
- イニシエーションキュー監視間隔 (-v オプション `itim` オペランド)

タイマ方式は，TP1/Message Queue 05-10 より前のバージョンとの互換のためにあります。新規に MQ システムを構築する場合は，イベント方式を使用することをお勧めします。

<イベント方式>

イニシエーションキューへのトリガメッセージの登録をイベント連絡によって検出し，チャネルを開始する方式です。

センダとサーバに，この方式を使用するには TCP 定義の `mqttcp` 定義コマンドで次に示す項目を指定します。

- イニシエーションキュー名 (-i オプション)
- イニシエーションキュー監視方式 (-z オプションに `event` を指定)

2.5.5 転送キュー監視と切断時間間隔

MQT サーバは，送信が終了して転送キューの中のメッセージがなくなってからも，指定された間隔で転送キューを監視します。転送キュー監視中に次のメッセージが転送キューに登録されなければチャネルは終了します。

転送キューは，TCP 定義の `mqtalcca` 定義コマンドの -q オプションで各送信側 MCA に一つ指定できます。一つの転送キューを複数のチャネルには割り当てられないので注意してください。

転送キューの監視方法には，次に示す方式があります。

<タイマ方式>

一定間隔で転送キューへのメッセージの登録を確認する方式です。この方式を使用するには TCP 定義の mqtalccha 定義コマンドで次に示す項目を指定します。

- 転送キュー監視方式 (-z オプションに time を指定)
- 転送キュー監視間隔 (-v オプション tim4 オペランド)
- 転送キュー監視回数 (-v オプション tim4cnt オペランド)

タイマ方式は、TP1/Message Queue 05-10 より前のバージョンとの互換のためにあります。新規に MQ システムを構築する場合は、イベント方式を使用することをお勧めします。

<イベント方式>

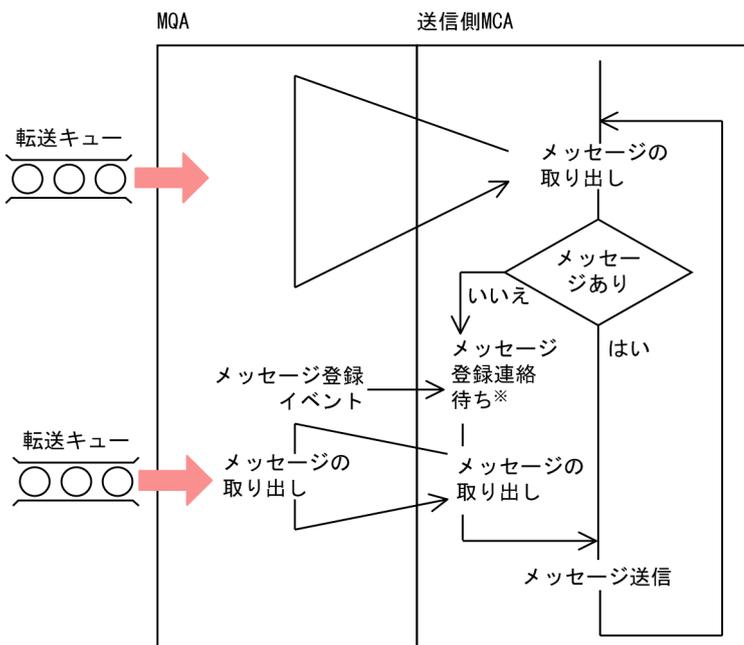
転送キューへのメッセージの登録をイベント連絡によって検出する方式です。

この方式を使用するには TCP 定義の mqtalccha 定義コマンドで次に示す項目を指定します。

- 転送キュー監視方式 (-z オプションに event を指定)
- 切断時間間隔での監視の要否 (-v オプション dtimefct オペランド)
- 切断時間間隔 (-v オプション dtim オペランド)

イベント方式による転送キュー監視について、次の図に示します。

図 2-34 イベント方式による転送キュー監視



注※

登録連絡待ちは TCP 定義の mqtalccha 定義コマンドの -v dtim オペランドに指定した切断時間間隔の間、実行します。切断時間間隔が経過するとチャンネルを終了します。

2.5.6 バッチ終了待ちタイマ（バッチインターバル）

バッチインターバル機能を使用すると、送信側 MCA がバッチサイズの個数に満たないメッセージを送信したところで転送キューの中のメッセージがなくなっても、送達確認の送信によってバッチを終了しないで転送キューへのメッセージ登録を待ちます。監視時間が経過するまでに転送キューにメッセージが登録されなければ、送達確認が送信されます。

この機能を使用するとバッチ内でアプリケーションからのメッセージ登録を待つことができるため、バッチ数を削減できます。また、ファーストチャンネルが非永続メッセージを送信する時に使用すると、不要な送達確認シーケンスによる待ち時間を削減できます。

この機能は転送キューの監視方式がイベント方式の場合にだけ使用できます。また、バッチインターバル中にはハートビートメッセージは送信されません。

2.5.7 ハートビート機能

ハートビート機能とは、送信側 MCA の転送キューにメッセージがない場合、受信側 MCA に対して一定間隔でハートビートメッセージを送信し、受信側 MCA の応答の有無によってお互いが正常に動作していることを確認する機能です。

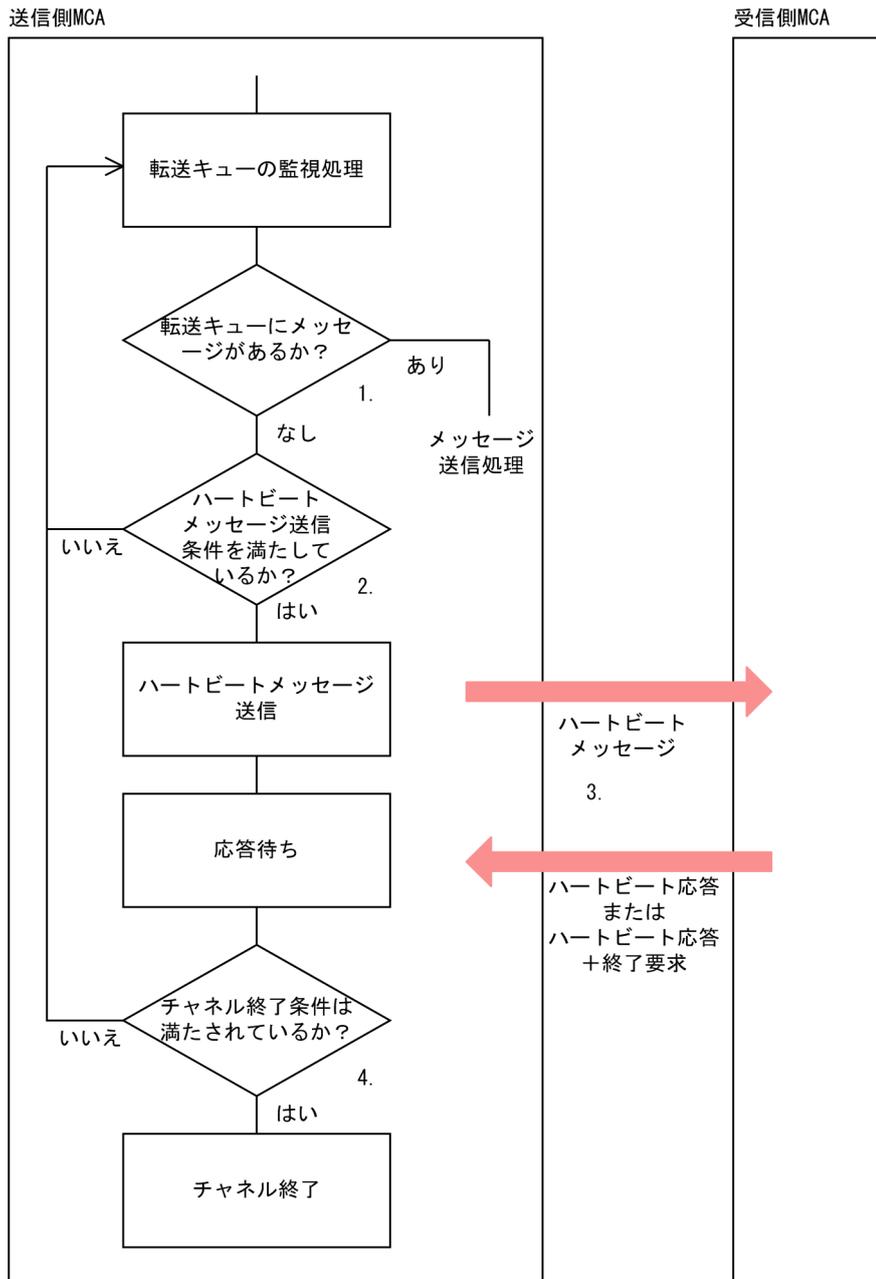
特に受信側 MCA はこのハートビートメッセージによって受信側の運用コマンドによるチャンネル終了契機を送信側 MCA に連絡できます。

ハートビート機能は、TP1/Message Queue 05-00 以降でサポートされる機能です。相手システムでもハートビート機能がサポートされていることを確認してご利用ください。サポートの有無については該当製品のマニュアルを参照してください。

(1) ハートビートメッセージ送信の契機

ハートビートメッセージ送信の契機と MCA の動作について、次の図に示します。

図 2-35 ハートビートメッセージ送信の契機と MCA の動作



1. 送信側 MCA は、転送キューにメッセージがないか確認します。転送キューにメッセージがある場合は、メッセージ送信を開始します。

2. 転送キューにメッセージがなく、次に示す条件が満たされている場合、送信側 MCA はハートビートメッセージを送信します。

前回のバッチまたはハートビート送信完了からの経過時間 \geq ハートビート間隔

3. 受信側 MCA は、ハートビートメッセージを受信すると、ハートビートメッセージへの応答を送信側 MCA に返信します。

この時、受信側 MCA が `mqtstpcha` コマンド（または `dcstop` コマンド）によるチャンネルの正常終了処理中だった場合、受信側 MCA はハートビートメッセージへの応答に加えてチャンネルの終了要求を返信し、終了します。

4.送信側 MCA は、次に示すどれかの条件を満たした場合、チャンネルを終了します。

- チャンネル終了を要求するコマンドを受け付けた場合
- タイマ方式の転送キュー監視で転送キュー監視回数を超えた場合
- イベント方式の転送キュー監視で切断時間間隔を超えた場合
- 受信側 MCA からハートビートメッセージに対する応答と一緒にチャンネルの終了を要求された場合
条件を満たさなかった場合、1.に戻ります。

(2) ハートビート機能の指定

ハートビート機能を使用する場合、ハートビートメッセージを送信する間隔を TCP 定義の `mqtalcccha` 定義コマンドの `-v htim` オペランドで指定します。指定できる値の範囲は 0~999999 (単位: 秒) です。ハートビート間隔は、すべてのチャンネルタイプに指定できます。

ハートビート間隔は、チャンネル確立時にコーラとレスポンド間のネゴシエーションでチェックされ、使用するかどうかが決まります。また、コーラ、レスポンド双方でハートビート間隔を指定した場合、大きい方の値が有効になります。

(3) 注意事項

- ハートビート機能を最も有効にするには、ハートビート間隔を送信側 MCA に指定した切断間隔 (転送キュー監視がタイマ方式の場合は転送キュー監視間隔×転送キュー監視回数、イベント方式の場合は切断時間間隔) よりも小さくしてください。
- ハートビート機能を使用すると、メッセージのない場合でも一定間隔でデータが送信されます。このため、ネットワーク上に不要なデータ転送が発生します。
- 転送キュー監視がタイマ方式の場合、送信側 MCA に指定した転送キュー監視回数 (TCP 定義の `mqtalcccha` 定義コマンドの `-v tim4cnt` オペランド指定値) が 1 のとき、ハートビートメッセージは送信されません。
- 受信側 MCA では一定時間 (ハートビート間隔×2)、ハートビートメッセージ、メッセージ転送、またはその他プロトコルメッセージの受信を待ちます。この時間を超えると通信状態が異常であると判断し、TCP/IP コネクションを解放します。特に、転送キュー監視がタイマ方式の場合、送信側 MCA に指定する転送キュー監視間隔 (TCP 定義の `mqtalcccha` 定義コマンドの `-v tim4` オペランド指定値) に、相手 MCA とチャンネル開始時にネゴシエーションした結果 (双方に指定されたハートビートタイマの大きい方の値) のハートビート間隔の 2 倍以上の値を設定すると、送信側がハートビートメッセージを送信する前に、受信側 MCA が通信状態に異常があると判断してコネクションを解放します。このため、送信側 MCA は TCP/IP コネクションにエラーを検出します。

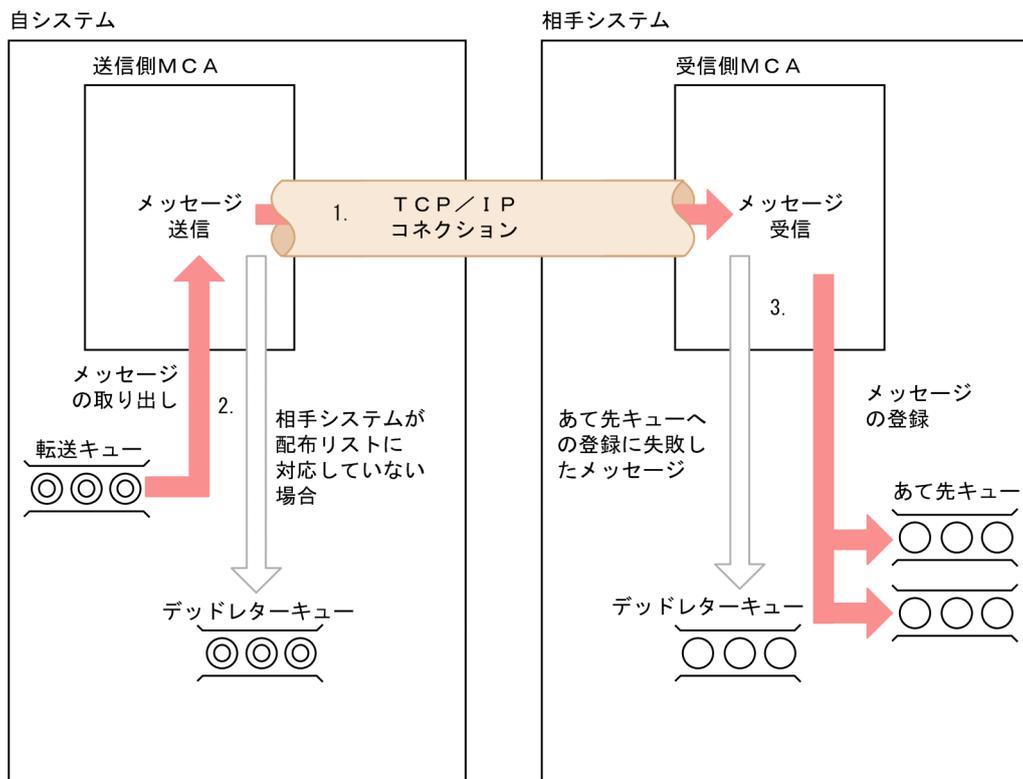
2.6 マルチキャスト機能

マルチキャスト機能とは、配布リストを使用することによって、アプリケーションからの1回のMQPUTまたはMQPUT1命令で複数のあて先にメッセージを送信する機能です。

マルチキャスト機能は、チャンネル定義のバッファ方式にメッセージ方式を指定（TCP定義のmqtalcccha定義コマンドの-g buftype オペランドにmsgを指定）した場合だけ有効です。バッファ方式がセグメント方式になっているチャンネルの転送キューに配布リスト付きのメッセージを登録した場合、TP1/Message Queueは、そのメッセージをデッドレターキューに登録します。

マルチキャスト機能の処理概要について、次の図に示します。

図 2-36 マルチキャスト機能の処理概要



(凡例)

- : 正常に配信されるメッセージの流れ
- : 配信に失敗したメッセージの流れ
- : メッセージ (配布リスト付き)
- : メッセージ (配布リストなし)

マルチキャスト機能の処理の流れについて、次に示します。

1. チャンネル確立時のネゴシエーションで、相手先システムが配布リストに対応しているかどうかを確認します。

2. 送信側の MCA は、配布リスト付きのメッセージを転送キューから取り出した場合、次に示すとおり処理します。
- 相手先システムが配布リストに対応しているときは、受信側 MCA に配布リスト付きのメッセージを送信します。
 - 相手先システムが配布リストに対応していないときは、配布リスト付きのメッセージをデッドレターキューに登録します。
3. 受信側の MCA は、配布リスト付きのメッセージを受信すると、配布リストに従ってあて先キューにメッセージに登録し、すべてのメッセージが登録されたことを確認します。
- どこかのあて先で登録に失敗したメッセージがあった場合、受信側 MCA はそのメッセージを配布リストなしでデッドレターキューに登録します。
- デッドレターキューにも登録できない場合は、チャンネルを終了します。この場合、メッセージは転送キューに残ります。

2.7 ファーストメッセージ機能

ファーストメッセージ機能は、MCA が非永続メッセージを高速に転送する機能です。この機能を持つチャンネルをファーストチャンネルといいます。また、このチャンネルが転送する非永続メッセージをファーストメッセージ、永続メッセージを通常メッセージといいます。この機能を使用する場合、TCP 定義の `mqtalcccha` 定義コマンドの `-s npmspeed` オペランドに `fast` を指定します。

この機能を使用すると、送信側 MCA が非永続メッセージ（MQMD 構造体に `MQPER_NOT_PERSISTENT` を指定）を転送キューから取り出すと、転送中のバッチとは関係なく送信します。受信側 MCA では受信したメッセージを、バッチの同期点外で登録します。ファーストメッセージはほかのメッセージと同様にバッチサイズとしてカウントされて相手システムに送信されますが、バッチには含まれず、バッチがコミットされなくても再送されることはありません。

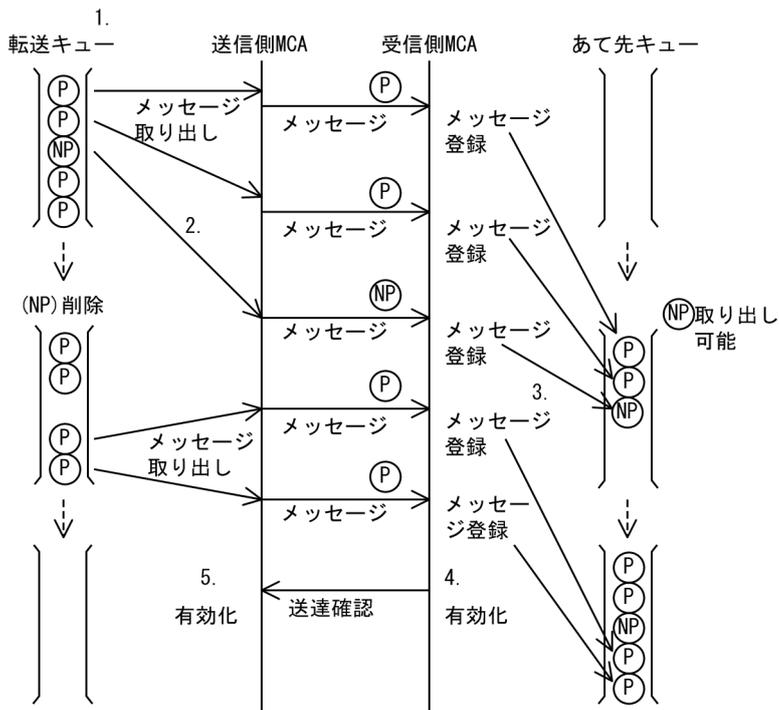
この機能の利点は、受信側 MCA が非永続性メッセージを登録するとすぐに、そのメッセージは受信アプリケーションによって取り出し可能になることです。このため、バッチの残りが送信されコミットされるのを待つ必要がありません。

しかし、送信中に通信障害やあて先キューへの登録に失敗しても再送されることはないため、メッセージの送達を保証する必要がある重要なメッセージの送信には使用できません。

また、受信側キューマネージャは、メッセージをセグメント分割してあて先キューに登録する機能を持っていますが、ファーストメッセージの場合はセグメント分割して登録できません。そのため、送信側のアプリケーションによって MQMD 構造体の `MsgFlags` フィールドに `MQMF_SEGMENTATION_ALLOWED` が指定されたメッセージが登録された場合、受信側ではセグメント分割して登録できません。このときメッセージは消滅します。メッセージのセグメント分割については、マニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。あわせて、MQMD 構造体の `MsgFlags` フィールドについて、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

ファーストメッセージの転送シーケンスの例について、次の図に示します。

図 2-37 ファーストメッセージの転送シーケンスの例



(凡例)

- Ⓟ : 永続メッセージ
- ⓃP : 非永続メッセージ

1. 非永続メッセージを含む5メッセージを転送キューに登録します。
2. 非永続メッセージを取り出すと転送キューから削除され、相手システムに送信されます。
3. 受信したメッセージをあて先キューに登録すると、すぐに有効化され取り出し可能になります。
4. 残りのメッセージを受信してバッチを有効化します。
5. 残りのメッセージが有効化によってコミット (削除) されます。

ファーストメッセージの障害処理については、「7.3 ファーストメッセージ障害」を参照してください。

2.8 キューマネージャクラスタの機能

TP1/Message Queue は IBM MQ のキューマネージャクラスタにパーシャルリポジトリとして参加できません。

この節では、IBM MQ で規定されるクラスタの概念と機能について説明します。

2.8.1 クラスタの概要

キューマネージャクラスタの概念について説明します。概念を説明するために、クラスタを使用する場合と分散キューイングの場合とを比較します。分散キューイングはクラスタを使用しない従来の手法です。

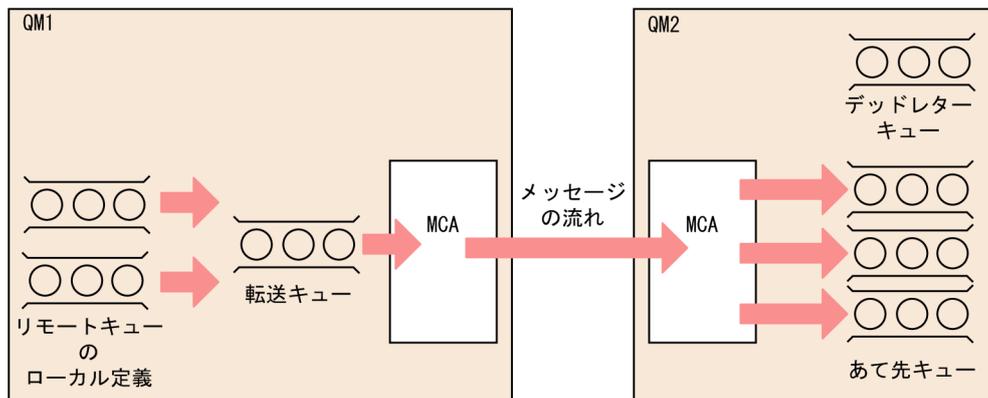
(1) 分散キューイングの概念

クラスタを使用しない場合、キューマネージャは相互に独立し、分散キューイングを使用して通信します。あるキューマネージャがほかのキューマネージャにメッセージを送信する場合、送信側キューマネージャには、次に示す定義が必要です。

- 転送キュー
- リモートキューマネージャへのチャンネル
- メッセージを送信したい各キューのリモートキューのローカル定義

分散キューイングのシステム構成について、次の図に示します。

図 2-38 分散キューイングのシステム構成



(2) クラスタの概念

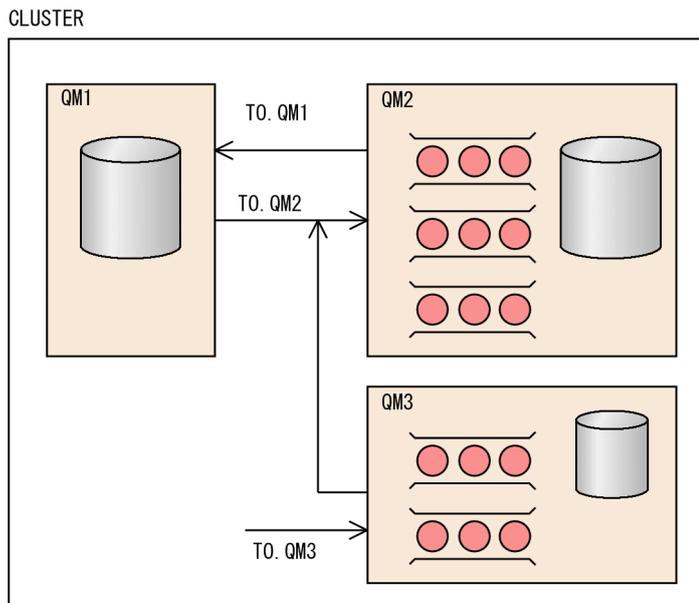
キューマネージャをクラスタにまとめた場合、キューマネージャは、自分が保持しているキューを、クラスタ内のすべての他キューマネージャに利用させることができます。すべてのキューマネージャは、明示的なチャンネル定義、リモートキューのローカル定義、および各あて先の転送キューがなくても、同じクラスタのほかのキューマネージャにメッセージを送信できます。クラスタ内のすべてのキューマネージャは、クラスタの

ほかのキューマネージャにメッセージを転送できる一つのクラスタ転送キューを持っています。クラスタの各キューマネージャは、次について定義します。

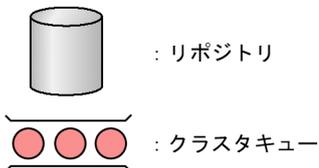
- メッセージを受信するための一つのクラスタレシーバチャンネル
- 自分をクラスタに紹介し、情報を取得するための一つのクラスタセンダチャンネル
- クラスタ環境のシステムキュー

CLUSTER というクラスタのシステム構成について、次の図に示します。

図 2-39 クラスタのシステム構成



(凡例)



- このクラスタには QM1, QM2, および QM3 という三つのキューマネージャがあります。
- QM1 と QM2 は、クラスタ内のキューマネージャ情報のリポジトリを保持し、管理しています。これらは、フルリポジトリキューマネージャといます (TP1/Message Queue はフルリポジトリキューマネージャとしての動作をサポートしていません)。QM3 は自キューマネージャ情報のほかに QM1 と QM2 から得た、他キューマネージャ情報を保持しています。QM3 をパーシャルリポジトリキューマネージャといます。
- QM2 と QM3 は、クラスタに参加しているほかのキューマネージャがアクセスできる幾つかのキューを保持しています。これらをクラスタキューといます。

分散キューイングの場合と同様に、アプリケーションは MQPUT 命令を使用して、メッセージを任意のキューマネージャのクラスタキューに登録します。アプリケーションは、MQGET 命令を使用して、ローカルキューマネージャのクラスタキューからメッセージを取り出します。

- 各キューマネージャには、メッセージを受信するための TO.QMx (x: 1~3) というチャンネルの受信端の定義があります。これは、クラスタレシーバチャンネルです。クラスタレシーバチャンネルは、分散キューイングで使用されるレシーバチャンネルのようなものですが、メッセージを転送する以外に、クラスタについての情報を通信することもできます。
- また、各キューマネージャには、一つのフルリポジトリキューマネージャのクラスタレシーバチャンネルに接続する、チャンネルの送信端の定義もあります。これは、クラスタセンダチャンネルです。QM1 と QM3 は、TO.QM2 に接続しているクラスタセンダチャンネルを持っています。QM2 は、TO.QM1 に接続しているクラスタセンダチャンネルを持っています。クラスタセンダチャンネルとは、分散キューイングで使用されるセンダチャンネルのようなものですが、このチャンネルは、メッセージを転送するほかに、クラスタについての情報を通信することもできます。
チャンネルのクラスタレシーバ端とクラスタセンダ端の両方が定義されれば、チャンネルは自動的に開始されます。

(3) クラスタの長所

クラスタを使用すると、次に示す利点が得られます。

(a) システム管理の軽減

クラスタを構築すると、システム管理が簡単になります。クラスタでキューマネージャのネットワークを構築すると、分散キューイングを使用してネットワークを構築する場合よりも、定義が少なくなります。定義が少なくなるので、ネットワークを迅速かつ簡単に設定および変更でき、定義で間違いをする危険も少なくなります。

(b) 可用性の向上と負荷分散

簡単なクラスタを構築するとシステム管理が容易になります。複雑なクラスタを構築する場合には、定義できるキューの数が増えて可用性が向上します。複数のキューマネージャで同じ名前のキューを定義できるので、負荷をクラスタ内のキューマネージャに分散できます。

(4) クラスタの考慮事項

- クラスタを最大限に活用するためには、ネットワークのすべてのキューマネージャが、クラスタをサポートしているプラットフォーム上にある必要があります。すべてのシステムが、クラスタをサポートしているプラットフォームに移植されるまで、クラスタの外のキューマネージャは、手作業で追加を定義してクラスタキューにアクセスすることがあります。
- 同じ名前の二つのクラスタがマージされた場合、それらを再度二つに分けることはできません。したがって、すべてのクラスタをユニークな名前にするをお勧めします。
- クラスタに参加するキューマネージャ名は、参加する各クラスタ内でユニークな名前にする必要があります。
- キューマネージャにメッセージが到着したときに、そのキューマネージャにメッセージを受信するキューがない場合、分散キューイングと同様に、デッドレターキューに登録されます。デッドレターキューがな

い場合、送信処理は失敗し、mqtalcccha 定義コマンドの-b bretrymcp オペランドが yes のときチャンネル確立再試行が発生します。

- 永続的メッセージの一貫性は保持されます。クラスタを使用した結果、メッセージが重複したり、喪失したりすることはありません。
- クラスタを使用すれば、システム管理が軽減されます。クラスタは、分散キューイングを使用するよりも、多くのキューマネージャと大きなネットワークの接続を容易にします。しかし、分散キューイングの場合と同様に、クラスタのすべてのキューマネージャ間での通信を有効にしようとした場合、ネットワークリソースが過度に消費されることがあります。
- 配布リストは、一回のMQPUT 命令で同じメッセージを複数のあて先に送信することを目的とします。配布リストはクラスタ環境でも使用できます。システム管理者にとっては、配布リストを使用する場合、複数のチャンネルや転送キューを手作業で定義する必要がないという利点があります。しかし、リモートのクラスタキューへメッセージを登録する目的で配布リストを使用する場合は、あて先ごとにメッセージが処理されます。ネットワークトラフィックの観点では、分散キューイング環境と比べて、利点がありません。
- 負荷分散のためにクラスタを使用する場合、自分のアプリケーションを調査して、メッセージが特定のキューマネージャまたは特定のシーケンスで処理されることがアプリケーションで必要でないか調べる必要があります。そのようなアプリケーションは、**メッセージ類似性**を持つといいます。複雑なクラスタで使用する前に、アプリケーションの修正が必要になることもあります。
- メッセージが特定のあて先に送信されるように MQOPEN 命令で MQOO_BIND_ON_OPEN オプションを使用した場合、あて先のキューマネージャが利用できないとき、そのメッセージは転送されません。複製の危険があるので、ほかのキューマネージャに向けてメッセージの送信経路が設定されることはありません。
- クラスタに参加するキューマネージャのクラスタセンダチャンネルでは mqtalcccha 定義コマンドの-o オプションに、フルリポジトリキューマネージャのホスト名または IP アドレスを指定します。しかし、DHCP を使用する場合は、IP アドレスが変更される可能性があります。システムの再開始時に DHCP が新しい IP アドレスを割り当てることがあるからです。そのため、この場合はクラスタセンダチャンネルの mqtalcccha 定義コマンドの-o オプションに IP アドレスを指定しないでください。すべてのクラスタセンダチャンネルで IP アドレスではなくホスト名を指定しても十分な信頼性はありません。DHCP はホストの DNS ディレクトリエントリを新しいアドレスで更新するとは限らないからです。

注

DNS ディレクトリが最新状態に更新されるよう保証するソフトウェアをインストールしていない場合は、DHCP を使用するシステム上のキューマネージャをフルリポジトリに割り当てないでください。

- メッセージの取り出しはローカルのクラスタキューだけから可能ですが、メッセージの登録はクラスタ内のすべてのキューへ可能です。MQGET 命令を使用するためにキューをオープンする場合は、キューマネージャはローカルキューだけを使用します。
- クラスタに新規に参加する場合、または再参加する場合、アプリケーションで使用するクラスタキューの情報がないことがあります。この状態で、クラスタキューへアクセスするアプリケーションを開始すると、MQOPEN 命令でエラーが発生することがあります。そのため、自システムで使用するクラス

タキューに対して MQOPEN 命令を発行するだけのアプリケーションを作成し、事前にクラスタキューの情報を収集することをお勧めします。

(5) まとめ

TP1/Message Queue は IBM MQ のクラスタにパーシャルリポジトリとして参加できます。クラスタに参加する場合、システム管理者は、対応する送信チャンネルとキューマネージャ上のリモートキューのローカル定義を作成する必要がなくなります。これらはキューマネージャによって自動的に作成されます。

クラスタに参加するすべてのキューマネージャは、自身および保持するキューについての情報をフルリポジトリキューマネージャに送信し、クラスタ内のほかのキューマネージャについての情報を受信します。

各キューマネージャにあるクラスタ転送キューは、情報をシステムメッセージとして送受信するときに使用されます。また、ユーザメッセージを送受信するときにも使用されます。

情報は、各キューマネージャのリポジトリに格納されます。また、クラスタ内のフルリポジトリキューマネージャでは、すべてのキューマネージャとキューについての情報をフルリポジトリとして管理します。

クラスタレシーバチャンネルは、レシーバチャンネルに似ています。クラスタレシーバチャンネルの定義の実行時には、キューマネージャ上にオブジェクトが作成されるだけでなく、保持するチャンネルとキューマネージャについての情報もリポジトリに格納されます。クラスタレシーバチャンネルの定義は、クラスタに対するキューマネージャの初期通知動作です。いったん定義すると、ほかのキューマネージャは対応するクラスタセンド側の定義を自動的に作成できます。

クラスタセンドチャンネルはセンドチャンネルに似ています。クラスタセンドチャンネルはほかのクラスタキューマネージャと通信するときに、対応するクラスタレシーバチャンネルの定義を参照して、自動的に作成されます。しかし、各キューマネージャには最初にクラスタに接続するための事前定義クラスタセンドチャンネルが一つ必要です。

クラスタをサポートするキューマネージャが、クラスタの一部になる必要はありません。クラスタを使用する代わりに、分散キューイングの手法を使い続けることができます。

2.8.2 クラスタの使用によるシステム管理の軽減

クラスタを使用して自分の環境でのシステム管理を軽減する方法を説明します。ここでは、簡単なクラスタを設定する方法と利点について説明します。

(1) クラスタの使用方法

クラスタにはキューマネージャがあり、通常、それらは論理的に関係を持ってデータやアプリケーションを共有します。例えば、自分の会社の各部門に一つのキューマネージャがあり、その部門に固有のデータやアプリケーションを管理していることがあります。すべてのキューマネージャが給与支払アプリケーションを使用できるように、これらのキューマネージャをすべて一つのクラスタにまとめることができます。または、チェーンストアの各支店に一つのキューマネージャを配置し、その支店の在庫などを管理できます。これら

のキューマネージャをクラスタにまとめた場合、本社のキューマネージャに集中されているように販売アプリケーションや購買アプリケーションにアクセスできます。

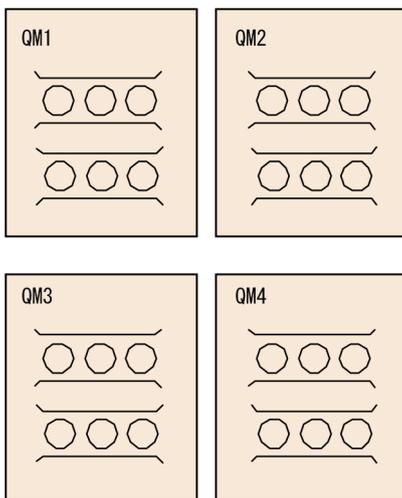
クラスタが設定されれば、そのクラスタ内のキューマネージャは、チャンネル定義やリモートキューのローカル定義なしで、相互に通信できます。

キューマネージャの既存のネットワークはクラスタに変換できます。また、新しいネットワークを構築しようとしている場合には、迅速にクラスタを構築できます。

(2) システム管理上の利点

クラスタを使用すると、ネットワークの管理が簡単になります。次の図は、それぞれ二つのキューを持っている四つのキューマネージャです。はじめに、分散キューイングでこれらのキューマネージャを接続するのに定義が幾つ必要か検討します。そのあと、同じネットワークをクラスタとして設定するのに定義が幾つ必要か検討します。

図 2-40 二つのキューを持っている四つのキューマネージャ



(a) 分散キューイングでネットワークを設定する定義

分散キューイングを使用してネットワークを設定するために必要な定義を次の表に示します。

表 2-13 分散キューイングでの定義

説明	キューマネージャ単位での定義数	定義総数
ほかのキューマネージャにメッセージを送信するチャンネルのセンドチャンネル定義	3	12
ほかのキューマネージャからメッセージを受信するチャンネルのレシーバチャンネル定義	3	12
ほかのキューマネージャへの転送キューの定義	3	12
各ローカルキューの定義	2	8

説明	キューマネージャ単位での定義数	定義総数
キューマネージャがメッセージを登録するためのリモートキューのローカル定義	6	24

(b) クラスタでネットワークを設定する定義

クラスタを使用する場合、次に示すとおり定義します。

- 各キューマネージャに一つのクラスタセンダチャンネル定義とクラスタレシーバチャンネル定義
- 別個に定義される転送キューはなし
- リモートキューのローカル定義はなし

したがって、クラスタを使用してネットワークを設定するためには、次の表に示す定義が必要です。

表 2-14 クラスタでの定義

説明	キューマネージャ単位での定義数	定義総数
フルリポジトリキューマネージャにメッセージを送信するチャンネルのクラスタセンダチャンネル定義	1	4
クラスタ内のほかのキューマネージャからメッセージを受信するチャンネルのクラスタレシーバチャンネル定義	1	4
各ローカルキューの定義	2	8
クラスタ環境のシステムキューの定義*	3	12

注※

詳細については、「[2.8.11\(6\) クラスタ環境のシステムキュー](#)」を参照してください。

二つのフルリポジトリを持ったキューマネージャでこのクラスタを設定するためには、各キューマネージャで七つの定義（合計で 28 の定義）をユーザが定義する必要があります。また、クラスタのリポジトリキューマネージャにするために、IBM MQ の二つのキューマネージャでキューマネージャ定義の変更が必要です。

クラスタセンダチャンネル定義とクラスタレシーバチャンネル定義は一つだけ作成します。クラスタがそろったら、ほかのキューマネージャを妨げることなく、（フルリポジトリキューマネージャ以外の）キューマネージャを追加したり、取り除いたりできます。

これによって、多数のキューマネージャによるネットワークの設定に必要な定義の数が少なくなります。定義が少なくなるので、次に示すとおりエラーのリスクも減少します。

- オブジェクト名（例えば、センダとレシーバのペアでのチャンネル名）の不一致がありません。
- チャンネル定義で指定されている転送キュー名が転送キューの定義と一致しているかどうか、リモートキューのローカル定義で指定されている転送キュー名と一致しているかどうか、考慮する必要はありません。
- リモートキューのローカル定義（MQA サービス定義の `mqaqueue` 定義コマンドの `-r` オプション）がリモートキューマネージャの間違ったキューを指すことはありません。

さらに、クラスタが設定されれば、ほかのキューマネージャでシステム管理作業をしなくても、クラスタ内でクラスタキューをあるキューマネージャから別のキューマネージャに移動できます。チャンネル定義、リモートキューのローカル定義、または転送キューの定義の削除や修正漏れはありません。既存のネットワークに影響を与えないで、新しいキューマネージャをクラスタに追加できます。

(3) アプリケーションの見直し

簡単なクラスタを設定する場合には、自分のアプリケーションを修正する必要はありません。アプリケーションでは、通常どおりに MQOPEN 命令でターゲットキューを指定し、キューマネージャの位置を意識する必要はありません。

しかし、クラスタ内に同じ名称のクラスタキューが複数存在する場合には、自分のアプリケーションを見直し、必要に応じて修正してください。

詳細については、「[2.8.6 ワークロード管理へのクラスタの使用](#)」を参照してください。

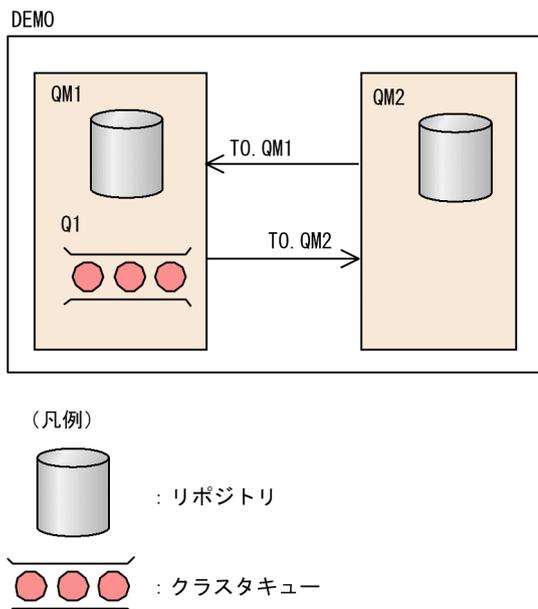
(4) クラスタの設定方法

複数のキューマネージャから成るクラスタを構築すると決定したら、どのキューマネージャにクラスタのフルリポジトリを持たせるかを検討する必要があります。そこで、任意の数のキューマネージャを選択できますが、二つに設定することをお勧めします。

小さなクラスタの場合、キューマネージャは二つしかありません。しかし、TP1/Message Queue はフルリポジトリをサポートしていないため、フルリポジトリをサポートした IBM MQ が必要になります。

二つのキューマネージャがあるクラスタについて、次の図に示します。

図 2-41 二つのキューマネージャがある小さなクラスタ



このようなクラスタの設定方法については、「[2.8.3 クラスタのセットアップ](#)」を参照してください。

2.8.3 クラスタのセットアップ

ここでは、具体的な作業例を挙げてクラスタ構築およびクラスタの使用方法について説明します。

TP1/Message Queue の環境を構築する手順については、4章の「[TP1/Message Queue の環境作成手順](#)」を参照してください。また、フルリポジトリを設定する場合は、IBM MQ のマニュアルを参照してください。

例では、キューマネージャに LONDON や NEWYORK などの名前を付けます。

(1) クラスタのセットアップ例の説明

作業で前提になる例について説明します。

- チェーン店用の新しいネットワークをセットアップします。この店には、二つの支店（ロンドンとニューヨークに1店ずつ）があることとします。
- 各店のデータとアプリケーションは、別々のキューマネージャが動作するシステムが保持します。この二つのキューマネージャをそれぞれ、LONDON と NEWYORK とします。次に示す製品で構築します。
キューマネージャ (LONDON) : IBM MQ で構築
キューマネージャ (NEWYORK) : TP1/Message Queue で構築
- キューマネージャ (NEWYORK) に接続されたニューヨークのシステム上では、在庫管理アプリケーションを実行します。このアプリケーションは、INVENTQ キュー (NEWYORK が保持する) 上の着信メッセージによって動作します。
- 二つのキューマネージャ (LONDON と NEWYORK) は、INVENTORY というクラスタにリンクされるため、両者ともメッセージを INVENTQ キューに登録できます。

(2) 作業を完了させるために必要な手順

このクラスタのセットアップは、次に示す手順に従って作業します。

1. キューマネージャ (LONDON) を IBM MQ で構築します。
2. キューマネージャ (NEWYORK) を TP1/Message Queue で構築します。
TP1/Message Queue の環境を構築する手順については、4章の「[TP1/Message Queue の環境作成手順](#)」を参照してください。
3. クラスタの編成およびその名前を定義します。
二つのキューマネージャ (LONDON と NEWYORK) へのリンクをクラスタ内に定義します。二つしかキューマネージャがリンクされていないクラスタでは、分散キューイングを使用するネットワークよりもそれほど有利ではありませんが、拡張の余地を残しておく点では、良い方法です。新しい支店を開店する場合には、クラスタに新しいキューマネージャを簡単に、かつ既存のネットワークに影響を与えないで追加できます。

この場合、実行中のアプリケーションは在庫管理アプリケーションになり、クラスタ名は INVENTORY になります。

4. フルリポジトリを保持するキューマネージャ（フルリポジトリキューマネージャ）を決定します。

どのクラスタでも、フルリポジトリを保持する最低一つ（望ましいのは二つ）のキューマネージャを指定します。

TP1/Message Queue はフルリポジトリをサポートしていないため、フルリポジトリをサポートした IBM MQ が必要です。この例では、キューマネージャ（LONDON）がフルリポジトリキューマネージャになります。

ポイント

以降の手順は、どの順序で実行してもかまいません。

5. クラスタレシーバチャンネルを定義します。

クラスタ内の各キューマネージャ上で、参加するクラスタごとに、キューマネージャがメッセージを受信するためのクラスタレシーバチャンネルを定義します。この定義によって、キューマネージャのネットワークアドレスが定義され、クラスタ内のほかのキューマネージャからメッセージを受信できることを通知する効果があります。

この定義は、クラスタ内のほかのキューマネージャに通知されます。コンピュータのネットワークアドレスは、ほかのキューマネージャが参照できるリポジトリ内に格納されます。

キューマネージャ（LONDON）での定義を次に示します。

```
DEFINE CHANNEL(TO.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(192.168.0.1) CLUSTER(INVENTORY)
```

キューマネージャ（NEWYORK）での定義を次に示します。

```
mqtalccha -c TO.NEWYORK -y" type=clusrcvr" ¥
           -r ipaddr=192.168.0.2           ¥
           -a "cluster = INVENTORY"
```

詳細については、「[2.8.9 クラスタ環境の通信構成](#)」を参照してください。

6. クラスタセンダチャンネルを定義します。

クラスタ内の各キューマネージャ上で、参加するクラスタごとに、クラスタセンダチャンネルを一つ定義します。このクラスタセンダチャンネルで、フルリポジトリキューマネージャの一つにメッセージを送信できます。この場合、フルリポジトリキューマネージャは LONDON です。パーシャルリポジトリキューマネージャは NEWYORK です。キューマネージャ（NEWYORK）には、キューマネージャ（LONDON）で定義されたクラスタレシーバチャンネルを接続先とする事前定義クラスタセンダチャンネル定義が必要です。事前定義クラスタセンダチャンネル定義に指定するチャンネル名は、対応するクラスタレシーバチャンネル定義に指定された名前と一致していなければいけないので、注意が必要です。

キューマネージャに、同じクラスタ内のクラスタレシーバチャンネルとクラスタセンダチャンネルの両方の定義が設定されると、クラスタセンダチャンネルが開始します。

キューマネージャ（NEWYORK）での定義を次に示します。

```
mqtalccha -c T0.LONDON -y" type=clusdr" ¥
           -o oipaddr=192.168.0.1      ¥
           oservername=listenerport    ¥
           -a "cluster = INVENTORY"
```

詳細については、「2.8.9 クラスタ環境の通信構成」を参照してください。

7. クラスタキュー INVENTQ を定義します。

INVENTQ キューをキューマネージャ (NEWYORK) で定義します。このキュー定義はクラスタのほかのキューマネージャに通知されます。

キューマネージャ (NEWYORK) でのコマンド入力を次に示します。

```
mqamkque -c INVENTORY INVENTQ モデルキューの定義名 キュー属性定義ファイル名
```

8. MQA サービス定義に定義コマンドを追加します。

次に示す項目を MQA サービス定義に追加します。

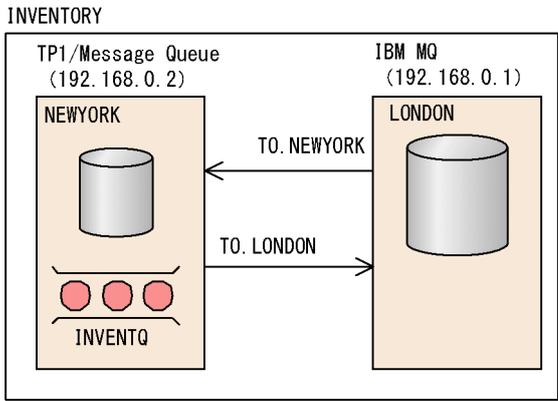
- クラスタセンダプロセスおよびクラスタレシーバプロセスに対応する mqamqtnam 定義コマンド
- システムキューに対応する mqaquegrp 定義コマンド

これで、すべての定義が完了します。TP1/Message Queue を開始してください。

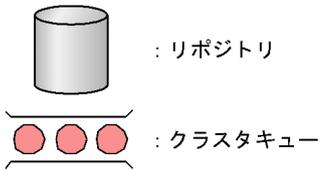
(3) 作業で構築されるクラスタ

この作業で構築されるクラスタについて、次の図に示します。

図 2-42 二つのキューマネージャがある INVENTORY クラスタ



(凡例)



これは、とても小さなクラスタです。しかし、将来に向けての拡張性があります。

(4) 作業で構築したクラスタを使用する

INVENTQ キューはクラスタに通知されているため、リモートキューのローカル定義は不要です。NEWYORK 上で動作するアプリケーションと LONDON 上で動作するアプリケーションは、メッセージを INVENTQ キューに登録できます。また、これらのアプリケーションは、キューにメッセージに登録する時に、応答キューを提供し、その名前を指定することによって、メッセージに対する応答を受信することもできます。

次に示す例では、LONDON が INVENTQ にメッセージに登録し、キュー LONDON_reply に応答を受信します。

LONDON 上で次に示すとおり動作します。

1. LONDON_reply という名前のローカルキューを定義する。
2. MQOPEN 命令のオプションに MQOO_OUTPUT を設定する。
3. MQOPEN 命令を発行し、キュー INVENTQ を開く。
4. メッセージ記述子の ReplyToQ に LONDON_reply を指定する。
5. MQPUT 命令を発行し、メッセージに登録する。

NEWYORK 上で次に示すとおり動作します。

1. MQOPEN 命令のオプションに MQOO_BROWSE を設定する。
2. MQOPEN 命令を発行し、キュー INVENTQ を開く。
3. MQGET 命令を発行し、INVENTQ からメッセージを取り出す。
4. メッセージ記述子から ReplyToQ 名を取得する。
5. オブジェクト記述子の ObjectName に ReplyToQ 名を指定する。
6. オブジェクト記述子の ObjectQMgrName にキューマネージャ (LONDON) を指定する。
7. MQOPEN 命令のオプションに MQOO_OUTPUT を設定する。
8. MQOPEN 命令を発行し、キューマネージャ (LONDON) にある LONDON_reply を開く。
9. MQPUT 命令を発行し、メッセージを LONDON_reply に登録する。

LONDON 上で次に示すとおり動作します。

1. MQOPEN 命令のオプションに MQOO_BROWSE を設定する。
2. MQOPEN 命令を発行し、キュー LONDON_reply を開く。
3. MQGET 命令を発行し、LONDON_reply からメッセージを取り出す。

注意事項

ローカルキュー LONDON_reply の定義には、ClusterName 属性は不要です。

(5) 既存のネットワークのクラスタへの変更

既存の分散キューイング環境を、このようなクラスタに変更した場合、「[2.8.3\(2\) 作業を完了させるために必要な手順](#)」の手順 7 では、既存のキューの定義を変更してください。また、LONDON で、INVENTQ キューに対するリモートキューのローカル定義を必ず削除してください。

2.8.4 クラスタへの新しいキューマネージャの追加

ここでは、次に示す作業を実行する方法について説明します。

フルリポジトリを設定する場合は、IBM MQ のマニュアルを参照してください。

(1) クラスタへの新しいキューマネージャの追加例の説明

作業で前提になる例について説明します。

- INVENTORY クラスタには二つのキューマネージャ、LONDON と NEWYORK が組み込まれていて、LONDON がフルリポジトリを保持しています。
- チェーン店の新しい支店をパリに設定します。このクラスタに、PARIS というキューマネージャを追加するとします。
- キューマネージャ (PARIS) は、INVENTQ キューにメッセージを登録することによって、在庫表の更新情報をニューヨークのシステムで動作するアプリケーションに送信します。
- 三つのシステムは互いに接続できます。
- ネットワークプロトコルは TCP です。

(2) 作業を完了させるために必要な手順

次に示す手順に従って作業します。

1. キューマネージャ (PARIS) を準備します。

キューマネージャの準備については、「[2.8.3\(2\) 作業を完了させるために必要な手順](#)」を参照してください。

2. PARIS が参照するフルリポジトリを決定します。

クラスタ内のすべてのキューマネージャは、クラスタについての情報を収集し、独自のパーシャルリポジトリを構築するために、フルリポジトリを参照する必要があります (クラスタ内に二つ以上のフルリポジトリがある場合は、どれか一つのフルリポジトリを参照する必要があります)。

自キューマネージャで変更した情報は、直接フルリポジトリに送信されます (クラスタ内に二つ以上のフルリポジトリがある場合は、直接二つのフルリポジトリに送信されます)。

この例では、フルリポジトリは LONDON だけです。キューマネージャ (PARIS) から LONDON への接続定義を作成します。

ポイント

以降の手順は、どの順序で実行してもかまいません。

3. キューマネージャ (PARIS) 上で、クラスタレシーバチャンネルを定義します。

クラスタ内の各キューマネージャ上で、参加するクラスタごとに、キューマネージャがメッセージを受信するためのクラスタレシーバチャンネルを定義します。

キューマネージャ (PARIS) での定義を次に示します。

```
mqtalcccha -c TO.PARIS -y" type=clusrcvr" ¥  
            -r ipaddr=192.168.0.3          ¥  
            -a "cluster = INVENTORY"
```

これによって、クラスタ INVENTORY 内のほかのキューマネージャからメッセージを受信できるという、キューマネージャの可用性が通知されます。

クラスタレシーバチャンネル TO.PARIS の受信端に送信するために、ほかのキューマネージャを定義する必要はありません。その定義は、必要に応じて自動的に実行されます。

4. キューマネージャ (PARIS) 上で、クラスタセンダチャンネルを定義します。

クラスタ内のすべてのキューマネージャでは、参加するクラスタごとに、事前定義クラスタセンダチャンネルを一つ以上定義する必要があります。事前定義クラスタセンダチャンネルを定義することによって、クラスタ制御用のシステムメッセージをフルリポジトリキューマネージャに送信できます。

PARIS では、TO.LONDON というチャンネルの定義を、キューマネージャ (LONDON) に対し作成します。

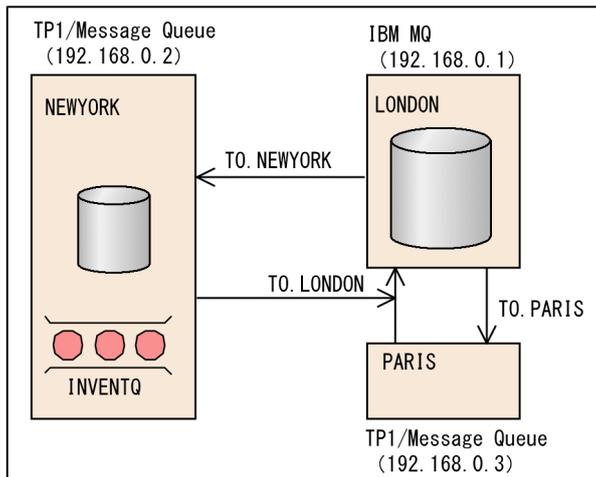
キューマネージャ (PARIS) での定義を次に示します。

```
mqtalcccha -c TO.LONDON -y" type=clusdr" ¥  
            -o oipaddr=192.168.0.1        ¥  
            oservername=listenerport     ¥  
            -a "cluster = INVENTORY"
```

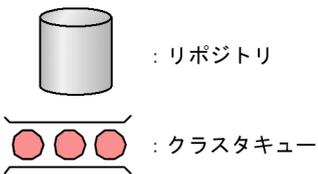
(3) 作業で構築されるクラスタ

この作業で構築されるクラスタについて、次の図に示します。

図 2-43 三つのキューマネージャがある INVENTORY クラスタ



(凡例)



二つの定義（クラスタレシーバチャンネル定義およびクラスタセンドチャンネル定義）を作成するだけで、キューマネージャ（PARIS）をクラスタに追加できました。

キューマネージャ（PARIS）は、LONDONにあるリポジトリから、クラスタキュー（INVENTQ）がキューマネージャ（NEWYORK）に存在することを知ります。PARISにあるアプリケーションがINVENTQにメッセージを登録するとき、PARISは自動的にクラスタセンドチャンネルを作成し、クラスタレシーバチャンネル TO.NEWYORK に接続します。キューマネージャ名がターゲットキューマネージャとして指定される場合、および応答キューが提供される場合は、アプリケーションは応答を受信できます。

2.8.5 クラスタの機能

クラスタについての詳細とその機能について説明します。内容は次に示すとおりです。

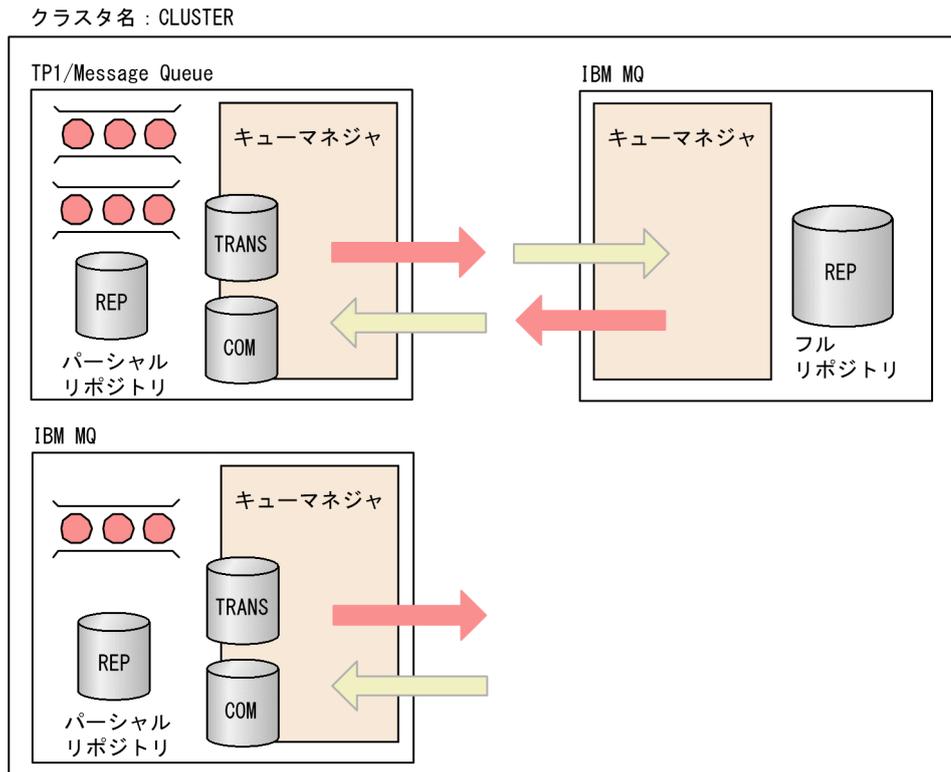
- クラスタの構成要素
- クラスタリングの動作
- クラスタに関連する別名の使用とリモートキューのローカル定義

(1) クラスタの構成要素

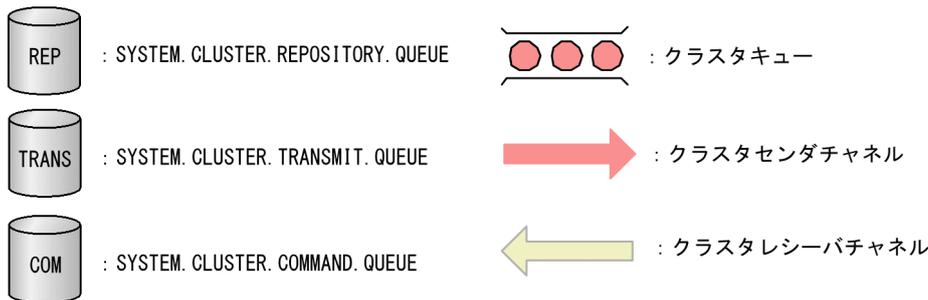
クラスタの構成要素がどのように連携して機能を果たすのか、クラスタの構成要素と機能について説明します。

クラスタの構成要素について、次の図に示します。

図 2-44 クラスタの構成要素



(凡例)



(a) キューマネージャとリポジトリ

すべてのクラスタには最低一つのフルリポジトリキューマネージャがあります。フルリポジトリキューマネージャはクラスタ内のキューマネージャ、キュー、およびチャンネルについての全情報（フルリポジトリ）を保持しています。また、クラスタ内のほかのキューマネージャから情報更新の要求があった場合も、その要求はフルリポジトリキューマネージャが保持します。

フルリポジトリキューマネージャ以外のキューマネージャ（パーシャルリポジトリキューマネージャ）はそれぞれの部分情報（パーシャルリポジトリ）を保持します。パーシャルリポジトリにはキューおよび通信する必要のあるキューマネージャの部分集合についての情報があります。キューマネージャは別のキューまたはキューマネージャにアクセスする必要が発生すると、フルリポジトリキューマネージャに問い合わせるパーシャルリポジトリを構築し、それ以降そのキューまたはキューマネージャについて新しい情報を通知するよう要求します。

各キューマネージャは SYSTEM.CLUSTER.REPOSITORY.QUEUE というキューにあるメッセージにリポジトリ情報を格納します。キューマネージャは SYSTEM.CLUSTER.COMMAND.QUEUE というキューにあるメッセージでリポジトリ情報を交換します。

クラスタのメンバである各キューマネージャは、一つのフルリポジトリキューマネージャにあるクラスタレシーバチャンネルに対応した、クラスタセンダチャンネルを定義します。これで、キューマネージャは、クラスタ内のどのキューマネージャがフルリポジトリを保持するかを知ることになります。以降、キューマネージャは、どのリポジトリにでも情報を要求できます。キューマネージャが自分自身についての情報を送信する場合（例えば、新しいクラスタキューを作成した場合）、この情報は事前定義クラスタセンダチャンネルが接続するフルリポジトリキューマネージャのほかにもう一つのフルリポジトリキューマネージャがあれば、そこにも送信されます。

フルリポジトリキューマネージャが、リンクされているキューマネージャの一つから情報を受信すると、フルリポジトリは更新されます。新しい情報は、フルリポジトリキューマネージャがリポジトリを更新できない場合のリスクを減らすため、別のフルリポジトリキューマネージャにも送信されます。すべての情報が2回送信されます。情報の各項目には、複製を識別するためのシーケンス番号が付いています。同じシーケンス番号である場合、フルリポジトリキューマネージャは2回目に受信した情報を破棄します。同様に、二つのフルリポジトリキューマネージャがある場合にパーシャルリポジトリキューマネージャに同じ情報が通知されたとき、シーケンス番号が同じであればパーシャルリポジトリは情報を破棄します。このように、すべてのリポジトリはメッセージを交換することによって内容を整合させます。

(b) クラスタキュー

クラスタキューを保持するキューマネージャは、自分のキューをクラスタに通知します。キューの作成時に mqamkque コマンドの -c オプションに指定したクラスタに対しリポジトリ管理サーバの開始時に通知します。

キューがいったん通知されると、クラスタ内のどのキューマネージャもそのクラスタキューにメッセージを登録できます。メッセージをクラスタキューに登録するには、キューマネージャは該当するクラスタキューを保持するキューマネージャをフルリポジトリから見つける必要があります。それからメッセージに送信経路情報を付けて、クラスタ転送キューにそのメッセージを登録します。

(c) クラスタ転送キュー

各クラスタキューマネージャは SYSTEM.CLUSTER.TRANSMIT.QUEUE というクラスタ転送キューを持っています。

クラスタの一部であるキューマネージャは、クラスタ転送キュー上のメッセージを同一クラスタ内のどのキューマネージャにも送信できます。

注意事項

アプリケーションでメッセージをクラスタ転送キューに直接登録しないでください。アプリケーションが名前の付いたキューにメッセージを登録すると、クラスタ転送キューに解決されます。

同様に、リモートキューのローカル定義に指定する転送キュー名に、クラスタ転送キューを指定しないでください。キューマネージャ定義に指定するデフォルトの転送キュー名に、クラスタ転送キューを指定しないでください。

キューマネージャは、クラスタの一部でない別のキューマネージャと通信することもできます。そのためにキューマネージャは、分散キューイング環境の場合と同様に、相手キューマネージャに対して転送キューとチャネルを定義する必要があります。

名称解決の間、クラスタ転送キューはデフォルトの転送キューに対して優先権を持ちます。クラスタの一部でないキューマネージャがリモートキューにメッセージを登録する場合、あて先キューマネージャに対応する転送キューがなければ、デフォルトアクションとしてデフォルトの転送キューが使用されます。クラスタの一部である送信側キューマネージャの場合、あて先キューマネージャと同名の転送キューがなければ、デフォルトアクションとして `SYSTEM.CLUSTER.TRANSMIT.QUEUE` が使用されます（ただし、あて先キューがクラスタ内にない場合を除きます）。つまり、通常の名称解決では、キュー名がフルリポジトリを使って解決される場合に `SYSTEM.CLUSTER.TRANSMIT.QUEUE` が使用されます。

注意事項

クラスタ転送キューには、ユーザメッセージとともにシステム制御用のシステムメッセージが登録されます。ユーザメッセージを削除する場合は、システム制御用のシステムメッセージを削除しないように注意してください。

(d) クラスタチャネル

クラスタ内では、クラスタレシーバチャネル定義とクラスタセンダチャネル定義という特殊なチャネルを使用して、キューマネージャ間でメッセージの送受信が行われます。

クラスタレシーバチャネル定義では、自キューマネージャがメッセージを受信できるチャネルを定義します。クラスタへの参加によって定義情報がフルリポジトリキューマネージャに保持されます。保持される主な内容は、連絡属性が「あり」の属性です。詳細については、「[3.3.2\(3\) チャネルデータ定義ブロック \(dcmtcq_uoc_mqcd\)](#)」を参照してください。これによって、ほかのキューマネージャは対応するクラスタセンダチャネル定義をそのキューマネージャに対して自動定義できます。ただし、最初は各キューマネージャでフルリポジトリキューマネージャ向けの事前定義クラスタセンダチャネルを手作業で定義する必要があります。この定義によってキューマネージャは、参加するクラスタに自システムの情報を送信できます。

(e) リモートキューのローカル定義

クラスタでは、同じクラスタ内のリモートキューに対応する、リモートキューのローカル定義が自動的にキューマネージャに作成されます。そのため、リモートキューのローカル定義を手作業で作成する必要はありません。クラスタキューマネージャはフルリポジトリキューマネージャからリモートキューの格納場所についての情報を取得します。格納場所が分かったら、送信経路情報をメッセージに追加して、クラスタ転送キューにメッセージを登録します。

(f) チャネルの自動定義

分散キューイングを使用する場合、キューマネージャがリモートの先へメッセージを送信するときに、センダチャネルの定義が必要です。

しかし、キューマネージャがクラスタセンダチャネル定義とクラスタレシーバチャネル定義を作成してクラスタに参加することによって、クラスタ内のほかのキューマネージャに対応するチャネル定義を作成する必要がなくなります。

クラスタセンダチャネル定義は必要に応じて自動的に作成されます。自動定義されたクラスタセンダチャネルは、受信側キューマネージャの持つクラスタレシーバチャネル定義に指定された属性を引き継ぎます。手作業で定義されたクラスタセンダチャネル（事前定義クラスタセンダチャネル）の場合もクラスタレシーバチャネル定義に指定された属性を引き継ぎ、対応するクラスタレシーバチャネル定義との一致が保たれます。また、メッセージ編集出口 UOC を使用すると、クラスタセンダチャネルまたはクラスタレシーバチャネルの属性を変更できます。

注意事項

TP1/Message Queue 独自の属性は引き継がれません。詳細については、「[2.8.13\(4\) クラスタチャネルの属性を変更](#)」を参照してください。

クラスタ転送キューにメッセージが登録され、そのメッセージを送信するクラスタセンダチャネルが存在しない場合は、クラスタセンダチャネルが自動定義され開始されます。自動定義されたチャネルは、設定された値の切断条件になるまで「チャネル動作中」状態が保持されます。

クラスタチャネルの状態は `mqtlscha` コマンドによって確認できます。

(2) クラスタリングの動作

クラスタセンダチャネルの定義には、キューマネージャをフルリポジトリキューマネージャに紹介する効果があります。フルリポジトリキューマネージャはそれによってフルリポジトリ内の情報を更新します。フルリポジトリキューマネージャはそのキューマネージャに対してクラスタセンダチャネルを作成し、クラスタについてのキューマネージャ情報を送信します。こうして、キューマネージャはクラスタについての情報が得られ、クラスタはキューマネージャについての情報が得られます。

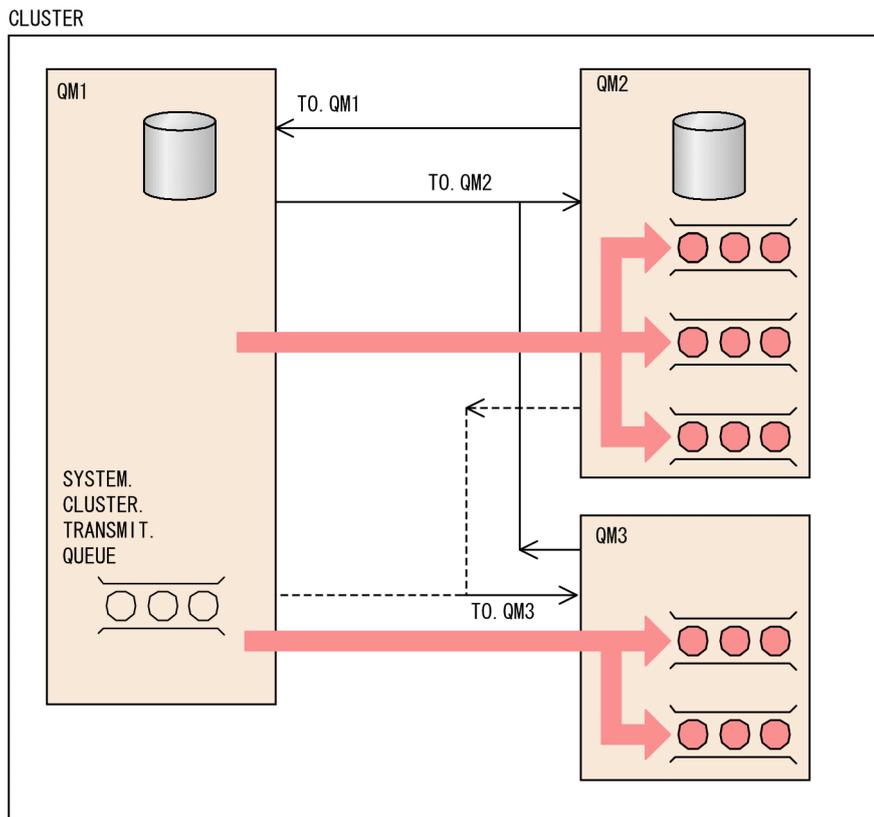
クラスタのシステム構成については、「[2.8.11\(1\) リポジトリ管理の構成](#)」を参照してください。キューマネージャ QM1 は QM2 にあるキューへメッセージを送信するとします。QM2 が QM1 に対してクラスタセンダチャネルを定義し、自分自身を紹介しているので、QM1 は QM2 でどのようなキューが利用できるかがわかっています。QM1 は QM2 に対してクラスタセンダチャネルを定義しているので、そのチャネルを使用してメッセージを送信できます。

QM3 は自分自身を QM2 に紹介します。QM1 もフルリポジトリを保持しているので、QM2 は QM3 についてのすべての情報を QM1 に渡しています。したがって、QM1 は QM3 ではどのキューが利用できるか、および QM3 が定義したクラスタレシーバチャネルがどれであることを認識します。QM1 が QM3

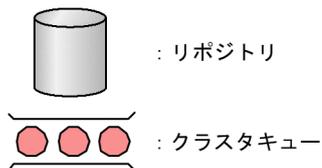
にあるキューへメッセージを送信したい場合には、QM3のクラスタレシーバチャンネルと接続するクラスタセンドチャンネルを自動的に作成します。

同一のクラスタおよび自動的に作成された二つのクラスタセンドチャンネルについて、次の図に示します。これらは、クラスタレシーバチャンネル TO.QM3 へと合流する 2 本の破線によって表されています。また、QM1 がメッセージ送信のために使用するクラスタ転送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE もあります。クラスタ内のすべてのキューマネージャはクラスタ転送キューを持っていて、クラスタ転送キューから同一クラスタ内のどのキューマネージャに対してもメッセージを送信できます。

図 2-45 自動定義のチャンネルのあるキューマネージャのクラスタ



(凡例)



クラスタセンドチャンネルの自動定義は、必要な時に自動的に定義されるので、クラスタの機能および効率にとってとても重要です。しかし、クラスタを機能させるために何を定義する必要があるかを示すため、この図では手作業で定義の対象になるチャンネル（またはチャンネルの受信側）だけを示します。

(3) クラスタに関連する別名の使用とリモートキューのローカル定義

別名にはキューマネージャの別名、応答キューの別名、および別名キューがあります。これらは分散キューイング環境だけでなくクラスタ環境にも適用されます。ここでは別名がクラスタとの関連でどのように使用されるかを説明します。

(a) キューマネージャの別名

キューマネージャの別名は、MQA サービス定義の `mqaremque` 定義コマンドの `-r` オプションの指定値を省略する場合に、コマンド引数であるリモートキューのローカル定義で指定され、次に示す用途があります。

■ メッセージ送信時のキューマネージャの再マッピング

MQOPEN 命令で指定したキューマネージャ名を別のキューマネージャに再マッピングするようキューマネージャの別名を使用できます。これは、クラスタキューマネージャでも可能です。例えば、キューマネージャは次に示すキューマネージャの別名を指定できます。

```
mqaremque -m CLUSQM YORK
```

これによって CLUSQM というキューマネージャに対する別名として使用できるキューマネージャ名として YORK が定義されます。この定義をしたキューマネージャ上のアプリケーションがキューマネージャ YORK に対してメッセージを送信すると、ローカルキューマネージャはそれを CLUSQM という名称に解決します。ローカルキューマネージャが CLUSQM という名称でない場合には、メッセージをクラスタ転送キューに登録し（そのあと CLUSQM へ転送される）、転送ヘッダを YORK ではなく CLUSQM に変更します。

■ 注意事項

クラスタ内のすべてのキューマネージャが YORK という名前を CLUSQM に解決するのではありません。定義が適用されるのはその定義をしたキューマネージャ上だけです。別名をクラスタ全体に通知するには、リモートキューのローカル定義の `-c` オプションで `ClusterName` 属性を指定してください。YORK へ向かうはずだったほかのメッセージキューからのメッセージは、別名の定義を持つキューマネージャへ送信され、別名が解決されます。

■ メッセージ送信時の転送キューの変更または指定

この別名を使用してクラスタを非クラスタシステムと結合できます。例えば、クラスタ ITALY 内のキューマネージャが PALERMO というキューマネージャ（クラスタの外部にある）と通信できるようにするには、クラスタ内のキューマネージャのうちの一つがゲートウェイとして機能する必要があります。このキューマネージャで次に示すとおり定義します。

```
mqaremque -c ITALY -m PALERMO -x X ROME
```

これはキューマネージャの別名の定義であり、ROME をキューマネージャとして定義し通知します。クラスタ ITALY 内のどのキューマネージャからのメッセージも ROME の名前で PALERMO のあて先に到着します。オープンハンドルにキューマネージャ名として ROME を設定してオープンされたキューに登録されたメッ

セージは、別名の定義元であるゲートウェイのキューマネージャに送信されます。メッセージがそこに届くと、転送キュー X に登録され、通常为非クラスタチャンネルによってキューマネージャ PALERMO へ転送されます。

この例では ROME という名前を選択していますが、これには特に意味がありません。リモートキューのローカル定義名と -m オプションの指定値は同じでもかまいません。

■ メッセージ受信時のあて先の決定

メッセージを受信すると、キューマネージャは転送ヘッダを参照し、あて先のキューとキューマネージャの名前を調べます。参照したキューマネージャと同じ名前のキューマネージャの別名の定義を持つ場合には、転送ヘッダ内のキューマネージャ名を対応するリモートキューマネージャ名で置き換えます。

キューマネージャ別名をこのように使用するには次に示す二つの理由があります。

- メッセージを別のキューマネージャへ向けて送信するため
- キューマネージャ名をローカルキューマネージャと同じ名前に変えるため

■ クラスタへのゲートウェイとしてのキューマネージャの使用

クラスタ外から送信されたメッセージに対する処理を負荷分散できます。

キュー EDINBURGH がクラスタ内の一つ以上のキューマネージャにあるとします。クラスタ外のキューマネージャには、転送キュー、およびクラスタ内のキューマネージャに対するセンダチャンネルが必要です。このクラスタ内のキューマネージャをゲートウェイキューマネージャといいます。キュー EDINBURGH をゲートウェイキューマネージャ以外に配置することによって、クラスタの負荷分散機能が有効になります。

詳細については、「(d) クラスタ内での別名の使用例」の「●クラスタ外のキューマネージャによるキューへのメッセージ登録 (別名使用)」を参照してください。

(b) 応答キューの別名

応答キューの別名の定義は、応答情報に代替名を指定する場合に使用します。応答キューの別名の定義は分散キューイング環境と同様にクラスタでも使用できます。例えば、次に示すような用途があります。

- キューマネージャ VENICE は、MQPUT 命令を使用して、メッセージ記述子に次に示すとおり指定して、メッセージをキューマネージャ PISA へ送信します。

```
ReplyToQ="QUEUE"  
ReplyToQMgr=""
```

- QUEUE に送信された応答が PISA にある OTHERQ で実際に受信されるように VENICE 上で応答キューの別名定義を作成します。

```
mqaremque -r OTHERQ -m PISA QUEUE
```

この定義によって応答キューの別名が作成されます。この別名は作成されたシステム上だけで有効です。

-m オプションの指定値がクラスタキューマネージャである場合でも、-m オプションとリモートキューのローカル定義名に同じ名前を指定してもかまいません。

(c) 別名キュー

別名キューの属性定義は、キューを識別する別名を作成するために使用します。例えば、アプリケーションを変更しないで別のキューを使用したい場合に使用します。または、メッセージが登録されるキューの本当の名前を何らかの理由でアプリケーションに知らせたくない場合や、キューの定義に使用されるものと異なる命名規則がある場合にも使用します。

キューマネージャ上の別名キューの属性定義を作成します。例えば、次に示す定義コマンドによって、PUBLIC というキューがクラスタ C 内のキューマネージャに対して通知されます。

```
mqaalsque -c C PUBLIC LOCAL
```

PUBLIC は実際は LOCAL という名前のキューとして解決される別名です。PUBLIC へ送信されたメッセージは LOCAL という名前のキューへ送信されます。

別名キューの属性定義を使用してクラスタキューとしてキュー名を解決することもできます。例えば、次に示す定義コマンドを使用すると、PUBLIC という名前でもクラスタ内の別のところで通知されているキューに対して、キューマネージャが PRIVATE という名前を使用してアクセスできます。

```
mqaalsque PRIVATE PUBLIC
```

この定義には ClusterName 属性は含まれないので、作成したキューマネージャでだけ適用されます。

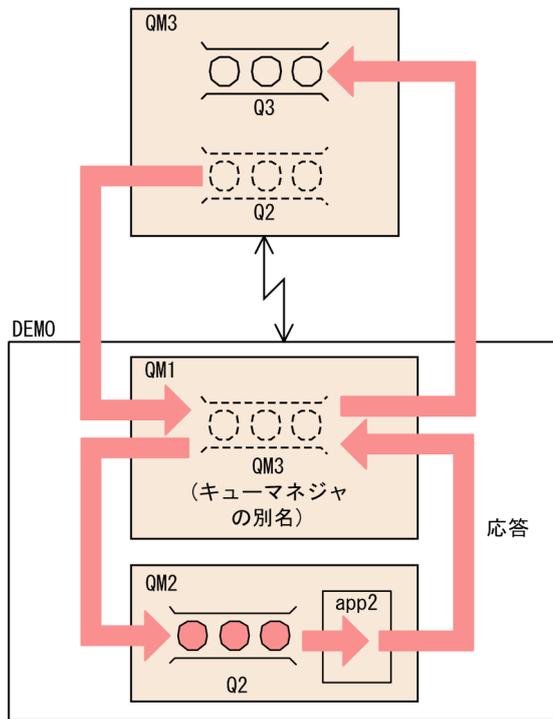
(d) クラスタ内での別名の使用例

DEMO というクラスタの外にある QM3 というキューマネージャについて、次の図に示します。QM3 はクラスタをサポートしないキューマネージャでもかまいません。QM3 は Q3 というキューを保持しています。

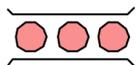
クラスタ内には QM1 と QM2 という二つのキューマネージャがあります。QM2 は Q2 というクラスタキューを保持しています。

クラスタ外のキューマネージャと通信するにはクラスタ内の一つ以上のキューマネージャがゲートウェイとして機能する必要があります。この例ではゲートウェイは QM1 です。

図 2-46 クラスタ外のキューマネージャからの登録



(凡例)

 : クラスタキュー

■ クラスタ外のキューマネージャによるキューへのメッセージ登録

クラスタ外のキューマネージャがクラスタ内の QM2 にあるキュー Q2 にどのようにメッセージを登録するかを説明します。

クラスタ外のキューマネージャ (QM3) は、メッセージを登録したいクラスタ内の各キューに対してリモートキューのローカル定義を持つ必要があります。例えば、次に示すとおり定義します。

```
mqaremque -r Q2 -m QM2 -x QM1 Q2
```

この図の QM3 上にこのリモートキュー Q2 が示されています。

QM3 はクラスタの一部ではないので、分散キューイングの方法で通信する必要があります。そのため、QM3 は QM1 に対応するセンダチャンネルと転送キューを持つことになります。QM1 には対応するレシーバチャンネルが必要です。ただし、この図ではチャンネルと転送キューは明示的には示していません。

QM3 にあるアプリケーションがメッセージを Q2 に登録するために MQPUT 命令を発行する場合、リモートキューのローカル定義によってメッセージはゲートウェイキューマネージャ QM1 を通って送信されます。

■ クラスタ外のキューマネージャへの応答

応答用の通信路を形成するため、ゲートウェイ (QM1) はクラスタ外のキューマネージャに対してキューマネージャの別名を通知します。QM1 はクラスタ属性をキューマネージャの別名の定義に付けることによって、

この別名をクラスタ全体に対して通知します (-r オプションが省略されていることに注意してください)。例えば、次に示すとおり定義します。

```
mqaremque -c DEMO -m QM3 -x X QM3
```

QM3 はクラスタの一部ではないので、分散キューイングの方法で通信する必要があります。したがって、QM1 は QM3 に対してセンダチャンネルと転送キュー X を持つ必要があります。QM3 には対応するレシーバチャンネルが必要です。この図ではチャンネルと転送キューは明示的には示していません。

QM2 にあるアプリケーション (app2) が QM3 にある Q3 へ応答を送信するために MQPUT 命令を発行すると、その応答はゲートウェイへ送信されます。ゲートウェイではキューマネージャの別名を使用して、あて先キューとキューマネージャの名前を決定します。

注意事項

クラスタ外のルートを二つ以上定義できます。

■ クラスタ外のキューマネージャによるキューへのメッセージ登録 (別名使用)

クラスタ外のキューマネージャからメッセージをキューに登録する別の方法があります。ゲートウェイキューマネージャ上で、例えば、ANY.CLUSTER というキューマネージャの別名を定義します。

```
mqaremque ANY.CLUSTER
```

これによって、キューマネージャ ANY.CLUSTER に対するどのような応答でも "null" とマッピングされます。つまり、クラスタ外のキューマネージャ内のリモートキューのローカル定義がキューマネージャ名 ANY.CLUSTER を使用できるので、正確なキューマネージャ名を使用する必要がなくなります。したがって、クラスタ外のキューマネージャ上では、次に示す定義ができます。

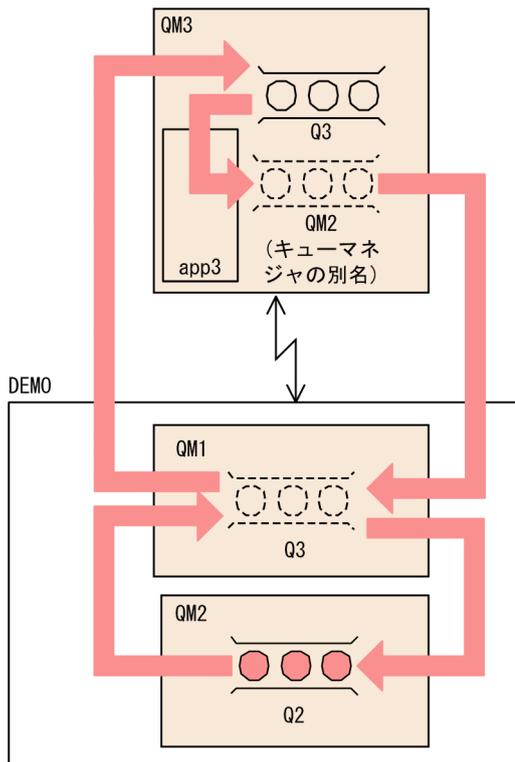
```
mqaremque -r Q2 -m ANY.CLUSTER -x QM1 Q2
```

この定義によってメッセージは最初に QM1 へ送信され、そこからクラスタキュー Q2 を保持するクラスタ内のキューマネージャへ転送されます。

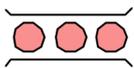
■ クラスタ外のキューマネージャへのメッセージ登録

クラスタ外へのメッセージの登録について次の図に示します。

図 2-47 クラスタ外のキューマネージャへの登録



(凡例)

 : クラスタキュー

ここでQM2 (クラスタ内) からメッセージをQM3にあるキューQ3 (クラスタ外) に登録する方法について検討します。

ゲートウェイ (この例ではQM1) はリモートキューのローカル定義を持ち、リモートキュー (Q3) をクラスタに対して次に示すとおり通知します。

```
mqaremque -r Q3 -c DEMO -m QM3 Q3
```

QM1のゲートウェイは、クラスタ外にあるキューマネージャに対するセンダチャンネルと転送キューも持っています。QM3には対応するレシーバチャンネルがありますが、この図ではこれらを示していません。

メッセージをキューに登録するため、QM2にあるアプリケーションは対象キュー名 (Q3) を指定し、さらに応答のあて先になるキューの名前 (Q2) を指定してMQPUT命令を発行します。メッセージはQM1に送信され、QM1はリモートキューのローカル定義を使用してQM3にあるQ3にキューの名称を解決します。

注意事項

クラスタ外のルートを二つ以上定義できます。

■ クラスタ外のキューマネージャからの応答

QM3 がクラスタ内部のキューマネージャに応答を送信するためには、通信したいクラスタ内の各キューマネージャに対してキューマネージャの別名を持つ必要があります。このキューマネージャの別名はメッセージが経由するゲートウェイの名前、つまりゲートウェイキューマネージャへの転送キューの名前を指定する必要があります。この例では、QM3 は QM2 に対して次に示すようなキューマネージャの別名の定義が必要です。

```
mqaremque -m QM2 -x QM1 QM2
```

QM3 も QM1 に対するセンダチャンネルと転送キューが必要であり、QM1 には対応するレシーバチャンネルが必要です。

QM3 にあるアプリケーション (app3) は MQPUT 命令を発行し、キュー名 (Q2) とキューマネージャ名 (QM2) を指定して応答を QM2 に送信できます。

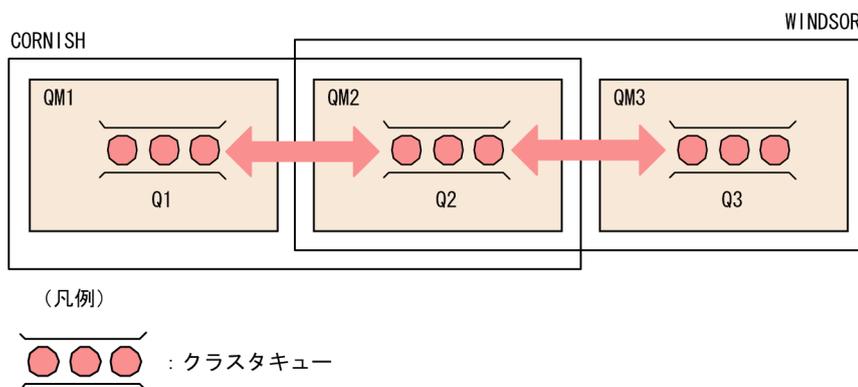
■ クラスタからクラスタへのメッセージ登録

すべてのキューマネージャを一つの大きなグループにまとめる代わりに、複数の小さなクラスタを構築できます。各クラスタでは一つ以上のキューマネージャを橋渡し役として動作させます。この方法では、利用できるキューおよびキューマネージャをクラスタ間で制限できます。この構成をマルチクラスタといいます。

別名を使用してキューおよびキューマネージャの名称を変えることによって、名称の衝突を回避したり、ローカルの命名規則に合わせたりすることができます。

クラスタ間の橋渡しとなるキューマネージャについて次の図に示します。

図 2-48 クラスタ間の橋渡し



QM1 から、別クラスタ (WINDSOR) に参加しているキューマネージャ (QM3) のクラスタキュー (Q3) へメッセージを登録する例を次に示します。

QM3 では Q3 を次に示すとおり作成します。

```
mqamkque -c WINDSOR Q3 モデルキューの定義名
```

QM2 は両方のクラスタに参加しており、両者の橋渡し役でもあります。QM2 を介して認識させる必要があるキューごとに、QM2 に別名キューの属性定義 (mqaalsque 定義コマンド) が必要です。この図の例では、QM2 で次に示す定義が必要です。Q3 の別名キュー名として、MYQ3 を指定します。

```
mqaalsque -w not_fixed -c CORNISH MYQ3 Q3
```

この定義によってクラスタ (CORNISH) 内のキューマネージャ (例えば QM1) で動作するアプリケーションは、MYQ3 という名前でも参照するキューにメッセージを登録できます。このメッセージはクラスタ (WINDSOR) 内のキューマネージャ (QM3) の Q3 に送信されます。

キューをオープンするときには、バインディングオプションに MQOO_BIND_NOT_FIXED または MQOO_BIND_AS_Q_DEF を指定する必要があります。MQOO_BIND_ON_OPEN を指定した場合、キューマネージャは別名キューの属性定義を橋渡し役のキューマネージャ (QM2) に解決し、QM2 はメッセージを転送しないからです。

この図は、橋渡し役として動作するキューマネージャが一つある二つのクラスタです。二つ以上のキューマネージャを橋渡し役として設定することもできます。

QM2 が動作するコンピュータでは、次に示すとおり定義します。

- MQT サーバ名 1 の定義

mqttcps 定義と複数の事前定義クラスタセンダチャンネルを定義します。

```
mqttcps -p tcp ...
mqtalcccha -c チャンネル名1 -y "type=cluscdr" -a"cluster=CORNISH" ...
mqtalced
mqtalcccha -c チャンネル名2 -y "type=cluscdr" -a"cluster=WINDSOR" ...
mqtalced
```

- MQT サーバ名 2 の定義

mqttcpcr 定義と一つの事前定義クラスタレシーバチャンネルを定義します。

```
mqttcpcr -p tcp ...
mqtalcccha -c チャンネル名3 -y "type=clusrcvr" -a"cluster=CORNISH" ...
mqtalced
```

- MQT サーバ名 3 の定義

mqttcpcr 定義と一つの事前定義クラスタレシーバチャンネルを定義します。

```
mqttcpcr -p tcp ...
mqtalcccha -c チャンネル名4 -y "type=clusrcvr" -a"cluster=WINDSOR" ...
mqtalced
```

- MQA サービス定義

MQT サーバ名 1, MQT サーバ名 2, および MQT サーバ名 3 を定義します。

```
mqamqtnam MQTサーバ名1 mqttcps 255
mqamqtnam MQTサーバ名2 mqttcpcr 100
mqamqtnam MQTサーバ名3 mqttcpcr 100
```

2.8.6 ワークロード管理へのクラスタの使用

複数の同名のクラスタキューをクラスタ内に設定することによって、システムの可用性を向上させ、メッセージの処理を負荷分散できます。ここでは、ワークロード管理を取り上げ、このようにクラスタを使用する意味を説明します。

(1) 複数キューマネージャにある同名クラスタキュー

複数のキューマネージャが、それぞれ同じ名前のキューを保持するクラスタを構築できます。これによって、キューマネージャが相互にクローンであるようにクラスタを構成できます。つまり、キューマネージャは、同じアプリケーションを実行でき、同じ名前のキューを持つということです。

このようにクラスタを使用する場合の利点は、次に示すとおりです。

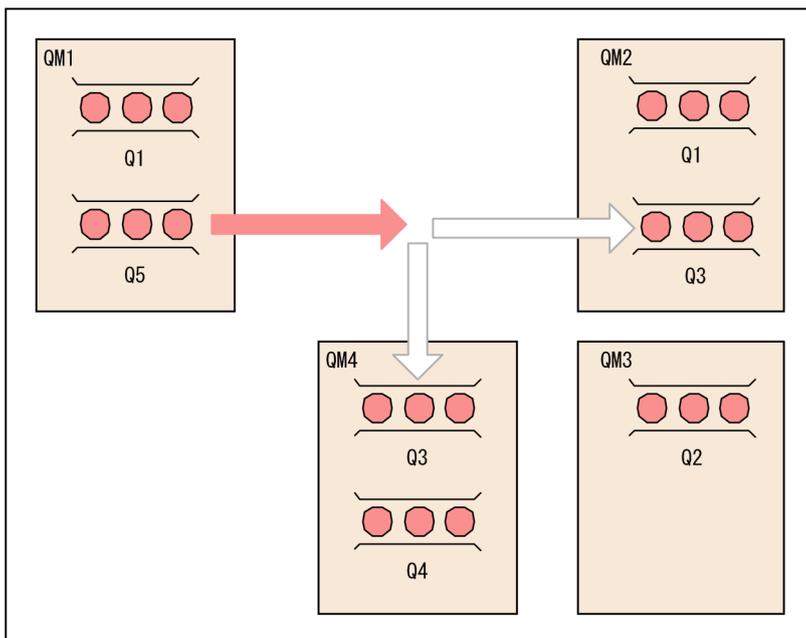
- キューとアプリケーションの可用性の向上
- メッセージのスループットの高速化
- ネットワークへの負荷分散の均等化

特定のキューあてのメッセージは、その名前のキューを保持するキューマネージャのどれかによって処理されます。つまり、アプリケーションは、メッセージを送信する時にキューマネージャを明示的に指定する必要はありません。ワークロード管理アルゴリズムが、メッセージを処理するキューマネージャを決めます。

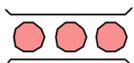
次の図は同名のキュー Q3 が複数あるクラスタを示しています。QM1 のアプリケーションが Q3 にメッセージを登録する場合、アプリケーションは、どのキューマネージャにある Q3 にメッセージが登録されるのか意識しません。しかし、QM2 のアプリケーションまたは QM4 のアプリケーションがメッセージを Q3 に登録する場合には、ローカルキューが使用されることに注意してください。

MQOO_BIND_NOT_FIXED オプションでクラスタ内のローカルキューがオープンされた場合、メッセージ転送中にキューが使用できなくなると、メッセージはほかのキューに転送されます。利用できるキューがない場合、メッセージはデッドレターキューに登録されます。クラスタ内のリモートキューや別名キューがメッセージの転送中に使用できなくなった場合、メッセージはデッドレターキューに登録されます。

図 2-49 複数の同名キューがあるクラスタ



(凡例)

 : クラスタキュー

クラスタ転送キューにメッセージがあるが、送信経路の通信障害などによってあて先キューマネージャに送信できない場合、システムはメッセージの送信経路を再設定します。しかし、メッセージを喪失する危険を冒したり、複製を作成したりすることによってメッセージの一貫性に影響を与えることはありません。キューマネージャが失敗し、メッセージが不確定なままになっている場合、そのメッセージに対して送信経路が再設定されることはありません。

メッセージの送信経路が再設定される状態については、「[2.8.7 経路の選択](#)」を参照してください。

注意事項

- 複数の同名キューがあるクラスタを設定する前に、自分のメッセージが相互依存性を持っていないこと、例えば、特定の順番で処理される必要があるとか、同じキューマネージャによって処理される必要があるとかいう条件について確認することが重要です。
- 同じ名前のキューについては、モデルキューの属性定義を同一にしておくことをお勧めします。そうしないと、異なる MQINQ 命令から異なる結果を得ることになります。

(2) 負荷分散

クラスタ内に複数の同名のクラスタキューがある場合、ワークロード管理アルゴリズムを使用して、メッセージのあて先になる最良のキューマネージャが決定されます。ワークロード管理アルゴリズムは、可能な場合、あて先としてローカルキューマネージャを選択します。ローカルキューマネージャに該当する名前のキューがない場合にアルゴリズムは、チャンネルに割り振った優先度を含むチャンネルの状態、およびキュー

マネージャやキューの利用可能性を基にして、どのあて先が適切か決定します。アルゴリズムはラウンドロビン方式で、最終的に適切なキューマネージャを選択します。

ローカルキューマネージャとリモートキューマネージャの両方、または一方に同名のクラスタキューがあり、そのクラスタキューに対してメッセージの登録を実行した時に解決されるキューについて、次の表に示します。

表 2-15 メッセージ登録時に解決されるキュー

クラスタキューの有無		メッセージ登録の有無	
ローカルキューマネージャ	リモートキューマネージャ	ローカルキューマネージャ	リモートキューマネージャ
○	×	○	×
○	○	○	×
○※1	○	×	○
○	○※1	○	×
○※2	○	書き込みエラー※3	
○※4	○	書き込みエラー※3	
○※5	○	×	○
○※5	○※5	書き込みエラー，または MQOPEN 命令時にエラー※3, ※6	
○※7	○	×	○
○	○※7	○	×
×	○	×	○
×	○※2	書き込みエラー※3	
×	○※4	書き込みエラー※3	
×	○※5	MQOPEN 命令時にエラー※3	

(凡例)

- ：該当します。
- ×：該当しません。

注※1

MQOPEN 命令発行後に作成されたキューです。

注※2

MQOPEN 命令発行後に書き込み禁止になったキューです。

注※3

ローカルキューマネージャで MQI がエラーになります。リモートキューマネージャにメッセージは送信されません。

注※4

MQOPEN 命令発行後に障害になったキューです。

注※5

MQOPEN 命令発行前に書き込み禁止になったキューです。

注※6

MQOO_BIND_ON_OPEN 指定時は書き込みエラーです。

MQOO_BIND_NOT_FIXED 指定時は MQOPEN 命令でエラーです。

注※7

MQOPEN 命令発行前に書き込み禁止，MQOPEN 命令発行後に書き込み可能になったキューです。

(3) プログラミングの検討事項

アプリケーションは，MQOPEN 命令を使用してクラスタキューをオープンし，MQPUT 命令を使用してクラスタキューにメッセージを登録します。または，MQPUT1 命令を使用して，まだオープンしていないクラスタキューに一つのメッセージを登録できます。

複数の同名のクラスタキューがないクラスタを設定する場合には，特にアプリケーションプログラミング上の検討事項はありません。しかし，クラスタのワークロード管理という利点を活用するには，アプリケーションの修正が必要なことがあります。同じ名前のキューが複数あるネットワークを構築する場合は，メッセージ類似性についてアプリケーションの見直しが必要です。

また，次に示す項目について注意してください。

- メッセージを取り出したり，属性を設定する目的でアプリケーションからキューをオープンしたりする場合は，MQOPEN 命令はローカルキューだけに有効です。
- メッセージを登録する目的でアプリケーションからキューをオープンする場合は，MQOPEN 命令はすべての利用できるクラスタキューから対象を選択します。ローカルキューが優先的に選択されます。そのためアプリケーションがクラスタの機能を十分に利用できないことがあります。

(a) メッセージ類似性についてのアプリケーションの見直し

同じキューの複数の定義を持ったクラスタを使用する前に，メッセージ類似性を持っていないか，つまり関連メッセージの交換要求がないか，自分のアプリケーションを調査する必要があります。クラスタでは，メッセージは，該当するキューを保持しているキューマネージャに転送されます。したがって，メッセージ類似性を持っているアプリケーションのロジックは無効になることがあります。

例えば，質疑応答の形での連続するメッセージフローに依存している二つのアプリケーションがあるとします。すべての質問を同じキューマネージャに送信し，すべての回答をほかのキューマネージャに送り返すことが重要です。また，ワークロード管理機能が，該当するキューを保持しているだけのキューマネージャにメッセージを送信しないことも重要です。

同様に，順番どおり取り出されるべきメッセージのバッチを送信するファイル転送アプリケーションやデータベース複製アプリケーションのように，メッセージが順番に処理されなければいけないアプリケーションがある場合もあります。

注意事項

セグメント化されたメッセージを使用しても、類似性の問題が発生することがあります。

(b) メッセージ類似性の処置

メッセージ類似性を持っているアプリケーションがある場合、できるだけクラスタを使用する前に、その類似性を取り除く必要があります。

メッセージ類似性を取り除くと、アプリケーションの可用性が改善されます。メッセージ類似性を持っているアプリケーションがメッセージのバッチをキューマネージャに送信し、バッチの一部を受け取ったあとでそのキューマネージャが失敗した場合、送信元のキューマネージャは、さらにメッセージを送信する前に送信先のキューマネージャの回復を待たなければいけません。

また、メッセージ類似性を取り除くと、アプリケーションのスケラビリティが改善されます。類似性を持っているメッセージのバッチによって、以後のメッセージを待っている間に、あて先のキューマネージャのリソースがロックすることがあります。このリソースが長期間ロックしたままであると、ほかのアプリケーションの業務を妨害することがあります。

さらに、メッセージ類似性によって、クラスタのワークロード管理機能は最良のキューマネージャを選択できなくなります。

類似性を取り除くには、次について検討します。

- メッセージ内に状態情報（シーケンス番号、または先頭と最終を表すデータフラグなど）を設定すること
- 状態情報をどのキューマネージャも利用できる不揮発性のストレージ（データベースなど）で保守すること

メッセージ類似性を取り除くために自分のアプリケーションを修正するのが適当でない場合、ほかにも多数の解決策があります。例えば、次に示すことを実行できます。

MQOPEN 命令で特定のあて先を指定する

各 MQOPEN 命令でリモートキューの名前とキューマネージャの名前を指定するのも、一つの解決策です。その場合、オブジェクトハンドルを使用してすべてのメッセージが同じキューマネージャに行くようにします。それが、ローカルキューマネージャのこともあります。

この方法には、次に示すような不都合があります。

- ワークロード管理が実行されません。これによって、ワークロード管理の利点を活用できなくなります。
- ターゲットキューマネージャがリモートであり、そこへの複数のチャンネルがある場合、メッセージは別のルートを通ることがあり、メッセージの連続性が保たれません。
- 自分のキューマネージャがあて先キューマネージャと同じ名前を持った転送キューの定義を持っている場合、メッセージはクラスタ転送キューではなくその転送キューに登録されます。

ReplyToQMGr フィールドにキューマネージャ名を返す

最初の解決策の変形として、最初のメッセージをバッチで受信するキューマネージャが応答でその名前を返すようにするという方法もあります。メッセージ記述子の ReplyToQMGr フィールドを使用して、これを実現します。送信側のキューマネージャは、このキューマネージャ名を取り出し、それを以後のすべてのメッセージで指定します。

前の方法と比べると、この方法は最初のメッセージを転送するために負荷分散が実行されるという利点があります。

しかし、この方法では、最初のキューマネージャは最初のメッセージに対する応答を待たなければいけなくて、以後のメッセージを送信する前に ReplyToQMGr の情報を見つけて使用しなければいけないというデメリットがあります。前の方法と同様に、キューマネージャへのルートが複数ある場合、メッセージの連続性が保たれないことがあります。

MQOPEN 命令で MQOO_BIND_ON_OPEN オプションを使用する

自分のすべてのメッセージを同じあて先に登録するというのも解決策として考えられます。そのためには、MQOPEN 命令で MQOO_BIND_ON_OPEN を使用します。キューをオープンし、MQOO_BIND_ON_OPEN を指定することによって、すべてのメッセージをこの同じキューに登録するようにします。MQOO_BIND_ON_OPEN は、すべてのメッセージを同じキューマネージャに、または同じルートにバインドします。したがって、例えば、同じあて先に複数のルートがある場合、キューがオープンされるときにどれかのルートが選択されると、取得したオブジェクトハンドルを使用して同じキューに登録されたすべてのメッセージは同じルートを選択します。

MQOO_BIND_ON_OPEN を指定することによって、すべてのメッセージが同じあて先に転送されるようにします。したがって、メッセージ類似性を持っているアプリケーションが壊されることはありません。あて先が利用できない場合、メッセージは、あて先が再度利用可能になるまで転送キューに残ります。

キューをオープンする時にキューマネージャ名がオブジェクト記述子で指定されている場合にも、MQOO_BIND_ON_OPEN が適用されます。名前付きキューマネージャへのルートが複数存在することがあります（例えば、複数のネットワークパスがある、またはほかのキューマネージャが別名を定義している）。MQOO_BIND_ON_OPEN を指定する場合、キューがオープンされる時にルートが選択されます。

注意事項

この方法をお勧めします。しかし、キューマネージャがクラスタキューの別名を通知しメッセージを中継する環境では使用できません。また、アプリケーションがメッセージグループごとに、同じキューマネージャの異なるキューを使用する場合にも使用できません。

MQOPEN 命令で MQOO_BIND_ON_OPEN を指定する代わりに、キューの定義を変更することによっても同じ効果を得られます。mqaqueatl 定義コマンドで -w オプションに fixed を指定して MQOPEN 命令での MQOO_BIND オプションが MQOO_BIND_AS_Q_DEF にデフォルト設定されるようにします。

(4) MQI とクラスタ

クラスタに関係する命令は次に示すとおりです。

- MQOPEN 命令
- MQPUT 命令と MQPUT1 命令
- MQINQ 命令
- MQSET 命令

各命令のクラスタに関連するオプションについて説明します。あわせて、マニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

ここでは、一般的なプログラミングインタフェース情報を含んでいます。

(a) MQOPEN 命令

MQOPEN 命令のオプションである MQOO_BIND_ON_OPEN オプションによって、クラスタ内に複数の同名のキューがある場合に、ターゲットキューマネージャを固定するよう指定できます。つまり、MQOPEN 命令から返されるオブジェクトハンドルを指定するキューに登録されるすべてのメッセージは、同じキューマネージャに向けられることとなります。

類似性を持ったメッセージがある場合に、このオプションを使用できます。例えば、メッセージのバッチをすべて同じキューマネージャによって処理する必要がある場合、キューをオープンする時に MQOO_BIND_ON_OPEN を指定します。これによって、キューマネージャを固定し、そのキューに登録されたすべてのメッセージがそのルートを採用するようになります。

必ずしもすべてのメッセージを同じあて先に登録したくない場合、MQOPEN 命令で MQOO_BIND_NOT_FIXED を指定します。これによって、MQPUT 命令発行時にあて先が選択されます。つまり、あて先はメッセージごとに選択されます。

注意事項

MQOO_BIND_NOT_FIXED と MQMF_SEGMENTATION_ALLOWED を同時に指定しないでください。指定した場合、メッセージのセグメントが異なるキューマネージャに転送されることがあります。

MQOO_BIND_ON_OPEN も MQOO_BIND_NOT_FIXED も指定しない場合、デフォルトオプションは MQOO_BIND_AS_Q_DEF です。MQOO_BIND_AS_Q_DEF を使用すると、キューハンドルに使用するバインドは、DefBind 属性から取得されます。

MQOPEN 命令のオブジェクト記述子で名前を指定することによっても、あて先キューマネージャを選択できます。このようにすれば、ローカルキューマネージャを含め、どのようなキューマネージャでも選択できます。

MQOPEN 命令で MQOO_BROWSE, MQOO_INPUT_*, または MQOO_SET の一つ以上を指定した場合、オープンが成功するためには、ローカルのクラスタキューである必要があります。MQOPEN 命令

で MQOO_OUTPUT, MQOO_BIND_*, または MQOO_INQUIRE の一つ以上を指定し, MQOO_BROWSE, MQOO_INPUT_*, または MQOO_SET (常にローカルのクラスタキューが選択されるようにする) のどれも指定しなかった場合, オープンされるクラスタキューは次に示すどちらかです。

- 存在する場合, ローカルキューマネージャにあるクラスタキュー
- ローカルキューマネージャにない場合, クラスタのどこかにあるクラスタキュー

■ 解決されたキューマネージャ名

キューマネージャ名が MQOPEN 命令発行時に解決される場合, その解決された名前がアプリケーションに返されます。

(b) MQPUT 命令と MQPUT1 命令

MQOO_BIND_NOT_FIXED が MQOPEN 命令で指定されている場合, 以後の MQPUT 命令は, ワークロード管理機能を起動して, メッセージを送信するキューマネージャを決定します。したがって, そこで指定されるあて先とルートはメッセージごとに選択されます。ネットワークの条件が変わった場合, メッセージが登録されたあとに, あて先とルートが変わることがあります。MQPUT1 命令は, 常に MQOO_BIND_NOT_FIXED が有効であるかのように動作します。つまり, 常にワークロード管理機能を起動します。

ワークロード管理機能がキューマネージャを選択した場合に, ローカルキューマネージャの登録オペレーションは完了します。ターゲットキューマネージャがローカルキューマネージャと同じクラスタのメンバである場合, ローカルキューマネージャは, メッセージをクラスタ転送キューに登録します。ターゲットキューマネージャがクラスタの外にあり, ローカルキューマネージャがターゲットキューマネージャと同じ名前の転送キューを持っている場合, メッセージはその転送キューに登録されます。

MQOO_BIND_ON_OPEN が MQOPEN 命令で指定されている場合, あて先とルートはすでに選択されているので, MQPUT 命令は, ワークロード管理機能を起動する必要がありません。

(c) MQINQ 命令

キューを照会する前に, MQOPEN 命令を使用して, MQOO_INQUIRE を指定してキューをオープンする必要があります。

複数の同名のクラスタキューがあるクラスタを持っている場合, 照会できる属性はクラスタキューがローカルにあるかどうか, キューがどのようにオープンされたのかによって異なります。

MQOPEN 命令で MQOO_INQUIRE を指定しているほかに, MQOO_BROWSE, MQOO_INPUT_*, または MQOO_SET オプションのどれかも指定している場合に, オープンが成功するには, ローカルのクラスタキューが必要です。この場合, ローカルキューに有効なすべての属性を照会できます。

MQOPEN 命令で, MQOO_INQUIRE だけ, または MQOO_INQUIRE と MQOO_OUTPUT を指定している場合 (しかし, 常にローカルのクラスタキューが選択されるようにする MQOO_BROWSE, MQOO_INPUT_*, または MQOO_SET のどれも指定していない), オープンされるクラスタキューは, 次に示すどちらかです。

- 存在する場合には、ローカルキューマネージャにあるクラスタキュー
- ローカルキューマネージャにない場合、クラスタのどこかにあるクラスタキュー

オープンされたキューがローカルキューでない場合、次に示す属性だけを照会できます。この場合、QType 属性は MQQT_CLUSTER の値を持っています。

- DefBind
- DefPersistence
- DefPriority
- InhibitPut
- QDesc
- QName
- QType

クラスタキューの DefBind 属性を照会するには、MQIA_DEF_BIND セレクタの MQINQ 命令を使用します。返される値は、MQBND_BIND_ON_OPEN または MQBND_BIND_NOT_FIXED のどちらかです。ローカルキューの ClusterName 属性を照会するには、MQCA_CLUSTER_NAME で MQINQ 命令を使用します。

注意事項

固定バインドにしないで (つまり MQOPEN 命令で MQOO_BIND_NOT_FIXED を指定する、またはキューの DefBind 属性が MQBND_BIND_NOT_FIXED の場合に MQOO_BIND_AS_Q_DEF を指定する)、クラスタキューをオープンした場合、以後の MQINQ 命令は、他キューマネージャのクラスタキューを照会することがあります。

(d) MQSET 命令

属性を設定するためにクラスタキューをオープンする場合 (MQOO_SET オプションを指定)、オープンが成功するためにはローカルのクラスタキューが必要です。したがって、MQSET 命令を使用してはクラスタのどこかほかにあるキューの属性を設定できません。しかし、クラスタ属性で定義されている別名キューまたはリモートキューをオープンする場合には、ターゲットキューまたはリモートキューが解決されてクラスタキューになっても、MQSET 命令を使用して別名キューまたはリモートキューの属性を設定できます。

(5) ワークロード管理アルゴリズム

ワークロード管理アルゴリズムを次に示します。

1. MQOPEN 命令でキューマネージャ名が指定されなかった場合、次のようになります。
 - 登録禁止状態のキューを選択対象から除外します。

- キューの属するクラスタがローカルキューマネージャの属するクラスタでないとき、キューを選択対象から除外します。
 - キューの経路となるリモートクラスタレシーバチャンネルがキューと同じクラスタに属していないとき、そのチャンネルを除外します。
2. MQOPEN 命令でキューマネージャ名称が指定された場合、次のようになります。
- 登録禁止状態であるキューマネージャ別名を選択対象から除外します。
 - リモートクラスタレシーバチャンネルの属するクラスタがローカルキューマネージャの属するクラスタでないとき、選択対象から除外します。
3. 1.の結果ローカルキューが選択対象として残っているとき、または 2.の結果ローカルキューマネージャ別名が選択対象として残っているとき、そのローカルキューマネージャ上のオブジェクトを選択します。残っていないとき、4.以降の処理をします。
4. リモートキューマネージャ上のオブジェクトだけ残っている場合、RESUME 状態のキューマネージャを、SUSPEND 状態のキューマネージャより優先して選択します。
5. 2つ以上のリモートキューマネージャ上のオブジェクトが残っている場合、次に示すクラスタセンダチャンネルの状態の順序でチャンネルを選択します。
- 「チャンネル停止」または「チャンネル動作中」
 - 「チャンネル確立リトライ中」
 - 「チャンネル使用不可」
6. 同一状態である 2つ以上のチャンネルから選択します。
- 同一のリモートキューマネージャをあて先とする 2つ以上のクラスタセンダチャンネルが存在する場合、その中で最も高い NetworkPriority 属性を持つチャンネル以外のチャンネルを除外します。
 - 上記で 2つ以上のリモートキューマネージャ上のオブジェクトが残っている場合、最後に使われた時刻がキューマネージャ単位で最も古いクラスタセンダチャンネルを選択します。

2.8.7 経路の選択

クラスタ内の複数のキューマネージャに同じ名前のクラスタキューがある場合に、メッセージを登録すると、ワークロード管理がチャンネルの状態などを考慮して、最適な経路を選択します。

名称解決によって、キューマネージャにメッセージが転送されたあと、次に示す状態が発生した場合は、該当するチャンネルに割り当てられたメッセージの送信経路が再設定され、最適なチャンネルでメッセージが送信されます。ただし、送達未確認状態のメッセージは経路選択の対象外になります。

- クラスタセンダチャンネルのチャンネル状態が「チャンネル確立リトライ中」または「チャンネル使用不可」になった場合
- 自動定義クラスタセンダチャンネルがクラスタレシーバの削除などによって削除された場合
自動定義クラスタセンダチャンネルが削除される条件を次に示します。

- 対象のクラスタレシーバチャンネルの情報の期限が切れた場合
対象のクラスタレシーバチャンネルを持つキューマネージャが最後にフルリポジトリキューマネージャにチャンネル情報を送信してから、チャンネル情報が保管される期限を超えた場合です。
- 対象のクラスタレシーバチャンネルが削除された場合
- 対象のチャンネルを持つキューマネージャ名を指定した RESET CLUSTER コマンドが実行された場合
- mqrremove -s コマンド、mqrrefresh -f コマンドまたは mqrrefresh -m コマンドを実行した場合
コマンド実行時に自動定義クラスタセンダチャンネルが削除され、再度作成されます。

これらの状態ではチャンネルに対する一部のコマンドの入力が抑止されることがあります。各コマンドの注意事項については、「6. システムの運用」を参照してください。

- クラスタセンダデーモンが使用できる MCA 数（チャンネル数）を超えたチャンネルを使用しようとした場合（クラスタセンダデーモンの MQT デーモン構成定義（mqamqtnam 定義コマンド）で指定する MCA 数が、同一クラスタ内にある自システム以外のクラスタレシーバチャンネルの総数よりも少ない場合）

チャンネル状態が「チャンネル使用不可」になった場合、および自動定義クラスタセンダチャンネルが削除された場合は KFCA16350-I および KFCA16351-I を出力し、その開始および終了の契機をログメッセージに出力します。しかし、チャンネル状態が「チャンネル確立リトライ中」になった場合は、リトライ処理が複数回実行され再設定処理も複数回実行されることからログメッセージを圧迫する可能性があり、ログメッセージを出力しません。また、クラスタセンダデーモンが使用できる MCA 数（チャンネル数）を超えたチャンネルを使用しようとした場合もログメッセージを出力しないので注意してください。

2.8.8 クラスタで使用する定義コマンド

多くのコマンドには、クラスタの操作に特に関係する属性があります。これらの属性について、コマンドの種類に応じて次に示すグループに分けて説明します。

- チャンネル定義コマンド
- キュー定義コマンド

コマンドでは、クラスタ名は ClusterName 属性を使用し、長さ 48 文字以下で指定します。参加できるクラスタは一つです。同一のキューマネージャ内で指定するクラスタ名は、すべて同じにする必要があります。

(1) チャンネル定義コマンド

TCP 定義の mqtalcccha 定義コマンドには、クラスタに適用できるパラメタと属性があります。クラスタに特別なチャンネルタイプとして clusrcvr と clussdr があります。クラスタレシーバチャンネルを定義するには、mqtalcccha 定義コマンドの -y type オペランドに clusrcvr を指定します。クラスタレシーバチャンネル定義に必要なそのほかの属性の多くは、レシーバチャンネルまたはセンダチャンネルの定義に適用されるものと同じです。クラスタセンダチャンネルを定義するには、mqtalcccha 定義コマンドの -y type オペランドに clussdr を指定し、さらに、センダチャンネルの定義に使用するものと同じ各種の属性を指定します。

TCP 定義の mqtalcccha 定義コマンドおよび mqttcpcr 定義コマンドに指定する、クラスタに関連する属性は次に示すとおりです。

- ClusterName (mqtalcccha -a cluster オペランド指定値)
- NetworkPriority (mqtalcccha -p netprty オペランド指定値)
- OwnIPAddr または OwnHostName (mqtalcccha -r オプション指定値)
- OwnPortNum または OwnServName (mqttcpcr -r オプション指定値)

ClusterName 属性は、クラスタセンダチャンネルまたはクラスタレシーバチャンネルを指定する場合だけ適用できます。NetworkPriority 属性は、クラスタレシーバチャンネルだけに適用できます。ClusterName 属性は、チャンネルに関連づけるクラスタの名前を指定するために使用します。NetworkPriority 属性は、チャンネルの優先順位を指定するために使用します。これは、ワークロード管理機能を補助するためのものです。あて先に達するのに複数のルートを選択できる場合、ワークロード管理機能は、優先順位が最も高いルートを選択します。

クラスタレシーバチャンネルに指定する OwnIPAddr 属性または OwnHostName 属性は、キューマネージャのネットワークアドレスを識別するためにクラスタ全体で使用されます。これらの値は、対応するクラスタセンダチャンネルの PartnerIPAddr 属性または PartnerHostName 属性に使用されるため、注意して指定してください。

クラスタレシーバプロセスに指定する OwnPortNum 属性または OwnServName 属性は、キューマネージャのポート番号を識別するためにクラスタ全体で使用されます。これらの値は、対応するクラスタセンダチャンネルの PartnerPortNum 属性または PartnerServName 属性に使用されるため、注意して指定してください。

注意事項

- OwnIPAddr 属性または OwnHostName 属性、および OwnPortNum 属性または OwnServName 属性は、クラスタセンダチャンネル側から認識できる値にしてください。不正な値が指定された場合、TCP/IP 障害が発生し、チャンネルを確立できません。
- IBM MQ ではクラスタレシーバチャンネル定義 (DEFINE CHANNEL) の CONNAME パラメータに指定するポート番号を省略できます。しかし、TP1/Message Queue 側が対応できませんので、必ず CONNAME パラメータにポート番号を指定してください。

これらの属性は、mqtdmped コマンドで確認できます。

クラスタセンダチャンネルまたはクラスタレシーバチャンネルの状態を表示するために mqtlscha コマンドを使用できます。このコマンドでは、手作業で作成された事前定義チャンネルだけでなく、MQT サーバが認識するすべてのチャンネルの状態を表示し、自動定義チャンネルの状態も表示できます。しかし、リポジトリで管理するすべてのチャンネルの状況が取得できるものではありません。

(2) キュー定義コマンド

モデルキューの属性定義 (MQA サービス定義の `mqaqueatl` 定義コマンド)、リモートキューのローカル定義 (MQA サービス定義の `mqaqueue` 定義コマンド)、および別名キューの属性定義 (MQA サービス定義の `mqaalsque` 定義コマンド) の各定義コマンドには、クラスタに関連する属性があります。

注意事項

MQA サービス定義の `mqaqueatl` 定義コマンドの `-u` オプションで、`transmission` を指定した場合、キューは転送用に使用されるので、クラスタに参加するキューは作成できません。

- `ClusterName` 属性は、キューが所属するクラスタの名前を指定するために使用します。
- `DefBind` 属性は、アプリケーションで、`MQOPEN` 命令に `MQOO_BIND_AS_Q_DEF` が指定されている場合に、バインディングを使用することを指定するために使用します。この属性のデフォルトは `fixed` です。これは、キューをオープンする時にキューハンドルを同名のクラスタキューのうち、特定のキューにバインドすることを指定します。これの代わりに、`not_fixed` を使用して、キューハンドルを特定のキューにバインドしないよう指定することもできます。MQA サービス定義の `mqaqueatl` 定義コマンドの `-w` オプションに `fixed` または `not_fixed` を指定すると、キューは `MQBND_BIND_ON_OPEN` または `MQBND_BIND_NOT_FIXED` の属性で定義されます。クラスタ内のすべての同名クラスタキューについて、`DefBind` 属性を同じ値に設定することをお勧めします。

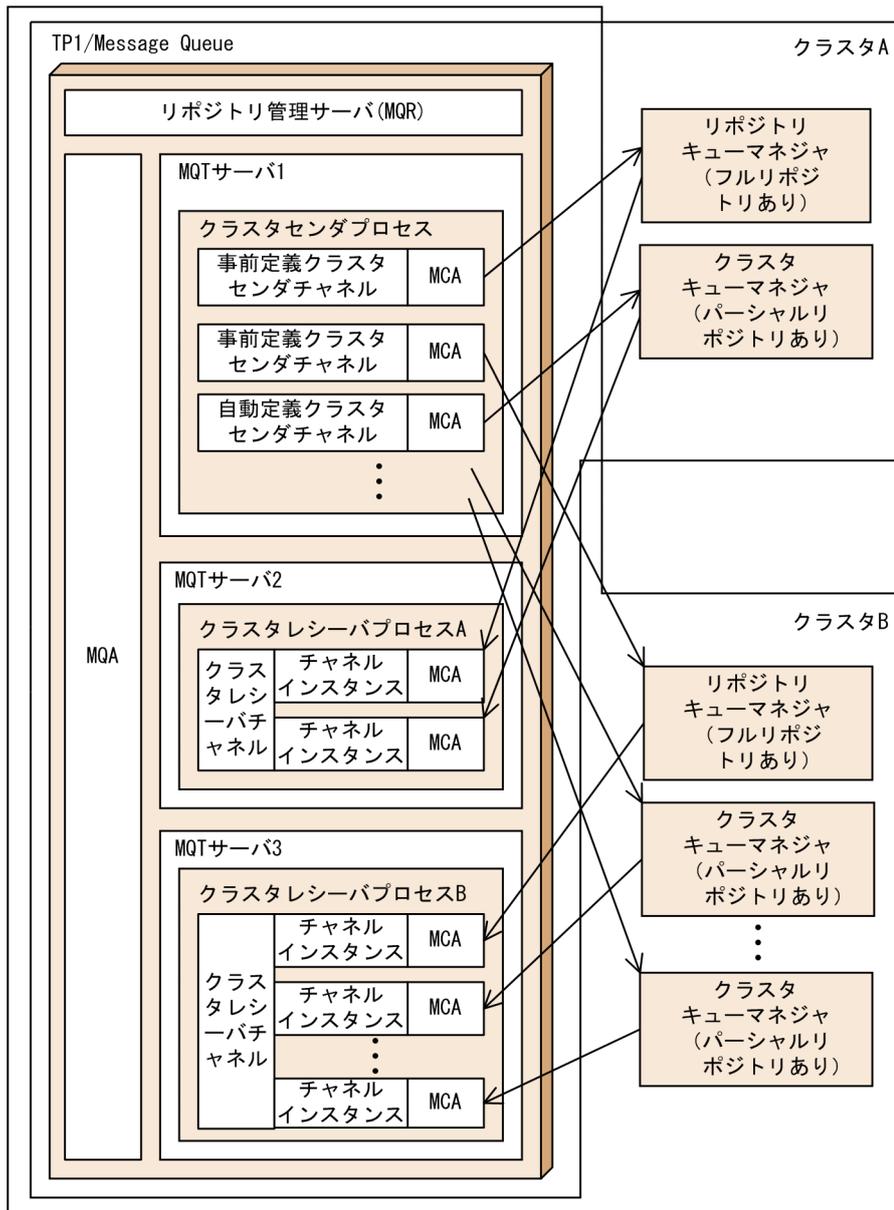
これらの属性は、`mqainq` コマンドの出力にも適用されます。`mqainq` コマンドにはキューマネージャ名を指定できます。

2.8.9 クラスタ環境の通信構成

クラスタ機能を使用する場合、ユーザは MQT プロセス構成に応じてシステム定義を作成します。

クラスタ環境での MQT プロセス構成について、次の図に示します。

図 2-50 クラスタ環境での MQT プロセス構成



- クラスタセンダプロセス

クラスタセンダチャンネルが動作するプロセスです。TP1/Message Queue システム内に一つ必要です。TCP 定義の mqttcps 定義コマンドで定義します。

注意事項

mqttcpsu で始まるシステムサービス定義名または mqttcps をプロトコル名称に指定する、mqamqtnam 定義コマンドは MQA サービス定義に一つだけ指定してください。複数指定した場合、OpenTP1 は開始できません。

- クラスタセンダチャンネル

フルリポジトリキューマネージャにクラスタのメンバである自システムの情報を送信します。また、他キューマネージャへ送信するメッセージの通信路になるチャンネルです。

事前定義クラスタセンダチャンネルは、ユーザが事前に定義します。参加するクラスタごとに一つ以上をチャンネル定義に指定（TCP 定義の `mqtalccha` 定義コマンドの `-y type` オペランドに `clussdr` を指定）します。この場合、`-o` オプションにはフルリポジトリキューマネージャについて設定してください。フルリポジトリがない場合、リポジトリ情報を取得できないため、クラスタに参加できません。参加するクラスタ内に複数のフルリポジトリキューマネージャがある場合は、最低一つの事前定義クラスタセンダチャンネル定義があれば十分です（必要に応じて自動定義クラスタセンダチャンネルをシステムが自動的に作成します）。どのフルリポジトリキューマネージャに接続してもかまいません。

事前定義クラスタセンダチャンネルのチャンネル属性は、通常は、接続先キューマネージャのクラスタレシーバで指定されたチャンネル属性で変更されます。ただし、TP1/Message Queue のクラスタセンダチャンネルに固有の情報は、クラスタレシーバチャンネルから入手できないため、TCP 定義の `mqtalccha` 定義コマンドの指定値を使用します。リポジトリ情報として TP1/Message Queue が保持している期間は、変更されたチャンネル属性が有効です。

このほかに、自動定義クラスタセンダチャンネルがあります。詳細については、「[2.8.9\(1\) クラスタセンダチャンネルの自動定義](#)」を参照してください。

- クラスタレシーバプロセス

クラスタレシーバチャンネルが動作するプロセスです。TP1/Message Queue システム内に一つ以上定義できます。

`mqttcp` 定義コマンドで定義します。

- クラスタレシーバチャンネル

フルリポジトリキューマネージャからクラスタにあるメンバの状態を受信します。また、他キューマネージャから受信するメッセージの通信路になるチャンネルです。

クラスタレシーバは、クラスタレシーバプロセスごとに一つだけ事前に定義できます。

チャンネル定義に指定（TCP 定義の `mqtalccha` 定義コマンドの `-y type` オプションに `clusrcvr` を指定）します。

クラスタレシーバプロセスを複数動作させることによってクラスタレシーバチャンネルを複数定義することもできます。クラスタレシーバチャンネルはチャンネルインスタンスという接続情報（メッセージシーケンス番号やチャンネル状態）を複数保持できるため、一つのチャンネル定義で複数のクラスタセンダチャンネルからの確立要求を受け入れることができます。

クラスタセンダプロセスおよびクラスタレシーバプロセスは、開始時にリポジトリ管理サーバの開始を待ち合わせます。このため、リポジトリ管理サーバを開始しないかぎり、チャンネルは動作できません。また、MQT のコマンドの多くは使用できません。該当するコマンドについては、6 章の「[TP1/Message Queue の運用コマンド](#)」を参照してください。

オンライン中にリポジトリ管理サーバが終了すると、終了中には新しいクラスタセンダチャンネルの開始はできません。リポジトリ管理サーバが再開始されると、クラスタセンダを開始します。

クラスタセンダプロセスが開始処理を完了したあとにリポジトリ管理サーバが終了した場合にも、チャンネルの開始ができなくなったり、送信経路の再設定処理が完了したりしないことがあります。クラスタ構成でチャンネルを動作させる場合は、リポジトリ管理サーバを必ず動作させてください。

(1) クラスタセンダチャンネルの自動定義

一つのクラスタセンダプロセスで自動定義クラスタセンダチャンネルは、MQA サービス定義の mqamqtnam 定義コマンドで MCA 数に指定した数まで自動的に定義されます。自動定義クラスタセンダチャンネルのチャンネル属性は、通常は接続先キューマネージャのクラスタレシーバチャンネルで指定されたチャンネル属性で動作します。ただし、TP1/Message Queue のクラスタセンダチャンネルに固有の属性は、リポジトリ情報として管理されないため、次に示す属性値を使用します。

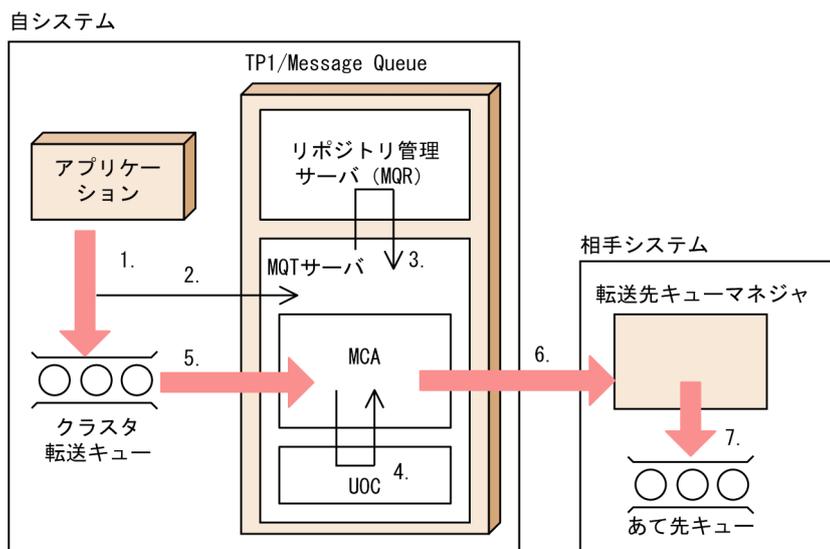
- デフォルトチャンネル定義 (mqtalcccha 定義コマンド) で指定があればその値
- 指定がなければ TP1/Message Queue 独自の属性のデフォルト値

接続先キューマネージャのクラスタレシーバチャンネルで指定されたチャンネル属性で動作する属性、および TP1/Message Queue 独自の属性のデフォルト値については「3.3.2(3) チャンネルデータ定義ブロック (dcmtcq_uoc_mqcd)」, および 4 章の「mqtalcccha (チャンネル定義)」を参照してください。

この値はメッセージ編集出口 UOC によって変更できます。変更方法については、「2.8.9(3) クラスタチャンネルの属性変更」を参照してください。

クラスタセンダチャンネルの自動定義の概要について、次の図に示します。

図 2-51 クラスタセンダチャンネルの自動定義の概要



1. アプリケーションがあて先キュー名を指定して、メッセージを登録します。
2. MQT サーバはメッセージの登録連絡を受けてチャンネルを開始します。
3. 接続するチャンネルがメッセージの転送元キューマネージャにない場合、MQT サーバは転送先キューマネージャのクラスタレシーバチャンネルのチャンネル属性を問い合わせ、それを基に、クラスタセンダチャンネルを自動作成して開始します。
4. メッセージ編集出口 UOC が登録されている場合、MCA はチャンネルを確立する前に UOC を呼び出します。メッセージ編集出口 UOC は、自動作成されたクラスタセンダチャンネルのチャンネル属性を変更できます。

5. 自動作成されたクラスタセンダチャンネルは、クラスタ転送キューからメッセージを取り出します。
6. クラスタセンダチャンネルはメッセージを転送します。
7. クラスタレシーバチャンネルは、受信したメッセージをあて先キューに登録します。

(2) クラスタレシーバチャンネルのマルチインスタンス

同じクラスタ内のすべてのクラスタセンダチャンネルからのメッセージ受信をクラスタレシーバチャンネルが実行します。クラスタレシーバチャンネルは、相手キューマネージャごとにチャンネル接続情報を保持できます。この機能をクラスタレシーバチャンネルのマルチインスタンス機能といいます。マルチインスタンス機能では、一つのクラスタレシーバチャンネルのチャンネル定義について、複数の接続情報（チャンネルインスタンス）を保持します。

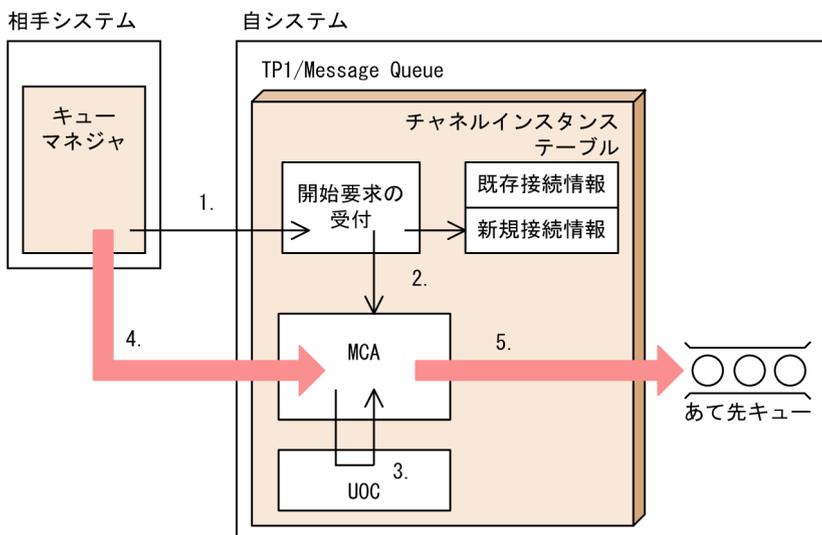
クラスタレシーバチャンネルのチャンネルインスタンスは、MQA サービス定義の mqamqtnam 定義コマンドで MCA 数に指定した数まで作成できます。

クラスタレシーバチャンネルのチャンネル属性はリポジトリ情報として相手キューマネージャに連絡され、相手キューマネージャのクラスタセンダチャンネルのチャンネル属性として使用されます。また、クラスタレシーバチャンネルの属性値をチャンネル開始時にメッセージ編集出口 UOC によってチャンネルインスタンス単位にユーザの任意の値に変更できます。

クラスタレシーバチャンネルの属性変更方法については、「2.8.9(3) クラスタチャンネルの属性変更」を参照してください。

クラスタレシーバチャンネルのマルチインスタンスの概要について、次の図に示します。

図 2-52 クラスタレシーバチャンネルのマルチインスタンスの概要



1. クラスタセンダチャンネルが、クラスタレシーバプロセスのポート (mqttcpcr 定義コマンド-r オプション指定値) へチャンネルの開始要求を送信します。
2. チャンネル接続情報を登録します。

相手キューマネージャ名に該当するインスタンスを検索し、すでに登録されている場合は該当するインスタンスの MCA を起動します。

インスタンス情報がない場合、新規にエントリを確保して MCA を起動します。

3. メッセージ編集出口 UOC が登録されている場合、MCA は UOC を呼び出します。

メッセージ編集出口 UOC は、クラスタレシーバチャンネルの属性値を必要に応じて変更できます。

MCA は UOC によって更新されたチャンネル属性でチャンネルを確立します。

4. MCA は相手システムのクラスタセンダチャンネルからメッセージを受信します。

5. MCA は、クラスタセンダチャンネルから受信したメッセージを、あて先キューに登録します。

(3) クラスタチャンネルの属性変更

クラスタチャンネルの属性値は、メッセージ編集出口 UOC を使用することでユーザの任意の値に変更できます。メッセージ編集出口 UOC で変更した属性値は、チャンネルが確立している間で有効です。再接続する時に、値はリセットされ UOC が変更する前の状態になります。このため、UOC で再度値を変更してください。

メッセージ編集出口 UOC とその用法については、「3. ユーザOWNコーディング」を参照してください。

クラスタ環境では、相手クラスタレシーバに指定したチャンネル属性をクラスタセンダチャンネルが使用して動作します。このため TP1/Message Queue 以外の MQ システムで設定したチャンネル属性を TP1/Message Queue のクラスタセンダチャンネルが使用し、チャンネル属性のサポート範囲の違いによって不正な動作をすることがあります。

TP1/Message Queue 以外の MQ システムで、クラスタレシーバチャンネルのチャンネル属性を設定する場合の注意事項について、次に示します。

- クラスタレシーバチャンネルのためのリスナ用ポート番号には 1024~65535 の値を使用してください。
- MQ システムが動作するコンピュータのホスト名は英字で始まる英数字を使用してください。
- 他 MQ システムのクラスタレシーバでサポートするチャンネル属性の上限値や下限値が TP1/Message Queue のサポート範囲と一致しない場合、TP1/Message Queue で動作ができる限界値を使用して動作します。

(4) クラスタチャンネルの削除

特定のチャンネルを使用するメッセージ送信が長期間にわたって発生しないで、キューマネージャ内でリポジトリ情報の要求が保持される期間を超えた場合、次に示すチャンネルは TP1/Message Queue によって自動的に削除されます。

- 自動定義クラスタセンダチャンネル
- 対応するクラスタレシーバチャンネルのチャンネルインスタンス

これらのチャンネルの有無については、`mqtlscha -n "*"` コマンドを入力して確認できます。定義種別として AUTOINST または MULTINST が出力されます。また相手システムのキューマネージャ名が出力されます。

例えば、あて先キューの名前を間違えて指定したために不要なチャンネルが作成された場合は、期間を超えてメッセージを送信しないことでチャンネルを削除できます。リポジトリ情報の要求が保持される期間については、「2.8.10(2)(g) リポジトリ情報の保持期間」を参照してください。

事前定義定義クラスタセンダチャンネルを削除する場合は、クラスタからキューマネージャを削除する必要があります。詳細については、「2.8.13(2) クラスタからキューマネージャを削除および再参加」を参照してください。

2.8.10 クラスタの管理

ここでは次に示すシステム管理者のための情報を提供します。

- クラスタ設計の検討事項
- クラスタ管理の検討事項

(1) クラスタ設計の検討事項

新しいクラスタをどのように設計するか説明します。

(a) フルリポジトリを保持するキューマネージャの選択

各クラスタで、フルリポジトリを保持するためには、少なくとも一つ、できれば二つ以上のフルリポジトリキューマネージャを選択する必要があります。クラスタは一つのフルリポジトリキューマネージャでも問題なく動作しますが、二つ使用すると可用性が向上します。その場合、フルリポジトリキューマネージャ間のクラスタセンダチャンネルを定義することによって、相互接続します。

図 2-53 一般的な 2 フルリポジトリキューマネージャのトポロジ



(凡例)



これは、一般的な 2 フルリポジトリキューマネージャのトポロジを表しています。モデルになるクラスタのシステム構成については、「2.8.1(2) クラスタの概念」を参照してください。

- 最も重要な検討事項は、選択したフルリポジトリキューマネージャは信頼性が高くよく管理されていなければならないということです。

- また、キューマネージャのロケーションを検討し、中央の位置、またはクラスタのほかの多数のキューマネージャと同じシステムに位置するものを選択します。
- さらに、キューマネージャがすでにほかのクラスタのためのフルリポジトリを持っているかどうか検討します。一度決定し、あるキューマネージャをクラスタのフルリポジトリとして設定するのに必要な定義を作成したら、同じキューマネージャを使い続けて、メンバであるほかのクラスタのフルリポジトリを保持することをお勧めします。

キューマネージャが自分自身についての情報を送信したり、ほかのキューマネージャについての情報を要求したりすると、その情報や要求は二つのフルリポジトリキューマネージャに送信されます。

事前定義クラスタセンダチャンネル定義を一本しか作成していない場合は、次のチャンネルを使用して、クラスタ情報を送信します。

- 事前定義クラスタセンダチャンネル
- システムが自動的に作成したどれか一つの自動定義クラスタセンダチャンネル

事前定義クラスタセンダチャンネル定義を二本以上作成した場合は、自動的に作成されたクラスタセンダチャンネルは使用しません。クラスタへ初回参加時に使用した二本の事前定義クラスタセンダチャンネルだけを使用して、クラスタ情報を送信します。

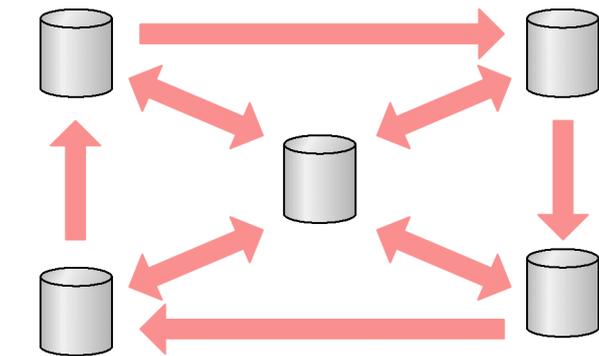
送信された情報や要求は、事前定義クラスタセンダチャンネル定義上で指定されているフルリポジトリキューマネージャが、要求を処理します。そのフルリポジトリキューマネージャが使用できない場合でも、もう一つの送信先フルリポジトリキューマネージャが要求を処理します。これによって一つのフルリポジトリキューマネージャが使用できなくなっても、最新のクラスタ情報が保持されます。

最初のフルリポジトリキューマネージャが再度利用できるようになると、そのフルリポジトリキューマネージャは最新の変更された情報を収集し、遅れを取り戻します。

クラスタセンダチャンネル定義上で指定されているフルリポジトリキューマネージャが、可能な場合要求を処理しますが、選択されたフルリポジトリキューマネージャが利用できない場合、ほかのフルリポジトリキューマネージャが使用されます。最初のフルリポジトリキューマネージャが再度利用可能になると、そのフルリポジトリキューマネージャは最新の変更された情報を収集し、遅れを取り戻します。

とても巨大なクラスタでは千単位のキューマネージャがあり、二つより多くのフルリポジトリキューマネージャが必要になります。その場合、例えば、次に示すようなトポロジにできます。

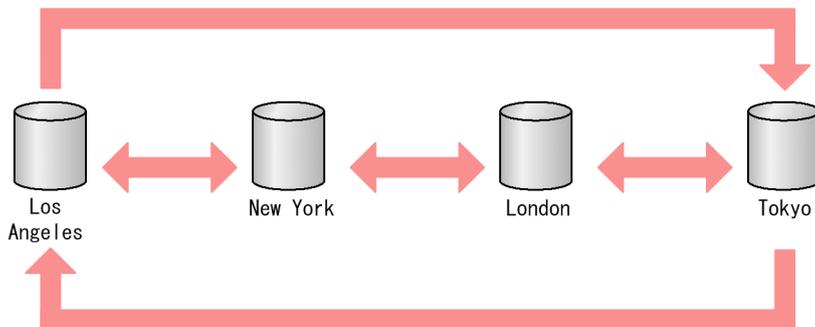
図 2-54 フルリポジトリキューマネージャをハブアンドスポーク構成にした配置



(凡例)



図 2-55 複雑なフルリポジトリキューマネージャのトポロジ



(凡例)



すべてのフルリポジトリキューマネージャが同時に使用できなくなる場合、自キューマネージャはパーシャルリポジトリにある情報を使用して処理を続行します。新しい情報と更新要求は処理されません。フルリポジトリキューマネージャがネットワークに再接続するとき、すべてのリポジトリ（フルおよびパーシャル）を最新にするようメッセージが交換されることになります。

(b) クラスタを構成

フルリポジトリを保持するための幾つかのキューマネージャを選択したら、どのキューマネージャをどのフルリポジトリとリンクするか決める必要があります。クラスタセンダチャンネル定義は、クラスタのほかのフルリポジトリが見つかるフルリポジトリキューマネージャにリンクします。そのあと、キューマネージャは、メッセージを二つのフルリポジトリキューマネージャに送信します。この時、必ず最初にクラスタセンダチャンネル定義を持っているフルリポジトリキューマネージャに送信します。どのフルリポジトリキューマネージャを選んだかは問題になりません。しかし、自分の構成のトポロジを検討する必要があります。また、キューマネージャの物理的、地理的位置も検討する必要があります。すべてのクラスタ情報は二つのフルリポジト

リキューマネージャに送信されるので、2番目のクラスタセンダチャンネル定義を作成したいような場合があります。多数のフルリポジトリが広範囲に広がるクラスタで、自分の情報がどのフルリポジトリキューマネージャに送信されるかコントロールするために定義します。

(c) 命名規則

新しいクラスタを設定する場合に、キューマネージャの命名規則を検討します。すべてのキューマネージャは、異なる名前を持つ必要があります。なお、似たような名前を付ければ、どのキューマネージャがグループになっているか覚えやすくなります。

また、すべてのクラスタレシーバチャンネルは、ユニークな名前を持つように設定してください。この場合、キューマネージャ名の前に TO を付けるのも、一つの方法です（例えば、TO.QM1、TO.QM2 など）。同じキューマネージャへの複数のチャンネルを持っており、それぞれの優先度が異なり、異なるプロトコルを使用している場合には TO.QM1.A1、TO.QM1.N3、および TO.QM1.T4 などの名前を使用して命名規則を拡張できます。

すべてのクラスタセンダチャンネルは、対応のクラスタレシーバチャンネルと同じ名前を持つことを覚えておいてください。

(d) マルチクラスタ

複数のクラスタをまとめてマルチクラスタを構成できます。次に示す場合にマルチクラスタの作成を検討してください。

- 異なる組織が独自の管理を行えるようにする場合
- 独立したアプリケーションを別々に管理する場合
- サービス階層を設定する場合
- テスト用と設計用の環境を作成する場合

キューマネージャが複数のクラスタのメンバである場合、定義の数を減らすためにネームリストを利用できます。ただし、TP1/Message Queue 07-50、07-51 および 07-52 ではネームリストをサポートしていません。そのため次に示す項目に注意してください。

- クラスタセンダチャンネル
各クラスタを管理するフルリポジトリキューマネージャに接続するために、フルリポジトリキューマネージャごとに事前定義クラスタセンダチャンネル（mqtalcccha 定義コマンド）が必要です。
- クラスタレシーバチャンネル
各クラスタを管理するフルリポジトリキューマネージャに接続するために、フルリポジトリキューマネージャごとにクラスタレシーバプロセスとクラスタレシーバチャンネルが必要です。
- クラスタキュー
mqamkque -c オプションに指定したクラスタに参加していないキューマネージャからは、アクセスできません。

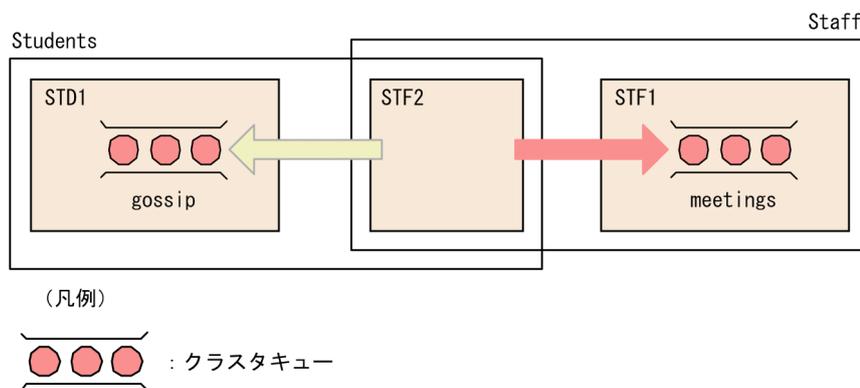
複数のクラスタがネットワークにある場合、それぞれに異なる名称を付ける必要があります。同じ名称の二つのクラスタをいったん合併した場合は、分離できません。クラスタ、およびチャンネルに異なる名称を付け、mqrls コマンドの出力を参照する時に容易に区別できるようにすることをお勧めします。正常動作をするためには、キューマネージャ名はクラスタ内でユニークである必要があります。

- サービス階層の設定

ある大学で各スタッフと各学生にキューマネージャがあるとします。スタッフ間のメッセージは優先度が高く、広帯域のチャンネルで転送されます。学生間のメッセージは狭くて遅いチャンネルで転送されます。通常の分散キューイングでこのネットワークを設定することができます。TP1/Message Queue はあて先キュー名とキューマネージャ名を参照し、どのチャンネルを使用するか検知します。

スタッフと学生を明確に分離するためには、次の図に示すとおりキューマネージャを二つのクラスタに分けることができます。TP1/Message Queue は Staff クラスタで定義されたチャンネルだけを使用して、Staff クラスタの meetings キューへメッセージを転送します。Students クラスタの gossip キューへのメッセージは Students クラスタで定義されたチャンネルを使用して転送します。こうして適切なサービス階層が設定されます。

図 2-56 サービス階層



(e) オブジェクト

クラスタを使用するには次に示すオブジェクトが必要です。

SYSTEM.CLUSTER.REPOSITORY.QUEUE

各キューマネージャは、SYSTEM.CLUSTER.REPOSITORY.QUEUE というローカルキューを持っています。このキューは、すべてのリポジトリ情報を保存するのに使用されます。

このキューのメッセージは削除しないでください。削除した場合、OpenTP1 が異常終了することがあります。また、削除したことによって、以前のクラスタ参加情報が失われるため、再度クラスタに参加する必要があります。

SYSTEM.CLUSTER.COMMAND.QUEUE

クラスタの各キューマネージャは、SYSTEM.CLUSTER.COMMAND.QUEUE というローカルキューを持っています。このキューは、メッセージをフルリポジトリに転送するのに使用されます。キューマネージャはこのキューをあて先として使用して、自分についての新しい情報または変更された情報をフルリポジトリキューマネージャに送信したり、ほかのキューマネージャについての情報の要求を送信したりします。

また、要求した情報の応答、および取得済みの情報の変更された情報を受信するキューとして使用します。

SYSTEM.CLUSTER.TRANSMIT.QUEUE

各キューマネージャは、SYSTEM.CLUSTER.TRANSMIT.QUEUE というローカルキューを持っています。これは、クラスタ内のすべてのキューやキューマネージャへのすべてのメッセージに使用されるクラスタ転送キューです。

注意事項

SYSTEM.CLUSTER.TRANSMIT.QUEUE には、ユーザメッセージとともにシステム制御用のシステムメッセージが登録されます。ユーザメッセージを削除する必要がある際には、システムメッセージを削除しないように注意してください。

(2) クラスタ管理の検討事項

システム管理者に関する検討事項を説明します。

(a) キューマネージャのリフレッシュ

キューマネージャが保持するクラスタについての情報をリフレッシュできます。ただし、通常の運用ではほとんど実行する必要がありません。リフレッシュするには `mqrrefresh` コマンドを入力します。自システム以外のキューマネージャに関連するすべてのクラスタキューマネージャオブジェクトとすべてのクラスタキューオブジェクトを、パーシャルリポジトリから削除できます。また、このコマンドはクラスタ転送キューにメッセージがなく、フルリポジトリキューマネージャに接続されていない自動定義クラスタチャンネルを削除します。

注意事項

通常、`mqrrefresh` コマンドを入力する必要はありません。`mqrrefresh` コマンドを入力するのは、フルリポジトリキューマネージャが保持するリポジトリ情報と TP1/Message Queue がパーシャルリポジトリで保持するリポジトリ情報とに不整合が発生した場合です。次に示す場合にコマンドを入力します。

- リポジトリ情報の破棄
フルリポジトリキューマネージャと通信するリポジトリ情報が破棄された場合です。例えば、運用やアプリケーションの誤りによって SYSTEM.CLUSTER.COMMAND.QUEUE のメッセージを削除した場合や、SYSTEM.CLUSTER.COMMAND.QUEUE に登録すべき自キューマネージャあてのメッセージが、他キューマネージャの SYSTEM.CLUSTER.TRANSMIT.QUEUE から削除された場合です。
- 米国 IBM 社からの指示
米国 IBM 社によって REFRESH CLUSTER コマンドを入力することが指示された場合です。
- クラスタレシーバチャンネル名の重複解消

クラスタ内の複数のキューマネージャに、同じ名前のクラスタレシーバチャンネルがある場合、意図しないキューマネージャにメッセージが転送されることがあります。この場合、クラスタレシーバチャンネルの名前を重複しないように変更したあと、重複したクラスタレシーバチャンネルを使用している他キューマネージャから `mqrrefresh` コマンドまたは `REFRESH CLUSTER` コマンドを入力する必要があります。

- `RESET CLUSTER` コマンドの誤入力
誤って `RESET CLUSTER ACTION(FORCEREMOVE)` コマンドを入力した場合です。
- 復元後のキューマネージャ再開始
キューファイルのバックアップを復元するなどして、キューマネージャを最後に終了した時点よりも前の状態で再度開始した場合です。

(b) キューマネージャの脱退

キューマネージャをクラスタから脱退できます。ただし、通常の運用ではほとんど実行する必要がありません。

脱退するには `mqrremove` コマンドを入力します。自システムのクラスタレシーバチャンネルについてフルリポジトリに削除依頼を送信します。

注意事項

通常、`mqrremove` コマンドを入力する必要はありません。`mqrremove` コマンドを入力するのは、使用されなくなる TP1/Message Queue 環境を削除する場合や、フルリポジトリキューマネージャが保持するリポジトリ情報と TP1/Message Queue がパーシャルリポジトリで保持するリポジトリ情報に不整合が発生した場合です。次に示す場合にコマンドを入力します。

- 復元後のキューマネージャ再開始
キューファイルのバックアップを復元するなどして、キューマネージャを最後に終了した時点よりも前の状態で再度開始した場合です。
- クラスタセンダチャンネルの情報の消滅
システム時刻を戻したことなどによってクラスタセンダチャンネルの情報が消滅し、リポジトリ情報をフルリポジトリに送信できない場合です。
- `SYSTEM.CLUSTER.TRANSMIT.QUEUE` に障害が発生した場合
- `SYSTEM.CLUSTER.REPOSITORY.QUEUE` に障害が発生した場合
- `SYSTEM.CLUSTER.COMMAND.QUEUE` に障害が発生した場合

(c) クラスタ転送キューの保守

クラスタ転送キューの可用性とパフォーマンスは、クラスタのパフォーマンスにとって重要です。満杯にならないようにする必要があります。また、`mqaset` コマンドを間違えて発行して取り出し不可や登録不

可にしないように注意してください。なお、クラスタ転送キューが格納されるキューファイルは、満杯にならないように注意してください。

(d) キューマネージャ失敗時の処理

メッセージがバッチ転送によって特定のキューマネージャに送信されたがそのキューマネージャが利用不可になった場合、次に示すような対策ができます。

- ファーストチャネルのファーストメッセージ（喪失されることがある）を除き、未転送のメッセージバッチは、送信したキューマネージャのクラスタ転送キューに戻されます。
 - 戻されたメッセージバッチに問題がなく、メッセージが特定のキューマネージャとバインドされていない場合、メッセージ送信経路の再設定が行われ、適切な代替キューマネージャを選択し、メッセージを送信します。
 - すでにキューマネージャに転送されたり（送達未確認状態）、問題があったり、適切な代替のキューマネージャが見つからないメッセージは、元のキューマネージャが再度利用可能になるまで待つ必要があります。

(e) リポジトリへの登録失敗時の処理

クラスタ情報は、SYSTEM.CLUSTER.COMMAND.QUEUE というローカルキューを介してパシャリリポジトリに反映されます。このキューが満杯の状態である場合、リポジトリ管理サーバは動作を停止しているので、クラスタ情報メッセージはデッドレターキューに転送されます。ログメッセージから発生がわかったら、デッドレターキューからメッセージを回復し、それを正しいあて先へと転送し直すアプリケーションを実行する必要があります。

リポジトリ情報は SYSTEM.CLUSTER.REPOSITORY.QUEUE というローカルキューに格納されています。このキューが満杯の状態である場合、クラスタ情報の追加と更新ができないでエラーログが出力されます。その場合、キューマネージャを停止し、キューの容量を拡大したあと、開始してください。

キューを再作成した場合は、以前のクラスタ参加情報は失われ、再度クラスタ参加を実行します。この場合、mqrsup と mqrspp の起動をする前に、フルリポジトリキューマネージャから RESET CLUSTER コマンドを実行して、自キューマネージャをクラスタから削除してください。

キューマネージャのリポジトリの共用メモリ不足が発生した場合（とてもまれな場合）、共用メモリ確保エラーがログ出力されます。この場合、キューマネージャを停止し、使用共用メモリを拡大したあと、開始します。

(f) クラスタキュー登録不可時の処理

クラスタキューが登録不可になると、この状況は、そのキューに関係している各キューマネージャのフルリポジトリに反映されます。ワークロード管理機能は、可能な場合メッセージを登録できるあて先に送信しようとしています。登録できるあて先がなく、ローカルキューもない場合、MQOO_BIND_ON_OPEN を指定した MQOPEN 命令はリターンコード MQRC_CLUSTER_PUT_INHIBITED をアプリケーションに返します。MQOO_BIND_NOT_FIXED が指定されている場合、またはローカルキューがある場合、

MQOPEN 命令は成功しますが、以後の MQPUT 命令はリターンコード MQRC_PUT_INHIBITED で失敗します。

(g) リポジットリ情報の保持期間

各リポジットリ情報の保持期間を次の表 2-16、表 2-17、表 2-18 に示します。

表 2-16 リモートキューマネージャ情報の保持期間

リポジットリ情報名	内容	保持期間※	備考
クラスタキューマネージャ情報	リモートのキューマネージャに存在するクラスタレシーバチャンネル情報	60～90 日	フルリポジットリキューマネージャから、リモートキューマネージャの情報が通知されます（キューマネージャ問い合わせ情報またはキューマネージャ識別子問い合わせ情報の応答として受信します）。
クラスタキュー情報	リモートのキューマネージャに存在するクラスタキュー情報	60～90 日	フルリポジットリキューマネージャから、クラスタキューの情報が通知されます（キュー問い合わせ情報の応答として受信します）。

注※

ローカルキューマネージャ上の保持期間で、リモートのキューマネージャの情報を受信したときからローカルキューマネージャ上で削除されるまでの期間。

この期間は、フルリポジットリキューマネージャから通知された有効期間に 60 日の猶予期間を加えた合計値です。

ローカルキューマネージャ情報の送信については、「[自キューマネージャでのクラスタキューの追加](#)」を参照してください。

表 2-17 ローカルキューマネージャ情報の保持期間

リポジットリ情報名	内容	保持期間※	備考
クラスタキューマネージャ情報	ローカルキューマネージャに存在するクラスタレシーバチャンネル情報	90 日	クラスタ参加時に送信したあとは、27 日周期で有効期間を更新し、フルリポジットリキューマネージャに再送信します。
クラスタキュー情報	ローカルキューマネージャに存在するクラスタキュー情報	90 日	

注※

フルリポジットリキューマネージャ上の保持期間で、ローカルキューマネージャの情報を送信したときからフルリポジットリキューマネージャで削除されるまでの期間。

この期間は、クラスタ転送キュー（SYSTEM.CLUSTER.TRANSMIT.QUEUE）に登録したときの有効期間（30 日）とフルリポジットリキューマネージャが受信した際に加算する猶予期間（60 日）の合計値です。

リモートキューマネージャ情報の応答については、「[クラスタキューへのメッセージ登録](#)」を参照してください。

表 2-18 問い合わせ情報の保持期間

リポジットリ情報名	内容	保持期間※	備考
問い合わせ情報（キューマネージャ、キューマネージャ識別子、キュー）	次の情報を問い合わせます。	30 日	ローカルキューマネージャがリモートのキューマネージャに存在するクラスタレシーバチャンネル情報またはクラスタキュー情報を必要とした（MQOPEN

リポジトリ情報名	内容	保持期間※	備考
問い合わせ情報 (キューマネージャ, キューマネージャ識別子, キュー)	<ul style="list-style-type: none"> リモートのキューマネージャに存在するクラスタレシーバチャンネル情報 リモートのキューマネージャに存在するクラスタキュー情報 	30日	<p>命令, MQPUT 命令, MQPUT1 命令などが実行された) 場合に送信されます。</p> <p>問い合わせ情報の応答として, クラスタキューマネージャ情報またはクラスタキュー情報が通知されます。</p> <p>問い合わせ情報の有効期限内にクラスタキューマネージャ情報またはクラスタキュー情報を使用した場合には, 27日周期で問い合わせ情報の有効期間を更新し, フルリポジトリキューマネージャに再送信します。</p>

注※

ローカルキューマネージャおよびフルリポジトリキューマネージャ上の保持期間で, ローカルキューマネージャが送信した情報がフルリポジトリキューマネージャで削除されるまでの期間。

この期間は, ローカルキューマネージャが問い合わせ情報をクラスタ転送キュー (SYSTEM.CLUSTER.TRANSMIT.QUEUE) に登録したときの有効期間 (30日) です。

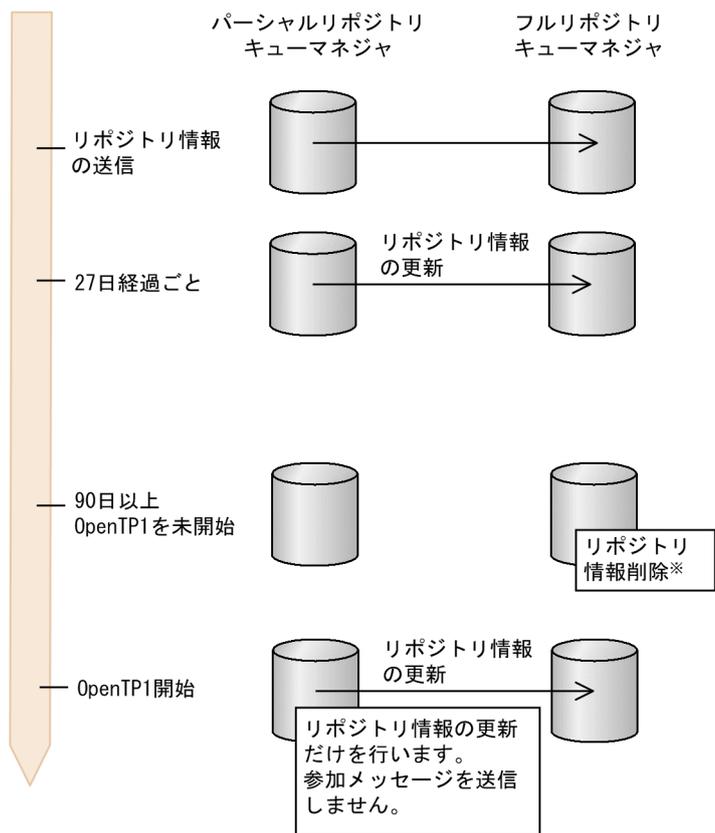
リモートキューマネージャ情報の要求については, 「[クラスタキューへのメッセージ登録](#)」を参照してください。

自キューマネージャでのクラスタキューの追加

キューマネージャが, 例えば, 新しいキューの作成を通知するためフルリポジトリキューマネージャに, 自キューマネージャのクラスタキュー情報を送信すると, フルリポジトリキューマネージャおよびパーシャルリポジトリキューマネージャは, その情報を30日間保管します。リポジトリの情報の期限が切れないように, 自キューマネージャは, 27日後に自動的に自分についてのすべての情報を再送信します。有効期間が過ぎても, 情報はフルリポジトリキューマネージャおよびパーシャルリポジトリキューマネージャ上のリポジトリから即座には削除されません。60日の猶予期間で保管されます。猶予期間中に更新情報を受け取らなかった場合, 情報は削除されます。有効期日にキューマネージャが一時的に使用できなくなるおそれを考慮して猶予期間があります。キューマネージャが90日を超えてクラスタから切断された場合, キューマネージャはクラスタの一部ではなくなります。

リポジトリ情報の保持期間について次の図に示します。

図 2-57 リポジトリ情報の保持期間



注※

90 日以上経過したことでリポジトリ情報が削除され、パーシャルリポジトリキューマネージャはクラスタから除去されます。

OpenTP1 開始時に、パーシャルリポジトリキューマネージャは参加した情報を保持しているため、開始時に参加メッセージを送信しないで、自マネージャ情報だけを送信します。そのため、mqremove コマンドによって脱退しないとクラスタに参加できません。

リポジトリ情報の有効期間が切れないように、キューマネージャは 27 日を経過したことを検出したときに、自動的に自分についてのすべての情報を、フルリポジトリに対応するクラスタセンダチャンネルを使用して再送します。クラスタセンダチャンネルで送信中は、チャンネル状態は「チャンネル動作中」になります。

なお、事前定義クラスタセンダチャンネルやデフォルトチャンネル定義のチャンネルの確立方法 (mqtalcccha 定義コマンド-i オプション指定値) に manual を指定している場合、フルリポジトリに対するクラスタセンダチャンネルが動作中でないとき自動的にリポジトリ情報は再送されません。このためユーザがチャンネルを動作させる必要があります。

最後のリポジトリ情報の送信から 90 日以上チャンネルを確立しない場合 (OpenTP1 未開始など)、リポジトリ情報は失われます。

TP1/Message Queue は次に示す契機に有効期間の確認を行い、有効期間が過ぎている場合は更新情報を送信します。

- リポジトリ管理サーバ開始時
- リポジトリ管理サーバ開始後の一定間隔

- リポジトリ管理サーバ終了時

クラスタキューへのメッセージ登録

自キューマネージャ上のアプリケーションが、例えば、任意のパーシャルリポジトリキューマネージャのクラスタキューにメッセージを登録しようとした場合、自キューマネージャはフルリポジトリキューマネージャに対して、問い合わせ情報を送信します。問い合わせ情報の存続期間は 30 日です。自キューマネージャは、27 日後に問い合わせ情報をチェックします。27 日の間に問い合わせ情報で取得した情報が参照された場合、その問い合わせ情報は自動的にリフレッシュされます。この期間に参照されなかった場合、その問い合わせ情報は期限満了の対象として各キューマネージャから除去されます。

フルリポジトリキューマネージャは、問い合わせ情報に対応するリポジトリ情報（クラスタレシーバチャンネル情報およびクラスタキュー情報）を送信します。キューマネージャは、その情報を受信後 30 日間保管します。有効期間が過ぎても、これらの情報はリポジトリから即座には削除されません。60 日の猶予期間で保管されます。猶予期間中に更新情報を受け取らなかった場合、リポジトリ情報は削除されます。有効期日にキューマネージャが一時的に使用できなくなるおそれを考慮して猶予期間があります。

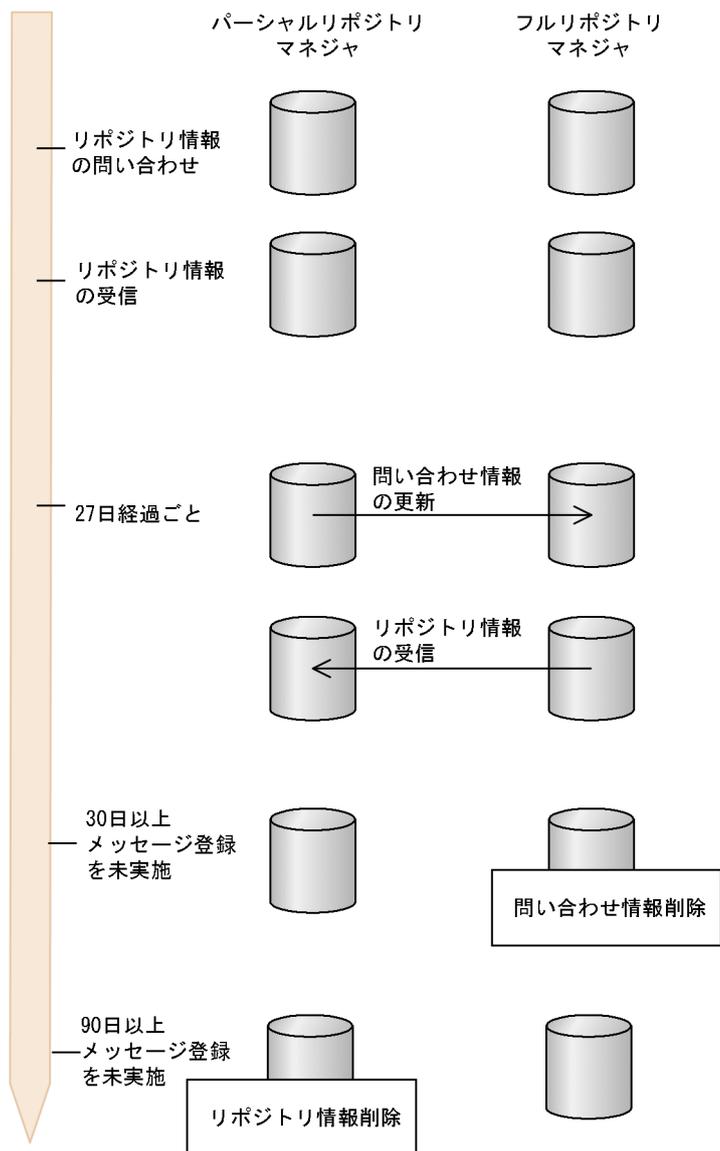
問い合わせ情報がリフレッシュされずに、フルリポジトリキューマネージャから削除された場合、フルリポジトリキューマネージャはリポジトリ情報を送信しなくなります。

自キューマネージャ上のアプリケーションは、MQOPEN 命令、MQPUT 命令、または MQPUT1 命令で、受信したこれらのリポジトリ情報を使用します。

通常、定期的（30 日以内）に MQOPEN 命令、MQPUT 命令、または MQPUT1 命令を実行している場合は、問い合わせ情報は自動的にリフレッシュされ、リポジトリ情報は最新に更新されます。一方、MQOPEN 命令を実行後、長期間 MQPUT 命令を実施しないまま放置するなど、定期的に使用されない場合は、自キューマネージャのリポジトリ上からリポジトリ情報は削除されます。

リポジトリ情報の保持期間について次の図に示します。

図 2-58 リポジトリ情報の保持期間



最後の情報の受信から 90 日以上チャンネルを確立しない場合（OpenTP1 未開始など）、フルリポジトリキューマネージャ上から情報が失われます。

TP1/Message Queue は次に示す契機で有効期間を確認し、有効期間が過ぎている場合は、情報を除去します。

- リポジトリ管理サーバ開始時
- リポジトリ管理サーバ開始後の一定間隔
- リポジトリ管理サーバ終了時

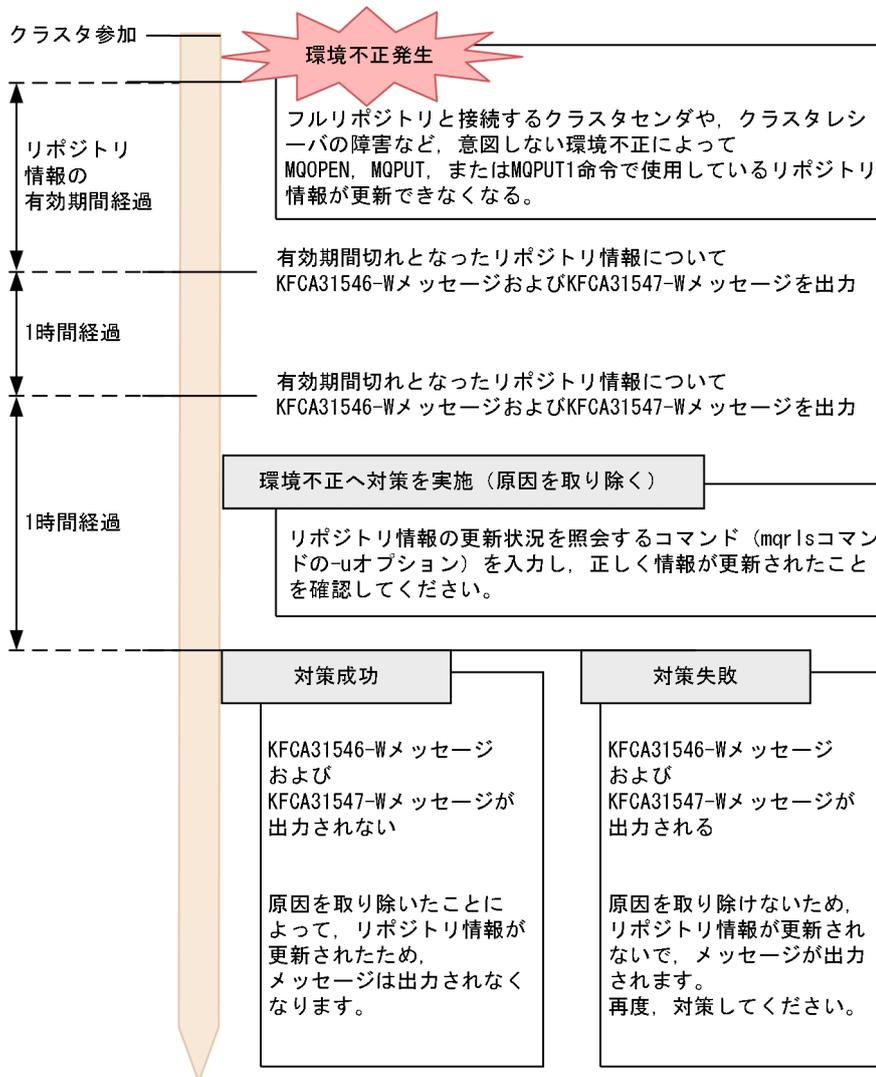
(h) 環境不正によるリポジトリ情報の有効期限切れの通知

フルリポジトリキューマネージャと接続するチャンネル（クラスタセンダまたはクラスタレシーバ）の障害などの環境不正が発生すると、定期的（30 日以内）に使用しているリポジトリ情報が更新されなくなります。

対策しない場合にリポジトリ情報の保持期間が満了すると、リポジトリ情報は使用できなくなり、MQOPEN 命令、MQPUT 命令、またはMQPUT1 命令が失敗します。そのため、障害が解決されないままリポジトリ情報が有効期間切れとなった場合、保持期限を通知する KFCA31546-W および KFCA31547-W メッセージを出力します。

メッセージが出力された場合、必要なリポジトリ情報が更新されるように対策してください。

図 2-59 リポジトリ情報有効期限切れ通知機能



注意事項

- リポジトリ管理サーバ（mqrsup）が停止している間はメッセージが出力されません。リポジトリ管理サーバ（mqrsup）停止中に出力契機に達した場合、リポジトリ管理サーバ（mqrsup）を再起動したときにメッセージが出力されます。
- OpenTP1 を長期間停止していた場合、OpenTP1 の開始時にメッセージが1度だけ出力されることがあります。この場合は、リポジトリ管理サーバ（mqrsup）によってリポジトリ情報が更新されるため対策不要です。必要に応じて、リポジトリ情報の更新状況を照会するコマンド

(mqrls コマンドの-u オプション) を入力し、リポジトリ情報の保持期限が更新されたことを確認してください。

- メッセージが出力されたりリポジトリ情報について対策しなかった場合、保持期限を過ぎるとリポジトリ情報が削除されます。削除されたりリポジトリ情報についてはメッセージが出力されなくなります。

(i) リポジトリ情報送信契機の通知

TP1/Message Queue は、リポジトリ情報の有効期限が切れる前に、フルリポジトリキューマネージャに対してリポジトリ情報の更新のため、自動的に問い合わせ情報を再送信します。更新の対象となるのは有効期限内だけです。

リポジトリ情報送信通知機能では、リポジトリ情報更新時の問い合わせ情報の再送信契機をメッセージ出力 (リポジトリ情報送信通知メッセージ (KFCA31544-I), およびリポジトリ情報送信完了メッセージ (KFCA31545-I)) によって事前に確認できるようになります。これによって、メッセージ出力されているリポジトリ情報は有効期限内で使用されているものであることが確認できます。

ただし、更新対象が複数存在する場合は、最も早くに送信契機に達するリポジトリ情報に対してだけ、メッセージが出力されます。そのため、その時点で更新対象となるすべてのリポジトリ情報については、mqrls コマンドの-u オプションで、リポジトリ情報の有効期間を確認してください。

有効期限の更新が必要なリポジトリ情報は、アプリケーションで MQI 命令 (MQOPEN 命令, MQPUT 命令, または MQPUT1 命令) を発行してください。

定義

次の環境変数をリポジトリ管理サーバユーザサービス定義ファイル (\$DCCONFPATH/mqrsup) に定義してください。

- リポジトリ情報送信通知メッセージの出力要否: DCMQA_MQR_REFRESH_REPORT
この環境変数に Y を指定した場合、最も早くに送信契機に達するリポジトリ情報に対してだけ、このメッセージが出力されます。
- リポジトリ情報送信通知メッセージの出力時間設定: DCMQA_MQR_REFRESH_REPORT_TIME
リポジトリ情報送信時刻から換算してこの環境変数に指定した時間だけ前に KFCA31544-I メッセージが出力されます。また、KFCA31544-I メッセージで表示されたりリポジトリ情報の送信後に KFCA31545-I メッセージが出力されます。

注意事項

- リポジトリ管理サーバ (mqrsup) が停止している間はメッセージが出力されません。リポジトリ管理サーバ (mqrsup) 停止中に DCMQA_MQR_REFRESH_REPORT_TIME に指定した出力契機に達した場合、リポジトリ管理サーバ (mqrsup) を再起動したときに DCMQA_MQR_REFRESH_REPORT_TIME で指定した時刻より短い時間でメッセージが出力されます。

- DCMQA_MQR_REFRESH_REPORT の指定値を Y から N へ変更しリポジトリ管理サーバ (mqrsup) を再起動した場合、オペランド変更前に出力された KFCA31544-I メッセージは引き継がれません。オペランドを再度 N から Y に変更した場合に、DCMQA_MQR_REFRESH_REPORT_TIME に指定した出力契機に達したりリポジトリ情報に対して、再度 KFCA31544-I メッセージが出力されます。
- KFCA31544-I メッセージ出力後、リポジトリ情報送信前に KFCA31544-I メッセージで表示されたりリポジトリ情報が削除された場合、KFCA31545-I メッセージは出力されません。
- 次の条件のどれかに一致するリポジトリ情報に対してだけ、KFCA31544-I および KFCA31545-I メッセージが出力されます。
 - クラスタ状態が JOIN でフルリポジトリキューマネージャが存在する場合
 - クラスタ状態が REFRESH の場合
 - クラスタ状態が BINDING の場合 (ただし、mqrremove コマンドの-s オプション実行後にクラスタ状態が BINDING となった場合)
- KFCA31544-I メッセージ出力後、かつ KFCA31545-I メッセージ出力前に OpenTP1 を停止させた場合、OpenTP1 再起動後のリポジトリ管理サーバ (mqrsup) 開始時に、以前と同じリポジトリ情報に対して再度 KFCA31544-I メッセージが出力されます。
- リポジトリ情報の有効期限を定期的 (1 時間間隔) に確認しているため、KFCA31544-I メッセージは DCMQA_MQR_REFRESH_REPORT_TIME で指定した時間より最大で 1 時間遅れて出力される場合があります。
また、KFCA31545-I メッセージは、KFCA31544-I メッセージで表示されたりリポジトリ情報の送信までの時間より最大で 1 時間遅れて出力される場合があります。
DCMQA_MQR_REFRESH_REPORT_TIME オペランドには 1 時間の遅れを考慮した値を指定するようにしてください。
- DCMQA_MQR_REFRESH_REPORT_TIME の値を以前の値とは異なる値に変更した場合、リポジトリ管理サーバ (mqrsup) を再起動してから有効となりますが、オペランド変更前に出力された KFCA31544-I メッセージは引き継がれません。新しく指定した値 (時刻) で再度 KFCA31544-I メッセージが出力されます。

各リポジトリ情報の保持期間と、リポジトリ情報の送信契機を次に示します。

表 2-19 リポジトリ情報の保持期間と情報送信契機

リポジトリ情報名称	保持期間	情報送信契機
ローカルキューマネージャ情報 (自キューマネージャに存在するクラスタレシーバチャンネル情報)	90 日*1	27 日
ローカルキュー情報 (自キューマネージャに存在するクラスタキュー情報)	90 日*1	27 日

リポジトリ情報名称	保持期間	情報送信契機
リモートキューマネージャ情報 (リモートのキューマネージャに存在するクラスタレシーバチャンネル情報) (送信しない)	60~90 日 ^{※2}	なし
リモートキュー情報 (リモートのキューマネージャに存在するクラスタキュー情報) (送信しない)	60~90 日 ^{※2}	なし
システム用問い合わせ情報	無期限	27 日
問い合わせ情報 (キューマネージャ, キューマネージャ識別子, キュー)	30 日	27 日 ^{※3}

注※1

ローカルキューマネージャの情報を送信したときからフルリポジトリキューマネージャで削除されるまでの期間。
この期間は、クラスタ転送キュー (SYSTEM.CLUSTER.TRANSMIT.QUEUE) に登録したときの有効期間 (30 日) とフルリポジトリキューマネージャが受信した際に加算する猶予期間 (60 日) の合計値です。なお、ローカルキューマネージャ上ではローカルキューマネージャ情報の保持期間は無期限です。

注※2

リモートのキューマネージャの情報を受信したときからローカルキューマネージャ上で削除されるまでの期間。この期間は、フルリポジトリキューマネージャが設定した有効期間 (最大 30 日) に 60 日の猶予期間を加えた合計値です。

注※3

情報送信契機で問い合わせ情報が送信されたあと、ユーザアプリケーションから同様の問い合わせ情報を使用したメッセージ登録を発行していない場合は、次の情報送信契機で問い合わせ情報は送信されません。この状態では KFCA31544-I メッセージの出力対象とはなりません。

(j) クラスタチャンネル

クラスタを使用すればチャンネルを定義する必要がなくなりますが、分散キューイングで使われているのと同じチャンネル技術が、クラスタのキューマネージャ間の通信に使用されています。次に示すことを理解しておいてください。

- チャンネルの動作方法
「2.8.9 クラスタ環境の通信構成」を参照してください。
- チャンネルのステータスの見つけ方
6 章の「mqtlscha (チャンネルの状態表示)」を参照してください。
- メッセージ編集出口の使い方
「3. ユーザオウンコーディング」を参照してください。

(k) IBM MQ で SUSPEND QMGR コマンドを実行した場合

IBM MQ のキューマネージャに SUSPEND QMGR コマンドを実行した場合、ワークロード管理機能は、該当するキューマネージャに対して、そこで処理する必要のあるメッセージだけを送信ようになります。その後、RESUME QMGR コマンドを実行すると、ワークロード管理機能は通常の動作に戻ります。

2.8.11 キューマネージャクラスタの準備

TP1/Message Queue でクラスタを使用する場合に必要な環境設定について説明します。

(1) リポジトリ管理の構成

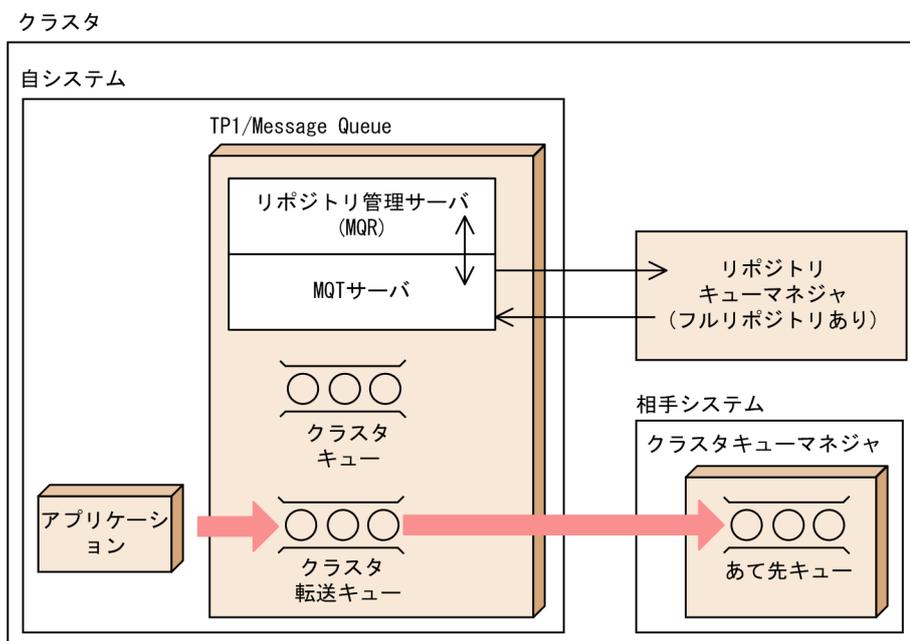
リポジトリの管理をする部分をリポジトリ管理サーバ (MQR) といいます。リポジトリ管理サーバは、クラスタに参加するためのリポジトリ情報を管理します。

(2) リポジトリ管理サーバの動作

パーシャルリポジトリを持つ TP1/Message Queue のリポジトリ管理サーバは、クラスタキューマネージャおよびクラスタキューの場所についてのリポジトリ情報を、フルリポジトリキューマネージャの持つフルリポジトリと MQT サーバを介して交換します。

リポジトリ情報の交換について、次の図に示します。

図 2-60 リポジトリ情報の交換



(3) リポジトリ管理サーバの作成

リポジトリ管理サーバは、OpenTP1 システムのユーザサーバです。

mqrmake コマンドを実行して、実行形式プログラム (mqr sup および mqr spp) とリポジトリ管理サーバユーザサービス定義ファイル (mqr sup および mqr spp) を作成してください。ただし、同一名称のユーザサービス定義ファイルがある場合、mqrmake コマンドはリポジトリ管理サーバユーザサービス定義ファイルを作成しません。

mqrmake コマンド実行時には、次に示す定義ファイルが必要です。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

- \$DCCONFPATH/betranrc

作成した mqr sup および mqr spp は、次に示すディレクトリに配置されます。

- \$DCDIR/aplib：実行形式プログラム
- \$DCCONFPATH：リポジトリ管理サーバユーザサービス定義ファイル

また、MQA サービス定義の mqa_mqr_conf オペランドに Y を指定してください。

mqrmake コマンドの延長で OpenTP1 に RM 登録をしています。したがって、dcsetup (-d の y 応答) *コマンドを実行し直した場合は、再度 mqrmake コマンドも実行する必要があります。

注※

Windows 版については、インストーラでアンインストール後、再インストールをした場合となります。

注意事項

mqr sup, mqr spp で設定できる値を次に示します。

1. mqrmake コマンドで出力された定義ファイルのオペランド

この値は変更しないでください。

2. このマニュアルに記載されたオペランド

このマニュアルに従った値で設定してください。

3. 1. および 2. 以外のオペランド

OpenTP1 のユーザサービスデフォルト定義のデフォルト値で動作するように設定してください。

mqr sup, mqr spp のユーザサービス定義、またはユーザサービスデフォルト定義に trnrmid 定義コマンドを指定しないでください。指定した場合、mqr sup, mqr spp は起動時に KFCA00906-E メッセージを出力して異常終了します。

(4) リポジトリ管理サーバの環境変数

リポジトリ管理機能が使用する環境変数は、リポジトリ管理サーバユーザサービス定義ファイル (\$DCCONFPATH/mqr sup) に定義してください。形式について、次に示します。

(a) 形式

```
[putenv CLUSTSSTARTWAIT リポジトリ管理サーバ起動待ち合わせ時間]
[putenv CHAINFWAIT クラスチャネル情報待ち時間]
[putenv DCMQA_TRACE_STATUS Y | N]
[putenv DCMQA_TRACE_PATH MQRトレースファイルのパス]
[putenv DCMQA_TRACE_FILE_NUM MQRトレースファイルの最大数]
[putenv DCMQA_TRACE_MAX_LEN MQRトレースファイルの最大長]
[putenv SPPGRPNAME mqrspgのサービスグループ名]
[putenv DCMQA_MQR_REFRESH_REPORT リポジトリ情報送信通知メッセージの出力可否]
[putenv DCMQA_MQR_REFRESH_REPORT_TIME リポジトリ情報送信通知メッセージの出力時間設定]
```

(b) オペランド

- CLUSTSSTARTWAIT リポジトリ管理サーバ起動待ち合わせ時間
～<符号なし整数>((1~999999999))《180》(単位:秒)
mqrspg 起動時に mqrspg 起動待ち合わせをする時間を指定します。指定時間を過ぎても mqrspg が起動されない場合、mqrspg はエラーになり起動できません。
KFCA31501-E エラーメッセージが出力された場合はこの値を見直してください。
- CHAINFWAIT クラスチャネル情報待ち時間
～<符号なし整数>((1~999999999))《180000》(単位:ミリ秒)
mqrspg 起動時にチャネル情報を取得するまでの待ち時間を指定します。指定時間を過ぎてもチャネル情報が取得できない場合、mqrspg はエラーになり起動できません。
KFCA31501-E エラーメッセージが出力された場合はこの値を見直してください。この値はチャネル定義の監視間隔よりも大きい値を指定してください。
- DCMQA_TRACE_STATUS Y | N
～《Y》
リポジトリ管理サーバの MQR トレース情報を出力するかどうかを指定します。
- DCMQA_TRACE_PATH MQR トレースファイルのパス
～<パス名>《\$DCDIR/spool/》
MQR トレース情報を出力する MQR トレースファイルのパス名を指定します。
- DCMQA_TRACE_FILE_NUM MQR トレースファイルの最大数
～<符号なし整数>((1~99))《3》
MQR トレースファイルの最大数を指定します。
- DCMQA_TRACE_MAX_LEN MQR トレースファイルの最大長
～<符号なし整数>((10000~1000000))《1000000》(単位:バイト)
MQR トレースファイルの容量を指定します。
- SPPGRPNAME mqrspg のサービスグループ名
～<1~31 文字の識別子>《mqrspg》

mqrsp サービスグループ名を指定します。この値を変更した場合は、リポジトリ管理サーバユーザサービス定義（\$DCCONFPATH/mqrsp）の service_group オペランドに指定するサービスグループ名も同じ値に変更してください。

- DCMQA_MQR_REFRESH_REPORT リポジトリ情報送信通知メッセージの出力要否
Y | N ~ 《N》

リポジトリ情報は、リポジトリ情報の保持期間更新時にキューマネージャによって、フルリポジトリキューマネージャに対して送信されます。この契機に KFCA31544-I および KFCA31545-I メッセージを出力するかどうかを指定します。

このオペランドの値を変更した場合、リポジトリ管理サーバ（mqrsp）を再起動すると有効となります。

- DCMQA_MQR_REFRESH_REPORT_TIME リポジトリ情報送信通知メッセージの出力時間設定
~ <符号なし整数> ((3600~2246400)) 《259200》 (単位：秒)

リポジトリ情報送信時刻からどのくらい前に KFCA31544-I メッセージを出力するかを指定します。

DCMQA_MQR_REFRESH_REPORT に Y が指定されていれば、リポジトリ情報送信時刻よりこのオペランドで指定した時間だけ前に KFCA31544-I メッセージが出力されます。

このオペランドの値を変更した場合、リポジトリ管理サーバ（mqrsp）を再起動すると有効となります。

(c) mqrsp のサービス定義に関する注意事項

- message_buflen を次に示す計算式で算出し、ユーザサービスデフォルト定義（\$DCCONFPATH/usrrc）の指定値、または省略値（4096）と比較して、message_buflen の値の方が大きい場合は、mqrsp サービス定義に message_buflen の値を指定してください。

$$568+W+ (X \times Y) \quad (\text{単位：バイト})$$

- message_store_buflen を次に示す計算式で算出し、ユーザサービスデフォルト定義（\$DCCONFPATH/usrrc）での指定値、または省略値（4096）と比較して、message_store_buflen の値の方が大きい場合は、mqrsp サービス定義に message_store_buflen の値を指定してください。

$$568 \times A+W+ (X \times Y) \times Z \quad (\text{単位：バイト})$$

message_buflen および message_store_buflen の計算式の記号の説明

- A：同時問い合わせ発生数
次の式で見積もってください。
 $A = B + C + D + (E \times F) + 1$
- B：同時に MQOPEN/MQPUT/MQINQ を行う UAP のプロセス数
- C：クラスタ内に存在するリモートのキューマネージャ名称の数
- D：クラスタ内に存在するリモートのクラスタキュー名称の数
- E：MQA サービス定義の mqa_maxque_expiry オペランドの指定値
- F：MQA サービス定義の mqa_maxmsg_expiry オペランドの指定値
- W：32 ビット版の場合 44, 64 ビット版の場合 56
- X：クラスタセンダチャンネル*の最大バッチ数

- Y: 32ビット版の場合 8, 64ビット版の場合 16
- Z: クラスタセンダチャンネル[※]数

注※

事前定義クラスタセンダチャンネルおよび自動定義クラスタセンダチャンネル。

(d) mqrspに parallel_count オペランドを指定する場合の注意事項

parallel_count オペランドで指定する値, およびこのオペランドについての注意事項を次に示します。

- parallel_count 常駐プロセス数 [, 最大プロセス数]
~ 〈符号なし整数〉 ((0~128))

このサービスグループ内のサービスを並行処理するプロセス数を指定します。mqrsp サービス定義の parallel_count オペランドの常駐プロセス数に 0 を指定しないでください。0 を指定した場合は、mqrsp 起動時に KFCA31501-E のメッセージを出力して mqrsp が異常終了します。

(5) リポジトリ管理サーバの開始方法

リポジトリ管理サーバは、OpenTP1 システムのユーザサーバです。

ユーザサーバの開始方法については、マニュアル「OpenTP1 運用と操作」を参照してください。

(6) クラスタ環境のシステムキュー

クラスタ環境でシステムが使用するメッセージキューには、ユーザが定義するものがあります。また、システムキューにはユーザが登録したメッセージ以外にキューマネージャが登録するメッセージも格納される場合があります。

クラスタ環境のシステムキューについて、次の表に示します。

表 2-20 クラスタ環境のシステムキュー

システムキュー名	用途	キュータイプ	定義の作成者	キューの最大メッセージ長	キューの最大メッセージ数	キュー共用性
SYSTEM.CLUSTER.TRANSMIT.QUEUE	メッセージの転送用	ローカルキュー (クラスタ転送キュー)	ユーザ	4088 バイト ^{※1}	最大 ^{※2}	共用
SYSTEM.CLUSTER.COMMAND.QUEUE	リポジトリ管理の受信用	ローカルキュー	ユーザ	4088 バイト ^{※3}	最大 ^{※2}	共用
SYSTEM.CLUSTER.REPOSITORY.QUEUE	リポジトリ管理情報格納用	ローカルキュー	ユーザ	4088 バイト以上	最大 ^{※2}	共用
SYSTEM.CLUSTER.REMOTE.QUEUE	クラスタ転送メッセージの登録用	リモートキュー	キューマネージャ	—	—	—

(凡例)

－：該当しません。

注※1

ユーザの最大メッセージ長（転送メッセージなので転送ヘッダ長を考慮してください）と比較して、大きい方の値を指定してください。

注※2

MQA サービス定義の mqaqueatl 定義コマンドの -d オプションで指定する最大メッセージ登録数を最大にしても、リソース（メモリおよびディスク）が増加することはないため最大を指定してください。

注※3

次に示す計算式の値と比較して、大きい方の値を指定してください。

32 ビット版の場合：(A×8) + 44

64 ビット版の場合：(A×16) + 56

A：事前定義クラスタセンダチャンネルと自動定義クラスタセンダチャンネル（リモートのクラスタレシーバチャンネル）の最大バッチサイズ

システムキューで使用するリソースはメッセージを格納するキューファイルに依存するため、キューファイル作成時の mqainit コマンドのオプションには、次に示す値を指定することをお勧めします。

クラスタ環境のキューファイル作成時のオプションについて、次の表に示します。

表 2-21 クラスタ環境のキューファイル作成時のオプション

キューファイル内のシステムキュー名	mqainit コマンドのオプション -q キュー数 -m メッセージ数 -s メッセージ長	備考
SYSTEM.CLUSTER.TRANSMIT.QUEUE	-q 1	1 キューファイルグループの 1 キューファイルが 1 キューを含む場合
	-m 1024 以上	次に示すどちらか大きい方の値を -m に設定してください。 1. 同時に相手に送信するメッセージ総数 2. リフレッシュ実行時のメッセージ総数 = (1 + 自システムのクラスタレシーバ総数) × 2 + 自システムのクラスタキュー総数 × 4 + 自システムを除くクラスタ全体のクラスタキュー総数 × 2 + 自システムを除くクラスタ全体のキューマネージャ数 × 2
	-s 4088 以上	ユーザメッセージの平均メッセージ長に合わせてください。 1 レコードに入らない場合を考慮して MQA サービス定義の mqa_max_msg_recnum オペランドに 1 メッセージの最大レコード数を指定してください。
SYSTEM.CLUSTER.COMMAND.QUEUE	-q 1	1 キューファイルグループの 1 キューファイルが 1 キューを含む場合
	-m 500*	クラスタ内のキューマネージャ数、クラスタチャンネル数、およびクラスタキューの数に比例して増やしてください。
	-s 4088 以上	－

キューファイル内のシステムキュー名	mqainit コマンドのオプション -q キュー数 -m メッセージ数 -s メッセージ長	備考
SYSTEM.CLUSTER.REPOSITORY.QUEUE	-q 1	1 キューファイルグループの 1 キューファイルが 1 キューを含む場合
	-m 500*	クラスタ内のキューマネージャ数、クラスタチャンネル数、およびクラスタキューの数に比例して増やしてください。
	-s 4088 以上	—

(凡例)

—：ありません。

注※

$$2 + A + B + C + E + (G \times 2) + I + J + (3 \times I - 2) \times (D \times 2 + K + F) + 3 \times I \times H + L$$

A：ローカルのクラスタチャンネル数（90 日以内に削除されたチャンネル数を含みます）

B：ローカルのクラスタキュー数（90 日以内に削除されたキュー数を含みます）

C：リモートのクラスタレシーバチャンネル数（90 日以内に削除されたチャンネル数を含みます）

- キューマネージャを再作成（リポジトリキューを再作成）した場合、再作成前のチャンネル、キューマネージャ、およびキュー情報は再作成後の情報とは別に数えてください。

D：リモートのキューマネージャ数

E：リモートのクラスタキュー数（90 日以内に削除されたキュー数を含みます）

F：クラスタキュー名の数（ローカルおよびリモートの両方にあるキューのうち、重複する名前は数えません）

G：参加するフルリポジトリ数（クラスタ内のフルリポジトリ数が 1 の場合は 1、クラスタ内のフルリポジトリが 2 以上の場合は 2）。複数のクラスタに参加する場合はクラスタごとに算出した総和。

H：SYSTEM.CLUSTER.REPOSITORY.QUEUE に登録されるエラーメッセージの数。次に示す場合に登録されます。

- アプリケーションで、クラスタ内にないキューマネージャ名を指定して、MQOPEN 命令を発行した場合
- アプリケーションで、クラスタ内にないキュー名を、キューマネージャ名を指定しないで MQOPEN 命令で指定した場合
- 事前定義クラスタセンダチャンネルの接続先でクラスタ名称が異なる場合

I：参加するクラスタ数

J：クラスタ内のフルリポジトリ数（複数のクラスタに参加する場合は、クラスタごとに算出した総和になります）

K：90 日以内に削除されたりリモートキューマネージャの数

- キューマネージャを再作成（リポジトリキューを再作成）した場合、再作成前のチャンネル、キューマネージャ、およびキュー情報は再作成後の情報とは別に数えてください。

L: クラスタごとに計算式 $(A + B + I + J + D \times 2 + K + F + 3 \times H)$ を算出し、そのうちの最大値システムキューの運用時の注意事項について、次に示します。

- システムキューはキューマネージャが使用して管理するキューであるため、ユーザまたはアプリケーションから障害回復以外の目的で使用しないでください。システムキューに対して運用コマンドや MQI で、オブジェクトの属性の設定、メッセージの登録、またはメッセージの取り出しなどを実行した場合、動作の保証はできません。
- システムキューに障害が発生した場合、デッドレターキューをユーザが定義および作成しているときに限り、メッセージがデッドレターキューに登録されることがあります。障害の原因に対策したあと、デッドレターキューに登録されたメッセージをあて先キューに再登録してください。

(7) リポジトリ管理サーバに関連する定義

リポジトリ管理サーバ (mqrspu および mqrspj) プロセスはトランザクションの範囲に入ります。

トランザクションとして動作することを考慮し、トランザクションサービス定義の `trn_tran_process_count` オペランドなどの関連する定義を見積もってください。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

2.8.12 クラスタ運用時の注意事項

クラスタを運用する上で注意する点を次に示します。

- 次に示す条件でリモートキューマネージャに存在するクラスタキューに MQOPEN または MQPUT1 命令を発行した場合、MQRC_CLUSTER_RESOLUTION_ERROR が返却されます。
 - mqrspu が 1 回も開始されていないとき
 - クラスタに参加できていないとき
 - mqrspu または mqrspj が開始されていないとき（ローカルキューマネージャにリモートキューマネージャの情報が存在しないとき）
- 次に示す条件でリモートキューマネージャに存在するクラスタキューに MQPUT 命令を発行した場合、MQRC_NO_DESTINATIONS_AVAILABLE が返却されます。
 - MQOO_BIND_ON_OPEN 指定の MQOPEN 命令を発行後、MQPUT 命令発行時にクラスタセクタチャンネルが削除されているとき
 - MQOO_BIND_ON_OPEN 指定の MQOPEN 命令を発行後、MQPUT 命令発行時にクラスタセクタチャンネルの情報の期限が切れているとき
- MQINQ 命令でリモートキューマネージャに存在するクラスタキューを指定した場合、次にメッセージが登録されるクラスタキューの属性が照会されます。

ただし、クラスタキューが書き込み禁止になった場合は、次に示す MQOPEN 命令のオプションによって動作が異なります。

- MQOO_BIND_ON_OPEN
常に MQOPEN 命令発行時に決定したクラスタキューの属性を照会します。
- MQOO_BIND_NOT_FIXED
すべてのクラスタキューが書き込み禁止であった場合には、すべてのクラスタキューを書き込み許可にしたときに、次にメッセージが登録されるクラスタキューの属性を照会します。
- クラスタ機能を使用する UAP の実行中に mqr sup を再度開始しないでください。
再度開始した場合、UAP の MQI がエラーリターンする場合があります。
- OpenTP1 開始後に 1 回でもリポジトリ管理サーバ (mqr sup および mqr spp) を開始させればクラスタチャネルの動作は可能となります。しかし、オンライン中にリポジトリ管理サーバを終了させた場合、クラスタ固有のメッセージの受信やメッセージ送信経路の再設定ができなくなります。オンライン中にリポジトリ管理サーバを終了しないでください。
- mqr spp サービス定義の parallel_count の常駐プロセス数に 0 を指定しないでください。
0 を指定した場合は、mqr sup 開始時に KFCA31501-E のメッセージを出力して mqr sup が異常終了します。
- 事前定義クラスタセンダチャネルの接続先がパーシャルリポジトリであり、接続後にフルリポジトリに変更する場合は、mqr sup を再度開始してください。
- クラスタ参加後、キューマネージャ名を変更しないでください。
変更した場合、以前のクラスタ参加情報は失われ、再度クラスタ参加を実行します。
- キューマネージャがクラスタに参加すると、「KFCA31509-I クラスタに参加しました」というメッセージが出力されます。このメッセージが出力されたあとに、mqtalcccha 定義コマンドの -a オプションでクラスタ名を変更する場合は、クラスタからの脱退および再参加が必要になります。クラスタからの脱退および再参加については、「2.8.13(2) クラスタからキューマネージャを削除および再参加」を参照してください。
- クラスタ環境のシステムキューのメッセージを削除しないでください。削除した場合、OpenTP1 が異常終了することがあります。
- 次に示す条件でリモートキューマネージャに存在するクラスタキューに MQOPEN 命令または MQPUT1 命令を発行する場合、MQI 命令の内部で dc_rpc_call 関数を使用します。
 - 未知のクラスタキュー名を指定
 - 未知のクラスタキューマネージャ名を指定上記の条件に一致する場合は、mqr spp が開始済みであり、dc_rpc_call 関数が発行できる環境である必要があります。
- クラスタセンダチャネルをフルリポジトリに接続し、クラスタレシーバチャネルがフルリポジトリから接続されている場合に於て先不正などのエラーによってクラスタに参加できないときは、mqr sup を終了したあと再び開始することによって、再度クラスタへの参加を試みるすることができます。

- システム時刻を変更した場合、自システムを再構築してください。また、参加していたクラスタの全フルリポジトリから RESET CLUSTER コマンドに自システムのキューマネージャ名を指定して実行してください。
- クラスタにフルリポジトリが2つある環境で、フルリポジトリのクラスタレシーバチャンネルが削除された場合、次のどちらかのメッセージが出力されます。
 - KFCA16350-I (種別に DELETE, かつチャンネル名にフルリポジトリ向けのクラスタセンダチャンネル名)
 - KFCA31533-I

フルリポジトリが1つになっていないかを `mqrsls -c` コマンドで確認してください。

`mqrsls -c` コマンドで出力される2つの「参加時使用チャンネル名」のどちらかが「****」になっている場合、フルリポジトリが1つだけであることを示しています。この場合、フルリポジトリの状態を確認してください。

2.8.13 高度な作業

クラスタについての高度な作業として次の内容について説明します。

- キューマネージャからクラスタキューを削除
- クラスタからキューマネージャを削除および再参加
- クラスタキューの属性を変更
- クラスタチャンネルの属性を変更

(1) キューマネージャからクラスタキューを削除

キューマネージャからクラスタキューを削除する方法を次に示します。

1. キューが空になるまでキューをモニタします。

`mqainq` コマンドを使用し、入力オープン数、出力オープン数、およびメッセージ登録数の三つの属性を監視します。

入力オープン数、出力オープン数、およびメッセージ登録数がすべてゼロの時、キューが空であることを確認できます。

2. チャンネル状態を確認して、送達未確認メッセージがないことを確認します。

チャンネルの送達未確認メッセージがないことを確認するには、`mqtlscha` コマンドで、クラスタセンダチャンネルの状態を確認します。

それぞれのキューマネージャから、`mqtlscha` コマンドを実行します。

```
mqtlscha -n クラスタセンダチャンネル名
```

送達未確認メッセージがある場合は、必ずそれを解決してから作業を進めてください。

3. クラスタキューを削除します。

削除するクラスタキューが存在する OpenTP1 を正常終了して、mqadeldque コマンドでクラスタキューを削除します。

```
mqadeldque 削除するクラスタキュー名 キューファイル名
```

(2) クラスタからキューマネージャを削除および再参加

クラスタからキューマネージャを削除し、再参加する方法を次に示します。IBM MQ から RESET CLUSTER コマンドを実行できる場合とできない場合に分けて説明します。

キューマネージャの削除および再参加の手順一覧を次の表に示します。必要な参照先を確認してください。

表 2-22 キューマネージャの削除および再参加の手順一覧

手順	RESET CLUSTER コマンドを実行できる場合	RESET CLUSTER コマンドを実行できない場合
クラスタからの削除	(a) RESET CLUSTER コマンドを実行できる場合の削除	(c) RESET CLUSTER コマンドを実行できない場合の削除
クラスタへの再参加 (キューファイルを再作成するとき)	(b) RESET CLUSTER コマンドを実行できる場合の再参加 • キューファイルを再作成するとき	(d) RESET CLUSTER コマンドを実行できない場合の再参加 (オンライン) • キューファイルを再作成するとき
		(e) RESET CLUSTER コマンドを実行できない場合の再参加 (オフライン) • キューファイルを再作成するとき
クラスタへの再参加 (キューファイルを再作成しないとき)	(b) RESET CLUSTER コマンドを実行できる場合の再参加 • キューファイルを再作成しないとき	(d) RESET CLUSTER コマンドを実行できない場合の再参加 (オンライン) • キューファイルを再作成しないとき
		(e) RESET CLUSTER コマンドを実行できない場合の再参加 (オフライン) • キューファイルを再作成しないとき

(a) RESET CLUSTER コマンドを実行できる場合の削除

IBM MQ から RESET CLUSTER コマンドを実行できる場合に、クラスタからキューマネージャを削除する方法を次に示します。

1. クラスタを管理しているフルリポジトリから RESET CLUSTER コマンドに自システムのキューマネージャ名を指定して実行します。
2. クラスタから削除する OpenTP1 を正常終了します。
手順 1., 2.の順序は逆でもかまいません。
3. MQA サービス定義からクラスタに関する定義を削除します。

- マルチクラスタでない場合、またはマルチクラスタですべてのクラスタから削除する場合
mqa_mqr_conf オペランド
クラスタに関する mqamqtnam 定義コマンド
- マルチクラスタで一部のクラスタから削除する場合
該当するクラスタに属するクラスタレシーバチャンネルが定義された mqamqtnam 定義コマンド
該当するクラスタに属するクラスタセンダチャンネル定義

4. OpenTP1 を正常開始します。

マルチクラスタで一部のクラスタから削除する場合に、クラスタセンダチャンネル定義を削除しなかったときは、削除後の最初のリポジトリ管理サーバ開始時に削除情報がフルリポジトリに送信されます。

送信を確認するには、OpenTP1 の開始後にクラスタセンダチャンネルを終了させ、SYSTEM.CLUSTER.TRANSMIT.QUEUE に登録されたメッセージの情報を mqalsmsg コマンドで表示します。クラスタセンダチャンネルの名前が関連識別子に設定されているメッセージがないことを確認してください。

該当するメッセージがある場合はクラスタセンダチャンネルを開始し、SYSTEM.CLUSTER.TRANSMIT.QUEUE に存在しないようにしてください。

(b) RESET CLUSTER コマンドを実行できる場合の再参加

IBM MQ から RESET CLUSTER コマンドを実行した場合に、クラスタヘキューマネージャを再参加させる方法を次に示します。

●キューファイルを再作成するとき

1. OpenTP1 を正常終了します。
2. MQA サービス定義からクラスタに関する定義を設定します。
 - mqa_mqr_conf オペランド
 - クラスタに関する mqamqtnam 定義コマンド
3. システムキュー (SYSTEM.CLUSTER.XXXX.QUEUE) が登録されているキューファイルを再作成後^{*}、再度クラスタキューを作成します。
XXXX：英字
4. OpenTP1 を正常開始します。

注※

システムキューの登録されているキューファイルが mqaquegrp 定義コマンドによって、複数のキューファイルで定義されている場合、mqaquegrp 定義コマンドで定義されているすべてのキューファイルを再作成してください

●キューファイルを再作成しないとき

1. RESET CLUSTER コマンド実行後 10 秒以上経過してから、mqrrefresh コマンドを実行します。

(c) RESET CLUSTER コマンドを実行できない場合の削除

IBM MQ から RESET CLUSTER コマンドを実行できない場合に、クラスタからキューマネージャを削除する方法を次に示します。

1. アプリケーションを終了します。

2. mqtstpcha コマンドでクラスタレシーバチャンネルを終了します。

キューマネージャを削除する、クラスタに関係するすべてのクラスタレシーバチャンネルを終了します。

3. mqrremove コマンドを実行します。

KFCA31526-I および KFCA31527-I が出力されることを確認してください。KFCA31526-I が出力されない場合、コマンドが異常終了しているおそれがあります。原因を取り除き、再度 mqrremove コマンドを実行してください。

KFCA31527-I が出力されない場合、リポジトリ管理サーバが開始していることを確認してください。

4. SYSTEM.CLUSTER.TRANSMIT.QUEUE にメッセージが滞留していないことを確認します。

5. mqtstpcha コマンドでクラスタセンダチャンネルを終了します。

KFCA16350-I, KFCA16351-I または KFCA16352-W が出力されることを確認してください。

クラスタセンダチャンネルは、mqtstpcha コマンドを入力すると、該当チャンネルに割り当てられたメッセージを別のチャンネルで送信するため、メッセージ送信経路の再設定処理を行います。再設定処理では、クラスタ転送キューに登録されているメッセージを再登録します。

KFCA16350-I, KFCA16351-I または KFCA16352-W が出力されない場合は、mqtstpcha -f コマンドを入力してチャンネルを強制解放してください。

6. チャンネル状態を確認して、送達未確認メッセージがないことを確認します。

チャンネルの送達未確認メッセージがないことを確認するには、mqtlscha コマンドで、クラスタセンダチャンネルの状態を確認します。

```
mqtlscha -n クラスタセンダチャンネル名
```

送達未確認メッセージがある場合は、必ずそれを解決してから作業を進めてください。

7. OpenTP1 を正常終了します。

8. MQA サービス定義からクラスタに関係する定義を削除します。

- マルチクラスタでない場合、またはマルチクラスタですべてのクラスタから削除する場合
mqa_mqr_conf オペランド
クラスタに関係する mqamqtnam 定義コマンド
- マルチクラスタで一部のクラスタから削除する場合
該当するクラスタに属するクラスタレシーバチャンネルが定義された mqamqtnam 定義コマンド
該当するクラスタに属するクラスタセンダチャンネル定義

9. OpenTP1 を正常開始します。

マルチクラスタで一部のクラスタから削除する場合には、クラスタセンダチャンネル定義を削除しなかったときは、削除後の最初のリポジトリ管理サーバ開始時に削除情報がフルリポジトリに送信されます。

送信を確認するには、OpenTP1 の開始後にクラスタセンダチャンネルを終了させ、SYSTEM.CLUSTER.TRANSMIT.QUEUE に登録されたメッセージの情報を mqalsmsg コマンドで表示します。クラスタセンダチャンネルの名前が相関識別子に設定されているメッセージがないことを確認してください。

該当するメッセージがある場合はクラスタセンダチャンネルを開始し、SYSTEM.CLUSTER.TRANSMIT.QUEUE に存在しないようにしてください。

(d) RESET CLUSTER コマンドを実行できない場合の再参加 (オンライン)

クラスタへキューマネージャをオンラインで再参加させる方法を次に示します。「(c) RESET CLUSTER コマンドを実行できない場合の削除」の手順 6 から再参加します。

●キューファイルを再作成するとき

1. リポジトリ管理サーバ (mqrsup, mqrspp) を終了します。
2. システムキュー (SYSTEM.CLUSTER.XXXX.QUEUE) が登録されているキューファイルを切り離します。
XXXX: 英字
3. システムキューを再作成^{*}します。
4. キューファイルをオンラインに組み込みます。
5. キューファイルの閉塞を解除します。
6. リポジトリ管理サーバを開始します。
7. クラスタチャンネルを開始します (mqstacha コマンドを実行します)。

注※

システムキューの登録されているキューファイルが mqaquegrp 定義コマンドによって、複数のキューファイルで定義されている場合、mqaquegrp 定義コマンドで定義されているすべてのキューファイルを再作成してください。

●キューファイルを再作成しないとき

1. mqremove コマンドに -s オプションを指定して実行します。
KFCA31526-I および KFCA31527-I が出力されることを確認してください。
KFCA31526-I が出力されない場合、コマンドが異常終了しているおそれがあります。原因を取り除き、再度 mqremove コマンドを実行してください。
KFCA31527-I が出力されない場合、リポジトリ管理サーバが開始されていることを確認してください。
2. クラスタチャンネルを開始します (mqstacha コマンドを実行します)。

注

「(c) RESET CLUSTER コマンドを実行できない場合の削除」の手順 8 または手順 9 まで実行したあと再参加する場合は、MQA サービス定義にクラスタに関する定義を追加し、OpenTP1 の開始後 `-f` オプションを指定した `mqrrefresh` コマンドを実行してください。

(e) RESET CLUSTER コマンドを実行できない場合の再参加（オフライン）

クラスタへキューマネージャをオフラインで再参加させる方法を次に示します。

●キューファイルを再作成するとき

1. OpenTP1 を正常終了します。
2. MQA サービス定義からクラスタに関する定義を設定します。
 - ・ `mqa_mqr_conf` オペランド
 - ・ クラスタに関する `mqamqtnam` 定義コマンド
3. システムキュー (SYSTEM.CLUSTER.XXXX.QUEUE) が登録されているキューファイルを再作成後^{*}、再度クラスタキューを作成します。
XXXX：英字
4. OpenTP1 を正常開始します。

注※

システムキューの登録されているキューファイルが `mqaquegrp` 定義コマンドによって、複数のキューファイルで定義されている場合、`mqaquegrp` 定義コマンドで定義されているすべてのキューファイルを再作成してください。

●キューファイルを再作成しないとき

1. OpenTP1 を正常終了します。
2. MQA サービス定義からクラスタに関する定義を設定します。
 - ・ `mqa_mqr_conf` オペランド
 - ・ クラスタに関する `mqamqtnam` 定義コマンド
3. MQT 定義結合ユーティリティで MQT 定義オブジェクトを再作成します。
4. OpenTP1 を正常開始します。

(3) クラスタキューの属性を変更

クラスタキューの属性を変更する方法を次に示します。

1. OpenTP1 を正常終了します。
2. クラスタキューの属性を変更します。
3. OpenTP1 を正常開始します。

この作業で次に示す点に注意してください。

- 必ず正常終了で OpenTP1 を終了してください。
- システムクラスタキューの属性は変更しないでください。

(4) クラスタチャンネルの属性を変更

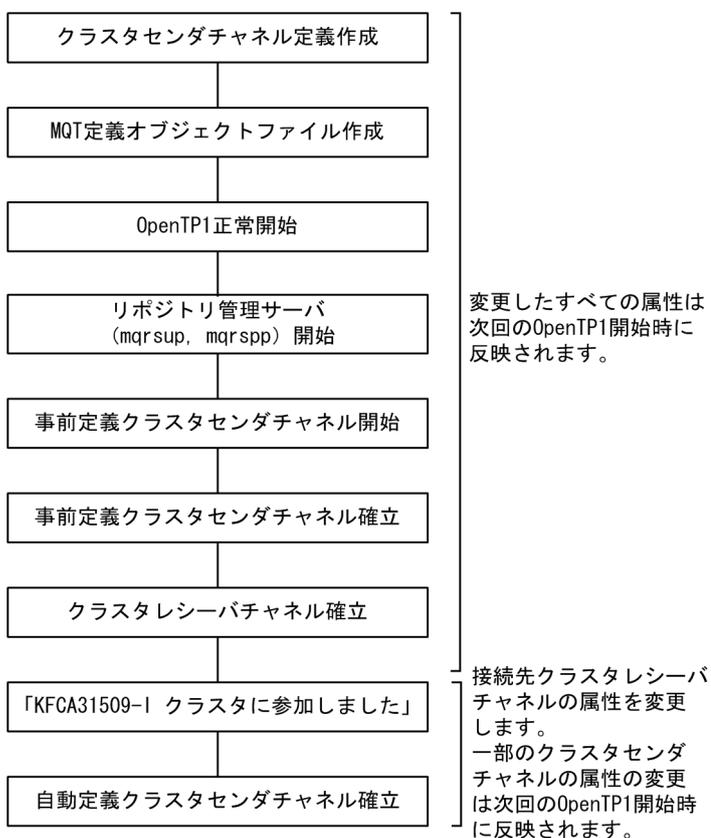
クラスタチャンネルの属性を変更する方法を次に示します。

(a) クラスタセンダチャンネルの属性の変更

クラスタセンダチャンネルの属性を変更する場合、クラスタに参加済みであるかどうかによって変更できる属性が異なります。

クラスタに参加するまでの手順を次の図に示します。

図 2-61 クラスタに参加するまでの手順



- クラスタ参加前の属性変更

クラスタに参加して「KFCA31509-I クラスタに参加しました」が出力される前に行った、事前定義クラスタセンダチャンネルの全属性の変更は次回の OpenTP1 開始時に反映されます。

1. OpenTP1 を正常終了します。
2. 事前定義クラスタセンダチャンネルの属性を変更後、MQT 定義オブジェクトファイルを再作成します。

3. OpenTP1 を正常開始します。

事前定義クラスタセンダチャンネルの属性がこの時点で反映されます。

- クラスタ参加後の属性変更

クラスタ参加後の事前定義クラスタセンダチャンネルおよび自動定義クラスタセンダチャンネルは、接続先クラスタレシーバチャンネルの属性で動作します。そのため、属性を変更する場合は接続先クラスタレシーバチャンネルの属性を変更してください。ただし、TP1/Message Queue 独自の属性については、事前定義クラスタセンダチャンネルのクラスタセンダチャンネル定義またはデフォルトチャンネル定義を変更することで、次の OpenTP1 開始時に反映されます。

クラスタ参加後に変更可能な属性について次の表に示します。

表 2-23 クラスタ参加後に変更が可能な属性

定義コマンド	オプション	オペランド	定義内容	備考
mqtalcccha (チャンネル定義)	-j	なし	バッチサイズ	事前定義クラスタセンダチャンネルの場合は mqtalcccha 定義コマンドの-j オプションで変更します。 自動定義クラスタセンダチャンネルの場合は mqamqtnam 定義コマンドで変更します。*1*2
	-m	maxmsg	最大メッセージ長	事前定義クラスタセンダチャンネルの場合は mqtalcccha 定義コマンドの-m maxmsg オペランドで変更します。 自動定義クラスタセンダチャンネルの場合はバッファ方式がメッセージ方式のときだけ mqttbuf 定義コマンドの-g length オペランドで変更します。*2
		maxseg	最大セグメントサイズ*2	TP1/Message Queue 独自の属性です。 事前定義クラスタセンダチャンネルの場合は mqtalcccha 定義コマンドの-m maxseg オペランドで変更します。 自動定義クラスタセンダチャンネルの場合はバッファ方式がセグメント方式のときだけ mqttbuf 定義コマンドの-g length オペランドで変更します。*2
	-g	sndbuf	送信バッファグループ番号	TP1/Message Queue 独自の属性です。

定義コマンド	オプション	オペランド	定義内容	備考
mqtalcccha (チャンネル定義)	-g	buftype	バッファ方式	TP1/Message Queue 独自の属性です。
	-e	medbuf	メッセージ編集出口編集用バッファグループ番号	TP1/Message Queue 独自の属性です。
		medcnt	メッセージ編集出口編集用バッファ数	TP1/Message Queue 独自の属性です。
	-i	なし	チャンネルの確立方法	TP1/Message Queue 独自の属性です。
	-b	bretrymcp	MQ プロトコル障害再試行の要否	TP1/Message Queue 独自の属性です。
	-r	ipaddr	自システムの IP アドレス	TP1/Message Queue 独自の属性です。
		hostname	自システムのホスト名	TP1/Message Queue 独自の属性です。
		portno	自システムのポート番号	TP1/Message Queue 独自の属性です。
		servname	自システムのサービス名	TP1/Message Queue 独自の属性です。
		portnum	自システムのポート番号使用数	TP1/Message Queue 独自の属性です。
	-v	tim1	チャンネル確立応答受信監視タイム値	TP1/Message Queue 独自の属性です。
		tim2	確認メッセージ受信監視タイム値	TP1/Message Queue 独自の属性です。
		vretry	タイムアウト時のチャンネル確立再試行の要否	TP1/Message Queue 独自の属性です。
	-t	tcpsndbuf	TCP/IP 出力用バッファサイズ	TP1/Message Queue 独自の属性です。
		tcprcvbuf	TCP/IP 入力用バッファサイズ	TP1/Message Queue 独自の属性です。
		tretrycnt	TCP/IP 確立確認回数	TP1/Message Queue 独自の属性です。
		tretryint	TCP/IP 確立確認間隔	TP1/Message Queue 独自の属性です。
	-d	cnvccsid	変換 CCSID	TP1/Message Queue 独自の属性です。

定義コマンド	オプション	オペラント	定義内容	備考
mqtalccha (チャンネル定義)	-jn	sndjnl	送信チャンネルジャーナル出力条件	TP1/Message Queue 独自の属性です。

注※1

ネゴシエーション時の自動定義クラスタセンダチャンネルのバッチサイズは mqamqtnam 定義コマンド -b オプション指定値および事前定義クラスタセンダチャンネルに指定したバッチサイズに依存します。詳細については、4章の「チャンネルのネゴシエーション」を参照してください。

事前定義クラスタセンダチャンネルのバッチサイズを変更すると、自動定義クラスタセンダチャンネルのバッチサイズが小さくなる場合があります。自動定義クラスタセンダチャンネルのバッチサイズが0になった場合、MQT サーバ開始時に「KFCA16306-E 通信構成定義に誤りがあります。理由コード=01010004」が出力され、MQT サーバは異常終了します。この場合は変更したバッチサイズを再度修正するか、mqamqtnam 定義コマンドの -b オプションの指定値を大きくしてください。

注※2

チャンネル確立時にクラスタセンダチャンネルは接続先クラスタレシーバチャンネルとネゴシエーションします。その結果、「チャンネル動作中」の状態では、チャンネル属性値は定義コマンドの指定値より小さくなる場合があります。

クラスタ参加後に TP1/Message Queue 独自の属性を変更する方法を次に示します。

1. OpenTP1 を正常終了します。
2. 事前定義クラスタセンダチャンネルまたはデフォルトチャンネル定義の属性を変更後、MQT 定義オブジェクトファイルを再作成します。
3. OpenTP1 を正常開始します。
事前定義クラスタセンダチャンネルまたは自動定義クラスタセンダチャンネルの属性がこの時点で反映されます。

(b) クラスタレシーバチャンネルの属性の変更

クラスタレシーバチャンネルの属性を変更する場合、すべての属性の変更は次回の OpenTP1 開始時に反映されます。クラスタレシーバチャンネルの属性を変更する方法を次に示します。

1. OpenTP1 を正常終了します。
2. クラスタレシーバチャンネルの属性を変更後、MQT 定義オブジェクトファイルを再作成します。
3. OpenTP1 を正常開始します。
変更した属性がこの時点で反映されます。
4. リポジトリ管理サーバ (mqrsup, mqrspp) を開始します。

5. フルリポジトリに接続する事前定義クラスタセンダチャンネルが開始していない場合は、mqststacha コマンドで開始します。

フルリポジトリにクラスタレシーバチャンネルの属性が連絡され、クラスタに参加している他キューマネージャのクラスタセンダチャンネル属性に反映されます。

(5) クラスタ使用時の障害対策

クラスタ使用時の障害について現象と原因を示します。

(a) MQRC_UNKNOWN_OBJECT_NAME のリターン

現象：

アプリケーションがクラスタキューをオープンする際に MQRC_UNKNOWN_OBJECT_NAME (2085L) がリターンされます。

原因：

オブジェクトのあるキューマネージャ、またはアプリケーションの動作するキューマネージャがクラスタに正常に参加できていません。これらのキューマネージャがクラスタ内のすべてのフルリポジトリを表示できることを確認してください。また、フルリポジトリに対応する手動定義クラスタセンダチャンネルが再試行中でないことを確認してください。該当するキューがクラスタ内にあることを確認してください。クラスタ内にキューがある場合は、適切なオープンオプションが指定されていることを確認してください。リモートのクラスタキューからはメッセージを取り出せないで、登録用のオープンオプションだけを指定できます。

(b) MQRC_CLUSTER_RESOLUTION_ERROR のリターン

現象：

アプリケーションがクラスタキューをオープンする際に MQRC_CLUSTER_RESOLUTION_ERROR (2189L) がリターンされます。

原因：

キューのオープンが初回であり、キューマネージャがフルリポジトリと通信できません。フルリポジトリに対応する手動定義クラスタセンダチャンネルが再試行中でないことを確認してください。該当するキューがクラスタ内にあることを確認してください。

クラスタ内にキューがある場合は、適切なオープンオプションが指定されていることを確認してください。リモートのクラスタキューからはメッセージを取り出せないで、登録用オプションだけを指定できます。

(c) メッセージ不到達

現象：

メッセージがあて先キューにありません。

原因：

メッセージが登録元キューマネージャに滞留しています。SYSTEM.CLUSTER.TRANSMIT.QUEUE が空であり、あて先キューマネージャへのチャンネルが動作していることを確認してください。

(d) 別名キューへのメッセージがデッドレターキューに登録される

現象：

アプリケーションが別名キューにメッセージを登録すると、別名キューのベースキューではなくデッドレターキューに登録されます。

原因：

別名キューが MQOO_BIND_NOT_FIXED ではなく MQOO_BIND_ON_OPEN でオープンされています。MQOO_BIND_ON_OPEN が指定された場合、メッセージの転送キューヘッダには、空白ではなくあて先キューマネージャ名が設定されます。

別名が定義されたキューマネージャにメッセージが到着するときに、そのキューマネージャ名が使用され、あて先キューを発見できません。そしてメッセージはデッドレターキューに登録されます。すべての別名キューの属性定義で、DefBind 属性に MQOO_BIND_NOT_FIXED を指定するよう変更するか、キューをオープンする際に MQOO_BIND_NOT_FIXED を使用してください。

(e) キューマネージャの情報が古い

現象：

クラスタ内のキューおよびチャンネルについての最新情報がキューマネージャにありません。コマンドを入力すると古いオブジェクトが表示されます。

原因：

オブジェクトのあるキューマネージャ、およびコマンドを入力したキューマネージャについて、クラスタに接続されていることを確認してください。これらのキューマネージャがクラスタ内のすべてのフルリポジトリを表示できることを確認してください。また、フルリポジトリに対応する手動定義クラスタセンダチャンネルが再試行中でないことを確認してください。

次に、フルリポジトリを相互接続するのに十分な数のクラスタセンダチャンネルがあることを確認してください。クラスタの更新は手動定義クラスタセンダチャンネルを使ってフルリポジトリ間だけで通信されます。クラスタが作成されたあとで、これらは IBM MQ 上で DEFTYPE(CLUSSDRB)チャンネルとして表示されます。手動と自動の両方のチャンネルがあるためです。すべてのフルリポジトリ間を接続するのに十分な数のチャンネルが必要です。

(6) 事前定義クラスタセンダチャンネルの接続先変更

事前定義クラスタセンダチャンネルの接続先を変更するための方法を次に示します。

また、次の手順 12 を実施するまでは、事前定義クラスタセンダチャンネルの接続先を変更する前の、フルリポジトリキューマネージャは停止させないでください。

1. アプリケーションを終了します。

2. SYSTEM.CLUSTER.TRANSMIT.QUEUE にメッセージが滞留していないことを確認してください。
3. チャンネル状態を確認して、送達未確認メッセージがないことを確認してください。
4. OpenTP1 を正常終了します。
5. 変更したいフルリポジトリキューマネージャへの事前定義クラスタセンダチャンネル定義を作成してください。
6. MQT 定義オブジェクトファイルを作成してください。
7. OpenTP1 を正常開始してください。
8. リポジトリ管理サーバ (mqrsup, mqrspp) を開始してください。
9. クラスタレシーバチャンネルの状態が DISABLED 以外の場合、mqtstpcha コマンドでクラスタレシーバチャンネルを終了してください。クラスタに関係するすべてのクラスタレシーバチャンネルを終了させ、定義種別が MULTINST となっているチャンネルの状態が DISABLED となっていることを確認してください。
10. クラスタセンダチャンネルの状態が DISABLED の場合、mqtstacha コマンドでクラスタセンダチャンネルを開始してください。
11. mqrremove -q -c クラスタ名を実行してください。
KFCA31526-I および KFCA31527-I が出力されることを確認してください。KFCA31526-I が出力されない場合、コマンドが異常終了しているおそれがあります。原因を取り除き、再度 mqrremove コマンドを実行してください。KFCA31527-I が出力されない場合、リポジトリ管理サーバが開始していることを確認してください。
12. SYSTEM.CLUSTER.TRANSMIT.QUEUE にメッセージが滞留していないことを確認してください。
13. 事前定義クラスタセンダチャンネルを変更した後の接続先フルリポジトリキューマネージャを用意してください。
14. mqrremove -s -c クラスタ名を実行し、クラスタに再参加してください。
KFCA31526-I が出力されない場合、コマンドが異常終了しているおそれがあります。原因を取り除き、再度 mqrremove コマンドを実行してください。KFCA31527-I が出力されない場合、リポジトリ管理サーバが開始していることを確認してください。
15. フルリポジトリキューマネージャへ接続するクラスタセンダチャンネルの状態が DISABLED の場合、mqtstacha コマンドで開始してください。
16. クラスタレシーバチャンネルの状態が DISABLED の場合、mqtstacha コマンドでクラスタレシーバチャンネルを開始してください。
17. mqrpls -c を実行してください。
参加状態が JOIN、表示された参加時使用チャンネル名が手順 5 で作成したチャンネル名となっていることを確認してください。

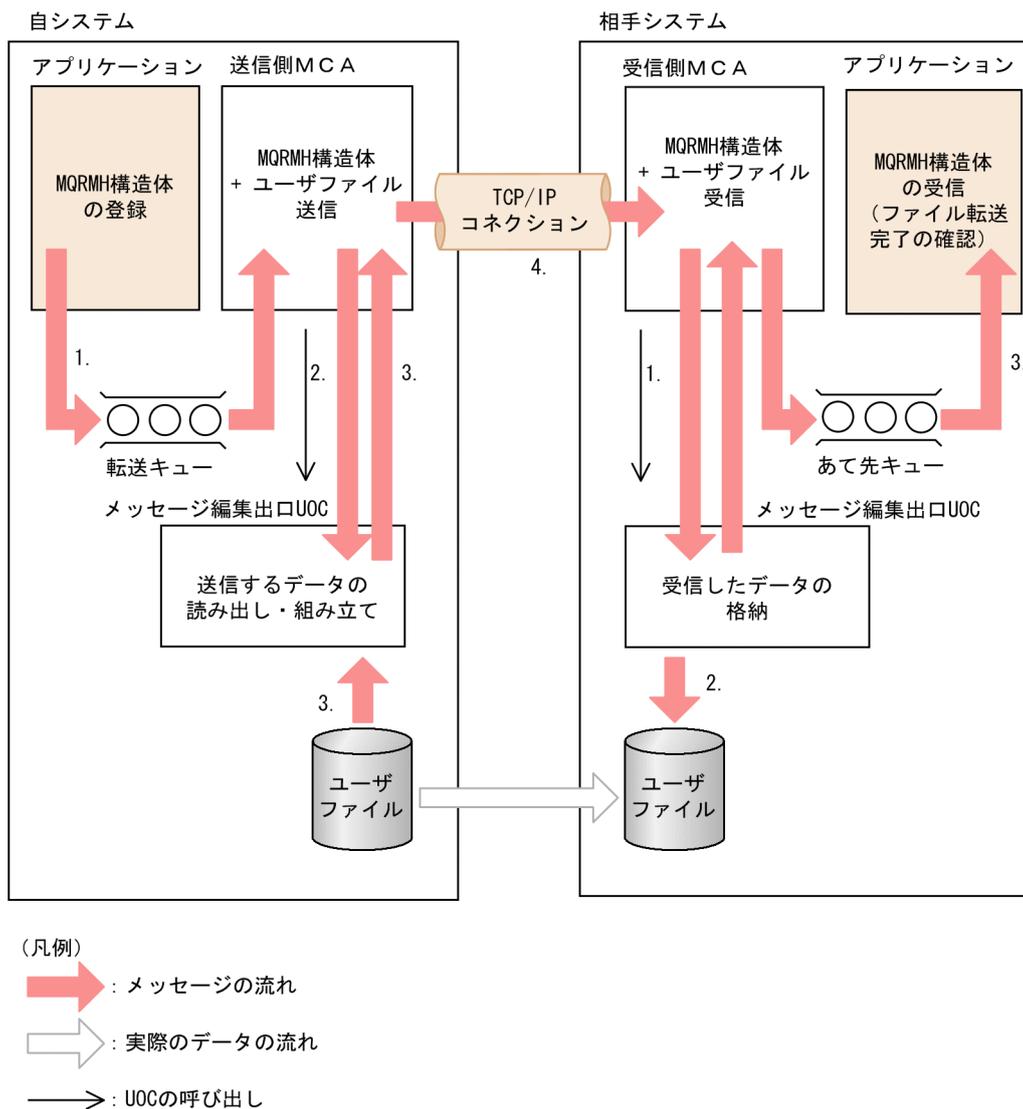
2.9 メッセージ編集出口 UOC を使用したファイル転送

メッセージ編集出口 UOC を使用することで、データを TP1/Message Queue のキューに登録しないでユーザファイルから直接、転送できます。この方法は、大量のデータが MQ システムのメッセージ以外の形式で存在する場合に便利です。

2.9.1 処理の概要

メッセージ編集出口 UOC を使用したファイル転送処理の概要について、次の図に示します。

図 2-62 メッセージ編集出口 UOC を使用したファイル転送処理の概要



送信側システムでの処理の流れ

1. UAP は、MQRMH (Message reference header) 構造体をメッセージとして登録します。

MQRMH 構造体については、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

- 2.MCA は、メッセージ編集出口 UOC を呼び出し、メッセージを渡します。
- 3.メッセージ編集出口 UOC は、MQRMH 構造体の情報を基に、転送するデータを読み出して MCA の送信／編集バッファに登録します。
- 4.MCA は、MQRMH 構造体、および送信／編集バッファのデータを相手システムに送信します。

受信側システムでの処理の流れ

- 1.MCA は、メッセージ編集出口 UOC を呼び出し、受信したメッセージを渡します。
- 2.メッセージ編集出口 UOC は、メッセージ中の MQRMH 構造体の情報を基に、転送されたデータを格納します。
- 3.UAP は、MQRMH 構造体を受信し、ファイル転送の完了を確認します。

2.9.2 使用方法

メッセージ編集出口 UOC を使用したファイル転送を使用するには、送信側システム、受信側システムの両方でメッセージ編集出口 UOC を指定する必要があります。メッセージ編集出口 UOC の使用方法については、「3. ユーザオウンコーディング」を参照してください。

メッセージ編集出口 UOC は、チャンネル定義のバッファ方式指定にメッセージ方式を指定（TCP 定義の mqtalccha 定義コマンドの-g buftype オペランドに msg を指定）した場合だけ有効です。バッファ方式にセグメント方式を指定（TCP 定義の mqtalccha 定義コマンドの-g buftype オペランドに seg を指定）した場合、UOC が呼び出されません。

(1) 注意事項

- 1.送信側メッセージ編集出口 UOC が参照メッセージに付けられるオブジェクトデータの長さは、次に示す計算式を満たす必要があります。

チャンネルの最大メッセージ長 \geq MQXQH構造体を除く転送メッセージ長

- 2.MCA、およびメッセージ編集出口 UOC は、OpenTP1 管理者のユーザ ID およびグループ ID の下で実行されます。そのため、メッセージ編集出口 UOC を使用してファイルを転送する場合は、OpenTP1 管理者のユーザ ID およびグループ ID の権限を、転送するファイルおよびディレクトリに設定してください。
- 3.受信側のメッセージ編集出口 UOC が受信したファイルを作成する作業は、MCA の作業単位と同期しません。このため、バッチがバックアウトされた場合、MQRMH 構造体が到着していないのにファイルが受信側に存在している状態になることがあります。この場合、ファイル転送は完了していません。
- 4.参照メッセージを配布リストとして書き込む場合、結果として得られる配布リスト、またはそのノードのあて先ごとにオブジェクトが取り出せるようにする必要があります。このため、ファイルの使用回数を考慮する必要があります。

また、あるあて先に対しては中間ノードであり、別のあて先に対しては最終ノードであるようなノードも存在します。このようなノードについてもファイルの使用回数を考慮して使用してください。

5. 転送するファイルデータは、送信側でのデータ変換が実行されません。

ただし、参照メッセージが中間ノードを経由する場合、中間ノードの転送キューに登録されたファイルデータが送信されるときに、データ変換を実行できます。

2.10 MQC サーバ機能

この節では、関連製品である TP1/Message Queue Access の使用時に必要な、MQC サーバ機能の定義、運用、および障害対策について説明します。

2.10.1 MQC サーバ機能のセットアップ

MQC サーバ機能を構成するファイルは、TP1/Message Queue に添付されています。MQC サーバ機能のセットアップは、TP1/Message Queue のセットアップ手順に従ってください。

MQC サーバ機能のインストール時に作成されるファイルを次の表に示します。

表 2-24 MQC サーバ機能のインストール時に作成されるファイル

名称	ディレクトリ	ファイル名 (AIX の場合)	ファイル名 (HP-UX 11i V3 (IPF) 以降の場合)	ファイル名 (Linux の場合)	ファイル名 (Windows の場合)
MQC リスナサーバ	\$DCDIR/lib/servers	mqcdtcp	mqcdtcp	mqcdtcp	mqcdtcp.exe
コマンド	\$DCDIR/bin	mqcmkgwp	mqcmkgwp	mqcmkgwp	mqcmkgwp.bat
定義解析用ファイル	\$DCDIR/lib/sysconf	mqctcp	mqctcp	mqctcp	mqctcp
	\$DCDIR/lib/sysdef	mqc.def	mqc.def	mqc.def	mqc.def
ライブラリ	\$DCDIR/lib	libmqcs.a	libmqcs.so libmqcs.a	libmqcs.so libmqcs.a	libmqcs.lib
	\$DCDIR/bin	—	—	—	libmqcs.dll
MQC ゲートウェイサーバ	\$DCDIR/lib	mqcgwp.o libmqcgwp.a	mqcgwp.o libmqcgwp.so libmqcgwp.a	mqcgwp.o libmqcgwp.so libmqcgwp.a	libmqcgwp.lib
	\$DCDIR/bin	—	—	—	libmqcgwp.dll

(凡例)

—：該当しません。

2.10.2 MQC サーバ機能の環境設定

MQC サーバ機能の環境設定について説明します。

(1) MQC サーバ機能の定義

ユーザは、MQC サーバ機能の定義について、OS のテキストエディタを使用して作成および更新する必要があります。MQC サーバ機能の定義および定義ファイル名を次に示します。詳細については、「4. 準備作業」を参照してください。

- MQC サービス定義
\$DCCONFPATH/mqc
- MQC ゲートウェイサーバユーザサービス定義
\$DCCONFPATH/MQC ゲートウェイサーバ名
- MQA サービス定義
\$DCCONFPATH/mqa
- トランザクションサービス定義
\$DCCONFPATH/trn

(2) MQC ゲートウェイサーバの作成

MQC ゲートウェイサーバは、OpenTP1 システムのユーザサーバです。

mqcmsgwp コマンドを実行して、実行形式プログラム mqcmsgwp と MQC ゲートウェイサーバユーザサービス定義ファイル mqcmsgwp を作成してください。

mqcmsgwp コマンド実行時には、次の定義ファイルが必要です。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

- \$DCCONFPATH/betranrc
- \$DCCONFPATH/trn

なお、作成した mqcmsgwp は、次に示すディレクトリに配置されます。

- \$DCDIR/aplib : 実行形式プログラム
- \$DCDIR/conf : MQC ゲートウェイサーバユーザサービス定義ファイル

mqcmsgwp コマンドの延長で OpenTP1 に RM 登録をしています。したがって、dcsetup (-d の y 応答) *コマンドを実行し直した場合は、再度 mqcmsgwp コマンドも実行する必要があります。

注※

Windows 版については、インストーラでアンインストール後、再インストールをした場合となります。

(3) MQC ゲートウェイサーバの環境設定での注意事項

MQC ゲートウェイサーバの環境を設定する上での注意事項について説明します。

(a) MQC サーバ機能が動作する OpenTP1 のトランザクションサービス定義

\$DCCONFPATH/trn に次の設定が行われている場合、MQC ゲートウェイサーバが異常終了（アポートコード：Mqcgw03）します。

トランザクションサービス定義

```
set trn_rm_open_close_scope=transaction
```

このため、MQC ゲートウェイサービス定義に次の定義を追加してください。

MQC ゲートウェイサービス定義

```
set trn_rm_open_close_scope=process
```

(b) MQC ゲートウェイサーバのユーザサービス定義

MQC ゲートウェイサーバのユーザサービス定義は、OpenTP1 のユーザサービスデフォルト定義のデフォルト値で動作するように設定してください。

2.10.3 MQC サーバ機能の運用

MQC サーバ機能は、MQC リスナサーバと MQC ゲートウェイサーバから構成されます。ここでは、各サーバの開始と終了について説明します。

なお、MQC サーバ機能の運用コマンドについては、「6. システムの運用」を参照してください。

(1) MQC リスナサーバの開始と終了

MQC リスナサーバの開始と終了について説明します。

(a) MQC リスナサーバの開始

MQC リスナサーバは、MQA サーバと同時に開始します。

MQA サーバと同時に開始させるためには、MQA サービス定義の mqa_mqc_conf オペランドに Y を指定しておく必要があります。ここで Y を指定すると MQC リスナサーバが MQA サーバと同時に開始されます。

■ 正常開始

MQC リスナサーバは、OpenTP1 システムの正常開始の延長で開始します。

■ 再開始

OpenTP1 システムに障害が発生した場合、MQC リスナサーバは OpenTP1 システムの再開始の延長で再開始します。再開始すると前回の終了時点の状態に回復されます。

メッセージ登録中および削除中などのキューアクセス中に強制停止した場合は、キューファイルの内容が矛盾することがあるため、必ず再開してください。

また、MQC リスナサーバではクライアント機能と MQC サーバ機能のトランザクションを管理するために、トランザクション ID などの情報をジャーナルに取得し再開時に回復しています。このため、強制停止やシステムダウンのあとに正常開始したときは、ジャーナルによって回復されないため、TP1/Message Queue との間に矛盾が発生する可能性があります。

(b) MQC リスナサーバの終了

MQC リスナサーバの終了モードを次に示します。

- 正常終了
- 計画停止 A
- 計画停止 B
- 強制停止

終了モードの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

正常終了、計画停止 A、または計画停止 B の場合、MQC クライアント機能の切断を時間監視できます。

終了処理監視タイマ値は、MQC サービス定義の mqcenv 定義コマンドで指定します。終了処理監視でタイムアウトが発生した場合、送受信中の処理を中断し、MQC クライアント機能との接続は強制解放されます。

(2) MQC ゲートウェイサーバの開始と終了

MQC ゲートウェイサーバの開始と終了について説明します。

(a) MQC ゲートウェイサーバの開始

MQC クライアント機能のクライアントアプリケーションは、OpenTP1 システムに登録されていない UAP です。MQI を使用するためには、MQC サーバ機能がある OpenTP1 システムで、MQC ゲートウェイサーバをユーザサーバとして開始する必要があります。

ユーザサーバの開始については、マニュアル「OpenTP1 運用と操作」を参照してください。

なお、MQC ゲートウェイサーバを開始しない場合は、MQC クライアント機能が提供するライブラリ関数を使用できません。

(b) MQC ゲートウェイサーバの終了

MQC ゲートウェイサーバは、OpenTP1 のユーザサーバです。

3

ユーザOWNコーディング

この章では、ユーザOWNコーディングの使用方法や作成例について説明します。

3.1 ユーザOWNコーディング

TP1/Message Queue の機能を、より多様な業務に対応させるために補助するプログラムを、ユーザOWNコーディング (UOC) といいます。TP1/Message Queue で使用できる UOC を次に示します。

- メッセージ編集出口 UOC
メッセージのデータ変換や、参照メッセージを処理する UOC です。

3.2 メッセージ編集出口 UOC

メッセージ編集出口 UOC は、メッセージのデータ変換や参照メッセージを処理する UOC です。また、クラスタセンダチャンネルとクラスタレシーバチャンネルでは、チャンネルの開始時にチャンネル属性を変更できます。

ここでは、メッセージ編集出口 UOC の開始条件、および渡されるデータ形式について説明します。

3.2.1 UOC の開始条件

送信側メッセージ編集出口 UOC は、送信側 MCA によって転送キューからメッセージが取り出された直後に、呼び出されます。

受信側メッセージ編集出口 UOC は、受信側 MCA によってメッセージがあて先キューに登録される直前に、呼び出されます。

なお、メッセージ編集出口 UOC は、該当するチャンネルのバッファ方式にメッセージ方式を指定（TCP 定義の mqtalcccha 定義コマンドの -g buftype オペランドに msg を指定）した場合だけ、呼び出されます。チャンネルのバッファ方式にセグメント方式を指定した場合、メッセージ編集出口 UOC は呼び出されません。

3.2.2 UOC に渡されるデータ形式

MCA は UOC の起動時に、メッセージが格納されている送信バッファ、受信バッファ、および TCP 定義の mqtalcccha 定義コマンドの -e medbuf オペランドで指定した編集バッファを渡します。UOC では、これらのバッファを使用してメッセージの編集ができます。

メッセージ編集出口 UOC に渡されるデータの形式について、次の図に示します。

図 3-1 メッセージ編集出口 UOC に渡されるデータの形式



MCA は、メッセージ編集出口 UOC の呼び出し時、転送キューに格納されているメッセージを送受信バッファに格納します。メッセージ編集出口 UOC では、送受信バッファの使用バッファ長から、該当セグメントの長さを知ることができます。

メッセージ編集出口 UOC でデータ変換や参照メッセージ処理を実施した場合、送受信バッファの使用バッファ長を変更できます。ただし、使用バッファ長が、相手システムとの間で最適な値として決定された最

大メッセージ長を超えないようにしてください。最大メッセージ長は、チャンネルデータ定義ブロック (dcmtcq_uoc_mqcd) の最大メッセージ長 (MaxMsgLength) で確認してください。

また、メッセージ記述子 (MQMD 構造体) を格納している転送キューヘッダ (MQXQH 構造体) については、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

3.3 UOC とのインタフェース

UOC とのインタフェースについて説明します。

3.3.1 UOC の呼び出し形式

UOC の呼び出し形式について、次に示します。

<メッセージ編集出口 UOC >

- ANSI C, C++言語の場合

```
#include <cmqc.h>
#include <dcmtcquo.h>
long med_uoc01(dcmtcq_uoc_parmlist *parm)
```

- K&R 版 C 言語の場合

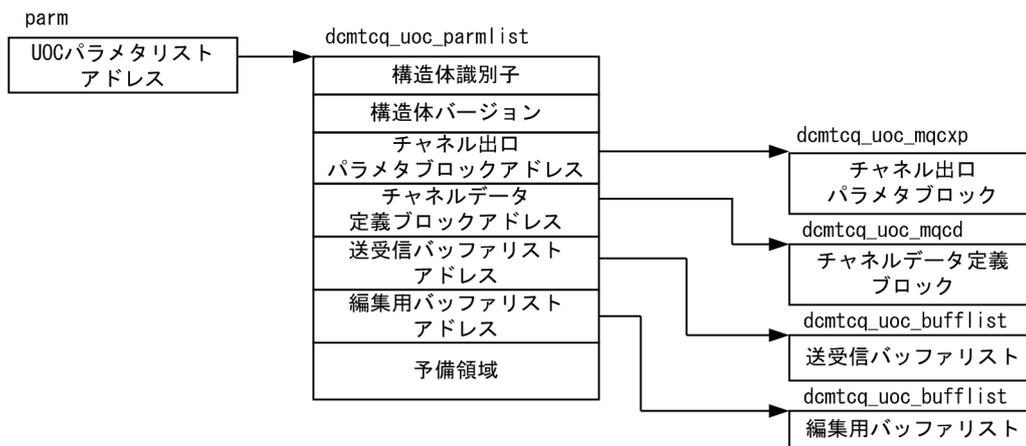
```
#include <cmqc.h>
#include <dcmtcquo.h>
long med_uoc01(parm)

dcmtcq_uoc_parmlist *parm;
```

3.3.2 パラメタ

MCA は、次の図に示すパラメタをパラメタリストアドレス (parm) に設定し、UOC の呼び出し時に渡します。

図 3-2 UOC に渡されるパラメタの形式



各パラメタの内容について説明します。

各領域の後ろの括弧は、その領域のデータタイプです。データタイプについては、マニュアル「TP1/Message Queue プログラム作成リファレンス」を参照してください。

データタイプの次に、その領域の「向き」を示します。つまり、その領域を何が使用し、何が設定するかです。

input

MQT から UOC へ、値が入力される領域です。ユーザは値を変更しないでください。

output

UOC から MQT へ、値を出力するための領域です。

(1) UOC パラメタリスト (dcmtcq_uoc_parmlist)

UOC パラメタリストは、UOC に渡されるパラメタを管理しています。

dcmtcq_uoc_parmlist の宣言形式

```
typedef struct {
    MQCHAR4          StructId;          /* 構造体識別子 */
    MQLONG           Version;          /* 構造体バージョン番号 */
    dcmtcq_uoc_mqcxp *mqcxp_adr;      /* チャンネル出口パラメタブロックアドレス */
    dcmtcq_uoc_mqcd  *mqcd_adr;        /* チャンネルデータ定義ブロックアドレス */
    dcmtcq_uoc_bufflist *tbuflist_adr; /* 送受信バッファリストアドレス */
    dcmtcq_uoc_bufflist *ebuflist_adr; /* 編集バッファリストアドレス */
    MQCHAR           Reserve1[8];      /* 予備 */
} dcmtcq_uoc_parmlist;
```

注

64 ビット版の場合は、次の形式になります。

```
    :
    MQCHAR           Reserve1[24];      /* 予備 */
} dcmtcq_uoc_parmlist;
```

- StructId (MQCHAR4 型) –input
構造体識別子 (DCMTCQ_MQCXPL_STRUC_ID)
UOC パラメタリストのための識別子が格納されます。
定数 DCMTCQ_MQCXPL_STRUC_ID_ARRAY は、DCMTCQ_MQCXPL_STRUC_ID と同じ値です。ただし、前者は文字配列、後者は文字列です。
- Version (MQLONG 型) –input
構造体バージョン番号 (32 ビット版の場合：DCMTCQ_MQCXPL_VERSION_1, 64 ビット版の場合：DCMTCQ_MQCXPL_VERSION_2)
UOC パラメタリストのためのバージョン番号が格納されます。

- mqcxp_adr (dcmtcq_uoc_mqcxp ポインタ型) -input
チャンネル出口パラメタブロックのアドレス
チャンネル出口パラメタブロックについては、「3.3.2(2) チャンネル出口パラメタブロック (dcmtcq_uoc_mqcxp)」を参照してください。
- mqcd_adr (dcmtcq_uoc_mqcd ポインタ型) -input
チャンネルデータ定義ブロックのアドレス
チャンネルデータ定義ブロックについては、「3.3.2(3) チャンネルデータ定義ブロック (dcmtcq_uoc_mqcd)」を参照してください。
- tbuflist_adr (dcmtcq_uoc_bufplist ポインタ型) -input
送受信バッファリストのアドレス
送受信バッファリストについては、「3.3.2(4) バッファリスト (dcmtcq_uoc_bufplist)」を参照してください。
出口への ExitReason が DCMTCQ_MQXR_ALT の場合、このアドレスは設定されません。
- ebuflist_adr (dcmtcq_uoc_bufplist ポインタ型) -input
編集用バッファリストのアドレス
編集用バッファリストについては、「3.3.2(4) バッファリスト (dcmtcq_uoc_bufplist)」を参照してください。
出口への ExitReason が DCMTCQ_MQXR_ALT の場合、このアドレスは設定されません。

(2) チャンネル出口パラメタブロック (dcmtcq_uoc_mqcxp)

チャンネル出口パラメタブロックは、UOC の処理を決定したり、UOC が MCA の動作を制御したりするために使用されます。

チャンネル出口パラメタブロックには、UOC への入力領域と UOC からの出力領域が両方とも存在するので注意してください。

dcmtcq_uoc_mqcxp の宣言形式

```
typedef struct {
    MQCHAR4   StructId;           /* 構造体識別子 */
    MQLONG    Version;           /* 構造体バージョン番号 */
    MQLONG    ExitId;           /* 出口のタイプ */
    MQLONG    ExitReason;       /* 出口を呼び出すための理由 */
    MQLONG    ExitReason2;      /* 出口を呼び出すための二次理由 */
    MQLONG    ExitResponse;     /* 出口からの応答 */
    MQLONG    ExitResponse2;    /* 出口からの二次応答 */
    MQLONG    Feedback;        /* フィードバックコード */
    MQLONG    MaxSegmentLength; /* セグメント最大長 */
    MQCHAR    Reserve1[4];     /* 予約領域 */
    MQBYTE16  ExitUserArea;     /* 出口ユーザ領域 */
    MQCHAR32  ExitData;        /* 出口データ */
    MQCHAR    PartnerName[52]; /* 相手システムのキューマネージャ名 */
    MQLONG    FAPLevel;        /* プロトコルレベル */
}
```

```

MQLONG    CapabilityFlags; /* 機能フラグ          */
} dcmtcq_uoc_mqcxp;

```

注

64 ビット版の場合は、次の形式になります。

```

      :
MQLONG    CapabilityFlags; /* 機能フラグ          */
MQCHAR    Reserve1[8];    /* 予備                */
} dcmtcq_uoc_mqcxp;

```

- StructId (MQCHAR4 型) -input
構造体識別子 (DCMTCQ_MQCXP_STRUC_ID)
チャンネル出口パラメタブロックのための識別子が格納されます。
定数 DCMTCQ_MQCXP_STRUC_ID_ARRAY は、DCMTCQ_MQCXP_STRUC_ID と同じ値です。ただし、前者は文字配列、後者は文字列です。
- Version (MQLONG 型) -input
構造体バージョン番号 (32 ビット版の場合: DCMTCQ_MQCXP_VERSION_2, 64 ビット版の場合: DCMTCQ_MQCXP_VERSION_3)
チャンネル出口パラメタブロックのためのバージョン番号 (バージョン 2) が格納されます。バージョン 2 の構造体だけに存在する領域については、各領域の説明で示します。
なお、新しいバージョンの構造体では、旧バージョンの領域名、および属性は変わりません。したがって、Version の値が UOC に必要な領域を含む最低バージョンの番号以上かどうか、UOC 側で検査する必要があります。
- ExitId (MQLONG 型) -input
出口のタイプ
呼び出される UOC のタイプが格納されます。
 - DCMTCQ_MQXT_CHANNEL_MSGEDT_EXIT
メッセージ編集出口 UOC を示します。
- ExitReason (MQLONG 型) -input
出口を呼び出すための理由
UOC が呼び出された理由が格納されます。
なお、各理由で UOC が呼び出されるタイミングについては、「[3.4 UOC 呼び出しの流れ](#)」を参照してください。
 - DCMTCQ_MQXR_ALT
属性変更
開始するチャンネルが使用するチャンネル属性を変更する必要がある場合に、変更するための契機を出口に与えるために呼び出されます。
この呼び出しはクラスタチャンネル (クラスタセンダおよびクラスタレシーバ) にだけ実行されます。
 - DCMTCQ_MQXR_INIT

初期化

UOC が使用する作業領域の確保、およびイニシャライズを実行するために呼び出されたことを示します。

- DCMTCQ_MQXR_TERM

終了

UOC が使用した作業領域の解放のために呼び出されたことを示します。

- DCMTCQ_MQXR_MSG

メッセージ処理

メッセージ処理のために呼び出されたことを示します。

- ExitReason2 (MQLONG 型) -input

出口を呼び出すための二次理由

UOC がメッセージ処理のために呼び出された場合 (ExitReason=DCMTCQ_MQXR_MSG), 呼び出された理由の詳細が格納されます。メッセージ編集出口 UOC では、この値は意味を持ちません。

- ExitResponse (MQLONG 型) -output

出口からの応答

UOC が、MCA への応答を格納する領域です。

- DCMTCQ_MQXCC_OK

正常に続行することを MCA に依頼します。

UOC がメッセージ処理のために呼び出された場合 (ExitReason=DCMTCQ_MQXR_MSG) は、ExitResponse2 領域に追加情報を設定してください。

- DCMTCQ_MQXCC_SUPPRESS_FUNCTION

メッセージをデッドレターキューに登録することを MCA に依頼します。

この値は、UOC がメッセージ処理のために呼び出された場合 (ExitReason=DCMTCQ_MQXR_MSG) に有効です。

デッドレターキューに登録できない場合、送信側の処理中メッセージは転送キューに格納され、受信側の処理中メッセージは破棄され、チャンネルが終了されます。

また、メッセージをデッドレターキューに登録する場合は、次に示す二つの領域を設定してください。

- Feedback 領域

- ExitResponse2 領域

- DCMTCQ_MQXCC_SUPPRESS_EXIT

UOC の呼び出しを抑止することを MCA に依頼します。

MCA は、チャンネル終了時 (ExitReason=DCMTCQ_MQXR_TERM) まで、UOC の呼び出しを抑止します。ただし、属性変更時 (ExitReason=DCMTCQ_MQXR_ALT) にこの応答が返された場合、チャンネル終了時 (ExitReason=DCMTCQ_MQXR_TERM) にも、UOC を呼び出しません。

UOC がメッセージ処理のために呼び出された場合 (ExitReason=DCMTCQ_MQXR_MSG) は、ExitResponse2 領域に追加情報を設定してください。

- DCMTCQ_MQXCC_CLOSE_CHANNEL
チャンネルの終了を MCA に依頼します。
送信側の処理中メッセージは転送キューに格納され、受信側の処理中メッセージは破棄され、チャンネルが終了されます。
- ExitResponse2 (MQLONG 型) -output
出口からの二次応答
UOC がメッセージ処理のために呼び出された場合 (ExitReason=DCMTCQ_MQXR_MSG), UOC から MCA へのメッセージの返却領域を指定する情報を格納します。
 - DCMTCQ_MQXR2_USE_TRAN_BUFFER
返却するメッセージが送受信バッファに格納されていることを示します。
 - DCMTCQ_MQXR2_USE_EDIT_BUFFER
返却するメッセージが編集用バッファに格納されていることを示します。
 DCMTCQ_MQXR2_USE_TRAN_BUFFER と DCMTCQ_MQXR2_USE_EDIT_BUFFER は、どちらか一方を指定できます。
- Feedback (MQLONG 型) -output
フィードバックコード
UOC がメッセージ処理のために呼び出された場合 (ExitReason=DCMTCQ_MQXR_MSG), メッセージがデッドレターキューに登録される理由を識別するためのフィードバックコードを格納します。
 - MQFB_STOPPED_BY_MSG_EXIT
メッセージが UOC によって止められたことを示します。
- MaxSegmentLength (MQLONG 型) -input
セグメント最大長
単一伝送で送信できる最大バイト長が格納されます。
セグメント最大長は、通信構成定義で指定されたセグメント長を基に、チャンネル確立時に MCA 間のネゴシエーションによって決定された値です。
ExitReason が DCMTCQ_MQXR_INIT または DCMTCQ_MQXR_ALT の場合、この値は意味を持ちません。
- ExitUserArea (MQBYTE16 型) -input/output
出口ユーザ領域
UOC が自由に利用できる領域です。
出口ユーザ領域は、UOC が最初に呼び出された時 (ExitReason=DCMTCQ_MQXR_ALT または DCMTCQ_MQXR_INIT), 2 進数の 0 で初期化されています。UOC 内で出口ユーザ領域を更新すると、その値は次の呼び出しまで保持されます。ただし、再度チャンネルが開始され、UOC を属性更新指示または初期化指示 (ExitReason=DCMTCQ_MQXR_ALT または DCMTCQ_MQXR_INIT) で呼び出す時、出口ユーザ領域は初期化されます。
- ExitData (MQCHAR*32 型) -input
出口データ

mqtalcccha 定義コマンドの-x metexdata オペランドで指定された、メッセージ編集出口ユーザデータが格納されます。

ただし、チャンネルタイプがクラスタセンダの場合でリポジトリ情報を取得している時は、接続相手のクラスタレシーバの定義で指定されているユーザデータ（データ変換なし）が格納されます。

この領域の長さは、DCMTCQ_MQ_EXIT_DATA_LENGTH から取得できます。ユーザデータの長さが領域の長さよりも短かった場合、ユーザデータの後ろにスペースが補われます。また、指定を省略した場合、領域はスペースで埋められます。

- PartnerName (MQCHAR*52 型) -input

相手システムのキューマネージャ名

チャンネルの接続先になる相手システムのキューマネージャ名です。

コーラ側で ExitReason が DCMTCQ_MQXR_ALT または DCMTCQ_MQXR_INIT である場合は、このフィールドの値は意味を持ちません。これは、チャンネルの接続が完了するまで相手システムのキューマネージャ名を特定できないからです。

領域は名称の後ろに¥0 が付けられ、定義で指定できるキューマネージャ名の最大サイズより 4 バイト大きくなっているため注意が必要です。

この領域は、バージョン番号 2 (Version=DCMTCQ_MQCXP_VERSION_2) 以降の構造体で使用できます。

- FAPLevel (MQLONG 型) -input

プロトコルレベル

接続先とのネゴシエーションで決定された、このチャンネルで使用する通信プロトコル (FAP) のレベルです。

DCMTCQ_MQFL_FPL: プロトコルレベル 1

DCMTCQ_MQFL_FPL4: プロトコルレベル 4

コーラ側で ExitReason が DCMTCQ_MQXR_ALT または DCMTCQ_MQXR_INIT である場合は、このフィールドの値は意味を持ちません。これは、チャンネルの確立が完了するまで相手システムとの間で使用する FAP レベルが決定しないからです。

この領域は、バージョン番号 2 (Version=DCMTCQ_MQCXP_VERSION_2) 以降の構造体で使用できます。

- CapabilityFlags (MQLONG 型) -input

機能フラグ

相手システムとの間で使用する機能によって、次に示す値が定義されます。

- DCMTCQ_MQCF_NONE

フラグなし

- DCMTCQ_MQCF_DIST_LIST

マルチキャスト機能を使用することを示します。

コーラ側で ExitReason が DCMTCQ_MQXR_ALT または DCMTCQ_MQXR_INIT である場合は、このフィールドの値は意味を持ちません。これは、チャンネルの確立が完了するまで相手システムとの間で使用する機能が決定しないからです。

この領域は、バージョン番号 2 (Version=DCMTCQ_MQCXP_VERSION_2) 以降の構造体で使用できます。

(3) チャンネルデータ定義ブロック (dcmtcq_uoc_mqcd)

チャンネルデータ定義構造体はチャンネルの属性を示すパラメータを持ちます。各領域の内容は特に記述のないかぎり、チャンネル定義の内容と同じです (ここでは、TCP 定義の mqtalcccha 定義コマンドの該当するオプションを [] で示します)。また、名称を格納する領域は後ろに¥0 を格納し、定義で指定できるサイズより大きくなるので注意してください。

また、このパラメータのすべての領域は出口への ExitReason が DCMTCQ_MQXR_ALT の場合を除き出口の入出力領域です。ExitReason が DCMTCQ_MQXR_ALT でない場合に UOC が更新すると、その後のチャンネルの動作は保証しません。

DCMTCQ_MQXR_ALT の場合に更新できる領域については次の表を参照してください。UOC が更新した値が各領域の指定としては不正であるときは、更新された値を無視して変更前の値でチャンネル処理を行います。

表 3-1 チャンネルタイプと有効領域

属性	sender	server	receiver	requester	cluster	clusrcvr	連絡属性 ※1	自動定義のクラスタセンダの仮定値	注意事項
ChannelName	●	●	●	●	●	●	あり	—	—
Version	●	●	●	●	●	●	—	DCMTCQ_MQCD_VERSION_7	—
ChannelType	●	●	●	●	●	●	あり	—	—
ClusterName	—	—	—	—	●	●	あり	—	—
BatchSize	●	●	●	●	○	○	あり	—	事前定義クラスタセンダチャンネルの場合は、定義値よりも大きくすることはできません。 自動定義クラスタセンダチャンネルの場合は、計算値※2よりも大きくすることはできません。
XmitQName	●	●	—	—	●	●	あり	—	クラスタチャンネルは SYSTEM.CLUSTER.TRANSMIT.QUEUE 固定です。

属性	sender	server	receiver	requester	cluster	cluster	連絡属性 ※1	自動定義のクラスタセンダの仮定値	注意事項
MaxMsgLength	●	●	●	●	○	○	あり	—	BufType がメッセージ方式 (DCMTCQ_MQB_UFT_MSG) の場合は、メッセージ送受信バッファのバッファ長よりも大きくすることはできません。
MaxSegLength	●	●	●	●	○	○	—	8192	BufType がセグメント方式 (DCMTCQ_MQB_UFT_SEG) の場合は、メッセージ送受信バッファのバッファ長よりも大きくすることはできません。
SeqNumberWrap	●	●	●	●	○	○	あり	—	—
BufType	●	●	●	●	●	●	—	msg	—
SendBufGroup	●	●	—	—	○	—	—	1	—
ReceiveBufGroup	—	—	●	●	—	○	—	—	—
MsgEdtBufGroup	●	●	●	●	○	○	—	—	—
MsgEdtBufCount	●	●	●	●	○	○	—	—	—
StartType	●	●	●	●	●	●	—	auto	—
LongRetryType	●	●	—	●	○	●	あり	—	—
LongRetryCount	●	●	—	●	○	●	あり	—	—
LongRetryInterval	●	●	—	●	○	●	あり	—	—
OwnIPAddr	●	●	—	●	○	●	あり ※3	—	両方指定された場合 OwnIPAddr 指定値が優先されます。
OwnHostName	●	●	—	●	○	●	あり ※3	—	

属性	sender	server	receiver	requester	clusdr	clusrcvr	連絡属性 ※1	自動定義のクラスタセンダの仮定値	注意事項
OwnPortNum	●	●	—	●	○	●	あり ※3	—	両方指定された場合 OwnPortNum 指定値が優先されます。
OwnServName	●	●	—	●	○	●	—	—	
PartnerIPAddr	●	●	—	●	○	—	あり	—	両方指定された場合 PartnerIPAddr 指定値が優先されます。 事前定義クラスタセンダだけ変更できます。
PartnerHostName	●	●	—	●	○	—	あり	—	
PartnerPortNum	●	●	—	●	○	—	あり	—	両方指定された場合 PartnerPortNum 指定値が優先されます。 clusdr では PartnerServName はポート番号で相手システムへ連絡します。 事前定義クラスタセンダだけ変更できます。
PartnerServName	●	●	—	●	○	—	—	—	
ConRspRecvInterval	●	●	—	●	○	—	—	30	—
RspMsgRecvInterval	●	●	—	—	○	—	—	30	—
NextSegRecvInterval	—	—	●	●	—	○	—	—	—
DiscInterval	●	●	—	—	—	—	—	—	転送キュー監視方式がタイマ方式の場合に有効です。
DiscIntervalCount	●	●	—	—	—	—	—	—	転送キュー監視方式がタイマ方式の場合に有効です。

属性	sender	server	receiver	requester	clusdr	clusrcvr	連絡属性※1	自動定義のクラスタセンダの仮定値	注意事項
DiscInterval2	●	●	—	—	○	●	あり	—	転送キュー監視方式がイベント方式の場合に有効です。 またはクラスタチャネルで有効です。
DiscInterval2Effect	●	●	—	—	●	●	あり	—	
TimeoutConRetry	●	●	—	●	○	—	—	YES	—
NextMsgRecvInterval	—	—	●	●	—	○	—	—	—
HeartbeatInterval	●	●	●	●	○	○	あり	—	—
BatchInterval	●	●	—	—	○	●	あり	—	—
MsgEdtUserData	●	●	●	●	○	○	あり	—	バッファ方式がメッセージ方式の場合に有効です。
SendUserData	—	—	—	—	—	●	あり	—	—
ReceiveUserData	—	—	—	—	—	●	あり	—	—
TcpSendBufLength	●	●	●	●	○	○	—	—	—
TcpRecvBufLength	●	●	●	●	○	○	—	—	—
ConvCCSID	●	●	—	—	○	●	あり	—	クラスタレシーバチャネルでは、システム定義に値を指定すると相手システムが TP1/Message Queue の CCSID に変更します。 ただし、相手システムは IBM MQ だけです。
NonPersistentMsgSpeed	●	●	●	●	○	○	あり	—	—
TranQMonType	●	●	—	—	●	—	—	event	クラスタセンダチャネルの場合は、event 固定です。
NetworkPriority	—	—	—	—	—	●	あり	—	—

属性	sender	server	receiver	requester	clusdr	clusrcvr	連絡属性※1	自動定義のクラスタセンダの仮定値	注意事項
AlterTime	●	●	●	●	●	●	あり	—	—
MsgChaProRetry	●	●	—	●	○	●	—	—	—
AdoptCheck	●	●	●	●	—	●	—	—	—
ShortRetryType	●	●	—	●	○	●	あり	—	—
ShortRetryCount	●	●	—	●	○	●	あり	—	—
ShortRetryInterval	●	●	—	●	○	●	あり	—	—
SecurityUserData	—	—	—	—	—	●	あり	—	—
SendExit	—	—	—	—	—	●	あり	—	—
RcvExit	—	—	—	—	—	●	あり	—	—
MsgExit	—	—	—	—	—	●	あり	—	—
ScyExit	—	—	—	—	—	●	あり	—	—
Descr	●	●	●	●	○	○	あり	—	—
MCAUser	—	—	—	—	—	●	あり	—	—
MCAType	—	—	—	—	—	●	あり	—	—
TcpRetryCount	●	●	—	●	○	—	—	10	—
TcpRetryInterval	●	●	—	●	○	—	—	1	—
StrucLength	△	△	△	△	△	△	—	—	—
SndJnlIO	●	●	—	—	○	—	—	flush	—
BatchMillInterval	●	●	—	—	○	●	あり	—	—
OwnPortAltNum	●	●	—	●	○	—	—	1	OwnPortNum 属性（自システムのポート番号）または OwnServName 属性（自システムのサービス名）の指定がある場合に有効です。

(凡例)

●：有効なチャネル属性です。UOC での変更はできません。

3. ユーザOWNコーディング

○：有効なチャンネル属性です。UOC での変更はできません。

△：参照だけです。

－：ありません。

注※1

相手クラスタレシーバチャンネルの属性によって値が設定されます。また、自クラスタレシーバチャンネルの値が相手に連絡されま

す。
UOC で変更した属性は、次にチャンネルを開始するまで有効です。

注※2

計算値については、「4. 準備作業」の「チャンネルのネゴシエーション」を参照してください。

注※3

この属性は clussdr では相手クラスタレシーバチャンネルから連絡されません。

dcmtcq_uoc_mqcd の宣言形式

```
typedef struct {
    MQCHAR  ChannelName[24];          /* チャンネル名 */
    MQLONG  Version;                 /* 構造体バージョン番号 */
    MQLONG  ChannelType;             /* チャンネルタイプ */
    MQCHAR  ClusterName[52];         /* クラスタ名 */
    MQLONG  BatchSize;               /* バッチサイズ */
    MQCHAR  XmitQName[52];           /* 転送キュー名 */
    MQLONG  MaxMsgLength;            /* 最大メッセージ長 */
    MQLONG  MaxSegLength;            /* 最大セグメントサイズ */
    MQLONG  SeqNumberWrap;           /* メッセージシーケンス番号の
                                     最大値 */
    MQLONG  BufType;                 /* バッファ方式 */
    MQLONG  SendBufGroup;            /* 送信バッファグループ番号 */
    MQLONG  ReceiveBufGroup;        /* 受信バッファグループ番号 */
    MQLONG  MsgEdtBufGroup;         /* メッセージ編集出口バッファ
                                     グループ番号 */
    MQLONG  MsgEdtBufCount;          /* メッセージ編集出口バッファ数 */
    MQLONG  StartType;               /* チャンネル確立方式 */
    MQLONG  LongRetryType;           /* 長期確立再試行の要否 */
    MQLONG  LongRetryCount;         /* 長期確立再試行の回数 */
    MQLONG  LongRetryInterval;      /* 長期確立再試行の間隔 */
    MQCHAR  OwnIPAddr[16];           /* 自ホストのIPアドレス */
    MQCHAR  OwnHostName[256];        /* 自ホストのホスト名 */
    MQCHAR  OwnPortNum[8];           /* 自ホストのポート番号 */
    MQCHAR  OwnServName[36];         /* 自ホストのサービス名 */
    MQCHAR  PartnerIPAddr[16];       /* 相手ホストのIPアドレス */
    MQCHAR  PartnerHostName[256];    /* 相手ホストのホスト名 */
    MQCHAR  PartnerPortNum[8];       /* 相手ホストのポート番号 */
    MQCHAR  PartnerServName[36];     /* 相手ホストのサービス名 */
    MQLONG  ConRspRecvInterval;      /* 開始要求応答受信監視
                                     タイマ値 */
    MQLONG  RspMsgRecvInterval;     /* 確認メッセージ受信監視
                                     タイマ値 */
    MQLONG  NextSegRecvInterval;    /* 継続セグメント受信監視
                                     タイマ値 */
    MQLONG  DiscInterval;            /* 転送キュー監視間隔 */
    MQLONG  DiscIntervalCount;       /* 転送キュー監視回数 */
    MQLONG  DiscInterval2;          /* 切断時間間隔 */
    MQLONG  DiscInterval2Effect;    /* 切断時間間隔使用の有無 */
    MQLONG  TimeoutConRetry;        /* タイムアウト時チャンネル確立
```

```

MQLONG NextMsgRecvInterval; /* 継続メッセージ受信監視
                               再試行の要否 */
MQLONG HeartbeatInterval; /* ハートビート間隔 */
MQLONG BatchInterval; /* バッチ終了待ちタイマ値 */
MQCHAR MsgEdtUserData[32]; /* メッセージ編集出口
                               タイマ値 */
MQCHAR SendUserData[32]; /* 送信出口ユーザデータ */
MQCHAR ReceiveUserData[32]; /* 受信出口ユーザデータ */
MQLONG TcpSendBufLength; /* TCP/IP出力用バッファサイズ */
MQLONG TcpRecvBufLength; /* TCP/IP入力用バッファサイズ */
MQLONG ConvCCSID; /* 変換後CCSID */
MQLONG NonPersistentMsgSpeed; /* 非永続メッセージ転送速度 */
MQLONG TranQMonType; /* 転送キュー監視方式 */
MQLONG NetworkPriority; /* ネットワーク接続優先順位 */
MQCHAR AlterTime[12]; /* チャネル定義の更新時刻 */
MQLONG MsgChaProRetry; /* MQプロトコルエラー時の
                               チャネル確立再試行の有無 */
MQLONG AdoptCheck; /* チャネル状態の不一致時の
                               強制再接続条件 */
MQLONG ShortRetryType; /* 短期確立再試行の要否 */
MQLONG ShortRetryCount; /* 短期確立再試行の回数 */
MQLONG ShortRetryInterval; /* 短期確立再試行の間隔 */
MQCHAR SecurityUserData[32]; /* セキュリティ出口ユーザデータ */
MQCHAR SendExit[128]; /* チャネル送信出口名 */
MQCHAR RcvExit[128]; /* チャネル受信出口名 */
MQCHAR MsgExit[128]; /* メッセージ出口名 */
MQCHAR ScyExit[128]; /* セキュリティ出口名 */
MQCHAR Descr[64]; /* 注釈 */
MQCHAR MCAUser[64]; /* MCAユーザID */
MQLONG MCAType; /* MCAタイプ */
MQLONG TcpRetryCount; /* TCP/IP確立確認の回数 */
MQLONG TcpRetryInterval; /* TCP/IP確立確認の間隔 */
MQLONG StrucLength; /* チャネルデータ定義ブロック
                               (dcmtcq_uoc_mqcd)の長さ */
MQLONG SndJnlIO; /* 送信チャネルジャーナル出力条件 */
MQLONG BatchMilInterval; /* バッチ終了待ちタイマ(ミリ秒) */
MQLONG FastSndMsg; /* 予約領域 */
MQLONG OwnPortAltNum; /* 自システムのポート番号使用数 */
MQCHAR Reserve[288]; /* 予約領域 */
} dcmtcq_uoc_mqcd;

```

- ChannelName (MQCHAR*24 型)

チャネル名 [-c]

領域の長さは DCMTQCQ_MQ_CHANNEL_NAME_LENGTH によって与えられます。

- Version (MQLONG 型)

構造体バージョン番号 (DCMTQCQ_MQCD_VERSION_7)

構造体のバージョンは7です。バージョン2以降の構造体だけに存在するフィールドについては、後述のフィールドの説明で示します。新しいバージョンの構造体では旧バージョンの予約領域以外のフィールド名および属性は変わりません。したがって、出口ではバージョン番号が、出口に必要なフィールドを含む最低バージョンの番号以上であるかどうかを検査する必要があります。

予約領域については予告なく変更することがあります。

- ChannelType (MQLONG 型)
チャンネルタイプ [-y type]
次に示すどれかの値が設定されます。
 - DCMTCQ_MQCHT_SENDER：センダ
 - DCMTCQ_MQCHT_SERVER：サーバ
 - DCMTCQ_MQCHT_RECEIVER：レシーバ
 - DCMTCQ_MQCHT_REQUESTER：リクエスタ
 - DCMTCQ_MQCHT_CLUSSDR：クラスタセンダ
 - DCMTCQ_MQCHT_CLUSRCVR：クラスタレシーバ
- ClusterName (MQCHAR*52 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
クラスタ名 [-a cluster]
この領域の長さは DCMTCQ_MQ_CLUSTER_NAME_LENGTH によって与えられます。
領域は名称の後ろに¥0 を格納し、定義で指定できるクラスタ名の最大サイズよりも大きくなるので注意してください。
- BatchSize (MQLONG 型)
バッチサイズ [-j]
チャンネル開始時のネゴシエーション前に出口が呼び出された場合、チャンネル定義のバッチサイズが設定されます。ネゴシエーション後に呼び出された場合は、ネゴシエーションの結果が設定されます。
- XmitQName (MQCHAR*52 型)
転送キュー名 [-q]
TCP 定義の mqtalcca 定義コマンドの-q オプションで定義した値が格納されます。
この領域の長さは DCMTCQ_MQ_Q_NAME_LENGTH によって与えられます。
- MaxMsgLength (MQLONG 型)
最大メッセージ長 [-m maxmsg]
チャンネル開始時のネゴシエーション前に出口が呼び出された場合、チャンネル定義の最大メッセージ長が設定されます。ネゴシエーション後に呼び出された場合は、ネゴシエーションの結果が設定されます。
- MaxSegLength (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
最大セグメントサイズ [-m maxseg]
チャンネル開始時のネゴシエーション前に出口が呼び出された場合、チャンネル定義最大セグメントサイズが設定されます。ネゴシエーション後に呼び出された場合は、ネゴシエーションの結果が設定されます。
- SeqNumberWrap (MQLONG 型)
メッセージシーケンス番号の最大値 [-w]
- BufType (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
バッファ方式 [-g buftype]
次に示すどちらかの値が設定されます。

- DCMTCQ_MQBUFT_SEG：セグメント方式
- DCMTCQ_MQBUFT_MSG：メッセージ方式
- SendBufGroup (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
送信バッファグループ番号 [-g sndbuf]
- ReceiveBufGroup (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
受信バッファグループ番号 [-g rcvbuf]
- MsgEdtBufGroup (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
メッセージ編集出口バッファグループ番号 [-e medbuf]
- MsgEdtBufCount (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
メッセージ編集出口バッファ数 [-e medcnt]
- StartType (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
チャンネル確立方式 [-i]
次に示すどれかの値が設定されます。
 - DCMTCQ_MQSTART_AUTO：トリガ起動またはネットワークリクエストによってチャンネルを自動確立します。
 - DCMTCQ_MQSTART_AUTO2：MQT サーバのサービス開始直後にチャンネルを自動確立します。
 - DCMTCQ_MQSTRAT_MANUAL：チャンネルを mqtstacha コマンド入力によって確立します。
- LongRetryType (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
長期確立再試行の要否 [-b bretrylg]
次に示すどちらかの値が設定されます。
 - DCMTCQ_MQLRTY_YES：長期確立再試行を実行します。
 - DCMTCQ_MQLRTY_NO：長期確立再試行を実行しません。
- LongRetryCount (MQLONG 型)
長期確立再試行の回数 [-b bretrylgcnt]
- LongRetryInterval (MQLONG 型)
長期確立再試行の間隔 [-b bretrylgint]
- OwnIPAddr (MQCHAR*16 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
自ホストの IP アドレス [-r ipaddr]
領域の長さは DCMTCQ_MQ_IP_ADDR_LENGTH によって与えられます。
形式は"nnn.nnn.nnn.nnn"の文字型です。nnn には 0～255 の文字が設定されます。nnn の値として先頭に 0 を指定した場合は、8 進数値として扱われます。
また、データを格納する領域のあとに¥0 を付けます。
- OwnHostName (MQCHAR*256 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
自ホストのホスト名 [-r hostname]

領域の長さは DCMTTCQ_MQ_HOST_NAME_LENGTH によって与えられます。

また、データを格納する領域のあとに¥0 を付けます。

- OwnPortNum (MQCHAR*8 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
自ホストのポート番号 [-r portno]
領域の長さは DCMTTCQ_MQ_PORT_NUM_LENGTH によって与えられます。
- OwnServName (MQCHAR*36 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
自ホストのサービス名 [-r servname]
領域の長さは DCMTTCQ_MQ_SERV_NAME_LENGTH によって与えられます。
また、データを格納する領域のあとに¥0 を付けます。
- PartnerIPAddr (MQCHAR*16 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
相手ホストの IP アドレス [-o oipaddr]
領域の長さは DCMTTCQ_MQ_IP_ADDR_LENGTH によって与えられます。
形式は"nnn.nnn.nnn.nnn"の文字型です。nnn には 0~255 の文字が設定されます。nnn の値として先頭に 0 を指定した場合は、8 進数値として扱われます。
また、データを格納する領域のあとに¥0 を付けます。
- PartnerHostName (MQCHAR*256 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
相手ホストのホスト名 [-o ohostname]
領域の長さは DCMTTCQ_MQ_HOST_NAME_LENGTH によって与えられます。
また、データを格納する領域のあとに¥0 を付けます。
- PartnerPortNum (MQCHAR*8 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
相手ホストのポート番号 [-o oportno]
領域の長さは DCMTTCQ_MQ_PORT_NUM_LENGTH によって与えられます。
- PartnerServName (MQCHAR*36 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
相手ホストのサービス名 [-o oservname]
領域の長さは DCMTTCQ_MQ_SERV_NAME_LENGTH によって与えられます。
また、データを格納する領域のあとに¥0 を付けます。
- ConRspRecvInterval (MQLONG 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
開始要求応答受信監視タイマ値 [-v tim1]
- RspMsgRecvInterval (MQLONG 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
確認メッセージ受信監視タイマ値 [-v tim2]
- NextSegRecvInterval (MQLONG 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)
継続セグメント受信監視タイマ値 [-v tim3]
- DiscInterval (MQLONG 型)
転送キュー監視間隔 [-v tim4]
- DiscIntervalCount (MQLONG 型) (Version=DCMTTCQ_MQCD_VERSION_4 で追加)

転送キュー監視回数 [-v tim4cnt]

- DiscInterval2 (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_3 で追加)
切断時間間隔 (秒数) [-v dtim]

- DiscInterval2Effect (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_3 で追加)
切断時間間隔使用の有無 [-v dtimefct]

次に示すどちらかの値が設定されます。

- DCMTCQ_DINT2EFC_YES: 切断時間間隔 (DiscInterval2) による監視をします。
- DCMTCQ_DINT2EFC_NO: 切断時間間隔 (DiscInterval2) による監視をしません。

- TimeoutConRetry (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
タイムアウト時チャンネル確立再試行の要否 [-v vretry]

次に示すどちらかの値が設定されます。

- DCMTCQ_MQTIMO_YES: チャンネル確立再試行を実行します。
- DCMTCQ_MQTIMO_NO: チャンネル確立再試行を実行しません。

- NextMsgRecvInterval (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
継続メッセージ受信監視タイマ値 [-v mtim]

- HeartbeatInterval (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_2 で追加)
ハートビート間隔 (秒数) [-v htim]

チャンネル開始時のネゴシエーション前に出口が呼び出された場合、チャンネル定義のハートビート間隔が設定されます。ネゴシエーション後に呼び出された場合は、ネゴシエーションの結果が設定されます。

- BatchInterval (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_3 で追加)
バッチ終了待ちタイマ値 (秒数) [-v btim]

- MsgEdtUserData (MQCHAR*32 型) (Version=DCMTCQ_MQCD_VERSION_2 で追加)
メッセージ編集出口ユーザデータ [-x metexdata]

これは、dcmtcq_uoc_mqcxp パラメタの ExitData 領域に設定されて出口に渡されます。

メッセージ編集出口ユーザデータは、32 バイト固定です。32 バイトに満たない場合は、スペースで補われます。

この領域の長さは DCMTCQ_MQ_EXIT_DATA_LENGTH によって与えられます。

- SendUserData (MQCHAR*32 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
送信出口ユーザデータ [-x sndexdata]

これは、dcmtcq_uoc_mqcxp パラメタの ExitData 領域に設定されて出口に渡されます。

送信出口ユーザデータは、32 バイト固定です。32 バイトに満たない場合は、スペースで補われます。この領域の長さは、DCMTCQ_MQ_EXIT_DATA_LENGTH によって与えられます。

- ReceiveUserData (MQCHAR*32 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
受信出口ユーザデータ [-x rcvexdata]

これは、dcmtcq_uoc_mqcxp パラメタの ExitData 領域に設定されて出口に渡されます。

受信出口ユーザデータは、32 バイト固定です。32 バイトに満たない場合は、スペースで補われます。この領域の長さは、DCMTCQ_MQ_EXIT_DATA_LENGTHによって与えられます。

- TcpSendBufLength (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
TCP/IP 出力用バッファサイズ [-t tcpsndbuf]
- TcpRecvBufLength (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
TCP/IP 入力用バッファサイズ [-t tcprcvbuf]
- ConvCCSID (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
変換後 CCSID [-d cnvccsid]
- NonPersistentMsgSpeed (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_2 で追加)
非永続メッセージ転送速度 [-s npmspeed]

チャンネル開始時のネゴシエーション前に出口が呼び出された場合、チャンネル定義の非永続メッセージの送信速度が設定されます。ネゴシエーション後に呼び出された場合は、ネゴシエーションの結果が設定されます。

次に示すどちらかの値が設定されます。

- DCMTCQ_MQNPMS_NORMAL：標準速度
- DCMTCQ_MQNPMS_FAST：高速度
- TranQMonType (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
転送キュー監視方式 [-z]

次に示すどちらかの値が設定されます。

- DCMTCQ_MQTRNQS_TIMER：タイマ方式
- DCMTCQ_MQTRNQS_EVENT：イベント方式

これは、ChannelType が DCMTCQ_MQCHT_SENDER, または DCMTCQ_MQCHT_SERVER の場合に有効です。

- NetworkPriority (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
ネットワーク接続優先順位 [-p netprty]
- AlterTime (MQCHAR*12 型) (Version=DCMTCQ_MQCD_VERSION_4 で追加)
チャンネル定義の更新時刻
チャンネル定義の作成および更新の日付/時刻が、通算の秒数で設定されます。
- MsgChaProRetry (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
MQ プロトコルエラー時のチャンネル確立再試行の有無 [-b bretrymcp]

次に示すどちらかの値が設定されます。

- DCMTCQ_MQMCP_NO：チャンネル確立再試行をしません。
- DCMTCQ_MQMCP_YES：チャンネル確立再試行をします。
- AdoptCheck (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
チャンネル状態の不一致時の強制再接続条件 [-f adoptchk]

次に示すどれかの値が設定されます。

- DCMTCQ_MQADPCK_OFF：強制再接続はできません。
- DCMTCQ_MQADPCK_NAME：強制再接続を受け付けます。
- DCMTCQ_MQADPCK_QMGR：同じキューマネージャ名の要求で強制再接続を受け付けます。
- DCMTCQ_MQADPCK_IPADDR：同じ IP アドレスの要求で強制再接続を受け付けます。
- DCMTCQ_MQADPCK_ALL：同じキューマネージャ名と IP アドレスの要求で強制再接続を受け付けます。

- ShortRetryType (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)

短期確立再試行の要否 [-b bretry]

次に示すどちらかの値が設定されます。

- DCMTCQ_MQSRTY_YES：短期確立再試行を実行します。
- DCMTCQ_MQSRTY_NO：短期確立再試行を実行しません。
- ShortRetryCount (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
短期確立再試行の回数 [-b bretrycnt]
- ShortRetryInterval (MQLONG 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
短期確立再試行の間隔 [-b bretryint]
- SecurityUserData (MQCHAR*32 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
セキュリティ出口ユーザデータ [-x scyexdata]
これは、dcmtcq_uoc_mqcxp パラメタの ExitData 領域に設定されて出口に渡されます。
セキュリティ出口ユーザデータは、32 バイト固定です。32 バイトに満たない場合は、スペースで補われます。この領域の長さは、DCMTCQ_MQ_EXIT_DATA_LENGTH によって与えられます。
- SendExit (MQCHAR*128 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
チャンネル送信出口名 [-x sendexit]
チャンネル送信出口名は、128 バイト固定です。128 バイトに満たない場合は、スペースで補われます。
この領域の長さは、DCMTCQ_MQ_EXIT_NAME_LENGTH によって与えられます。
- RcvExit (MQCHAR*128 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
チャンネル受信出口名 [-x rcvexit]
チャンネル受信出口名は、128 バイト固定です。128 バイトに満たない場合は、スペースで補われます。
この領域の長さは、DCMTCQ_MQ_EXIT_NAME_LENGTH によって与えられます。
- MsgExit (MQCHAR*128 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
メッセージ出口名 [-x msgexit]
メッセージ出口名は、128 バイト固定です。128 バイトに満たない場合は、スペースで補われます。
この領域の長さは、DCMTCQ_MQ_EXIT_NAME_LENGTH によって与えられます。
- ScyExit (MQCHAR*128 型) (Version=DCMTCQ_MQCD_VERSION_5 で追加)
セキュリティ出口名 [-x scyexit]

セキュリティ出口名は、128バイト固定です。128バイトに満たない場合は、スペースで補われます。この領域の長さは、DCMTCQ_MQ_EXIT_NAME_LENGTHによって与えられます。

- Descr (MQCHAR*64型) (Version=DCMTCQ_MQCD_VERSION_5で追加)
注釈 [-h]
注釈は、64バイト固定です。64バイトに満たない場合は、スペースで補われます。この領域の長さは、DCMTCQ_MQ_DESCR_LENGTHによって与えられます。
- MCAUser (MQCHAR*64型) (Version=DCMTCQ_MQCD_VERSION_5で追加)
MCA ユーザ ID [-mc mcauser]
MCA ユーザ ID は、64バイト固定です。64バイトに満たない場合は、スペースで補われます。この領域の長さは、DCMTCQ_MQ_MCAUSER_LENGTHによって与えられます。
- MCAType (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_5で追加)
MCA タイプ [-mc mcatype]
次に示すどちらかの値が設定されます。
 - DCMTCQ_MQMCTYPE_PROCESS : MCA は独立のプロセスとして動作します。
 - DCMTCQ_MQMCTYPE_THREAD : MCA は独立のスレッドとして動作します。
- TcpRetryCount (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_5で追加)
TCP/IP 確立確認の回数 [-t tretrycnt]
- TcpRetryInterval (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_5で追加)
TCP/IP 確立確認の間隔 (秒数) [-t tretryint]
- StrucLength (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_5で追加)
チャンネルデータ定義ブロック (dcmtcq_uoc_mqcd) の長さ (バイト)
- SndJnlIO (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_6で追加)
送信チャンネルジャーナル出力条件 [-jn sndjnl]
次に示すどちらかの値が設定されます。
 - DCMTCQ_SNDJNL_FLUSH : 即時フラッシュ (ディスク出力) します。
 - DCMTCQ_SNDJNL_NOFLUSH : 即時フラッシュ (ディスク出力) しません。
- BatchMilInterval (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_6で追加)
バッチ終了待ちタイム値 (ミリ秒) [-v bmtim]
- FastSndMsg (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_7で追加)
システムで使用するための予約領域です。
- OwnPortAltNum (MQLONG型) (Version=DCMTCQ_MQCD_VERSION_7で追加)
自システムのポート番号使用数 [-r portnum]

(4) バッファリスト (dcmtcq_uoc_bufflist)

バッファリストは、メッセージを格納した送受信バッファや、それを編集する場合に使用する編集バッファをやり取りするために使用されます。

dcmtcq_uoc_bufflist の宣言形式

```
typedef struct {
    MQLONG          Buff_num;      /* バッファ数 */
    MQLONG          Data_buff_num; /* データを格納する編集用バッファエントリ番号 */
    dcmtcq_uoc_buffinf Buffinf_array[DCMTCQ_UOC_BUFF_MAX];
                                     /* バッファ情報領域 */
} dcmtcq_uoc_bufflist;

typedef struct {
    MQLONG Buff_length;          /* バッファ長 */
    MQLONG Used_buff_length;    /* 使用バッファ長 */
    MQCHAR *Buffer_addr;        /* バッファアドレス */
    MQCHAR Reserve1[4];         /* 予約領域 */
} dcmtcq_uoc_buffinf;
```

注

64 ビット版の場合は、次の形式になります。

```
    :
    MQCHAR *Buffer_addr;          /* バッファアドレス */
} dcmtcq_uoc_buffinf;
```

- Buff_num (MQLONG 型) -input
バッファ数
バッファ情報の数が格納されます。
送受信バッファの場合、必ず 1 が格納されます。
編集用バッファの場合、MQT 通信構成定義に編集用バッファを指定しないときは、0 が格納されます。
- Data_buff_num (MQLONG 型) -output
データを格納する編集用バッファエントリ番号
UOC からメッセージを編集用バッファに格納した
(ExitResponse2=DCMTCQ_MQXR2_USE_EDIT_BUFFER) 場合、データを格納したバッファのエントリ番号を格納します。
指定が省略された場合 (Data_buff_num=0) は、編集用バッファの第 1 エントリがメッセージとして扱われます。また、複数の編集バッファに分割して格納できません。
- Buffinf_array (dcmtcq_uoc_buffinf 型) -input/output
バッファ情報領域
UOC との間でやり取りされるメッセージが格納される領域です。
次に示す領域から構成されます。
 - Buff_length (MQLONG 型) -input

チャンネル出口パラメタブロックの ExitReason に、DCMTCQ_MQXR_MSG が指定された場合にだけ有効な領域です。

UOC 内でバッファとして使用できる長さが格納されます。

送受信バッファリストの場合、MQXQH 構造体の長さを含みます。

- Used_buff_length (MQLONG 型) –input/output

チャンネル出口パラメタブロックの ExitReason に、DCMTCQ_MQXR_MSG が指定された場合にだけ有効な領域です。

Buff_length で指定された長さのうち、使用している有効な領域長（先頭からの長さ）が格納されます。UOC 処理後に有効な領域長が変化した場合は、UOC のリターン前に更新してください。更新する値には、MQXQH 構造体の長さを含めてください。

Buff_length の値よりも大きい値を格納した場合は、UOC が異常終了します。

- Buffer_addr (MQCHAR ポインタ型) –input

バッファアドレス

データが格納されている領域のアドレスが格納されます。格納されているデータは、UOC から作成、および更新できます。データ形式については、「[3.2.2 UOC に渡されるデータ形式](#)」を参照してください。

3.3.3 UOC パラメタ指定についての注意事項

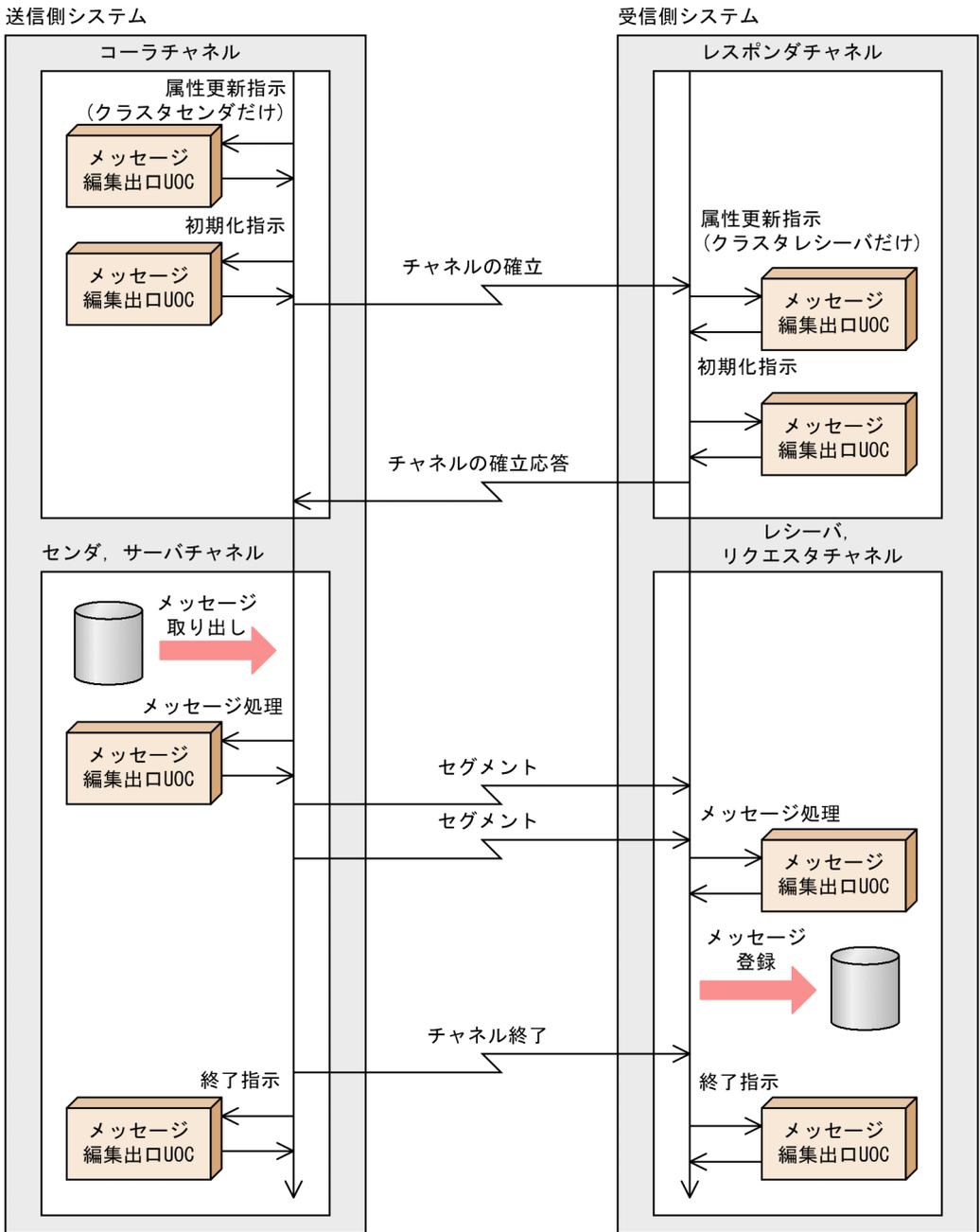
UOC で設定したパラメタに不正があった場合、エラーログメッセージ (KFCA16347-E) が出力され、チャンネル状態は「チャンネル使用不可」にして終了します。

ただし、ExitReason が DCMTCQ_MQXR_ALT の場合、UOC が設定したチャンネルデータ定義 (dcmtcq_uoc_mqcd) に不正な値が設定されても、チャンネルは設定された値を無視して処理を続行します。

3.4 UOC 呼び出しの流れ

メッセージ編集出口 UOC が呼び出されるタイミングについて、次の図に示します。

図 3-3 メッセージ編集出口 UOC が呼び出されるタイミング



3.5 UOC の組み込み

TP1/Message Queue に UOC を組み込むには、次に示す手順で MQT 実行形式プログラムを作成する必要があります。

1. UOC を作成する
2. MQT メイン関数を作成し、メイン関数で UOC の関数アドレスを指定する
3. UOC, および MQT メイン関数をコンパイルし、OpenTP1 および TP1/Message Queue が提供する各種のライブラリと結合する
MQT 実行形式プログラムが作成されます。
4. MQT のシステムサービス情報定義を作成する
3. で作成した実行形式プログラムを TP1/Message Queue から起動できるようになります。

3.5.1 MQT メイン関数の作成

MQT メイン関数は、UOC 関数アドレスを指定し、MQT 通信プロセスメイン関数を呼び出すプログラムです。MQT メイン関数と UOC は、同じ言語（ANSI C, C++, または K&R 版 C 言語）で作成する必要があります。

MQT メイン関数のコーディングについて、次に示します。

ANSI C, C++言語の場合

```
#include <dcmtcquo.h> /* TP1/MQ用ヘッダファイル */ ...1.
extern long med_uoc01(dcmtcq_uoc_parmlist *);
/* メッセージ編集出口UOC関数 */ ...2.
dcmtcq_uoc_def_t dcmtcq_uoctbl =
    {NULL, NULL, NULL, NULL}; ...3.
int main(void)
{
    dcmtcq_uoctbl.med_uoc = (dcmqt_uocfunc)med_uoc01;
/* メッセージ編集出口UOC関数アドレス設定 */ ...4.
    dc_mtcq_main(); /* MQT通信プロセスメイン関数コール */ ...5.
    return(0); /* MQTメイン関数リターン */ ...6.
}
```

1. プロトコル提供ヘッダを取り込みます。
2. 使用する UOC 関数のプロトタイプ宣言をします。UOC のリターン値は long 型にしてください。
3. UOC テーブルを確保して、初期化します。必ずこのとおりコーディングしてください。
4. UOC 関数のアドレスを、UOC テーブルに設定します。

dcmtcq_uoctbl.med_uoc : メッセージ編集出口 UOC 関数のアドレス

5. MQT 通信プロセスメイン関数を呼び出します。必ずこのとおりコーディングしてください。

6. MQT メイン関数をリターンします。

K&R 版 C 言語の場合

```
#include <dcmtcquo.h>      /* TP1/MQ用ヘッダファイル */ ...1.
extern long med_uoc01();  /* メッセージ編集出口UOC関数 */ ...2.
dcmtcq_uoc_def_t dcmtcq_uoctbl; ...3.

main()
{
    dcmtcq_uoctbl.med_uoc = (dcmtcq_uocfunc)med_uoc01;
                          /* メッセージ編集出口UOC関数アドレス設定 */ ...4.
    dc_mtcq_main();      /* MQT通信プロセスメイン関数コール */ ...5.
}
```

1. プロトコル提供ヘッダを取り込みます。

2. 使用する UOC 関数の extern 宣言をします。UOC のリターン値は long 型にしてください。

3. UOC テーブルを確保します。必ずこのとおりコーディングしてください。

4. UOC 関数のアドレスを、UOC テーブルに設定します。

dcmtcq_uoctbl.med_uoc : メッセージ編集出口 UOC 関数のアドレス

5. MQT 通信プロセスメイン関数を呼び出します。必ずこのとおりコーディングしてください。

3.5.2 MQT 実行形式プログラムの作成 (UNIX)

MQT メイン関数と UOC をコンパイルし、OpenTP1 および TP1/Message Queue で提供する各種ライブラリを結合して、UOC を含む MQT 実行形式プログラムを作成します。UNIX を使用する場合の MQT 実行形式ファイルの作成方法、および作成時の注意事項について、次に示します。

なお、ここでは、C 言語で作成した UOC オブジェクトファイルおよび MQT メイン関数オブジェクトファイルから MQT 実行形式ファイルを作成する方法を示します。C 言語の形式 (ANSI 形式、K&R 形式)、C++言語で作成する場合、コンパイラの種類によってはコマンドオプションが異なりますので、詳細はコンパイラのマニュアルを参照してください。

(1) C 言語で作成する例

MQT メイン関数オブジェクトファイルを作成するコマンド

```
cc -c※1 -I$DCDIR※2/include MQTメイン関数ソースファイル名
```

UOC オブジェクトファイルを作成するコマンド

```
cc -c※1 -I$DCDIR※2/include UOCソースファイル名
```

MQT 実行形式プログラムを作成するコマンド

```
mqtpltcp -A -o 実行形式プログラム名※3  
-m MQTメイン関数オブジェクトファイル名  
[-i UOCオブジェクトファイル名1 -i UOCオブジェクトファイル名2 …]  
[-l ユーザ固有ライブラリ] [-P] ※4 [-Q] ※4 [-e] ※5
```

注※1

AIX で 64 ビット版 TP1/Message Queue を使用する場合は、オプション「-q64」を追加してください。

HP-UX (IPF) で 64 ビット版 TP1/Message Queue を使用する場合は、オプション「+DD64」を追加してください。

64 ビット版 Linux で 32 ビット版 TP1/Message Queue を使用する場合には、オプション「-m32」を追加してください。

注※2

\$DCDIR は、OpenTP1 の格納ディレクトリです。

注※3

MQT 実行形式プログラム名は、先頭が mqtu で始まる 8 文字以内の名称です。

作成した MQT 実行形式プログラムは、\$DCDIR/lib/servers 配下に格納してください。

注※4

-P オプションと-Q オプションを指定することで、実行するリンケージコマンドが決まります。使用する環境に合わせて、オプションを指定してください。

言語仕様と想定するリンケージコマンドを次に示します。

OS	言語仕様と想定するリンケージコマンド	
	C 言語の場合	C++言語の場合
AIX	xlc	xlC
HP-UX	cc	CC, aCC
Linux	gcc	gcc

-P オプションと-Q オプション指定時に実行するリンケージコマンドを次に示します。

OS	オプション指定時に実行するリンケージコマンド		
	-P 指定の場合	-Q 指定の場合	-P, -Q 指定の場合
AIX	xlC	xlc	xlc
HP-UX	CC	aCC	cc
Linux	gcc	gcc	gcc

なお、-P オプションと-Q オプションを同時に指定した場合、あとに指定したオプションが有効になりますので注意してください。

注※5

64ビット版 Linux で 32 ビット版 TP1/Message Queue を使用する場合には、-e オプションを追加してください。

(2) MQT 実行形式プログラム作成時の注意事項

- OpenTP1, およびユーザ固有ライブラリは、次に示す順序で検索されます。

1. \$DCDIR
2. ライブラリパス環境変数の指定ディレクトリ

ユーザ固有ライブラリを\$DCDIR 以外の場所に格納している場合、ライブラリパス環境変数にユーザ固有ライブラリの格納ディレクトリを指定したあと、mqtpltcp コマンドを実行してください。なお、ライブラリパス環境変数の詳細は、ld コマンドの仕様を参照してください。

- mqtpltcp コマンドの実行結果は、次に示すログメッセージで表示されます。

- 正常終了の場合

Completed to make the uCosminexus TP1/Message Queue(MQT) executable file 実行形式プログラム名

- 異常終了の場合

Failed to make the uCosminexus TP1/Message Queue(MQT) executable file 実行形式プログラム名

3.5.3 MQT 実行形式プログラムの作成 (Windows)

MQT メイン関数と UOC をコンパイルし、OpenTP1 および TP1/Message Queue で提供する各種ライブラリを結合して、UOC を含む MQT 実行形式プログラムを作成します。Windows を使用する場合の MQT 実行形式ファイルの作成方法、および作成時の注意事項について、次に示します。

(1) ANSI C または C++ で作成する場合

MQT メイン関数オブジェクトファイルを作成するコマンド

```
cl /c /MD MQTメイン関数ソースファイル名 /I %DCDIR%\include
```

注

%DCDIR%は、OpenTP1 の格納ディレクトリです。

環境によって、コンパイル時に warning C4996 が出力される場合があります。この場合は、コンパイルオプションに-D_CRT_SECURE_NO_DEPRECATED を追加してください。

UOC オブジェクトファイルを作成するコマンド

```
cl /c /MD UOCソースファイル名 /I %DCDIR%\include
```

MQT 実行形式プログラムを作成するコマンド

```
Link /NODEFAULTLIB  
/stack:4194304  
MQTメイン関数オブジェクトファイル名  
UOCオブジェクトファイル名1  
 [UOCオブジェクトファイル名2...]  
 [ユーザ固有ライブラリ]  
%DCDIR%\lib\libbetran.lib  
%DCDIR%\lib\libtactk.lib  
%DCDIR%\lib\libmtcq.lib  
%DCDIR%\lib\libmqa.lib  
kernel32.lib※1  
msvcrt.lib※1  
ucrt.lib※1, ※2  
vcruntime.lib※1, ※2  
oldnames.lib※1  
MsWSock.lib※1  
version.lib※1  
WS2_32.lib※1  
"/subsystem:console"  
-out:実行形式プログラム名※3
```

注※1

これらのライブラリの格納ディレクトリを、環境変数 LIB に適切に設定してください。

注※2

Visual Studio 2013 を使用している場合は指定しないでください。

注※3

MQT 実行形式プログラム名は、先頭が mqtu で始まる 8 文字以内の名称です。

作成した MQT 実行形式プログラムは、%DCDIR%\lib\servers 配下に格納してください。

3.5.4 システムサービス情報定義の作成

MQT のシステムサービス情報定義を作成して、UOC を組み込んだ MQT 実行形式プログラムを TP1/Message Queue から起動できるようにします。

システムサービス情報定義の作成方法について、次に示します。

1. 標準で提供されているシステムサービス情報定義を、同一ディレクトリ内に別の名称でコピーする
ファイル名は次に示すとおりに変更してください。また、このファイル名を、MQA サービス定義の mqamqtnam 定義コマンドに指定するプロトコル名称として指定してください。

- 標準デーモンの場合

\$DCDIR/lib/sysconf/mqttcp が標準で提供されます。ファイル名を mqtu で始まる 14 文字以内の識別子に変更してください。

- クラスタセンダデーモンの場合

\$DCDIR/lib/sysconf/mqttcpcs が標準で提供されます。ファイル名を mqttcpcsu で始まる 14 文字以内の識別子に変更してください。

- クラスタレシーバデーモンの場合

\$DCDIR/lib/sysconf/mqttcpcr が標準で提供されます。ファイル名を mqttcpcru で始まる 14 文字以内の識別子に変更してください。

2. システムのテキストエディタで、システムサービス情報定義中の実行形式プログラム名を修正する

実行形式プログラム名は、次に示すとおり指定されています。

```
..... :  
set module = "mqtdtcp"  
..... :
```

「mqtdtcp」の部分を、作成した MQT 実行形式プログラムの名称に変更してください。なお、MQT 実行形式プログラム名は、mqtu で始まる 8 文字以内の名称にしてください。

3.6 UOC の作成についての注意事項

UOC の作成についての注意事項について説明します。

3.6.1 UOC の異常処理

UOC で異常を検知した場合、MQT で定められたリターンコードを使用して、MQT に異常の発生を通知するようにコーディングしてください。UOC でプロセス終了のシグナル、または abort 関数が発行された場合、TP1/Message Queue は異常終了します。

3.6.2 UOC が動作するプロセス

UOC は、メッセージを送受信している MQT サーバのプロセス内で動作します。このため、UOC が実行する処理は、同一プロセス内で動作しているすべてのチャンネルの動作に影響します。

3.6.3 UOC のスタック領域

UOC がスタック領域として使用できる領域は、1 キロバイトです。

1 キロバイトを超えるスタック領域を確保した場合、MQT サーバのメモリが破壊され、不正に動作することがあります。

3.6.4 使用ライブラリの制限

MQT の UOC 機能を使用する場合、UOC から次に示すアプリケーションインタフェースをコールできません。

- OpenTP1 システムの各プログラムプロダクトが提供するアプリケーションインタフェース
- TP1/NET/Library が提供するアプリケーションインタフェース

3.6.5 バージョンアップ時の注意

TP1/Message Queue をバージョンアップした場合、および修正版を組み込んだ場合、UOC を組み込んだ MQT デーモンを再度メイクしてください。

UOC および UOC メイン関数の再コンパイルおよび再リンケージを実行してください。

3.6.6 UOC で使用できる関数

UOC を作成する場合、UOC では次に示す関数を使用できます。ほかの関数を使用した場合の動作は保証できません。

- メモリを操作する関数
 - データ領域管理 (例: malloc, free)
 - 共有メモリ管理 (例: shmctl, shmget, shmop)
 - メモリ操作 (例: memcpy, memset)
 - 文字列操作 (例: strcpy, strcmp)
- 時間を取得する関数 (例: time)
- ファイルを操作する関数 (例: fopen, fseek, fread, fwrite, fclose)

3.7 UOC の作成例

UOC のサンプルとして、ファイル転送を実現するメッセージ編集 UOC のコーディング例について、次に示します。

3.7.1 サンプル UOC の機能

MQRMH 構造体に指定されたファイルを、送信時にメッセージ内に取り込み、受信時に取り出して格納する UOC です。

3.7.2 サンプル UOC の所在

ファイル転送を実現するメッセージ編集 UOC のソースファイルは、次に示す場所に格納されています。

```
$DCDIR/examples/mq/exit/c/mqmsgedt.c
```

3.7.3 サンプル UOC のコーディング例

```
0010 /*****/
0020 /* */
0030 /* Program name: MsgEdit */
0040 /* */
0050 /* Description: Sample program of the user exit */
0060 /* routine for message editing */
0070 /* */
0080 /*****/
0090 /* */
0100 /* Function: */
0110 /* */
0120 /* This sample program processes messages whose */
0130 /* reference header (MQRMH) has the object type */
0140 /* matching with the message editing exit user data of */
0150 /* the channel definition. */
0160 /* When the channel type is sender, server or clusdr, */
0170 /* the program takes the data of the file with the */
0180 /* send object name described in the message reference */
0190 /* header, and copies the whole data length from the */
0200 /* offset to the application data area of the */
0210 /* transmission buffer. */
0220 /* If the program cannot copy all the messages in the */
0230 /* file to the available area in the */
0240 /* transmission buffer, an error will be returned. */
0250 /* An error will also be returned when an error occurs */
0260 /* while accessing the file. */
0270 /* A successful return will set the following */
0280 /* information: Channel exit parameter block */
```

```

0290 /*          (dcmtcq_uoc_mqxcpc structure) */
0300 /*      ExitResponse   = DCMTQCQ_MQXCC_OK          */
0310 /*      ExitResponse2  = DCMTQCQ_MQXR2_USE_TRAN_BUFFER */
0320 /* Transmission buffer list                        */
0330 /*          (dcmtcq_uoc_buffinf structure) */
0340 /*      Used_buff_length += Data_length          */
0350 /* An error return will set the following information: */
0360 /* Channel exit parameter block                    */
0370 /*          (dcmtcq_uoc_mqxcpc structure) */
0380 /*      ExitResponse   = DCMTQCQ_MQXCC_SUPPRESS_FUNCTION */
0390 /*      Feedback       = MQFB_STOPPED_BY_MSG_EXIT      */
0400 /*          */
0410 /* When the channel type is requester, receiver or */
0420 /* clusrvcvr, the program takes the messages in the */
0430 /* application data area of the transmission buffer, */
0440 /* and outputs the whole data length from the offset */
0450 /* to the file with the destination object name */
0460 /* specified in the message reference header. */
0470 /* If the offset value is 0, the program creates a */
0480 /* new file. Otherwise, the messages are added to the */
0490 /* end of the file. */
0500 /* If the message being added encounters the condition */
0510 /* that the offset value is so small that the file's */
0520 /* trailing end is not reached, the program discards */
0530 /* the message. */
0540 /* If the offset value is greater, the program returns */
0550 /* an error. */
0560 /* The user exit routine will return an error when */
0570 /* accessing the file encounters an abnormality. */
0580 /* A successful return will set the following */
0590 /* information: */
0600 /* Channel exit parameter block                    */
0610 /*          (dcmtcq_uoc_mqxcpc structure) */
0620 /*      ExitResponse   = DCMTQCQ_MQXCC_OK          */
0630 /*      ExitResponse2  = DCMTQCQ_MQXR2_USE_TRAN_BUFFER */
0640 /* Transmission buffer list                        */
0650 /*          (dcmtcq_uoc_buffinf structure) */
0660 /*      Used_buff_length = Transmission header (MQXQH) */
0670 /*      length + Message_reference_header_length */
0680 /* An error return will set the following information: */
0690 /* Channel exit parameter block                    */
0700 /*          (dcmtcq_uoc_mqxcpc structure) */
0710 /*      ExitResponse   = DCMTQCQ_MQXCC_SUPPRESS_FUNCTION */
0720 /*      Feedback       = MQFB_STOPPED_BY_MSG_EXIT      */
0730 /*          */
0740 /****** */
0750 /*          */
0760 /* The program does not convert the application data */
0770 /* containing the message reference header. In order */
0780 /* to use this program for transmission between */
0790 /* different machines requiring data conversion, add */
0800 /* a data conversion process. */
0810 /*          */
0820 /****** */
0830
0840 #include <errno.h>
0850 #include <stdio.h>
0860 #include <stdlib.h>

```

```

0870 #include <stddef.h>
0880 #include <string.h>
0890 #include <ctype.h>
0900
0910 #include <cmqc.h>
0920 #include <dcmtcquo.h>
0930
0940 long MsgEdit(dcmtcq_uoc_parmlist *ParmList);
0950
0960 /*****
0970  */
0980 /*****
0990 #define MAX_FILENAME_LENGTH 256
1000 #define LOGICAL_OFFSET_VAL 1000000000L
1010
1020
1030 long MsgEdit(dcmtcq_uoc_parmlist *ParmList)
1040 {
1050     PMQXQH     pMQXQH;
1060     PMQRMH     pMQRMH;
1070     MQLONG     FileDataOffset;
1080                 /* offset of data within file */
1090     MQLONG     FeedbackCode = 0; /* feedback code */
1100     char       *pMsgData; /* pointer to data
1110                 within reference msg */
1120     char       DestFileName[MAX_FILENAME_LENGTH+1];
1130                 /* Name of file at receiving end */
1140     char       SrcFileName[MAX_FILENAME_LENGTH+1];
1150                 /* Name of file at sending end */
1160     FILE       *fd = NULL; /* file descriptor */
1170     char       mode[4]; /* file open mode */
1180     long       position; /* position within file. */
1190     int        rc; /* return code */
1200     size_t     itemswritten; /* returned by fwrite */
1210     size_t     bytesread; /* returned by fread */
1220     char       dummy; /* work area */
1230     MQLONG     MsgSpace;
1240                 /* Space remaining in input buffer */
1250     MQLONG     ReadLength; /* Number of bytes to read
1260                 from file */
1270     MQLONG     WriteLength; /* Number of bytes to write
1280                 to file */
1290     int        WriteData = 1; /* Do we write the data */
1300     MQLONG     InputDataLength; /* Length requested
1310                 in reference msg */
1320
1330     ParmList->mqcxp_adr->ExitResponse = DCMTCQ_MQXCC_OK;
1340     ParmList->mqcxp_adr->ExitResponse2 =
1350         DCMTCQ_MQXR2_USE_TRAN_BUFFER;
1360
1370     switch(ParmList->mqcxp_adr->ExitId)
1380     {
1390     case DCMTCQ_MQXT_CHANNEL_MSGEDT_EXIT:
1400         break;
1410     default:
1420         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
1430         goto UOC_EXIT;
1440     }

```

```

1450
1460 switch(ParmList->mqcxp_adr->ExitReason)
1470 {
1480     case DCMTCQ_MQXR_INIT:
1490         goto UOC_EXIT;
1500     case DCMTCQ_MQXR_TERM:
1510         goto UOC_EXIT;
1520     case DCMTCQ_MQXR_MSG:
1530         break;
1540     case DCMTCQ_MQXR_ALT:
1550         goto UOC_EXIT;
1560     default:
1570         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
1580         goto UOC_EXIT;
1590 }
1600
1610 /*****/
1620 /* Get addresses of transmission header and message */
1630 /* itself. The message should start with a reference */
1640 /* message header. */
1650 /*****/
1660
1670 pMQXQH = (PMQXQH)
1680     (ParmList->tbuflist_adr->Buffinf_array[0].Buffer_addr);
1690 pMQRMH = (PMQRMH)
1700     (ParmList->tbuflist_adr->Buffinf_array[0]
1710     .Buffer_addr + sizeof(MQXQH));
1720
1730 /*****/
1740 /* If the message format does not indicate a */
1750 /* reference message then ignore the message. */
1760 /*****/
1770 if (memcmp(pMQXQH->MsgDesc.Format,
1780           MQFMT_REF_MSG_HEADER,
1790           MQ_FORMAT_LENGTH))
1800 {
1810     goto UOC_EXIT;
1820 }
1830
1840 /*****/
1850 /* If the object type does not match the type */
1860 /* specified in the channel definition then ignore */
1870 /* the message. */
1880 /*****/
1890 if (memcmp(pMQRMH->ObjectType,
1900           ParmList->mqcxp_adr->ExitData,
1910           sizeof(pMQRMH->ObjectType)))
1920 {
1930     goto UOC_EXIT;
1940 }
1950
1960 /*****/
1970 /* Check that the SrcEnv field lies within the */
1980 /* structure */
1990 /*****/
2000 if (pMQRMH->SrcEnvLength < 0 ||
2010     (pMQRMH->SrcEnvLength > 0 &&
2020     (pMQRMH->SrcEnvLength + pMQRMH->SrcEnvOffset) >

```

```

2030     pMQRMH->StrucLength))
2040 {
2050     FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
2060     goto UOC_EXIT;
2070 }
2080
2090 /*****
2100 /* Check that the SrcName field lies within the */
2110 /* structure */
2120 /*****
2130 if (pMQRMH->SrcNameLength < 0 ||
2140     (pMQRMH->SrcNameLength > 0 &&
2150      (pMQRMH->SrcNameLength + pMQRMH->SrcNameOffset) >
2160       pMQRMH->StrucLength))
2170 {
2180     FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
2190     goto UOC_EXIT;
2200 }
2210
2220 /*****
2230 /* Check that the DestEnv field lies within the */
2240 /* structure */
2250 /*****
2260 if (pMQRMH->DestEnvLength < 0 ||
2270     (pMQRMH->DestEnvLength > 0 &&
2280      (pMQRMH->DestEnvLength + pMQRMH->DestEnvOffset) >
2290       pMQRMH->StrucLength))
2300 {
2310     FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
2320     goto UOC_EXIT;
2330 }
2340
2350 /*****
2360 /* Check that the DestName field lies within the */
2370 /* structure */
2380 /*****
2390 if (pMQRMH->DestNameLength < 0 ||
2400     (pMQRMH->DestNameLength > 0 &&
2410      (pMQRMH->DestNameLength + pMQRMH->DestNameOffset) >
2420       pMQRMH->StrucLength))
2430 {
2440     FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
2450     goto UOC_EXIT;
2460 }
2470
2480 /*****
2490 /* Check that the message contains a complete MQRMH */
2500 /* structure */
2510 /*****
2520 if (pMQRMH->StrucLength >
2530     (MQLONG)(ParmList->tbuflist_adr->
2540      Buffinf_array[0].Used_buff_length - sizeof(MQXQH)))
2550 {
2560     FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
2570     goto UOC_EXIT;
2580 }
2590
2600 /*****

```

```

2610 /* Extract the following values from the reference */
2620 /* message. */
2630 /* Offset of data within the file. */
2640 /* Length of data to be read or written */
2650 /* Name of file on destination system. */
2660 /* Name of file on source system. */
2670 /*****/
2680 FileDataOffset = pMQRMH->DataLogicalOffset2 *
2690                 LOGICAL_OFFSET_VAL +
2700                 pMQRMH->DataLogicalOffset;
2710 InputDataLength = pMQRMH->DataLogicalLength;
2720 memset(DestFileName, 0, sizeof(DestFileName));
2730 memset(SrcFileName, 0, sizeof(SrcFileName));
2740 memcpy(DestFileName,
2750        (char *)pMQRMH + pMQRMH->DestNameOffset,
2760        (size_t)pMQRMH->DestNameLength);
2770 memcpy(SrcFileName,
2780        (char *)pMQRMH + pMQRMH->SrcNameOffset,
2790        (size_t)pMQRMH->SrcNameLength);
2800
2810 /*****/
2820 /* Move the data to or from the file. */
2830 /*****/
2840 switch (ParmList->mqcd_adr->ChannelType)
2850 {
2860 /******/
2870 /* For server and sender channels append data from */
2880 /* the file to the reference message and return it */
2890 /* to the caller. */
2900 /******/
2910 case DCMTCQ_MQCHT_SENDER:
2920 case DCMTCQ_MQCHT_SERVER:
2930 case DCMTCQ_MQCHT_CLUSSDR:
2940
2950 /******/
2960 /* Calculate space remaining at end of input */
2970 /* buffer and pointer to start of the space. */
2980 /* If the length of the agent buffer is greater */
2990 /* than the MaxMsgLength for the channel plus */
3000 /* the transmission header size, then use the */
3010 /* MaxMsgLength in the calculation. */
3020 /* If InputDataLength > 0 this means that only */
3030 /* that length is required. In this case, if the */
3040 /* length specified is less than the space in */
3050 /* the buffer then read just this length. */
3060 /******/
3070 MsgSpace = ((ParmList->tbuflist_adr->
3080             Buffinf_array[0].Buff_length >
3090             (MQLONG)(ParmList->mqcd_adr->MaxMsgLength
3100                     + sizeof(MQXQH)))
3110             ? (ParmList->mqcd_adr->MaxMsgLength)
3120             : (MQLONG)(ParmList->tbuflist_adr->
3130             Buffinf_array[0].Buff_length -
3140             sizeof(MQXQH)))
3150             - pMQRMH->StrucLength;
3160 pMsgData = (PMQCHAR)pMQRMH + pMQRMH->StrucLength;
3170 ReadLength = (InputDataLength > 0 &&
3180             InputDataLength < MsgSpace)

```

```

3190         ? InputDataLength
3200         : MsgSpace;
3210
3220     /******
3230     /* Open the file for reading */
3240     /******
3250     fd = fopen(SrcFileName, "rb");
3260
3270     if (fd == NULL)
3280     {
3290         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
3300         goto UOC_EXIT;
3310     }
3320
3330     /******
3340     /* Position the file to the specified offset. */
3350     /******
3360     rc = fseek(fd, FileDataOffset, SEEK_SET);
3370
3380     if (rc)
3390     {
3400         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
3410         goto UOC_EXIT;
3420     }
3430
3440     /******
3450     /* Read the data from the specified offset into */
3460     /* the buffer */
3470     /******
3480     bytesread = fread(pMsgData, 1, (size_t)ReadLength, fd);
3490
3500     if (ferror(fd))
3510     {
3520         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
3530         goto UOC_EXIT;
3540     }
3550     else
3560     if (feof(fd) ||
3570         (InputDataLength > 0 &&
3580          InputDataLength == (MQLONG)bytesread))
3590     {
3600         /******
3610         /* End of file reached or all the requested */
3620         /* data has been read. */
3630         /* Set MQRMHF_LAST flag. */
3640         /******
3650         pMQRMH->Flags |= MQRMHF_LAST;
3660     }
3670     else
3680     {
3690         /******
3700         /* May have read last byte of the file (EOF is */
3710         /* only set when attempting to read beyond the */
3720         /* last byte). */
3730         /* Try reading one more byte. */
3740         /******
3750
3760         fread(&dummy, 1, 1, fd);

```

```

3770
3780     if (ferror(fd))
3790     {
3800         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
3810         goto UOC_EXIT;
3820     }
3830     else
3840     if (feof(fd))
3850     {
3860         /******  

3870         /* End of file reached before buffer was      */  

3880         /* filled. Set MQRMHF_LAST flag.              */  

3890         /******  

3900         pMQRMH->Flags |= MQRMHF_LAST;
3910     }
3920     else
3930     {
3940         /******  

3950         /* End of file not reached.                    */  

3960         /******  

3970         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
3980         goto UOC_EXIT;
3990     }
4000 }
4010
4020 /******  

4030 /* Set the MQRMH.DataLogicalLength field to the      */  

4040 /* number of bytes read.                              */  

4050 /* Set DataLength to the increased size of the      */  

4060 /* message.                                           */  

4070 /* The size of the message has increased by the     */  

4080 /* number of bytes read from the file plus the      */  

4090 /* difference in the sizes of the input and         */  

4100 /* converted reference messages.                    */  

4110 /******  

4120 pMQRMH->DataLogicalLength = bytesread;
4130 ParmList->tbuflist_adr->  

4140     Buffinfn_array[0].Used_buff_length += bytesread;
4150     break;
4160
4170 /******  

4180 /* For requester and receiver channels, copy data   */  

4190 /* from the reference message into the file.         */  

4200 /* If this is the first part of the file then       */  

4210 /* (re)create the file first.                       */  

4220 /* If this is the last part of the file then        */  

4230 /* return the reference message to the caller,      */  

4240 /* otherwise discard it.                            */  

4250 /******  

4260 case DCMTCQ_MQCHT_REQUESTER:  

4270 case DCMTCQ_MQCHT_RECEIVER:  

4280 case DCMTCQ_MQCHT_CLUSRCVR:  

4290
4300 /******  

4310 /* Calculate address of data appended to MQRMH     */  

4320 /* header.                                          */  

4330 /* Calculate length of data to be written to the   */  

4340 /* file.                                           */

```

```

4350      /*****/
4360      pMsgData = (PMQCHAR)(pMQXQH + 1) + pQRMH->StrucLength;
4370      WriteLength = ParmList->tbuflist_adr->
4380          Buffinf_array[0].Used_buff_length -
4390          sizeof(MQXQH) - pQRMH->StrucLength;
4400
4410      /*****/
4420      /* If the data offset within the file is zero */
4430      /* then the file is created for writing. If it */
4440      /* already exists, its contents are destroyed. */
4450      /* Otherwise it is opened for appending. */
4460      /*****/
4470      strcpy(mode,
4480          (FileDataOffset == 0) ? "wb" : "ab");
4490
4500      fd = fopen(DestFileName, mode);
4510
4520      if (fd == NULL)
4530      {
4540          FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
4550          goto UOC_EXIT;
4560      }
4570
4580      /*****/
4590      /* If the data offset within the file is not */
4600      /* zero, check the current size of the file. */
4610      /* If it matches the data offset, then write the */
4620      /* data to the file. */
4630      /* Otherwise ignore the message. */
4640      /* If the data offset is zero then write the */
4650      /* data to the file. */
4660      /*****/
4670      if (FileDataOffset != 0)
4680      {
4690
4700          rc = fseek(fd, 0, SEEK_END);
4710
4720          if (rc)
4730          {
4740              FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
4750              goto UOC_EXIT;
4760          }
4770
4780          position = ftell(fd);
4790
4800          if (position < 0)
4810          {
4820              FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
4830              goto UOC_EXIT;
4840          }
4850
4860          if (position > FileDataOffset)
4870          {
4880              /*****/
4890              /* File is bigger than the offset of this */
4900              /* segment. */
4910              /* This can happen during recovery when a */
4920              /* channel terminates half way through a */

```

```

4930      /* file transfer.                                */
4940      /* Our only response is to not write the         */
4950      /* data                                           */
4960      /*******/
4970      WriteData = 0;
4980  }
4990
5000  if (position < FileDataOffset)
5010  {
5020      /*******/
5030      /* File is smaller than the offset of this      */
5040      /* segment. We must have lost a segment,       */
5050      /* alert the user to the error                  */
5060      /*******/
5070      FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
5080      goto UOC_EXIT;
5090  }
5100 }
5110
5120 /*******/
5130 /* Write the data to the specified offset in the    */
5140 /* file.                                           */
5150 /* The amount of data to be written is the         */
5160 /* length of the message minus the transmission    */
5170 /* and reference message headers                   */
5180 /*******/
5190 if (WriteData && (WriteLength > 0))
5200 {
5210     itemswritten = fwrite(pMsgData, (size_t)WriteLength, 1, fd);
5220
5230     if (!itemswritten)
5240     {
5250         FeedbackCode = MQFB_STOPPED_BY_MSG_EXIT;
5260         goto UOC_EXIT;
5270     }
5280 }
5290
5300 /*******/
5310 /* ExitResponse is left at DCMTCCQ_MQXCC_OK which  */
5320 /* means that the reference message (minus the     */
5330 /* appended data) will be put to the target        */
5340 /* queue.                                           */
5350 /*******/
5360 ParmList->tbuflist_adr->
5370 Buffinf_array[0].Used_buff_length =
5380     sizeof(MQXQH) + pMQRMH->StrucLength;
5390 break;
5400
5410 /*******/
5420 /* For other channel types, ignore the message.    */
5430 /*******/
5440 default:
5450     goto UOC_EXIT;
5460 }
5470
5480 UOC_EXIT:
5490
5500 /*******/

```

```

5510  /* Close the file if necessary.                */
5520  /*****
5530  if (fd)
5540  {
5550      fclose(fd);
5560  }
5570
5580  /*****
5590  /* If an error has occurred, tell the caller to DLQ */
5600  /* the message.                                    */
5610  /* Set the feedback code to be returned in an      */
5620  /* exception report (if requested).                */
5630  /*****
5640  if (FeedbackCode)
5650  {
5660      ParmList->mqcxp_adr->ExitResponse =
5670          DCMTQC_MQXCC_SUPPRESS_FUNCTION;
5680      ParmList->mqcxp_adr->Feedback    = FeedbackCode;
5690  }
5700
5710  return(0) ;
5720  }

```

4

準備作業

TP1/Message Queue を使用するには、あらかじめ環境を定義しておく必要があります。TP1/Message Queue の定義には、MQA, MQT, および MQC の三つがあります。この章では、定義の作成方法、定義の詳細、および定義例について説明します。

TP1/Message Queue の環境作成手順

TP1/Message Queue を使用するための環境作成手順について、次の図に示します。

図 4-1 TP1/Message Queue の環境作成手順



1. OpenTP1 をインストールし、OpenTP1 管理者を設定します。

詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

キューマネージャクラスタを構成する場合は、あわせて IBM MQ でフルリポジトリを構築します。詳細については、IBM MQ のマニュアルを参照してください。

2. TP1/Message Queue で使用するメッセージキューの構成、メッセージ容量、および転送先チャネル構成などを考慮し、TP1/Message Queue の定義ファイルを作成してください。

詳細については、この章の「定義情報の作成手順」を参照してください。

キューマネージャクラスタを構成する場合は、あわせてクラスタセンダプロセスおよびクラスタレシーバプロセスについて定義ファイルを作成してください。詳細については、「[2.8.9 クラスタ環境の通信構成](#)」を参照してください。

3. TP1/Server Base の各定義を TP1/Message Queue 起動用に追加および変更してください。

詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

キューマネージャクラスタを構成する場合は、あわせてリポジトリ管理サーバユーザサービス定義を定義し、リポジトリ管理サーバを作成してください。詳細については、「[2.8.11\(3\) リポジトリ管理サーバの作成](#)」および「[2.8.11\(4\) リポジトリ管理サーバの環境変数](#)」を参照してください。

4. TP1/Server Base の提供するコマンドで OpenTP1 ファイルシステムをディスク上に作成してください。

詳細については、マニュアル「[OpenTP1 運用と操作](#)」を参照してください。

5. OpenTP1 ファイルシステム上に TP1/Message Queue で使用するキューファイルを作成してください。

詳細については、6章の「[mqainit \(キューファイルの割り当て\)](#)」を参照してください。

6. キューファイル上に TP1/Message Queue で使用するキューを作成してください。

詳細については、6章の「[mqamkque \(キューの作成\)](#)」を参照してください。

キューマネージャクラスタを構成する場合は、あわせてクラスタ環境のシステムキューを作成してください。詳細については、「[2.8.11\(6\) クラスタ環境のシステムキュー](#)」を参照してください。

7. 転送先のチャンネル構成に合わせて作成した MQT 通信構成定義から MQT 定義オブジェクトファイルを作成してください。

詳細については、この章の「[MQT 定義オブジェクトファイルの作成手順](#)」を参照してください。

8. TP1/Server Base を開始してください。

詳細については、マニュアル「[OpenTP1 運用と操作](#)」を参照してください。

キューマネージャクラスタを構成する場合は、あわせてリポジトリ管理サーバを開始します。詳細については、「[2.8.11\(5\) リポジトリ管理サーバの開始方法](#)」を参照してください。

定義情報の作成手順

TP1/Message Queue の定義は、MQA サービス定義、MQT サービス定義および MQT 通信構成定義に分けられます。

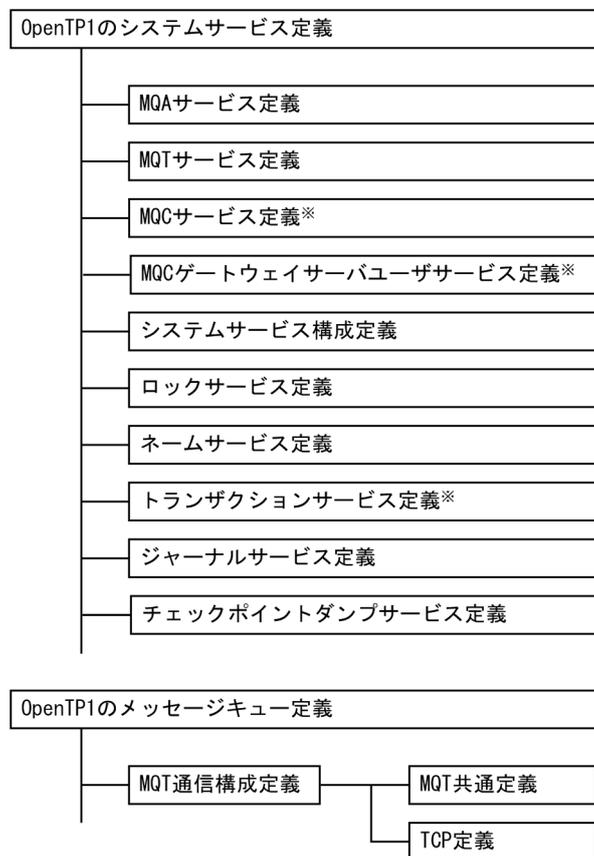
MQA サービス定義および MQT サービス定義は、OpenTP1 システム定義の中の、システムサービス定義の一部に当たります。

MQT 通信構成定義は、OpenTP1 システム定義の中の、メッセージキュー定義に当たります。

また、TP1/Message Queue を使用するために、OpenTP1 のシステムサービス定義のシステムサービス構成定義、ロックサービス定義、ネームサービス定義、ジャーナルサービス定義、およびチェックポイントダンプサービス定義を使用します。

TP1/Message Queue の定義の体系について、次の図に示します。

図 4-2 TP1/Message Queue の定義の体系



注※

関連製品である TP1/Message Queue Access の使用時に作成する定義です。OpenTP1 システム定義の中の、システムサービス定義の一部に当たります。TP1/Message Queue Access を使用しない場合は、作成不要です。

TP1/Message Queue の定義情報の作成手順

MQA サービス定義を作成する場合は、テキストエディタを使用して、定義ファイルを作成します。

MQT 通信構成定義を作成する場合は、オンライン開始前にオブジェクトファイルを作成しておく必要があります。オブジェクトファイルは、テキストエディタによって作成したソースファイルを、ユーティリティで変換して作成します。MQT 通信構成定義の中の MQT 共通定義および TCP 定義は、それぞれ個別のソースファイルに作成し、個別のオブジェクトファイルを作成します。

注意事項

- OpenTP1 ファイル（ジャーナルファイル、チェックポイントダンプファイル、ステータスファイル、キューファイル）および共用メモリの見積もりを行い、その結果に応じて定義情報を作成してください。見積もり値よりも定義で指定した値のほうが少ない場合、各種資源が不足し、オンライン中にエラーが発生したり OpenTP1 が強制停止したりします。
- MQT 通信構成定義は、オブジェクトファイルでのバージョン互換性を保証しません。そのため、バージョン、または OS が異なる環境で作成した MQT 定義オブジェクトファイルを使用した場合の動作は保証できません。

定義の規則

定義を指定する場合の規則を説明します。

MQA サービス定義

MQA サービス定義には、set 形式およびコマンド形式があります。どちらの定義形式も同一定義ファイルに格納します。

同一定義内では、set 形式を定義してからコマンド形式を定義してください。

set 形式では OpenTP1 システム開始時に値が決定されます。

コマンド形式では OpenTP1 システム開始時に値が決定されても、オンライン中に運用コマンドを入力すると、OpenTP1 システム開始時に決定された値を変更できます。

MQT サービス定義

MQT サービス定義には、set 形式があります。

set 形式では OpenTP1 システム開始時に値が決定されます。

MQC サービス定義

関連製品である TP1/Message Queue Access の使用時に作成する定義です。

MQC サービス定義には、コマンド形式があります。

コマンド形式では OpenTP1 システム開始時に値が決定されても、オンライン中に運用コマンドを入力すると、OpenTP1 システム開始時に決定された値を変更できます。

MQC ゲートウェイサーバユーザサービス定義

関連製品である TP1/Message Queue Access の使用時に作成する定義です。

MQC ゲートウェイサーバユーザサービス定義には、set 形式があります。

set 形式では OpenTP1 システム開始時に値が決定されます。

システム環境定義 (env)

default_value_option=システム定義のデフォルト値選択オプション ~ 〈符号なし整数〉 ((0~1)) 《1》

表 4-1、および表 4-2 に示す TP1/Message Queue のオペランドについて、07-50 より前のデフォルト値で動作させるか、07-50 以降のデフォルト値で動作させるかを指定します。

- 07-50 より前のデフォルト値で動作させる場合
default_value_option オペランドを 0 で指定します。

上記の指定によって、デフォルト値が変更される TP1/Message Queue のオペランドを次の表に示します。

表 4-1 デフォルト値が変更される TP1/Message Queue のオペランド (07-50 より前のデフォルト値で動作させる場合)

オペランド		変更後のデフォルト値
mqttcp 定義コマンド (TCP 構成定義)	-t disk	no
	-t bufcnt	100
mqttcps 定義コマンド (クラスタセンダ TCP 構成定義)	-t disk	no
	-t bufcnt	100
mqttcpcr 定義コマンド (クラスタレシーバ TCP 構成定義)	-t disk	no
	-t bufcnt	100

- 07-50 以降のデフォルト値で動作させる場合
default_value_option オペランドを 1 で指定します。

上記の指定によって、デフォルト値が変更される TP1/Message Queue のオペランドを次の表に示します。

表 4-2 デフォルト値が変更される TP1/Message Queue のオペランド (07-50 以降のデフォルト値で動作させる場合)

オペランド		変更後のデフォルト値
mqttcp 定義コマンド (TCP 構成定義)	-t disk	yes
	-t bufcnt	2500

オペランド		変更後のデフォルト値
mqttcps 定義コマンド (クラスタセンダ TCP 構成定義)	-t disk	yes
	-t bufcnt	2500
mqttpcr 定義コマンド (クラスタレシーバ TCP 構成定義)	-t disk	yes
	-t bufcnt	2500

システムサービス構成定義

TP1/Message Queue を使用するためには、mqa_conf オペランドに Y を指定してください。

ロックサービス定義

TP1/Message Queue を使用するためには、lck_limit_formqa オペランドに、MQA サーバの最大同時排他要求数を指定してください。ここで指定する値は、同一ノード上にある MQA サーバから発生する排他要求数の最大値です。

次の計算式で最大同時排他要求数を見積もってください。

$$A+S+T+E+1 \times 1 + R \times 2$$

A:

MQOO_INPUT_AS_Q_DEF, MQOO_INPUT_SHARED または MQOO_INPUT_EXCLUSIVE を指定した MQOPEN 発行数 (MQCLOSE 未発行) (mqcgwp クライアントアプリケーションを含みます)

S:

同時に確立する送信側のチャンネル数 (自動定義クラスタセンダチャンネル mqttcps を含みます)

T:

MQA サービス定義のプロトコル名称が mqttcp の mqamqtnam コマンド定義数

E:

- メッセージ保持時間機能を使用する場合、1 (set mqa_expiry=Y の場合)
- メッセージ保持時間機能を使用しない場合、0 (set mqa_expiry=N, または省略した場合)

注※1

mqadelmsg コマンドおよび、mqarm コマンドによる MQOPEN 発行数 (二つのコマンドを同時に実行した際は、動作するのはどちらか一つとなり、もう一方は待たされます)。

注※2

クラスタ機能を使用する場合、次の計算式の結果を R の値として加えてください。

$$R=5+ (4 \times \text{mqrspプロセス数})$$

なお、MQA サーバでの排他には、lck_wait_timeout オペランドを使用しません。このため、lck_wait_timeout オペランドの指定値によるエラーリターンは発生しません。

ネームサービス定義

TP1/Message Queue を使用するためには、name_total_size オペランドに MQA サーバ分として、次の式で見積もった値を加算してください。

(4+MQAサービス定義のmqa_ioproc_numオペランドの指定値
+mqamqtnam定義コマンドの定義数) ×528 (単位: バイト)

クラスタ機能を使用する場合は、OpenTP1 の name_total_size オペランドの指定値の見積もり式に含まれる「SPP 数」に 1 を加算してください。見積もり式については、マニュアル「OpenTP1 システム定義」の TP1/Server Base 用の共用メモリの見積もり式の、ネームサーバの説明を参照してください。

トランザクションサービス定義

関連製品である TP1/Message Queue Access の使用時に作成する定義です。

MQC サーバ機能で、TP1/Message Queue Access からグローバルトランザクションを使用するかどうかを指定します。

TP1/Message Queue Access からグローバルトランザクションを使用する場合は、trn_crm_use オペランドに Y を指定してください。Y を指定しないで TP1/Message Queue Access からグローバルトランザクションを使用した場合の処理の流れを次に示します。

1. TP1/Message Queue Access からグローバルトランザクションを開始します。
2. MQPMO_MQSYNCPOINT, または MQGMO_SYNCPOINT を指定して、MQPUT 命令, MQPUT1 命令, および MQGET 命令を発行します。
3. 理由コード MQRC_UNEXPECTED_ERROR が TP1/Message Queue Access へ返却されます。また、MQC サーバ側に KFCA00910-E メッセージが出力されます。

ジャーナルサービス定義

MQA サービス定義に mqamqtnam 定義コマンドで、MQT デーモンの構成を定義した場合、MQA サーバはチェックポイントダンプを取得します。MQA サーバ用のチェックポイントダンプサービス定義のファイル名を jnldfs 定義コマンドで指定してください。

チェックポイントダンプサービス定義

MQA サービス定義に mqamqtnam 定義コマンドで、MQT デーモンの構成を定義した場合、MQA サーバはチェックポイントダンプを取得します。MQA サーバ用のチェックポイントダンプサービス定義を作成してください。jnl_objservername オペランドには、対象となるシステムサービス名として_mqa を指定してください。

mqamqtnam 定義コマンドを指定しないで MQA サーバ用のチェックポイントダンプサービス定義を作成した場合、OpenTP1 起動時に、「KFCA01125-E 理由コード=908-0」が出力されて起動に失敗します。

MQT 通信構成定義

MQT 通信構成定義では、各定義内での記述順序が決まっています。

MQT 通信構成定義には、コマンド形式だけがあります。

旧製品の MQA サービス定義を使用する場合の注意事項

TP1/Message Queue 05-15 以降および 05-14 以降では、MQA サービス定義の指定を省略した場合の解釈値を変更しました。TP1/Message Queue 05-14 以前または 05-13 以前の定義を使用する場合、動作が異なることがありますので注意してください。変更内容をそれぞれ表 4-3 および表 4-4 に示します。

表 4-3 TP1/Message Queue 05-15 以降で変更した MQA サービス定義の省略時の解釈値

定義コマンド名およびオプション名	TP1/Message Queue 05-14 以前で省略した場合の解釈値	TP1/Message Queue 05-15 以降で省略した場合の解釈値
mqaquemgr -t	150	999999999

トリガ起動によるチャンネル確立を行う環境で次に示す状態の場合、運用に合ったトリガ間隔を指定するか、または 05-14 以前でトリガ間隔を省略した場合の解釈値（150 ミリ秒）を指定してください。

- 送信チャンネルが何らかの理由で転送キューにメッセージを滞留した状態で「チャンネル使用不可」になる。
- mqtstacha コマンドの -a オプションで送信チャンネルの状態を「チャンネル停止」に変更する。
トリガ間隔を指定していない場合は、チャンネルの確立が前回のトリガメッセージの発生時間から 999999999 ミリ秒後になります。

表 4-4 TP1/Message Queue 05-14 以降で変更した MQA サービス定義の省略時の解釈値

定義コマンド名およびオプション名	TP1/Message Queue 05-13 以前で省略した場合の解釈値	TP1/Message Queue 05-14 以降で省略した場合の解釈値
mqaquegrp -m	100	0

TP1/Message Queue 07-50 以降では、MQT 通信構成定義に指定する TCP 定義の省略値の解釈を変更しました。TP1/Message Queue 07-01 以前の定義を使用する場合、動作が異なることがありますので注意してください。変更内容を次の表に示します。

表 4-5 TP1/Message Queue 07-50 以降で変更した MQT 通信構成定義の省略時の解釈値

定義コマンド名およびオプション名	TP1/Message Queue 07-01 以前で省略した場合の解釈値	TP1/Message Queue 07-50 以降で省略した場合の解釈値
mqttcp -t disk	no	yes
mqttcp -t bufcnt	100	2500
mqttcpcs -t disk	no	yes
mqttcpcs -t bufcnt	100	2500
mqttcpcr -t disk	no	yes
mqttcpcr -t bufcnt	100	2500

定義の記述形式

定義の記述形式について説明します。

set 形式

set 形式ではオペランドに値を指定します。

形式

```
set オペランド = 値
```

コマンド形式

コマンド形式では、次に示す形式に従ってオプションおよび引数を指定します。

形式

```
コマンド名 オプション "オペランド" コマンド引数
```

用語の定義

● コマンド名

実行できるコマンドが登録されているファイル名です。

● オプション

マイナス記号 (-) で始まる文字列です。

引数を取らない形式 1 と引数を必要とする形式 2 の二つがあります。

- 形式 1
-オプションフラグ
- 形式 2
-オプションフラグ フラグ引数

オプションフラグは、マイナス記号 (-) に続く 1 文字の英数字です。英大文字と英小文字とは区別しません。フラグ引数は、オプションフラグに対する引数です。

● オペランド

オプションの次の、「=値」が付くものをオペランドといいます。

一つのオプションの中に複数のオペランドがある場合があります。一つのオプションの中のオペランドは、最初と最後を引用符 (") で囲みます。

● コマンド引数

マイナス記号 (-) 以外で始まる引数でコマンドの操作対象です。

コメント

コメントを記入する場合は、コメントの先頭に「#」を記述してください。行の先頭に「#」を記述すると1行全体がコメント扱いになります。

形式1 定義の記述 #コメント
形式2 #コメント

制限事項

- OpenTP1 のシステムサービス定義の場合

MQA サービス定義など、OpenTP1 のシステムサービス定義に属する定義を記述する場合、1行の長さなどの記述形式はOpenTP1システムの規定に従います。記述形式の制限事項については、マニュアル「OpenTP1システム定義」を参照してください。

- MQT 通信構成定義の場合

MQT 通信構成定義の1行の長さは最大512文字です。512文字を超える指定の場合または定義を読みやすくするために必要な場合は、複数行に分けて記述してください。この場合、行の末尾に継続符号「¥」を記述します。継続符号「¥」のあとには何も記述しないでください。なお、アポストロフィ（'）で囲まれた文字列は、複数行に分けて記述できません。

コメントを記述した行に継続行を続ける場合は、コメントを「#」で囲んだあと、その行の末尾に継続符号「¥」を記述してください。

MQA サービス定義

TP1/Message Queue の定義のうち、MQA サービス定義について説明します。

概要

MQA サービス定義は、OpenTP1 のシステムサービス定義の一つです。

MQA サービス定義の定義ファイルは、OS のテキストエディタを使用して、テキストファイルとして作成します。MQA サービス定義の定義ファイル名（完全パス名）を次に示します。

```
$DCCONFPATH/mqa
```

機能

キューを管理するための実行環境を定義します。

種類

MQA サービス定義の種類について、次の表に示します。

表 4-6 MQA サービス定義の種類

形式	オプション	定義内容	指定値
set watch_time	なし	最大応答待ち時間	<符号なし整数> ((0~65535)) (単位:秒)
set mqa_channel_inf_file_name_a	なし	MQT サーバ用チャンネル管理情報格納 A 系ファイルパス名	<1~80 バイトの文字列>または<1~63 バイトの文字列>
set mqa_channel_inf_file_name_b	なし	MQT サーバ用チャンネル管理情報格納 B 系ファイルパス名	<1~80 バイトの文字列>または<1~63 バイトの文字列>
set mqa_channel_inf_file	なし	MQT サーバ用チャンネル管理情報格納ファイル入出力形式	《normal》 tp1file
set mqa_max_quenum	なし	オンライン中に使用するキューの最大数	<符号なし整数> ((2~20480)) 《4096》
set mqa_max_grpnum	なし	オンライン中に使用するキューファイルグループの最大数	<符号なし整数> ((1~4095)) 《16》
set mqa_max_msg_recnum	なし	1 メッセージの最大レコード数	<符号なし整数> ((1~65535)) 《32》

形式	オプション	定義内容	指定値
set mqa_max_filnum	なし	オンライン中に使用するキューファイルの最大数	<符号なし整数> ((1~4095))《16》
set mqa_ioproc_num	なし	入出力プロセス起動数	<符号なし整数> ((1~64))《1》
set mqa_expiry	なし	メッセージ保持時間機能を使用するかどうか	Y 《N》
set mqa_maxque_expiry	なし	メッセージ保持時間対象サーチキュー数	<符号なし整数> ((1~20480))《3》
set mqa_maxmsg_expiry	なし	メッセージ保持時間対象削除メッセージ数	<符号なし整数> ((1~65535))《3》
set mqa_quefil_inf	なし	キュー／キューファイルのしきい値, および回復値に達した場合に, メッセージを出力するかどうか	Y 《N》
set mqa_mqr_conf	なし	クラスタ機能使用の要否	Y 《N》
set mqa_local_ccsid	なし	キューマネージャの文字セット識別子	<符号なし整数> ((1~65535))《850》
set mqa_jnl_conf	なし	ジャーナルへのメッセージデータ取得の要否	Y 《N》
set mqa_mqc_conf	なし	MQC サーバ機能の要否	Y 《N》
set mqa_mqc_clientnum	なし	オンライン中に同時に処理できるクライアントの最大数	<符号なし整数> ((1~256))《16》
set mqa_mqc_trnnum	なし	オンライン中に同時に処理できるトランザクションの最大数	<符号なし整数> ((0~256))《0》
set mqa_mqo_conf	なし	Message Queue - Operation - Agent の要否	Y 《N》
set mqa_mqo_queuefile_alarm_level	なし	キューファイル登録メッセージサイズしきい値	<符号なし整数> ((10~100))《80》 (単位: %)
set mqa_mqo_queuefile_recover_level	なし	キューファイル登録メッセージサイズ回復値	<符号なし整数> ((10~100))《70》 (単位: %)
set mqa_mqo_queue_alarm_level	なし	キュー登録メッセージ数しきい値	<符号なし整数> ((10~100))《80》 (単位: %)
set mqa_mqo_queue_recover_level	なし	キュー登録メッセージ数回復値	<符号なし整数> ((10~100))《70》

形式	オプション	定義内容	指定値
set mqa_mqo_queue_recover_level	なし	キュー登録メッセージ数回復値	(単位: %)
set mqa_msg_stay_time	なし	メッセージ滞留時間	<符号なし整数> ((0~2147483647)) 《0》 (単位: 秒)
set max_socket_descriptors	なし	ソケット用ファイル記述子の最大数	<符号なし整数> Linux 以外の場合((32~2032)) 《514》 Linux の場合((32~1008)) 《514》
set mqa_pool_attri	なし	MQA サービス専用共用メモリをメモリプールに固定するかどうか	《free》 fixed
set mqa_extend_rmshm_size	なし	MQA サービス専用共用メモリの拡張領域サイズ	<符号なし整数> ((0~2000000000)) 《0》 (単位: バイト)
set mqa_getwait_timeout	なし	MQGMO_WAIT 指定時のMQGET 命令の待ち合わせ最大時間精度	《low》 high
set mqa_filerr_continue	なし	入出力エラーが発生した場合にオンラインを継続させるかどうか	《Y》 N
set mqa_prf_trace_level	なし	性能検証用トレース情報取得レベル	< 16 進数字> ((00000000 00000001 00000010)) 《00000000》
mqaquegrp (キューファイルグループの構成定義) 指定数: 1 以上*	-g	キューファイルグループ名	< 1~31 文字の識別子>
	-b	バックアップキューファイル名	< 1~59 文字の文字列>
	-n	入出力バッファ数	<符号なし整数> ((0~1048560)) 《640》
	-m	メッセージエントリ確保率	<符号なし整数> ((0~100)) 《0》
	-k	キューファイルグループ監視の要否	Y 《N》
	-u	キューファイルグループ登録メッセージサイズしきい値	<符号なし整数> ((10~100)) 《80》 (単位: %)
	-l	キューファイルグループ登録メッセージサイズ回復値	<符号なし整数> ((10~100)) 《70》 (単位: %)
	-s	MQA サービス専用共用メモリ確保要求者識別子	< 16 進数字> ((00~ff))

形式	オプション	定義内容	指定値
mqaquegrp (キューファイルグループの構成定義) 指定数：1 以上*	なし	キューファイル名	< 1~59 文字の文字列 >
mqaqueatl (モデルキューの属性定義) 指定数：0 以上*	-q	キュー記述子	< 1~64 文字の文字列 >
	-g	取り出し許可	《allowed》 inhibited
	-p	登録許可	《allowed》 inhibited
	-y	省略時のメッセージ優先度	< 符号なし整数 > ((0~9)) 《0》
	-e	省略時のメッセージ永続性	《persistent》 notpersistent
	-a	プロセス名	< 1~48 文字の MQ 文字列 >
	-d	最大メッセージ登録数	< 符号なし整数 > ((0~2147483647)) 《128》
	-l	最大メッセージ長	< 符号なし整数 > ((0~128000000)) 《4096000》
	-s	共用性	shareble 《notshareble》
	-o	省略時の取り出しオプション	《exclusive》 shared
	-m	メッセージ配布順序	priority 《fifo》
	-r	キュー保持時間	< 符号なし整数 > ((0~65535)) 《8》 (単位：時間)
	-k	定義タイプ	《permanent》 temporary
	-u	使用種別	《normal》 transmission
	-i	イニシエーションキュー名	< 1~48 文字の MQ 文字列 >
	-c	トリガ制御	off 《on》
	-t	トリガタイプ	《every》 first depth none
	-b	トリガのためのメッセージ優先度の下限値	< 符号なし整数 > ((0~9)) 《0》
-n	トリガのためのメッセージ登録数の下限値	< 符号なし整数 > ((1~65535)) 《1》	
-f	トリガデータ	< 1~64 文字の文字列 >	
-j	配布リストのサポート	support 《nosupport》	

形式	オプション	定義内容	指定値
mqaqueatl (モデルキューの属性定義) 指定数：0以上*	-w	デフォルトのバインディング	《fixed》 not_fixed
	-v	メッセージ滞留時間監視の要否	《off》 on
	なし	モデルキューの定義名	< 1~48 文字の MQ 文字列 >
	なし	キューファイル名	< 1~59 文字の文字列 >
mqaremque (リモートキューのローカル定義) 指定数：0以上*	-q	キュー記述子	< 1~64 文字の文字列 >
	-p	登録許可	《allowed》 inhibited
	-y	省略時のメッセージ優先度	< 符号なし整数 > ((0~9)) 《0》
	-e	省略時のメッセージ永続性	《persistent》 notpersistent
	-r	リモートキュー名	< 1~48 文字の MQ 文字列 >
	-w	デフォルトのバインディング	《fixed》 not_fixed
	-c	クラスタ内共用クラスタ名称	< 1~48 文字の MQ 文字列 >
	-m	リモートキューマネージャ名	< 1~48 文字の MQ 文字列 >
	-x	転送キュー名	< 1~48 文字の MQ 文字列 >
	なし	リモートキューのローカル定義名	< 1~48 文字の MQ 文字列 >
mqaalsque (別名キューの属性定義) 指定数：0以上*	-q	キュー記述子	< 1~64 文字の文字列 >
	-g	取り出し許可	《allowed》 inhibited
	-p	登録許可	《allowed》 inhibited
	-y	省略時のメッセージ優先度	< 符号なし整数 > ((0~9)) 《0》
	-e	省略時のメッセージ永続性	《persistent》 notpersistent
	-w	デフォルトのバインディング	《fixed》 not_fixed
	-c	クラスタ内共用クラスタ名称	< 1~48 文字の MQ 文字列 >
	なし	別名キュー名	< 1~48 文字の MQ 文字列 >
	なし	ベースキュー名	< 1~48 文字の MQ 文字列 >
mqaprcdef (プロセス定義) 指定数：0以上	-p	プロセス記述子	< 1~64 文字の文字列 >
	-t	アプリケーションタイプ	< 符号なし整数 > ((15, または 65536~99999999)) 《15》
	-i	アプリケーション記述子	< 1~256 文字の文字列 >
	-e	環境データ	< 1~128 文字の文字列 >

形式	オプション	定義内容	指定値
mqaprcdef (プロセス定義) 指定数：0 以上	-u	ユーザデータ	< 1~128 文字の文字列 >
	なし	プロセス名	< 1~48 文字の MQ 文字列 >
mqaquemgr (キューマネージャ定義) 指定数：1	-n	キューマネージャ名	< 1~48 文字の MQ 文字列 >
	-q	キューマネージャ記述子	< 1~64 文字の文字列 >
	-t	トリガ間隔	< 符号なし整数 > ((0~999999999)) 《999999999》 (単位：ミリ秒)
	-d	デッドレターキュー名	< 1~48 文字の MQ 文字列 >
	-p	最大メッセージ優先度	< 符号なし整数 > ((0~9)) 《9》
	-x	省略時の転送キュー名	< 1~48 文字の MQ 文字列 >
	-h	最大ハンドル数	< 符号なし整数 > ((1~65535)) 《10》
	-l	最大メッセージ長	< 符号なし整数 > ((1~128000000)) 《4096000》
	-s	トランザクション有効性	《sync》 nosync
	-j	配布リストのサポート	support 《nosupport》
mqamqtnam (MQT デーモン構成定義) 指定数：0 以上	-b	総バッチメッセージ数	< 符号なし整数 > ((1~2549745)) 《255》
	なし	MQT サーバ名または MQC リスナサーバ名	< _ (アンダスコア) で始まる 2~8 文字の識別子 >
	なし	プロトコル名称	< 1~14 文字の識別子 >
	なし	MCA 数	< 符号なし整数 > ((1~255))

注※

これらの定義コマンドの指定数 (mqaquegrp 定義コマンドの場合、キューグループ内のキューの数) の合計は、TP1/Message Queue の動的キューの数と合わせて、20480 以内にしてください。

注意事項

- 各定義コマンドの指定数は、MQA サーバを開始するために必要な値を表します。
- MQA サービス定義に指定できる MQC リスナサーバの構成定義 (mqamqtnam 定義コマンド) の数は一つだけです。複数を指定した場合は、MQA サービス開始時に KFCA31183-E メッセージを出力して異常終了します。

set 形式の MQA サービス定義

set 形式の MQA サービス定義について説明します。

形式

```
[set watch_time = 最大応答待ち時間]
[set mqa_channel_inf_file_name_a = MQTサーバ用チャンネル
                                管理情報格納A系ファイルパス名]
[set mqa_channel_inf_file_name_b = MQTサーバ用チャンネル
                                管理情報格納B系ファイルパス名]
[set mqa_channel_inf_file = normal | tp1file]
[set mqa_max_quenum = オンライン中に使用するキューの最大数]
[set mqa_max_grpnum = オンライン中に使用する
                    キューファイルグループの最大数]
[set mqa_max_msg_recnum = 1メッセージの最大レコード数]
[set mqa_max_filnum = オンライン中に使用するキューファイルの最大数]
[set mqa_ioproc_num = 入出力プロセス起動数]
[set mqa_expiry = Y | N]
[set mqa_maxque_expiry = メッセージ保持時間対象サーチキュー数]
[set mqa_maxmsg_expiry = メッセージ保持時間対象削除メッセージ数]
[set mqa_quefil_inf = Y | N]
[set mqa_mqr_conf = Y | N]
[set mqa_local_ccsid = キューマネージャの文字セット識別子]
[set mqa_jnl_conf = Y | N]
[set mqa_mqc_conf = Y | N]
[set mqa_mqc_clientnum = オンライン中に同時に処理できる
                        クライアントの最大数]
[set mqa_mqc_trnnum = オンライン中に同時に実行できる
                     トランザクションの最大数]
[set mqa_mqo_conf = Y | N]
[set mqa_mqo_queuefile_alarm_level = キューファイル登録メッセージ
                                    サイズしきい値]
[set mqa_mqo_queuefile_recover_level = キューファイル登録メッセージ
                                       サイズ回復値]
[set mqa_mqo_queue_alarm_level = キュー登録メッセージ数しきい値]
[set mqa_mqo_queue_recover_level = キュー登録メッセージ数回復値]
[set mqa_msg_stay_time = メッセージ滞留時間]
[set max_socket_descriptors = ソケット用ファイル記述子の最大数]
[set mqa_pool_attri = free | fixed]
[set mqa_extend_rmshm_size = MQAサービス専用共用メモリの拡張領域サイズ]
[set mqa_getwait_timeout = low | high]
[set mqa_filerr_continue = Y | N]
[set mqa_prf_trace_level = 性能検証用トレース情報取得レベル]
```

オペランド

● watch_time=最大応答待ち時間

～<符号なし整数>((0~65535)) (単位: 秒)

RPC によってプロセス間で通信する場合に、サービス要求を送ってからサービスの応答が返るまでの待ち時間の最大値を指定します。

指定時間を過ぎても応答がない場合は、RPC は送信タイムアウトと判断してエラーにします。

0 を指定した場合は、応答を受信するまで無限に待ち続けます。

省略した場合は、OpenTP1 のシステム共通定義の watch_time オペランドの値が設定されます。

● mqa_channel_inf_file_name_a=MQT サーバ用チャンネル管理情報格納 A 系ファイルパス名

～< 1～80 バイトの文字列>または～< 1～63 バイトの文字列>

ユーザ情報を格納する A 系ファイルパス名を指定します。

MQT サーバのチャンネルについての情報を格納するファイルです。このファイルは、MQT サーバサービス開始時と MQT サーバサービス終了時に作成、更新される情報で、次のオンラインに引き継ぎます。

MQT サーバ用チャンネル管理情報格納ファイルを通常ファイルに作成する場合は、ファイルパス名は 80 バイト以内にしてください。OpenTP1 ファイルシステム上のファイルに作成する場合は、ファイルパス名は 63 バイト以内、ファイル名は OpenTP1 ファイル名にしてください。

ファイルパス名およびファイル名にスペースを指定しないでください。また、すべてのファイルパス名の"/"個所に"/"を追加したファイルのパス名を指定しないでください。指定した場合、動作は保証されません。

■ 注意事項

MQT サーバ用チャンネル管理情報格納ファイルを使用しない場合はチャンネルの運用に注意が必要です。

チャンネルを引き継がない場合の属性値および注意点について、「[5.3 開始時のチャンネル情報引き継ぎ](#)」を参照してください。

● mqa_channel_inf_file_name_b=MQT サーバ用チャンネル管理情報格納 B 系ファイルパス名

～< 1～80 バイトの文字列>または～< 1～63 バイトの文字列>

ユーザ情報を格納する B 系ファイルパス名を指定します。

MQT サーバのチャンネルについての情報を格納するファイルです。このファイルは、MQT サーバサービス開始時と MQT サーバサービス終了時に作成、更新される情報で、次のオンラインに引き継ぎます。

MQT サーバ用チャンネル管理情報格納ファイルを通常ファイルに作成する場合は、ファイルパス名は 80 バイト以内にしてください。OpenTP1 ファイルシステム上のファイルに作成する場合は、ファイルパス名は 63 バイト以内、ファイル名は OpenTP1 ファイル名にしてください。

ファイルパス名およびファイル名にスペースを指定しないでください。また、すべてのファイルパス名の"/"個所に"/"を追加したファイルのパス名を指定しないでください。指定した場合、動作は保証されません。

注意事項

MQT サーバ用チャンネル管理情報格納ファイルを使用しない場合はチャンネルの運用に注意が必要です。

チャンネルを引き継がない場合の属性値および注意点について、「[5.3 開始時のチャンネル情報引き継ぎ](#)」を参照してください。

● mqa_channel_inf_file=normal | tp1file

～《normal》

A 系および B 系の MQT サーバ用チャンネル管理情報格納ファイルの入出力形式を指定します。

mqa_channel_inf_file_name_a オペランドまたは mqa_channel_inf_file_name_b オペランドが指定されている場合だけ有効です。

- normal

指定された MQT サーバ用チャンネル管理情報格納ファイルを通常ファイルとして処理します。

- tp1file

指定された MQT サーバ用チャンネル管理情報格納ファイルを OpenTP1 ファイルシステム上のファイルとして処理します。

MQT サーバ開始前に、A 系および B 系の MQT サーバ用チャンネル管理情報格納ファイルを格納するための OpenTP1 ファイルシステムを初期設定してください。オンライン中に作成した場合は、チャンネル引き継ぎ情報を維持できなくなることがありますので注意してください。

normal を指定して作成した MQT サーバ用チャンネル管理情報格納ファイルと tp1file を指定して作成した MQT サーバ用チャンネル管理情報格納ファイルの互換性はありません。

tp1file を指定して作成した MQT サーバ用チャンネル管理情報格納ファイルがある OpenTP1 ファイルシステムの属性（セクタ長、容量、および最大ファイル数）を変更し、かつチャンネル引き継ぎ情報を維持したい場合には次に示す手順を実行してください。

1. OpenTP1 ファイルシステムのバックアップファイルを作成します。
2. バックアップ元の OpenTP1 ファイルシステムを初期設定します。
3. 2.で初期設定したファイルに、1.のバックアップファイルの内容をリストアします。

● mqa_max_quenum=オンライン中に使用するキューの最大数

～<符号なし整数>((2~20480))《4096》

オンライン中に使用するキューの最大数を指定します。

オンライン中に使用するキューには、定義済みキューおよび永続的動的キュー（キューファイルに登録するキュー）、一時的動的キュー（メモリキュー）、リモートキューのローカル定義で指定したキュー、および別名キューが含まれます。

このオペランドには、次に示す数の合計以上の値を指定してください。

- 各キューファイルの作成で指定した mqainit コマンドの-q オプションの値の合計
- mqaqueueatl 定義コマンド（モデルキューの属性定義）の数
- mqaremqque 定義コマンド（リモートキューのローカル定義）の数
- mqaalsque 定義コマンド（別名キューの属性定義）の数
- オンライン中に動的に作成するキュー（動的キュー）の数

クラスタ機能を使用する場合、システムで四つのキューを作成することを考慮した値を指定してください。

● mqa_max_grpnum=オンライン中に使用するキューファイルグループの最大数

～<符号なし整数>((1~4095))《16》

オンライン中に使用するキューファイルグループの最大数を指定します。

mqaqueuegrp 定義コマンドで指定するキューファイルグループの合計以上の値を指定してください。

● mqa_max_msg_recnum=1 メッセージの最大レコード数

～<符号なし整数>((1~65535))《32》

1 メッセージに登録する場合の最大レコード数を指定します。

最大レコード数は、次に示す計算式の最大値を指定してください。

$$\left\lceil \frac{\text{キューファイルに格納するメッセージの最大メッセージ長} + 512}{\text{キューファイルのレコード長} - 8} \right\rceil$$

(凡例) $\uparrow \uparrow$: 小数点以下を切り上げます。

キューファイルのレコード長は、mqainit コマンドの-s オプション（メッセージ長）と、キューファイルのセクタ長で算出した値です。レコード長の計算については、6章の「mqainit（キューファイルの割り当て）」を参照してください。

TP1/Message Queue は、開始時にこの値からジャーナルに取得するレコードデータの最大長を計算し、OpenTP1 のシステムジャーナルサービス定義 (jnl_max_datasize オペランド) で指定された値以下かどうかをチェックします。エラーになった場合、OpenTP1 のシステムジャーナルサービス定義 (jnl_max_datasize オペランド) の最大レコードデータ長の指定値を見直してください。

● mqa_max_filnum=オンライン中に使用するキューファイルの最大数

～<符号なし整数>((1~4095))《16》

オンライン中に使用するキューファイルの最大数を指定します。

mqaquegrp 定義コマンドで指定するキューファイルグループおよび mqaadd コマンドで追加するキューファイル (バックアップキューファイルを含む) の合計以上の値を指定してください。

● mqa_ioproc_num=入出力プロセス起動数

～<符号なし整数>((1~64))《1》

MQPUT 命令および MQPUT1 命令で登録したメッセージや MQGET 命令で削除したメッセージの属性情報をキューファイルに出力する、入出力プロセスの最大起動数を指定します。

mqaquegrp 定義コマンドに指定したキューファイルの数が、このオペランドの指定値よりも小さい場合は、キューファイルの数分の入出力プロセスが起動されます。ただし、二重化構成のキューファイルの場合、バックアップキューファイル (B 系ファイル) の数は含まれません。

● mqa_expiry=Y | N

～《N》

メッセージ保持時間機能を使用するかどうかを指定します。

• Y

メッセージ保持時間機能を使用します。監視サーバプロセスが起動します。

なお、監視サーバプロセスは、メッセージ保持時間終了の対象となったメッセージの削除だけを行います。メッセージの削除前に、メッセージ保持時間の終了は確認しません。

メッセージ保持時間は、MQMD 構造体の Expiry フィールドに指定します。メッセージ保持時間の対象となる契機については、マニュアル「TP1/Message Queue プログラム作成リファレンス」の MQMD 構造体の Expiry フィールドについての記載を参照してください。

• N

メッセージ保持時間機能を使用しません。

● mqa_maxque_expiry=メッセージ保持時間対象サーチキュー数

～<符号なし整数>((1~20480))《3》

メッセージ保持時間が終了したメッセージが存在するキューから、1 回の監視でメッセージ削除の対象とするキュー数を指定します。

TP1/Message Queue は一定間隔 (10 秒) でキューを監視して、メッセージ保持時間が終了したメッセージをキューから削除します。

例えば、mqa_maxque_expiry オペランドに 10、mqa_maxmsg_expiry オペランドに 5 が指定されている環境で、メッセージ保持時間が終了したメッセージが存在するキューが 100 個ある場合、1 回の監視で 10 個のキューが対象となり、1 キュー当たり最大で 5 個のメッセージが削除されます。

● mqa_maxmsg_expiry=メッセージ保持時間対象削除メッセージ数

～<符号なし整数>((1~65535))《3》

mqa_maxque_expiry オペランドで対象となったキューから、メッセージ保持時間が終了したメッセージを削除するメッセージ数を指定します。

メッセージ保持時間が終了したメッセージ数が、mqa_maxmsg_expiry オペランドで指定したメッセージ数を超えた場合は、超えた分のメッセージは、次の対象になるまで削除されません。

● mqa_quefil_inf=Y | N

～《N》

キュー/キューファイルのしきい値、および回復値に達した場合、メッセージを出力するかどうかを指定します。

- Y
メッセージを出力します。
- N
メッセージを出力しません。

● mqa_mqr_conf=Y | N

～《N》

このノードでクラスタ機能を使用するかどうかを指定します。

- Y
このノードでクラスタ機能を使用します。
- N
このノードでクラスタ機能を使用しません。

● mqa_local_ccsid=キューマネージャの文字セット識別子

～<符号なし整数>((1~65535))《850》

キューマネージャの文字セット識別子 (CodedCharSetId 属性) を指定します。

指定する文字セット識別子が表す文字セットには、SBCS (1 バイト文字セット) の場合も DBCS (2 バイト文字セット) の場合も、必ず ASCII 文字列が含まれている必要があります。

省略値 (850) を変更してチャネルを接続する場合、相手キューマネージャと同じ文字セット識別子、または相手キューマネージャが変換可能な文字セット識別子を指定してください。

指定した値は次に示す個所で使用されます。

- アプリケーションが MQMD 構造体の CodedCharSetId フィールドに MQCCSI_Q_MGR を指定した場合の MQMD 構造体の CodedCharSetId フィールド
- 相手キューマネージャとのチャンネル接続

相手キューマネージャとのチャンネル接続に使用される文字セットは、MQMD 構造体の CodedCharSetId フィールドで指定する文字セット識別子と関係ありません。アプリケーションが MQMD 構造体の CodedCharSetId フィールドに MQCCSI_Q_MGR 以外の値を指定してアプリケーションデータを作成してもチャンネル接続には影響しません。

● mqa_jnl_conf=Y | N

～ 《N》

メッセージ登録時にメッセージデータの内容をジャーナルに取得するかどうかを指定します。

- Y
メッセージデータの内容をジャーナルに取得します。
メッセージデータの内容をジャーナルに取得することによって、キューファイルがキューファイル格納媒体損傷の要因で障害となった場合に、mqafrc コマンドでキューファイルの回復ができます。
- N
メッセージデータの内容をジャーナルに取得しません。

メッセージデータの内容をジャーナルに取得する場合は、次に示す制限事項に従ってシステムを構築してください。

- キューファイルの二重化構成はできません。
- mqaquemgr 定義コマンドの-l オプションに指定する最大メッセージ長は次に示す計算式以下の値を設定してください。

<p>32ビット版の場合 jnl_max_datasizeオペランドの指定値 - 704 - 4× (mqa_max_msg_recnunオペランドの指定値-1)</p> <p>64ビット版の場合 jnl_max_datasizeオペランドの指定値 - 720 - 4× (mqa_max_msg_recnunオペランドの指定値-1)</p>

jnl_max_datasize オペランドはシステムジャーナルサービス定義のオペランドです。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

- mqtalcccha 定義コマンドのバッファ方式にはメッセージ方式を指定してください。セグメント方式を指定した場合は MQT が異常を検出し、TP1/Message Queue は正常開始できません。

● mqa_mqc_conf=Y | N

～ 《N》

このノードで MQC サーバ機能を使用するかどうかを指定します。MQC サーバ機能を使用する場合、必ず Y を指定してください。

- Y
このノードで MQC サーバ機能を使用します。
- N
このノードで MQC サーバ機能を使用しません。

関連製品である TP1/Message Queue Access を使用する場合に指定するオペランドです。

● mqa_mqc_clientnum = オンライン中に同時に処理できるクライアントの最大数

～<符号なし整数>((1~256))《16》

オンライン中に同時に処理できるクライアントの最大数を指定します。

mqcgwpsnam 定義コマンドの合計以上の値を指定してください。

指定値が mqcgwpsnam 定義コマンドの合計よりも小さい場合、MQC サービス開始時に KFCA30823-E メッセージを出力して異常終了します。

関連製品である TP1/Message Queue Access を使用する場合に指定するオペランドです。

● mqa_mqc_trnnum = オンライン中に同時に実行できるトランザクションの最大数

～<符号なし整数>((0~256))《0》

クライアントが XA 接続した場合、オンライン中に同時に実行できるトランザクションの最大数を指定します。

指定値以上のトランザクションを開始しようとした場合、KFCA30855-E メッセージを出力してトランザクションの開始を中断します。

この値を変更するときは、ジャーナルファイル容量の見積もりを見直してください。

関連製品である TP1/Message Queue Access を使用する場合に指定するオペランドです。

● mqa_mqo_conf=Y | N

～《N》

TP1/Message Queue では、Message Queue - Operation - Agent 機能をサポートしていません。mqa_mqo_conf 定義は過去のバージョンで作成された定義との互換性のためだけにあります。そのため、mqa_mqo_conf 定義に Y を指定した場合、MQ 管理機能インタフェースでエラー (KFCA31049-E) となりますが、この機能を縮退し OpenTP1 の起動には成功します。

- Y
Message Queue - Operation - Agent を使用します。

- N

Message Queue - Operation - Agent を使用しません。

● **mqa_mqo_queuefile_alarm_level=キューファイル登録メッセージサイズしきい値**

～<符号なし整数>((10~100))《80》(単位：%)

各キューファイルのメッセージ格納域全容量に対する、メッセージ格納域登録率のしきい値を指定します。

キューファイルの登録率がこのオペランドで指定したしきい値以上になった場合、mqa_quefil_inf オペランドに Y が指定されていれば、KFCA31055-I メッセージが出力されます。

● **mqa_mqo_queuefile_recover_level=キューファイル登録メッセージサイズ回復値**

～<符号なし整数>((10~100))《70》(単位：%)

各キューファイルのメッセージ格納域全容量に対する、メッセージ格納域登録率の回復値を指定します。

キューファイルの登録率がこのオペランドで指定した回復値を下回った場合、mqa_quefil_inf オペランドに Y が指定されていれば、KFCA31056-I メッセージが出力されます。

● **mqa_mqo_queue_alarm_level=キュー登録メッセージ数しきい値**

～<符号なし整数>((10~100))《80》(単位：%)

各キューに登録できる最大メッセージ数に対する、メッセージ登録率のしきい値を指定します。

キューのメッセージ登録率がこのオペランドで指定したしきい値以上になった場合、mqa_quefil_inf オペランドに Y が指定されていれば、KFCA31053-I メッセージが出力されます。

● **mqa_mqo_queue_recover_level=キュー登録メッセージ数回復値**

～<符号なし整数>((10~100))《70》(単位：%)

各キューに登録できる最大メッセージ数に対する、メッセージ登録率の回復値を指定します。

キューのメッセージ登録率がこのオペランドで指定した回復値を下回った場合、mqa_quefil_inf オペランドに Y が指定されていれば、KFCA31054-I メッセージが出力されます。

● **mqa_msg_stay_time=メッセージ滞留時間**

～<符号なし整数>((0~2147483647))《0》(単位：秒)

メッセージが滞留したと判断する時間を指定します。この指定した時間以上メッセージがキューにある場合に、キューに対応する KFCA31065-I メッセージが出力されます。メッセージ登録時から、メッセージの削除が有効になるまでが監視対象になります。0 を指定した場合、滞留状態はチェックされません。このオペランドに 0 以外を指定した場合に、監視サーバプロセスが起動します。滞留状態の検知は遅れる場合があります。KFCA31065-I メッセージが出力されたあとに、メッセージ滞留時間を経過したメッセージがキューからすべて取り出されると、KFCA31066-I メッセージが出力されます。

このオペランドを指定する場合は、監視対象にするキューについて、mqaqueatl 定義コマンドの-v オプションに on を指定してください。

● max_socket_descriptors=ソケット用ファイル記述子の最大数

～<符号なし整数> ((Linux 以外の場合：32～2032, Linux の場合：32～1008)) 《514》

OpenTP1 の制御下のプロセスで、ソケット用に使用するファイル記述子の最大数を指定します。次に示す計算式で求めた値を指定してください。

$\uparrow (mqa_ioproc_num \text{ の指定値} \times 2 + 7) \nearrow 0.8 \uparrow$

(凡例)

↑↑：小数点以下を切り上げます。

● mqa_pool_attri=free | fixed

～《free》

MQA サービス専用共用メモリをメモリプールに固定するかどうかを指定します。

このオペランドで fixed を指定する場合は、MQA サービス専用共用メモリがメモリプールに固定されるため、実装するメモリの容量や、全使用容量に対する共用メモリの割合などを考慮してください。

- free

MQA サービス専用共用メモリをメモリプールに固定しません。

この値を指定した場合は、共用メモリのページングが発生して処理速度が低下することがあります。

- fixed

MQA サービス専用共用メモリをメモリプールに固定します。

共用メモリのページングによって処理速度が低下することはありません。

ただし、実装するメモリの容量が少ない場合、共用メモリ以外のテキストやデータセグメントなどのページングが発生することがあります。

● mqa_extend_rmshm_size=MQA サービス専用共用メモリの拡張領域サイズ

～<符号なし整数> ((0～2000000000)) 《0》 (単位：バイト)

オンライン中に組み込むキューファイルで使用する共用メモリのサイズを指定します。TP1/Message Queue の開始時に指定したサイズで MQA サービス専用共用メモリの拡張領域を確保します。省略した場合、TP1/Message Queue の開始時に MQA サービス専用共用メモリの拡張領域は確保されないため、オンライン中に新しくキューファイルを組み込めません。

MQA サービス定義の mqaquegrp 定義コマンドで指定したキューファイル以外の新規キューファイルをオンライン中に組み込むと、組み込むキューファイルで使用するサイズ分の MQA サービス専用共用メモリの拡張領域を使用します。一度、新規キューファイルを組み込むと、mqarm コマンドでキューファイルを切り離しても MQA サービス専用共用メモリの拡張領域は予約された状態のままとなります。したがっ

て、同一のキューファイルを組み込む場合だけ MQA サービス専用共用メモリの拡張領域の予約された部分を使用できます。

● `mqa_getwait_timeout=low | high`

～ 《low》

MQGMO_WAIT を指定した MQGET 命令でメッセージが到着しなかった場合に、タイムアウトする時間の精度を指定します。

- low

MQGMO 構造体の WaitInterval フィールドに指定した待ち合わせ最大時間を、秒単位の精度で待ち合わせを行います。

タイムアウトするまでに最大で 1 秒の遅延が発生することがあります。

- high

MQGMO 構造体の WaitInterval フィールドに指定した待ち合わせ最大時間を、ミリ秒単位の精度で待ち合わせを行います。

タイムアウトするまでに最大で 1 秒の遅延が発生することがあります。

なお、メッセージがキューに登録されてから MQGET 命令の待ち合わせが解除されるまでに、数 10 ミリ秒の遅延が発生することがあります。

メッセージが到着してから MQGET 命令の待ち合わせが解除されるまでの遅延を小さくしたい場合は、low を指定してください。メッセージが到着しなかったときに、MQGET 命令がタイムアウトする時間をミリ秒単位で制御したい場合は、high を指定してください。

● `mqa_filerr_continue=Y | N`

～ 《Y》

キューファイルに入出力エラーが発生した場合、オンラインを継続させるかどうかを指定します。

このオペランドを省略した場合または Y を指定した場合、キューファイルに入出力エラーが発生すると、キューファイルの状態を一時的障害閉塞としてオンラインを継続させます。

このオペランドに N を指定する場合、キューファイル入出力エラー発生時に KFCA31191-E メッセージを出力してオンラインを異常終了させることができます。異常終了時は再開前に障害要因を取り除くことによって、再開時にキューファイルがジャーナルによって回復されます。そのため、障害閉塞しないでキューファイルの整合性が保たれます。

- Y

オンラインを継続させます。

- N

オンラインを継続せず、OpenTP1 を異常終了させます。

注

このオペランドに N を指定する場合、次に示す点に注意してください。

- キューファイル二重化構成の場合、両系が障害となった時点で OpenTP1 が異常終了します。
- OpenTP1 が異常終了したあと、このオペランドに N を指定したまま障害要因を取り除かないで OpenTP1 を再開始した場合、再び入出力エラーで OpenTP1 が異常終了するおそれがあります。
- OpenTP1 が異常終了したあと、このオペランドを Y に変更して OpenTP1 を再開始したときに再び入出力エラーが発生すると、キューファイルは障害閉塞になります。

● mqa_prf_trace_level=性能検証用トレース情報取得レベル

～< 16 進数字>((00000000 | 00000001 | 00000010))《00000000》

性能検証用トレース情報を取得するレベルを指定します。取得する場合はシステム共通定義の prf_trace オペランドに Y を指定してください。N を指定した場合、mqa_prf_trace_level オペランドに 00000001 または 00000010 を指定しても性能検証用トレース情報は取得されません。

- 00000000
性能検証用トレース情報は取得されません。
- 00000001
イベント ID が 0xd000～0xd003 の性能検証用トレース情報が取得されます。
- 00000010
イベント ID が 0xd100～0xd103 の性能検証用トレース情報が取得されます。

性能検証用トレース情報の詳細を見るためのトレース情報ファイルの取得 (prfget コマンド) およびトレース情報ファイルの編集出力 (prfed コマンド) など、性能検証用トレースに関する運用については、マニュアル「OpenTP1 運用と操作」を参照してください。OpenTP1 の開始モードに関係なく、OpenTP1 開始時に MQA サービス定義に指定されている値が有効となります。

イベント ID 別の取得内容を次に示します。

表 4-7 イベント ID 別の取得内容

イベント ID	キュー名称 (48)	メッセージトークン (内部登録時間) (8)	リターンコード (4)	メッセージ識別子 (32)	相関識別子 (32) ※	ユーザデータ (32) ※	メッセージ滞留時間 (16) ※	処理時間 (16) ※
0xd000	○	○	○	○	○	○	—	—
0xd001	○	○	○	○	○	○	○	—
0xd002	○	○	○	○	○	○	—	—
0xd003	○	○	○	○	○	○	○	—
0xd100	○	○	○	○	○	○	—	○

イベント ID	キュー名称 (48)	メッセージトークン (内部登録時間) (8)	リターンコード (4)	メッセージ識別子 (32)	関連識別子 (32) ※	ユーザデータ (32) ※	メッセージ滞留時間 (16) ※	処理時間 (16) ※
0xd101	○	○	○	○	○	○	○	○
0xd102	○	○	○	○	○	○	—	○
0xd103	○	○	○	○	○	○	○	○

(凡例)

- ：情報を取得します。
- ：情報を取得しません。
- 括弧内：取得バイト数

注※

prfed コマンドに-d オプションを指定した場合に出力される，詳細情報と出力順序を示しています。

TP1/Massge Queue の詳細情報の出力情報，および表示形式を次に示します。

表 4-8 イベント ID 別の詳細情報出力情報

イベント ID	先頭からのオフセット			
	0000 ~ 001f	0020 ~ 003f	0040 ~ 004f	0050 ~ 005f
0xd000	関連識別子	ユーザデータ	—	—
0xd001			メッセージ滞留時間	—
0xd002			—	—
0xd003			メッセージ滞留時間	—
0xd100			処理時間	—
0xd101			メッセージ滞留時間	処理時間
0xd102			処理時間	—
0xd103			メッセージ滞留時間	処理時間

出力形式

- 関連識別子
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
- ユーザデータ
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
- メッセージ滞留時間
cccccccccccccccc
- 処理時間

dddddddddddddddd

aa..aa

相関識別子 (16 進数)

次の形式で出力されます。

相関識別子 (24 バイト) + “¥0¥0¥0¥0¥0¥0¥0¥0” (予約領域)

bb..bb

ユーザデータ (16 進数)

ユーザデータの先頭 32 バイトが出力されます。

cc..cc

メッセージ滞留時間 (16 進数)

次の形式で出力されます。

メッセージ滞留時間 (8 バイト) + “¥0¥0¥0¥0¥0¥0¥0¥0” (予約領域)

dd..dd

処理時間 (16 進数)

次の形式で出力されます。

処理時間 (秒) (4 バイト) + 処理時間 (マイクロ秒) (4 バイト) + “¥0¥0¥0¥0¥0¥0¥0¥0” (予約領域)

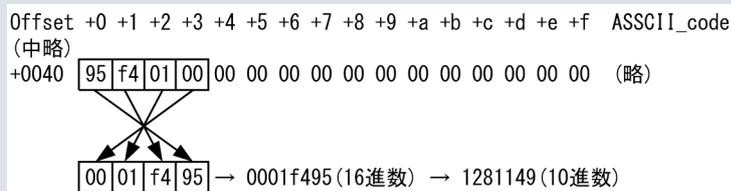
このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

注意事項

トレースを取得した OS が Linux または Windows の場合、メッセージ滞留時間、処理時間はリトルエンディアンのため、次の例のとおり 1 バイトずつ逆に解釈してください。

(例 1) イベント ID が 0xd001 の場合

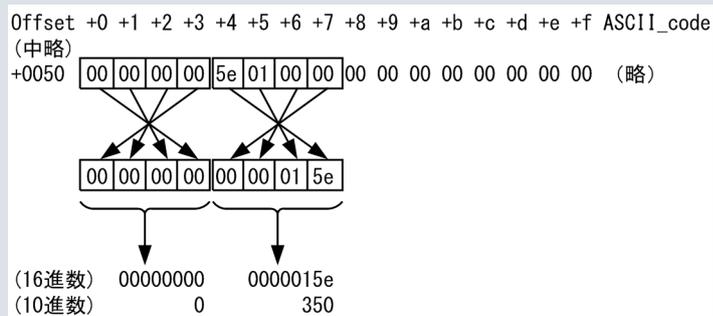
Linux で取得し、「prfcd -d -m」で編集した prf トレース情報の詳細を次に示します。



上記の場合、+ 0040 ~ + 0043 はメッセージ滞留時間のため、128149 秒となります。

(例 2) イベント ID が 0xd101 の場合

Linux で取得し、「prfcd -d -m」で編集した prf トレース情報の詳細を次に示します。



上記の場合、次のように合計 350 マイクロ秒になります。

- + 0050~+ 0053 は処理時間 (秒) のため 0 秒
- + 0054~+ 0057 は処理時間 (マイクロ秒) のため 350 マイクロ秒

mqaquegrp (キューファイルグループの構成定義)

形式

```
{ { mqaquegrp    -g キューファイルグループ名
                  [-b バックアップキューファイル名
                    [ [ , バックアップキューファイル名 ] ... ] ]
                  [-n 入出力バッファ数]
                  [-m メッセージエントリ確保率]
                  [-k キューファイルグループ監視の可否]
                  [-u キューファイルグループ登録メッセージサイズしきい値]
                  [-l キューファイルグループ登録メッセージサイズ回復値]
                  [-s MQAサービス専用共用メモリ確保要求者識別子]
                  キューファイル名 [ [ , キューファイル名 ] ... ] }
```

機能

オンライン中にアクセスしたメッセージを格納するキューファイル名とキューファイルの属性を合わせて、キューファイルグループ名として定義します。

TP1/Message Queue 全体で登録できるキューの最大数は 20,480 個です。

オプション

● -g キューファイルグループ名

～< 1～31 文字の識別子 >

キューファイル名とキューファイルの属性を定義するキューファイルグループの名称を指定します。

アプリケーション、またはチャネルによって、複数のキュー操作が同時に一つのキューファイルグループに集中すると、キュー操作が遅くなる場合があります。同時に使用するキューファイルは、別々のキューファイルグループに分割して定義することを推奨します。

● -b バックアップキューファイル名

～< 1～59 文字の文字列 >

キューファイル二重化構成を行う場合、指定したキューファイル名 (A 系ファイル名) に対するバックアップキューファイル名 (B 系ファイル名) を指定します。キューファイル (A 系ファイル) とバックアップキューファイル (B 系ファイル) は、大きさ (mqainit コマンドのオプション)、キュー作成の有無 (mqamkque コマンドのコマンド引数)、指定の順序、OpenTP1 ファイルシステムのセクタ長を同じにしてください。

指定例を次に示します。

```
mqaquegrp -g GR1 -b /dev/rdisk/MQA4,/dev/rdisk/MQA5,/dev/rdisk/MQA6
                /dev/rdisk/MQA1,/dev/rdisk/MQA2,/dev/rdisk/MQA3
```

- バックアップファイルとの対応

```
/dev/rdsk/MQA1のバックアップキューファイルは、 /dev/rdsk/MQA4
/dev/rdsk/MQA2のバックアップキューファイルは、 /dev/rdsk/MQA5
/dev/rdsk/MQA3のバックアップキューファイルは、 /dev/rdsk/MQA6
```

キューファイル二重化構成でアクセスをしたキューファイル名（A系ファイル名）とバックアップキューファイル名（B系ファイル名）との定義をオンライン終了後に入れ替えて定義しないでください。

- 1回目のオンライン定義

```
mqaquegrp -g GR1
          -b /dev/rdsk/MQA4, /dev/rdsk/MQA5, /dev/rdsk/MQA6
            /dev/rdsk/MQA1, /dev/rdsk/MQA2, /dev/rdsk/MQA3
```

- 2回目のオンライン定義

```
mqaquegrp -g GR1
          -b /dev/rdsk/MQA1, /dev/rdsk/MQA2, /dev/rdsk/MQA3
            /dev/rdsk/MQA4, /dev/rdsk/MQA5, /dev/rdsk/MQA6
```

上記の指定をした場合、処理は保証できません。

● -n 入出力バッファ数

～<符号なし整数> ((0～1048560)) 《640》

メッセージ本体を格納するキューファイルに対するキャッシュ用の入出力バッファの個数を指定します。キャッシュ用の入出力バッファには MQA サービス専用の共用メモリ（拡張領域も含まれます）が使用されます。

メッセージの読み込み（取り出し、参照および送信）時、対象のメッセージが入出力バッファにある場合、メッセージ本体のディスク入出力が発生しなくなります。

メッセージをキューファイルに蓄積しても、入出力バッファの数に空きがあるまでは、メッセージ読み込み時にメッセージ本体のディスク入出力は発生しません。

1メッセージが入出力バッファに存在するために必要な入出力バッファ数の計算式を次に示します。

↑ (メッセージ長 + 512) / (キューファイルのレコード長^{*} - 8) ↑

(凡例)

↑↑：小数点以下を切り上げます。

注※

キューファイルのレコード長については、mqainit コマンドの-s オプションの説明を参照してください。

一つのキューファイルグループに複数のキューファイルを指定した場合、それぞれのキューファイルにここで指定した数のバッファが割り当てられます。

入出力バッファ数には次に示す計算式以上の値を指定することをお勧めします。

$$\sum_{i=1}^j (A_i \times L_i) + \sum_{i=1}^k L_i + \sum_{i=1}^m (B_i \times M_i)$$

(凡例)

A_i

1 トランザクションの登録メッセージ数です。メッセージの読み込みだけを行う場合は 1 になります。トリガメッセージおよび報告メッセージを生成してキューに登録するメッセージ数もこの値に加算してください。

B_i

受信チャンネルのバッチサイズのうちでいちばん大きい値です。トリガメッセージおよび報告メッセージを生成してキューに登録するメッセージ数もこの値に加算してください。ただし、送信チャンネルの場合は 1 になります。

L_i

次に示す計算式の算出結果です。ただし、メッセージの読み込みだけを行う場合は 1 になります。

$$\uparrow (\text{メッセージ長} + 512) / (\text{キューファイルのレコード長} - 8) \uparrow$$

↑↑は小数点以下を切り上げることを示します。

M_i

次に示す計算式の算出結果です。ただし、送信チャンネルだけの場合は 1 になります。

$$\uparrow (\text{メッセージ長} + 512) / (\text{キューファイルのレコード長} - 8) \uparrow$$

↑↑は小数点以下を切り上げることを示します。

j

同じキューファイルに割り当てられているキューに同時にアクセスし、かつトランザクションを使用するアプリケーション数です。

トリガメッセージおよび報告メッセージを生成してキューに登録するアプリケーション数およびMQT サーバのプロセス数も計算に含めます。

また、MQC サーバ機能を使用する場合、TP1/Message Queue Access を使用するアプリケーション数も計算に含めます (MQC サービス定義の mqcgwpsnam 定義コマンドの指定値が最大値になります)。

k

同じキューファイルに割り当てられているキューに同時にアクセスし、かつトランザクションを使用しないアプリケーション数です。

トリガメッセージおよび報告メッセージを生成してキューに登録するアプリケーション数およびMQT サーバのプロセス数も計算に含めます。

また、MQC サーバ機能を使用する場合、TP1/Message Queue Access を使用するアプリケーション数も計算に含めます (MQC サービス定義の mqcgwpsnam 定義コマンドの指定値が最大値になります)。

m

同じキューファイルに割り当てられているキューに同時にアクセスする MQT サーバプロセス数です。トリガメッセージおよび報告メッセージを生成してキューに登録するアプリケーション数および MQT サーバのプロセス数も計算に含めます。

入出力バッファ数を上記の計算式より小さく設定した場合、次に示す状況が発生することがあります。

- 次に示す処理を実行するとき、1 秒間の空きバッファ待ちをしたあと、プロセス域にキューファイルのレコード長で入出力バッファを確保
 - MQPUT, MQPUT1 または MQGET 命令発行時
 - チャネル経由のメッセージの送信または受信
- バッファ面数に 0 を指定した場合は、次の処理を実行するとき空きバッファ待ちをしないで、プロセス域にキューファイルのレコード長で入出力バッファを確保
 - MQPUT, MQPUT1 または MQGET 命令発行時
 - チャネル経由のメッセージの送信または受信

どちらの場合もチャネル経由のメッセージ送受信時、または API リターン時に領域は解放されます。

● -m メッセージエントリ確保率

~<符号なし整数>((0~100))《0》

キューファイルに格納できるメッセージ管理情報 (MQMD 構造体など) の数に対して、何パーセント分のキャッシュエリアが確保されるかを指定します。メッセージエントリ確保率が高いほど、メッセージに対する MQGET 命令発行時のディスクに対する入出力回数が少なくなります。メッセージエントリ確保率が低いほど、使用するメモリ量が少なくなります。

一つのキューファイルグループに複数のキューファイルを指定した場合、それぞれのキューファイルにここで指定した数のメッセージエントリが割り当てられます。メッセージエントリ確保率には、キューファイルに格納できるメッセージ数に対する最大保留メッセージ数 (メッセージがキューファイルにある数) の割合を%で指定してください。

メッセージエントリ確保率に 0 を指定した場合、メッセージエントリ (292 バイト) ×メッセージ数 + 20 バイト分のメモリが、メッセージエントリ確保率に 100 を指定した時に比べて削減できます。ただし、バッファ長に 0 バイトを指定して、MQGET 命令を発行した場合もキューファイルの入出力が発生します。

● -k キューファイルグループ監視の要否

~《N》

キューファイルグループ監視を使用するかどうかを指定します。

- Y: キューファイルグループ監視を使用します。
キューファイルグループ監視を使用する場合、-u オプションおよび-l オプションの指定が有効になります。監視サーバプロセスが起動します。

- N: キューファイルグループ監視を使用しません。

省略する場合、キューファイルグループ監視を使用しません。

キューファイルグループ監視機能を使用する場合、監視されるキューファイルは正常状態のキューファイルだけです。mqarm コマンドを使用して切り離れたキューファイル、および mqaadd コマンドを使用して組み込まれたキューファイルは監視の対象外です。mqarles コマンドを使用して正常状態になった時点でキューファイルは監視の対象になります。

● -u キューファイルグループ登録メッセージサイズしきい値

~<符号なし整数>((10~100))《80》(単位: %)

キューファイルグループのメッセージ格納域全容量に対するメッセージ格納域登録率のしきい値を指定します。

キューファイルグループの登録率がこのオペランドで指定したしきい値以上になった場合、-k オプションに Y が指定されていれば、KFCA31107-I メッセージが出力されます。

● -l キューファイルグループ登録メッセージサイズ回復値

~<符号なし整数>((10~100))《70》(単位: %)

キューファイルグループのメッセージ格納域全容量に対するメッセージ格納域登録率の回復値を指定します。

キューファイルグループの登録率がこのオペランドで指定した回復値を下回った場合、-k オプションに Y が指定されていれば、KFCA31108-I メッセージが出力されます。

● -s MQA サービス専用共用メモリ確保要求者識別子

~<16進数字>((00~ff))

MQA サービス専用共用メモリ確保要求者識別子を指定します。指定値の先頭に'm'を付加した識別子が、実際に MQA サービス専用共用メモリを確保するときを使用される識別子となります。また、dcshmls -r コマンドで表示される共用メモリプールの確保要求者種別は、指定値の先頭に'm'を付加した値になります。dcshmls コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。識別子を指定したキューファイルグループに属するキューファイルは、MQA サービス専用共用メモリセグメントを識別子単位で確保します。なお、共用メモリの識別子の種類は、31 個以内に収まるようにしてください。

このオプションを使用するケースを次に示します。

1. MQA サービス専用共用メモリへのアタッチは、MQA サービス専用共用メモリ確保要求者識別子単位で実施されます。このオプションを指定すれば、MQA サービスで指定されるキューファイルの一部だけをアクセスする UAP が使用する共用メモリのアタッチサイズを減らすことができます。
2. MQA サービス専用共用メモリサイズが、トータルで 2147483583 バイトを超えた場合、-s オプションで分割すれば共用メモリを確保できます。

ほかの mqaquegrp 定義コマンドと同一の識別子を指定した場合、同一の識別子を持つキューファイルグループはすべて同一の MQA サービス専用共用メモリセグメントを確保します。

省略した場合、指定を省略した他の mqaquegrp 定義コマンドと同一の MQA サービス専用共用メモリセグメントを確保します。この場合の MQA サービス専用共用メモリ確保要求者識別子は、mqaquegrp 定義コマンドで指定されている値と重複しない 00~ff の範囲で自動的に付与されます。

この識別子を指定したキューファイルグループの MQA サービス専用共用メモリセグメントの確保で共用メモリのメモリ不足が発生した場合、KFCA31186-W のメッセージが出力され、共用メモリが分割して確保されます。指定した識別子または自動付与された識別子は、mqafills コマンドの -f オプションを実行した時に表示される内容で確認することができます。

コマンド引数

● キューファイル名

～< 1～59 文字の文字列 >

メッセージを格納するファイルの名称を指定します。

一つのキューファイルに、複数のキューが格納されます。

キューファイルを複数指定した場合、一つのキューに登録されるメッセージは、複数のキューファイルに分散して格納されます。そのためキューファイルを複数指定する場合は、それぞれのキューファイルを物理的に別ディスクにしてください。一つのキューに複数アプリケーションからの操作が重なった場合、性能が良くなります。

キューファイルパス名およびキューファイル名にスペースを指定しないでください。また、ファイルパス名の "/" 個所に "/" を追加したキューファイルパス名を指定しないでください。指定した場合、動作は保証されません。

mqaqueatl (モデルキューの属性定義)

形式

```
{ { mqaqueatl [-q キュー記述子] [-g 取り出し許可]
[-p 登録許可]
[-y 省略時のメッセージ優先度]
[-e 省略時のメッセージ永続性] [-a プロセス名]
[-d 最大メッセージ登録数] [-l 最大メッセージ長]
[-s 共用性] [-o 省略時の取り出しオプション]
[-m メッセージ配布順序] [-r キュー保持時間]
[-k 定義タイプ] [-u 使用種別]
[-i イニシエーションキュー名]
[-c トリガ制御] [-t トリガタイプ]
[-b トリガのためのメッセージ優先度の下限値]
[-n トリガのためのメッセージ登録数の下限値]
[-f トリガデータ]
[-j 配布リストのサポート]
[-w デフォルトのバインディング]
[-v メッセージ滞留時間監視の可否]
モデルキューの定義名 キューファイル名 }
```

機能

オンライン中に動的に作成されるキュー（動的キュー）のモデルになる属性を指定します。

MQOPEN 命令でモデルキューの定義名がオブジェクト名として指定された場合、作成される動的キューの属性は、モデルキューの属性定義の値になります。

この定義コマンドは複数指定できますが、モデルキューの定義名は重複して指定できません。重複して指定した場合、MQA サーバ開始処理でエラーになります。また、リモートキューのローカル定義名、および別名キュー名と重複した名称をモデルキューの定義名に指定した場合も、MQA サーバ開始処理でエラーになります。

mqaquegrp 定義コマンドでキューファイル二重化構成を定義しているキューファイルにキュー（動的キュー）を作成する場合、MQA サービス定義ファイルに A 系キューファイルを指定した mqaqueatl 定義を指定するようにしてください。

TP1/Message Queue 全体で登録できるキューの最大数は 20,480 個です。

オプション

● -q キュー記述子

～< 1～64 文字の文字列 >

キューについてのユーザに任意の情報 (QDesc 属性) を指定します。キューマネージャはこの指定値を、一切チェックしません。省略した場合、キュー記述子は空白になります。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-q "svrg01,svr01"
```

● -g 取り出し許可

～ 《allowed》

作成される動的キューに対する取り出し操作を許可するかどうか (InhibitGet 属性) を指定します。

- allowed
取り出し操作は許可されます (属性値: MQQA_GET_ALLOWED)。
- inhibited
取り出し操作は禁止されます (属性値: MQQA_GET_INHIBITED)。

● -p 登録許可

～ 《allowed》

作成されるキューに対する登録操作を許可するかどうか (InhibitPut 属性) を指定します。

- allowed
登録操作は許可されます (属性値: MQQA_PUT_ALLOWED)。
- inhibited
登録操作は禁止されます (属性値: MQQA_PUT_INHIBITED)。

● -y 省略時のメッセージ優先度

～ <符号なし整数> ((0~9)) 《0》

キューに登録されるメッセージの省略時の優先度 (DefPriority 属性) を指定します。優先度は、指定した値が小さいほど低く、大きいほど高くなります。

メッセージ記述子 (MQMD 構造体) に MQPRI_PRIORITY_AS_Q_DEF が指定され、メッセージがキューに登録される場合に、この指定値は使用されます。

● -e 省略時のメッセージ永続性

～ 《persistent》

キューに登録されるメッセージの省略時の永続性 (DefPersistence 属性) を指定します。

- persistent
メッセージを永続にします (属性値: MQPER_PERSISTENT)。
- notpersistent
メッセージを永続にしません (属性値: MQPER_NOT_PERSISTENT)。

● -a プロセス名

～< 1～48 文字の MQ 文字列>

プロセスオブジェクトの名称 (ProcessName 属性) を指定します。

ローカルキューマネージャの mqaprcdef 定義コマンドで指定したプロセス名と同じ名称を指定してください。

プロセスオブジェクトは、トリガモニタアプリケーションを決定するために使用されます。トリガを使用する場合は、必ず指定してください。トリガが使用される場合に省略したとき、イニシエーションキュー障害になります。トリガが使用されない場合に省略したとき、プロセス名は空白になります。

● -d 最大メッセージ登録数

～< 符号なし整数> ((0～2147483647)) 《128》

キューに登録できるメッセージの最大値 (MaxQDepth 属性) を指定します。

ここで指定した数のメッセージがすでに格納されているキューに対してさらにメッセージを登録しようとすると、理由コード MQRC_Q_FULL でエラーになります。

● -l 最大メッセージ長

～< 符号なし整数> ((0～128000000)) 《4096000》

キューに登録されるメッセージの中に、入れることができるアプリケーションデータの最大値 (MaxMsgLength 属性) を指定します。キューに対してこの指定値を超える長さのメッセージを登録すると、理由コード MQRC_MSG_TOO_BIG_FOR_Q でエラーになります。

この指定値は、mqaquemgr 定義コマンドの -l オプション (最大メッセージ長) の指定値以下にする必要があります。

転送キューの場合は、構造体データタイプを含んだ最大メッセージ長を設定してください。

● -s 共用性

～ 《notshareble》

キューが同時に何度も入力用にオープンできるかどうか (Shareability 属性) を指定します。

- shareble

キューを共用できます (属性値: MQQA_SHAREABLE)。

MQOO_INPUT_SHARED オプション指定の多重オープンができます。

- notshareble

キューを共用できません (属性値: MQQA_NOT_SHAREABLE)。

MQOO_INPUT_SHARED オプション指定の MQOPEN 命令は、MQOO_INPUT_EXCLUSIVE と同様に処理されます。

● -o 省略時の取り出しオプション

～ 《exclusive》

キューを入力用にオープンする場合の省略時解釈値 (DefInputOpenOption 属性) を指定します。

キューオープン時の MQOPEN 命令で, MQOO_INPUT_AS_Q_DEF オプションが指定された場合に, この値が使用されます。

- exclusive

排他アクセスのメッセージ取り出し用にオープンします (属性値: MQOO_INPUT_EXCLUSIVE)。MQOPEN 命令呼び出し元アプリケーション自身またはほかのアプリケーションによって, キューがすでに入力用にオープンされていると, そのタイプがどのようなものであっても (MQOO_INPUT_SHARED や MQOO_INPUT_EXCLUSIVE) 理由コード MQRC_OBJECT_IN_USE でエラーになります。

- shared

共用アクセスのメッセージ取り出し用にオープンします (属性値: MQOO_INPUT_SHARED)。MQOPEN 命令呼び出し元アプリケーション自身またはほかのアプリケーションによって, キューが MQOO_INPUT_SHARED ですでにオープンされている場合, 命令は成功します。しかし, MQOO_INPUT_EXCLUSIVE でオープンされている場合は, 理由コード MQRC_OBJECT_IN_USE でエラーになります。

● -m メッセージ配布順序

～ 《fifo》

メッセージが MQGET 命令でアプリケーションに返される順序 (MsgDeliverySequence 属性) を指定します。

- priority

優先度の順序でメッセージが返されます (属性値: MQMDS_PRIORITY)。

MQGET 命令は, 指定される選択基準を満足する最も優先度の高いメッセージを返します。各優先度のレベル内では, FIFO (先入れ先出し) の順序で返します。

- fifo

FIFO (先入れ先出し) の順序でメッセージが返されます (属性値: MQMDS_FIFO)。

MQGET 命令は, 優先度を無視して, 指定される選択基準を満足する先頭のメッセージを返します。

● -r キュー保持時間

～ <符号なし整数> ((0~65535)) 《8》 (単位: 時間)

キューを保持する時間 (RetentionInterval 属性) を指定します。キューを作成してからここで指定した時間が経過すると, キューは破棄対象になります。ただし, キューマネージャがこの指定値に基づいてキューを削除したり, 保持時間が終了していないキューの削除を制限したりはしません。

MQINQ 命令でこの指定値とキューの作成時間が返却されるため、必要な操作は、アプリケーションの責任でしてください。

● -k 定義タイプ

～《permanent》

作成するキューが永続的か一時的か (DefinitionType 属性) を指定します。

- permanent
永続的動的キュー (属性値: MQQDT_PERMANENT_DYNAMIC)
- temporary
一時的動的キュー (属性値: MQQDT_TEMPORARY_DYNAMIC)
一時的動的キューとは、MQA サーバの再開始または正常開始時に引き継がれないキューのことです。

● -u 使用種別

～《normal》

キューが何のために使用されるか (Usage 属性) を指定します。

- normal
通常に使用 (通常キュー) (属性値: MQUS_NORMAL)
通常のアプリケーションが、メッセージの登録または取り出しをする場合に指定します。
- transmission
転送キューとして使用 (属性値: MQUS_TRANSMISSION)
リモートキューマネージャあてのメッセージを保持する場合に指定します。

● -i イニシエーションキュー名

～< 1~48 文字の MQ 文字列 >

トリガメッセージを書き込むイニシエーションキューの名称 (InitiationQName 属性) を指定します。省略した場合、イニシエーションキュー名は空白になります。

イニシエーションキューは、ローカルキューマネージャで定義されるローカルキューとして定義されます。

イニシエーションキュー名を定義したキューにメッセージが到着し、アプリケーションの起動が要求されると、キューマネージャはここで指定したイニシエーションキューにトリガメッセージを送信します。イニシエーションキューは、トリガメッセージの受信後に適切なアプリケーションを起動するトリガモニタアプリケーションによって監視される必要があります。

● -c トリガ制御

～《on》

キューを処理するアプリケーションを起動させるために、トリガメッセージをイニシエーションキューに書き込むかどうか (TriggerControl 属性) を指定します。

この指定値は、MQSET 命令で変更できます。

- off
トリガメッセージをキューに書き込みません (属性値: MQTC_OFF)。
-t オプション (トリガタイプ) を指定してもトリガメッセージはキューに書き込まれません。
- on
トリガメッセージをキューに書き込みます (属性値: MQTC_ON)。

● -t トリガタイプ

~ 《every》

メッセージがキューに到着した場合、トリガメッセージをイニシエーションキューに書き込む条件 (TriggerType 属性) を指定します。

- every
メッセージごとに通知されます (属性値: MQTT_EVERY)。
1 個のメッセージがキューに到着するごとに、トリガメッセージが書き込まれます。
- first
キューが空でないときに通知されます (属性値: MQTT_FIRST)。
キューが空 (メッセージが 1 個もない) の状態から空でない (1 個以上のメッセージがある) 状態へ移行した場合に、トリガメッセージが書き込まれます。
このモデルキューを基に転送キューを作成する場合は、first を指定してください。
- depth
一定の格納数 (下限値) を超えたときに通知されます (属性値: MQTT_DEPTH)。
キューに一定数 (トリガのためのメッセージ登録数の下限値) のメッセージがある場合にトリガメッセージが書き込まれます。トリガメッセージが書き込まれたあとに、それ以上のトリガメッセージが書き込まれないようにするために、再度明示的にトリガ処理を再開する (MQSET 命令によるトリガ制御の変更) までトリガ制御属性に MQTC_OFF が設定されます。
- none
トリガメッセージは通知されません (属性値: MQTT_NONE)。
キューにメッセージが到着しても、トリガメッセージは書き込まれません。この指定は、-c オプション (トリガ制御) で off を指定した場合と同様の効果を持ちます。

● -b トリガのためのメッセージ優先度の下限値

~ <符号なし整数> ((0~9)) 《0》

トリガタイプが depth の場合の、トリガメッセージを書き込む下限値 (TriggerMsgPriority 属性) をメッセージ優先度で指定します。

キューに到着したメッセージの優先度が、ここで指定した優先度より低い場合は、トリガメッセージを書き込むほかの条件を満たしていても、トリガメッセージは書き込まれません。トリガメッセージを書き込む条件については、マニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

● **-n トリガのためのメッセージ登録数の下限値**

～<符号なし整数>((1~65535))《1》

トリガタイプが depth の場合の、トリガメッセージが書き込まれる前にキューになければならないメッセージの個数 (TriggerDepth 属性) を指定します。

この指定値には、-b オプションで指定した優先度より低い優先度のメッセージは含まれません。

● **-f トリガデータ**

～<1~64 文字の文字列>

キューにメッセージが到着すると、イニシエーションキューにトリガメッセージが書き込まれます。この時、キューマネージャによってトリガメッセージに挿入される、ユーザに任意のデータ (TriggerData 属性) を指定します。省略した場合、トリガデータは空白になります。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-f "svrg01,svr01"
```

● **-j 配布リストのサポート**

～《nosupport》

MQPUT 命令、および MQPUT1 命令で、配布リスト付きのメッセージをキューに書き込めるかどうかを指定します。

- support
配布リストに対応します。
- nosupport
配布リストに対応しません。

● **-w デフォルトのバインディング**

～《fixed》

あて先クラスタキューのバインディング (DefBind 属性) を制御します。オープンするキューがクラスタキューの場合に適用されます。

- fixed
キューがオープンされたハンドルにあて先を固定します。

- not_fixed

キューにメッセージが登録される時、メッセージ単位にあて先を決定します。

● -v メッセージ滞留時間監視の要否

～《off》

メッセージの滞留時間を監視するかどうかを指定します。

このオプションに on を指定する場合は、set 形式の mqa_msg_stay_time オペランドにメッセージ滞留時間を指定してください。

- off

メッセージ滞留時間監視を実行しません。

- on

メッセージ滞留時間監視を実行します。

コマンド引数

● モデルキューの定義名

～< 1～48 文字の MQ 文字列 >

モデルキューの定義名を指定します。

MQOPEN 命令で動的キューを作成する場合は、この引数で指定した名称をオブジェクト名として指定してください。

● キューファイル名

～< 1～59 文字の文字列 >

動的キューを格納するキューファイル名を指定します。mqaquegrp 定義コマンドで設定したキューファイル名を指定してください。

この引数は、-k オプション（定義タイプ）で temporary を指定した場合も指定してください。一時的動的キューの情報は、基本的にはメモリ上だけで管理されますが、一時的動的キューの情報を管理するバッファの都合で、ファイルに出力される場合があります。その場合、どのキューファイルに出力すればよいかをキューマネージャは知る必要があるため、この引数は一時的動的キューに対しても有効です。

メッセージをキューに登録する場合は、指定されたキューファイルが属するキューファイルグループのどれかのキューファイルに登録します。

キューファイルグループに複数のキューファイルが対応づけられている場合、どのキューファイルにメッセージに登録するかは、キューアクセス競合時の負荷分散を考慮して、MQA サービスが決定します。

キューファイルパス名およびキューファイル名にスペースを指定しないでください。また、ファイルパス名の"/"個所に"/"を追加したキューファイルパス名を指定しないでください。指定した場合、動作は保証されません。

mqaremque (リモートキューのローカル定義)

形式

```
{ { mqaremque  [-q キュー記述子]  [-p 登録許可]
                [-y 省略時のメッセージ優先度]
                [-e 省略時のメッセージ永続性]
                [-r リモートキュー名]
                [-w デフォルトのバインディング]
                [-c クラスタ内共用クラスタ名称]
                [-m リモートキューマネージャ名]
                [-x 転送キュー名]
                リモートキューのローカル定義名 } }
```

機能

リモートキューのローカル定義を指定します。

この定義コマンドは、MQA サービス定義に複数指定できますが、リモートキューのローカル定義名は重複して指定できません。重複して指定した場合は、MQA サーバ開始処理でエラーになります。また、モデルキューの定義名、別名キュー名、および存在するキュー名と重複した名称をリモートキューのローカル定義名に指定した場合も、MQA サーバ開始処理でエラーになります。

TP1/Message Queue 全体で登録できるキューの最大数は 20,480 個です。

オプション

● -q キュー記述子

～< 1～64 文字の文字列 >

キューについてのユーザに任意の情報 (QDesc 属性) を指定します。キューマネージャはこの内容を一切チェックしません。省略した場合、キュー記述子は空白になります。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-q "svrg01,svr01"
```

● -p 登録許可

～ 《allowed》

作成されるキューに対する登録操作を許可するかどうか (InhibitPut 属性) を指定します。

- allowed

登録操作は許可されます (属性値: MQQA_PUT_ALLOWED)。

- inhibited

登録操作は禁止されます（属性値：MQQA_PUT_INHIBITED）。

● -y 省略時のメッセージ優先度

～<符号なし整数> ((0~9)) 《0》

キューに登録されるメッセージの省略時の優先度（DefPriority 属性）を指定します。優先度は、指定した値が小さいほど低く、大きいほど高くなります。

メッセージ記述子（MQMD 構造体）に MQPRI_PRIORITY_AS_Q_DEF が指定され、メッセージがキューに登録される場合に、この指定値は使用されます。

● -e 省略時のメッセージ永続性

～ 《persistent》

キューに登録されるメッセージの省略時の永続性（DefPersistence 属性）を指定します。

- persistent

メッセージを永続にします（属性値：MQPER_PERSISTENT）。

- notpersistent

メッセージを永続にしません（属性値：MQPER_NOT_PERSISTENT）。

● -r リモートキュー名

～< 1~48 文字の MQ 文字列>

リモートキュー名（RemoteQName 属性）を指定します。-m オプション（リモートキューマネージャ名）のキューマネージャが認識できるキュー名を指定してください。

アプリケーションがリモートキューのローカル定義をオープンする場合、この指定値は省略できません。一方、この定義をキューマネージャの別名定義として使用する場合は、必ず省略してください。また、この定義を応答キューの別名として使用する場合は、この指定値は、応答キューの名称にする必要があります。

● -w デフォルトのバインディング

～ 《fixed》

あて先クラスタキューのバインディング（DefBind 属性）を制御します。オープンするキューがクラスタキューの場合に適用されます。

- fixed

キューがオープンされたハンドルにあて先を固定します。

- not_fixed

キューにメッセージが登録される時、メッセージ単位にあて先を決定します。

● -c クラスタ内共用クラスタ名称

～< 1～48 文字の MQ 文字列>

キューが属するクラスタの名称を指定します。

省略した場合、キューはクラスタに属しません。

● -m リモートキューマネージャ名

～< 1～48 文字の MQ 文字列>

r オプション（リモートキュー名）のリモートキューを定義しているキューマネージャの名称（RemoteQMGrName 属性）を指定します。

アプリケーションがリモートキューのローカル定義をオープンする場合、この指定値を省略したり、ローカルキューマネージャ名を指定したりできません。しかし、この定義をキューマネージャの別名定義として使用する場合は、この指定値を省略したり、ローカルキューマネージャ名を指定したりできます。

-x オプション（転送キュー名）を省略した場合は、ここで指定した名称と同じ名称のローカルキューが必要です。このローカルキューは、転送キューとして使用されます。このローカルキューが MQA サーバ開始時に存在しなくても、MQA サーバは正常に立ち上がります。アプリケーションがリモートキューのローカル定義をオープンしたときに、リモートキューマネージャ名と同じローカルキューが転送キューとして定義されているかがチェックされます。

● -x 転送キュー名

～< 1～48 文字の MQ 文字列>

リモートキューのローカル定義をオープンした場合、メッセージの転送に使用されるローカル転送キューの名称（XmitQName 属性）を指定します。省略した場合、-m オプション（リモートキューマネージャ名）で指定したリモートキューマネージャ名と同じ名称のキューが転送キューとして使用されます。

リモートキューのローカル定義がキューマネージャの別名として使用され、-m オプション（リモートキューマネージャ名）の指定値を省略したり、ローカルキューマネージャの名称を指定したりした場合、この指定値は無視されます。

コマンド引数

● リモートキューのローカル定義名

～< 1～48 文字の MQ 文字列>

リモートキューのローカル定義名を指定します。

MQOPEN 命令では、この引数で指定した名称をオブジェクト名として指定してください。

mqaalsque (別名キューの属性定義)

形式

```
{ { mqaalsque  [-q キュー記述子]    [-g 取り出し許可]
                [-p 登録許可]
                [-y 省略時のメッセージ優先度]
                [-e 省略時のメッセージ永続性]
                [-w デフォルトのバインディング]
                [-c クラスタ内共用クラスタ名称]
                別名キュー名  ベースキュー名 }
```

機能

別名キューの属性を指定します。

この定義コマンドで指定した別名キュー名を、MQOPEN 命令のオブジェクト名に指定すると、ベースキューに異なる名称を指定できます。

この定義コマンドは、MQA サービス定義に複数指定できますが、別名キュー名は重複して指定できません。重複して指定した場合は、MQA サーバ開始処理でエラーになります。また、モデルキューの定義名、およびリモートキューのローカル定義名と重複した名称を別名キュー名に指定した場合も、MQA サーバ開始処理でエラーになります。

TP1/Message Queue 全体で登録できるキューの最大数は 20,480 個です。

オプション

● -q キュー記述子

～< 1～64 文字の文字列>

キューについてのユーザに任意の情報 (QDesc 属性) を指定します。キューマネージャはこの内容を一切チェックしません。省略した場合、キュー記述子は空白になります。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-q "svrg01,svr01"
```

● -g 取り出し許可

～《allowed》

作成される動的キューに対する取り出し操作を許可するかどうか (InhibitGet 属性) を指定します。

- allowed

取り出し操作は許可されます (属性値: MQQA_GET_ALLOWED)。

- inhibited
取り出し操作は禁止されます（属性値：MQQA_GET_INHIBITED）。

● -p 登録許可

～《allowed》

作成されるキューに対する登録操作を許可するかどうか（InhibitPut 属性）を指定します。

- allowed
登録操作は許可されます（属性値：MQQA_PUT_ALLOWED）。
- inhibited
登録操作は禁止されます（属性値：MQQA_PUT_INHIBITED）。

● -y 省略時のメッセージ優先度

～<符号なし整数>((0~9))《0》

キューに登録するメッセージの省略時の優先度（DefPriority 属性）を指定します。優先度は、指定した値が小さいほど低く、大きいほど高くなります。

メッセージ記述子（MQMD 構造体）に MQPRI_PRIORITY_AS_Q_DEF が指定され、メッセージがキューに登録される場合に、この指定値は使用されます。

● -e 省略時のメッセージ永続性

～《persistent》

キューに登録するメッセージの省略時の永続性（DefPersistence 属性）を指定します。

- persistent
メッセージを永続にします（属性値：MQPER_PERSISTENT）。
- notpersistent
メッセージを永続にしません（属性値：MQPER_NOT_PERSISTENT）。

● -w デフォルトのバインディング

～《fixed》

あて先クラスタキューのバインディング（DefBind 属性）を制御します。オープンするキューがクラスタキューの場合に適用されます。

- fixed
キューがオープンされたハンドルにあて先を固定します。
- not_fixed
キューにメッセージが登録される時、メッセージ単位にあて先を決定します。

- **-c クラスタ内共用クラスタ名称**

～< 1～48 文字の MQ 文字列>

キューが属するクラスタの名称を指定します。

省略した場合、キューはクラスタに属しません。

コマンド引数

- **別名キュー名**

～< 1～48 文字の MQ 文字列>

別名キューの名称を指定します。

- **ベースキュー名**

～< 1～48 文字の MQ 文字列>

別名を付けるベースキューの名称 (BaseQName 属性) を指定します。

mqaprcdef (プロセス定義)

形式

```
{ { mqaprcdef  [-p プロセス記述子]
                [-t アプリケーションタイプ]
                [-i アプリケーション記述子]    [-e 環境データ]
                [-u ユーザデータ]    プロセス名 }
```

機能

ユーザが使用するプロセスの属性を指定します。

この定義コマンドは、MQA サービス定義に複数指定できますが、プロセス名は重複して指定できません。重複して指定した場合、MQA サーバ開始処理でエラーになります。

オプション

● -p プロセス記述子

～< 1～64 文字の文字列 >

トリガモニタアプリケーションに対する、ユーザに任意の情報 (ProcessDesc 属性) を指定します。キューマネージャはこの指定値を一切チェックしません。省略した場合、プロセス記述子は空白になります。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-p "svrg01,svr01"
```

● -t アプリケーションタイプ

～< 符号なし整数 > ((15, または 65536～99999999)) 《15》

トリガモニタアプリケーションのタイプ (ApplType 属性) を指定します。この指定値は、トリガメッセージの一部としてイニシエーションキューに書き込まれます。

不正な値を指定した場合は、MQA サーバ開始処理でエラーになります。

- 15
OpenTP1 アプリケーション (属性値: MQAT_OPEN_TP1)
- 65536～99999999
ユーザが使用できる値 (属性値: なし)

● -i アプリケーション記述子

～< 1～256 文字の文字列 >

トリガモニタアプリケーションを識別するための文字列 (ApplId 属性) を指定します。この文字列は、トリガメッセージの一部としてイニシエーションキューに書き込まれます。

キューマネージャはこの指定値を一切チェックしません。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-i "svrg01,svr01"
```

● -e 環境データ

～< 1～128 文字の文字列>

トリガモニタアプリケーションについての、環境に関連した情報を含む文字列 (EnvData 属性) を指定します。この文字列は、トリガメッセージの一部としてイニシエーションキューに書き込まれます。

キューマネージャはこの指定値を一切チェックしません。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-e "svrg01,svr01"
```

● -u ユーザデータ

～< 1～128 文字の文字列>

トリガモニタアプリケーションについての、ユーザ情報を含む文字列 (UserData 属性) を指定します。この文字列は、トリガメッセージの一部としてイニシエーションキューに書き込まれます。

キューマネージャはこの指定値を一切チェックしません。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-u "svrg01,svr01"
```

コマンド引数

● プロセス名

～< 1～48 文字の MQ 文字列>

トリガモニタアプリケーションのプロセス名を指定します。

mqaqueatl 定義コマンドの-a オプション (プロセス名) で指定するプロセス名と同じ名称を指定してください。

キューマネージャはこの名称をチェックしません。

mqaquemgr (キューマネージャ定義)

形式

```
mqaquemgr  -n キューマネージャ名  [-q キューマネージャ記述子]
            [-t トリガ間隔]        [-d デッドレターキュー名]
            [-p 最大メッセージ優先度]
            [-x 省略時の転送キュー名]  [-h 最大ハンドル数]
            [-l 最大メッセージ長]      [-s トランザクション有効性]
            [-j 配布リストのサポート]
```

機能

キューマネージャの属性を指定します。

この定義コマンドは、MQA サービス定義に一つだけ指定できます。複数指定した場合は、MQA サーバ開始処理でエラーになります。

オプション

● -n キューマネージャ名

～< 1～48 文字の MQ 文字列 >

キューマネージャ名 (QMgrName 属性) を指定します。アプリケーションをローカルキューマネージャに接続する場合に使用される名称です。

名称の先頭 12 文字は、ユニークなメッセージ識別子が生成される場合に使用されます。キューマネージャのネットワークでメッセージ識別子がユニークであるために、通信するキューマネージャ同士では、名称の先頭 12 文字が異ならなければなりません。

● -q キューマネージャ記述子

～< 1～64 文字の文字列 >

キューマネージャに対するユーザに任意の情報 (QMgrDesc 属性) を指定します。キューマネージャはこの指定値を一切チェックしません。省略した場合、キューマネージャ記述子は空白になります。

指定値に、(コンマ) を含む場合は文字列全体を" (引用符) で囲んでください。

指定例を次に示します。

```
-q "svrg01,svr01"
```

● -t トリガ間隔

～< 符号なし整数 > ((0～999999999)) 《999999999》 (単位：ミリ秒)

mqaqueatl 定義コマンドの-t オプション (トリガタイプ) で first を指定した場合の、トリガメッセージ数を制御するために使用される時間間隔 (TriggerInterval 属性) を指定します。

トリガタイプが first の場合、通常、キューが空の状態のときにメッセージが到着すると、トリガメッセージが生成されます。しかし特定の状況の下では、キューが空でなくても、追加のトリガメッセージが生成されることがあります。この追加のトリガメッセージの生成頻度を制御します。

チャンネルトリガを起動する場合、通信相手システムの起動状態によってイニシエーションキューが満杯になることがあります。そこで余分なトリガメッセージを発生させないために、次に示す計算式に従って定義を指定してください。

トリガ間隔 (mqaquemgr 定義コマンドの-t オプション指定値)
≥ イニシエーションキュー監視タイマ値
 (mqttcp 定義コマンドの-v itim オペランド指定値)
+ 基本タイマ値
 (mqtttim 定義コマンドの-t btim オペランド指定値 +
 bmtim オペランド指定値)

なお、上記の計算式に従って定義を指定した場合でも、トリガ起動チャンネル数の 3 倍のメッセージが一時的に発生することがありますので、注意してください。

● -d デッドレターキュー名

～< 1～48 文字の MQ 文字列>

ローカルキューマネージャで定義されているキューに登録できない場合に、そのメッセージを転送するキュー (デッドレターキュー) の名称 (DeadLetterQName 属性) を指定します。省略した場合、デッドレターキュー名は空白になります。デッドレターキューにメッセージに登録する条件については、マニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

● -p 最大メッセージ優先度

～< 符号なし整数> ((0～9)) 《9》

キューマネージャでサポートするメッセージ優先度の最大値 (MaxPriority 属性) を指定します。

● -x 省略時の転送キュー名

～< 1～48 文字の MQ 文字列>

省略時の転送キューの名称 (DefXmitQName 属性) を指定します。

このキューは、使用する転送キューの指示がほかにない場合、リモートキューマネージャへのメッセージ転送用に使用されます。

● -h 最大ハンドル数

～< 符号なし整数> ((1～65535)) 《10》

プロセスから同時に使用できるオープン状態のオブジェクトハンドルの最大数 (MaxHandles 属性) を指定します。

最大ハンドル数は、次に示すハンドル数のうちで最大のハンドル数以上になるように指定してください。

- 一つのアプリケーションが同時に使用するハンドル数
- クラスタ機能を使用する場合、リポジトリ管理サーバの最大ハンドル数 (最大ハンドル数: 6)
クラスタ機能を使用する場合、最大ハンドル数に 5 以下の値を指定すると、OpenTP1 の開始時に KFCA31204-E メッセージを出力し、OpenTP1 が異常終了します。
- MQC サーバ機能を使用する場合、クライアントアプリケーションから接続された接続で同時に使用するオブジェクトハンドル数
 - C 言語または COBOL 言語で作成したアプリケーションの場合
MQCONN 命令で取得した接続ハンドルで実行する同時 MQOPEN 命令の数
 - C++ 言語で作成したアプリケーションの場合
一つの ImqQueueManager クラスを使用して同時にオープンするオブジェクトの数
 - Java 言語で作成したアプリケーションの場合
一つの MQQueueManager クラスから生成するオブジェクトの数
 - JMS インタフェースで作成したアプリケーションの場合
一つの論理接続 (QueueSession) から生成するオブジェクトの数

● -l 最大メッセージ長

~ <符号なし整数> ((1 ~ 128000000)) 《4096000》

キューマネージャによって処理されるメッセージの最大長 (MaxMsgLength 属性) を指定します。

キューマネージャで作成されるすべてのキューに対して、この指定値がメッセージ長の最大値です。mqaqueatl 定義コマンドの -l オプション (最大メッセージ長) はこの指定値の大きさを超えて指定できません。

● -s トランザクション有効性

~ 《sync》

接続するキューマネージャが、MQGET 命令、MQPUT 命令、および MQPUT1 命令で同期点処理をサポートするかどうか (SyncPoint 属性) を指定します。

- sync
トランザクションの同期点処理を利用できます (属性値: MQSP_AVAILABLE)。
- nosync
トランザクションの同期点処理を利用できません (属性値: MQSP_NOT_AVAILABLE)。

● -j 配布リストのサポート

~ 《nosupport》

MQPUT 命令, および MQPUT1 命令で, 配布リスト付きのメッセージをキューに書き込めるかどうかを指定します。

- support
配布リストに対応します。
- nosupport
配布リストに対応しません。

mqamqtnam (MQT デーモン構成定義)

形式

```
{ { mqamqtnam  [-b 総バッチメッセージ数]
                MQTサーバ名またはMQCリスナサーバ名
                プロトコル名称 MCA数 } }
```

機能

メッセージの送受信をするための MQT デーモン (MQT サーバ) の構成を定義します。MQA サーバは、指定された MQT デーモンを開始処理で起動します。

関連製品である TP1/Message Queue Access を使用する場合は MQC リスナサーバの構成を定義します。MQA サーバは、指定された MQC リスナサーバを開始処理で起動します。

この定義を指定すると、MQA サーバはチェックポイントダンプを取得するため、MQA サーバ用のチェックポイントダンプの定義を指定する必要があります。MQA サーバ用のチェックポイントダンプの定義については、マニュアル「OpenTP1 システム定義」を参照してください。

オプション

● -b 総バッチメッセージ数

～<符号なし整数>((1～2549745))《255》

一つの TCP 定義に対するバッチメッセージの総数を指定します。

64bit 版の場合は MQA サービス定義に定義されている各 mqamqtnam 定義コマンドの-b オプション (総バッチメッセージ数) の合計値が 120,000,000 以下となるように定義してください。合計値が 120,000,000 よりも大きい場合、KFCA31210-E メッセージを出力して異常終了します。

コマンド引数

● MQT サーバ名または MQC リスナサーバ名

～<_ (アンダスコア) で始まる 2～8 文字の識別子>

MQT サーバ名または MQC リスナサーバ名を指定します。

MQT サーバ名は、MQT 通信構成定義のファイル名です。先頭のアンダスコアを含めて 8 文字までの英数字で指定してください。

関連製品である TP1/Message Queue Access を使用する場合は MQC リスナサーバ名を指定します。MQA サービス定義に指定できる MQC リスナサーバは一つだけです。複数を指定した場合、MQC サービス開始時に KFCA31183-E が出力されます。

● プロトコル名称

～< 1～14 文字の識別子>

プロトコル名称を指定します。

関連製品である TP1/Message Queue Access を使用する場合は mqctcp を指定します。

● MCA 数

～< 符号なし整数> ((1～255))

一つの MQT サーバ当たりの MCA の数を指定します。

関連製品である TP1/Message Queue Access を使用する場合は 1 を指定します。

起動する MQT デーモン種別と mqamqtnam 定義コマンドの指定値について、次の表に示します。

表 4-9 起動する MQT デーモン種別と mqamqtnam 定義コマンドの指定値

起動する MQT デーモン種別	総バッチメッセージ数 ※1	MQT サーバ名	プロトコル名称	MCA 数
標準デーモン 標準デーモン (UOC 組み込み)	チャンネル定義で sender または server に指定されたバッチサイズの合計。受信チャンネルだけの場合は"1"を指定します。	MQT 通信構成定義のファイル名	mqttcp mqtu で始まるシステムサービス定義名※2	MQT 通信構成定義で指定するチャンネル定義の数 (デフォルトチャンネル定義は含みません)
クラスタセングデーモン クラスタセングデーモン (UOC 組み込み)	次に示す計算式で求めた値 (事前定義チャンネルに指定したバッチサイズの合計) + (該当する MQT に接続されるクラスタレシーバ内で最大のバッチサイズ) × (MCA 数 - 事前定義チャンネル数)	MQT 通信構成定義のファイル名	mqttcps mqttcpsu で始まるシステムサービス定義名※2	事前定義チャンネル数 + 自動定義チャンネル数 (デフォルトチャンネル定義は含みません) (算出の目安: 同一クラスタ内の自システム以外のクラスタレシーバチャンネル数が最大になります)
クラスタレシーバデーモン クラスタレシーバデーモン (UOC 組み込み)	1	MQT 通信構成定義のファイル名	mqttcpcr mqttcpcru で始まるシステムサービス定義名※2	事前定義したクラスタレシーバで動作するインスタンス数 (算出の目安: 同一クラスタ内の自システム以外の MQ システム数)
MQC リスナサーバ	1	MQC リスナサーバ名	mqctcp	1

注※1

ネゴシエーション時の自動定義クラスタセンダチャンネルのバッチサイズは mqamqtnam 定義コマンド-b オプション指定値および事前定義クラスタセンダチャンネルに指定したバッチサイズに依存します。詳細については、この章の「[チャンネルのネゴシエーション](#)」を参照してください。

この値が不足していると、相手システムのクラスタレシーバで指定したバッチサイズより小さいバッチサイズでチャンネルが確立されることがあります。

注※2

詳細については、「[3.5.4 システムサービス情報定義の作成](#)」を参照してください。

MQT サービス定義

TP1/Message Queue の定義のうち、MQT サービス定義について説明します。

概要

MQT サービス定義は、OpenTP1 のシステムサービス定義の一つです。

MQT サービス定義の定義ファイルは、OS のテキストエディタを使用して、テキストファイルとして作成します。MQT サービス定義の定義ファイル名（完全パス名）を次に示します。

```
$DCCONFPATH/mqt
```

機能

通信するための実行環境を定義します。

種類

MQT サービス定義の種類について、次の表に示します。

表 4-10 MQT サービス定義の種類

形式	オプション	定義内容	指定値
<code>set max_socket_descriptors</code>	なし	ソケット用ファイル記述子の最大数	<符号なし整数> Linux 以外の場合((64~1998)) Linux の場合((64~974)) UNIX の場合《304》 Windows の場合《64》
<code>set mqt_trace_file_path</code>	なし	MQT トレースファイル格納ディレクトリパス名	<1~80 バイトの文字列> UNIX の場合《\$DCDIR/spool》 Windows の場合《%DCDIR% ¥spool》
<code>set mqt_tcpnodelay</code>	なし	MQT サーバのチャンネルの TCP/IP 通信のソケット属性に、TCP_NODELAY オプションを使用するかどうか	《Y》 N

set 形式の MQT サービス定義

set 形式の MQT サービス定義について説明します。

形式

```
[set max_socket_descriptors = ソケット用ファイル記述子の最大数]
[set mqt_trace_file_path= MQTトレースファイル格納ディレクトリパス名]
[set mqt_tcpnodelay = Y | N]
```

オペランド

● max_socket_descriptors=ソケット用ファイル記述子の最大数

～<符号なし整数> ((Linux 以外の場合：64～1998, Linux の場合：64～974))

UNIX の場合 《304》

Windows の場合 《64》

MQT サーバでソケット用に使用するファイル記述子の最大数を指定します。

この定義で指定した属性は、すべての MQT サーバで共通の属性となります。

MQT サーバの MCA はソケットを使用した TCP/IP 通信を行います。また、MQA サーバとの間でもソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、MCA の定義内容によっては、ソケット用のファイル記述子の最大数を変更する必要があります。

MQT サーバが使用するソケット用ファイル記述子の最大数の計算式を、次に示します。計算式の最大値が 64 より大きい場合は、その値を指定し、64 以下の場合は、64 を指定します。MQT サーバを複数定義する場合は、計算式が最大となる MQT サーバの値を指定してください。

$$\frac{\begin{array}{l} \uparrow \text{MQTサーバの最大MCA数 (mqamqtnam定義コマンドで指定した値)} \\ \uparrow \text{+ イベント方式の転送キュー監視 (mqtalocha -z event) をするMCA数} \\ \uparrow \text{+ MQTサーバに対して同時に処理要求をするMQT運用コマンド数} \\ \uparrow \text{+ イベント方式の転送キュー監視をするMCAの転送キューにメッセージを} \\ \uparrow \text{登録するUAPのプロセス数} \\ \uparrow \text{+32} \end{array}}{0.8}$$

(凡例) $\uparrow \uparrow$: 小数点以下を切り上げます。

注※

センダチャンネルまたはサーバチャンネルだけ該当します。

MQT サーバが使用中のソケット用ファイル記述子数が、MQT サービス定義の `max_socket_descriptors` オペランドに指定したソケット用ファイル記述子の最大数よりも多くなった場合、次に示すどれかのメッセージを出力して、OpenTP1 が異常終了することがあります。

出力メッセージ ID	内容	出力先
KFCA00107-E	"open(モジュール名)" failed errno=24 : Too many open files	標準エラー出力
KFCA00307-E	ソケット不足が発生したため RPC ができません。	標準エラー出力
KFCA16331-E	TCP/IP インタフェースエラー (エラーコード=24)	メッセージログファイル
KFCA18376-E	トレース出力失敗 (理由コード=-1)	メッセージログファイル

● `mqt_trace_file_path=MQT` トレースファイル格納ディレクトリパス名

～ < 1～80 バイトの文字列 >

UNIX の場合 《\$DCDIR/spool》

Windows の場合 《%DCDIR%\spool》

MQT トレースファイルを格納するディレクトリの絶対パス名（UNIX 版の場合は通常ファイルを作成できるディレクトリの絶対パス名）を指定します。パス名は 80 バイト以内になしてください。

● `mqt_tcpnodelay = Y | N`

～ 《Y》

MQT サーバのチャネルの TCP/IP 通信のソケット属性に、TCP_NODELAY オプションを使用するかどうかを指定します。この定義は、OpenTP1 システム上のすべての MQT サーバで有効になります。

TCP_NODELAY オプションを使用すると（このオペランドに Y を指定または省略すると）、Nagle アルゴリズムが無効になるので、送信済みパケットの応答待ちの状態でも遅延させることなくパケットを送信できます。

ただし、TCP_NODELAY オプションを使用することで、通信時の送信効率が低下し、ネットワークの負荷が大きくなる場合があります。これは、Nagle アルゴリズムが無効にすると、データ量が小さい電文でも個々のパケットとして送信され、全体的なパケット量が増加することがあるからです。このため、オペランドを指定する場合は、ネットワークの帯域などを考慮し、この機能の必要性を十分に検討してください。

- Y
チャネルのソケット属性に TCP_NODELAY オプションを使用する。
- N
チャネルのソケット属性に TCP_NODELAY オプションを使用しない。

MQT 通信構成定義

TP1/Message Queue の定義のうち、MQT 通信構成定義について説明します。

概要

MQT 通信構成定義は、OpenTP1 のネットワークコミュニケーション定義の一つです。MQT 通信構成定義は、MQT 共通定義と TCP 定義に分けられます。

MQT 通信構成定義の定義ファイルは、OS のテキストエディタを使用して、定義ソースファイルを作成します。定義ソースファイルは、MQT 共通定義と TCP 定義にそれぞれの一つずつ作成します。定義ファイルの作成方法については、この章の「MQT 定義オブジェクトファイルの作成手順」を参照してください。

種類

MQT 通信構成定義の種類について、次の表に示します。

表 4-11 MQT 通信構成定義の種類

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
MQT 共通定義	mqttenv (MQT 環境定義) 指定数：1	-s	なし	MQT 通信プロセス識別子	< 16 進数字 > ((01~ff))
		-p	なし	ポート番号	< 符号なし整数 > ((1024~65535))
	mqtttim (タイマ定義) 指定数：0~1	-t	btim	基本タイマ値	< 符号なし整数 > ((0~60)) 《1》 (単位：秒)
			bmtim	基本タイマ値 (ミリ秒)	< 符号なし整数 > ((0~999)) 《0》 (単位：ミリ秒)
mqttbuf (バッファグループ定義) 指定数：1~512	-g	groupno	バッファグループ番号	< 符号なし整数 > ((1~512))	
		length	バッファ長	< 符号なし整数 > ((1024~1073741824)) (単位：バイト)	
		count	バッファ数	< 符号なし整数 > ((1~65535))	
		extend	拡張バッファ数	< 符号なし整数 > ((0~65535)) 《0》	

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
TCP 定義	mqttcp (TCP 構成定義) 指定数：1	-i	なし	イニシエーション キュー名	< 1~48 文字の MQ 文字列 >
		-p	なし	プロトコル種別	tcp
		-g	rcvbuf	受信バッファグループ 番号	< 符号なし整数 > ((1~512))
		-r	portno	自システムのポート 番号	< 符号なし整数 > ((1024~65535))
			servname	自システムのサービ ス名	< 1~32 文字の識別子 (ハイフン (-) を含む) >
		-v	itim	イニシエーション キュー監視タイマ値	< 符号なし整数 > ((0~65535)) 《60》 (単位：秒)
			etim	終了処理監視タイマ 値	< 符号なし整数 > ((0, 10~65535)) 《360》 (単位：秒)
			rtim	チャンネル確立完了監 視タイマ値	< 符号なし整数 > ((0~65535)) 《60》 (単位：秒)
		-t	disk	MQT トレースの ディスク出力要否	《yes》 no
			bufcnt	バッファ格納個数	< 符号なし整数 > ((10~2147483647)) 《2500》
			trccnt	トレースファイル 個数	< 符号なし整数 > ((3~99)) 《3》
			bufsize	トレースバッファサ イズ	< 符号なし整数 > ((20480~128000000)) 《20480》 (単位：バイト)
	-z	なし	イニシエーション キュー監視方式	《time》 event	
	mqttcps (クラスタセンダ TCP 構 成定義) 指定数：1	-p	なし	プロトコル種別	tcp
		-v	stim	チャンネル要求監視 間隔	< 符号なし整数 > ((1~65535)) 《5》 (単位：秒)
			etim	終了処理監視タイマ 値	< 符号なし整数 > ((0, 10~65535)) 《360》 (単位：秒)

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
TCP 定義	mqttcps (クラスタセンダ TCP 構成定義) 指定数：1	-t	disk	MQT トレースの ディスク出力要否	《yes》 no
			bufcnt	バッファ格納個数	<符号なし整数> ((10~2147483647)) 《2500》
			trccnt	トレースファイル 個数	<符号なし整数> ((3~99)) 《3》
			bufsize	トレースバッファサ イズ	<符号なし整数> ((20480~128000000)) 《20480》 (単位：バイト)
	mqttcpcr (クラスタレシーバ TCP 構成定義) 指定数：1	-p	なし	プロトコル種別	tcp
			-g	rcvbuf	受信バッファグルー プ番号
		-r	portno	自システムのポート 番号	<符号なし整数> ((1024~65535))
			servname	自システムのサービ ス名	< 1~32 文字の識別子 (ハイフン (-) を含む) >
		-v	stim	チャンネル要求監視 間隔	<符号なし整数> ((1~65535)) 《5》 (単位：秒)
			etim	終了処理監視タイマ 値	<符号なし整数> ((0, 10~65535)) 《360》 (単位：秒)
			rtim	開始要求受信監視タ イマ値	<符号なし整数> ((0~65535)) 《60》 (単位：秒)
		-t	disk	MQT トレースの ディスク出力要否	《yes》 no
			bufcnt	バッファ格納個数	<符号なし整数> ((10~2147483647)) 《2500》
			trccnt	トレースファイル 個数	<符号なし整数> ((3~99)) 《3》
			bufsize	トレースバッファサ イズ	<符号なし整数> ((20480~128000000)) 《20480》 (単位：バイト)
		mqtalccha	-c	なし	チャンネル名

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
TCP 定義	(チャンネル定義) 指定数：1～255*	-y	type	チャンネルタイプ	sender server receiver requester clusdr clusrchr defsnd defsrv defrcv defreq defcsn
		-j	なし	バッチサイズ	<符号なし整数> ((1～9999)) チャンネルタイプが sender, server, receiver, または requester の場合 《1》 チャンネルタイプが clusdr または cluschr の場合 《50》
		-q	なし	転送キュー名	< 1～48 文字の MQ 文字列>
		-m	maxmsg	最大メッセージ長	<符号なし整数> ((1024～127996000)) (単位：バイト)
			maxseg	最大セグメントサイズ	<符号なし整数> ((1024～1073741824)) (単位：バイト)
		-w	なし	メッセージシーケンス番号の最大値	<符号なし整数> ((100～999999999)) 《999999999》
		-g	sndbuf	送信バッファグループ番号	<符号なし整数> ((1～512))
			rcvbuf	受信バッファグループ番号	<符号なし整数> ((1～512))
			buftype	バッファ方式	seg 《msg》
		-e	medbuf	メッセージ編集出口編集用バッファグループ番号	<符号なし整数> ((1～512))
			medcnt	メッセージ編集出口編集用バッファ数	<符号なし整数> ((1～65535))
		-i	なし	チャンネルの確立方法	auto manual auto2 チャンネルタイプが sender, server, receiver, または requester の場合 《manual》 チャンネルタイプが clusdr, または cluschr の場合 《auto》

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
TCP 定義		-b	bretry	短期確立再試行の 要否	《yes》 no
			bretrycnt	短期確立再試行の 回数	<符号なし整数> ((1~999999999)) 《10》 (単位:回)
			bretryint	短期確立再試行の 間隔	<符号なし整数> ((0~999999999)) 《60》 (単位:秒)
			bretrylg	長期確立再試行の 要否	《yes》 no
			bretrylgcnt	長期確立再試行の 回数	<符号なし整数> ((1~999999999)) 《999999999》 (単位:回)
			bretrylgint	長期確立再試行の 間隔	<符号なし整数> ((0~999999999)) 《1200》 (単位: 秒)
			bretrymcp	MQ プロトコル障害 再試行の要否	《yes》 no
		-r	ipaddr	自システムの IP アド レス	<符号なし整数> ((0~255)) (形式:nnn.nnn.nnn.nnn)
			hostname	自システムのホスト 名	< 1~255 バイトの文字列>
			portno	自システムのポート 番号	<符号なし整数> ((1024~65535))
			servname	自システムのサービ ス名	< 1~32 文字の識別子 (ハイフン(-)を含む) >
			portnum	自システムのポート 番号使用数	<符号なし整数> ((1~64512)) 《1》
		-o	oipaddr	相手システムのホス トの IP アドレス	<符号なし整数> ((0~255)) (形式:nnn.nnn.nnn.nnn)
			ohostname	相手システムのホス ト名	< 1~255 バイトの文字列>
			oportno	相手システムのホス トのポート番号	<符号なし整数> ((1024~65535))
			oservname	相手システムのホス トのサービス名	< 1~32 文字の識別子 (ハイフン (-) を含む) >

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
TCP 定義		-v	tim1	チャンネル確立応答受信監視タイマ値	<符号なし整数> ((0~8191)) 《30》 (単位: 秒)
			tim2	確認メッセージ受信監視タイマ値	<符号なし整数> ((0~8191)) チャンネルタイプが sender, または server の場合 《10》 チャンネルタイプが clussdr の場合 《30》 (単位: 秒)
			tim3	継続セグメント受信監視タイマ値	<符号なし整数> ((0~8191)) チャンネルタイプが receiver, または requester の場合 《10》 チャンネルタイプが clusrdrv の場合 《30》 (単位: 秒)
			tim4	転送キュー監視間隔	<符号なし整数> ((0~8191)) 《60》 (単位: 秒)
			tim4cnt	転送キュー監視回数	<符号なし整数> ((0~2147483647)) 《1》 (単位: 回)
			dtimefct	切断時間間隔での監視の要否	《yes》 no
			dtim	切断時間間隔	<符号なし整数> ((0~999999)) チャンネルタイプが sender, または server の場合 《60》 チャンネルタイプが clussdr, または clusrdrv の場合 《6000》 (単位: 秒)
			vretry	タイムアウト時のチャンネル確立再試行の要否	yes no 《yes》
			htim	ハートビート間隔	<符号なし整数> ((0~999999)) 《300》 (単位: 秒)
			btim	バッチ終了待ちタイマ値	<符号なし整数>

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
TCP 定義		-v	btim	バッチ終了待ちタイム値	((0~999999)) 《0》 (単位：秒)
			bmtim	バッチ終了待ちタイム値 (ミリ秒)	<符号なし整数> ((0~999)) 《0》 (単位：ミリ秒)
			mtim	継続メッセージ受信監視タイム値	<符号なし整数> ((0~999999)) 《tim3 オペランドの指定値》 (単位：秒)
		-x	sndexdata	送信出口ユーザデータ	< 1~32 バイトの文字列> 《32 バイトのスペース文字》
			rcvexdata	受信出口ユーザデータ	< 1~32 バイトの文字列> 《32 バイトのスペース文字》
			metexdata	メッセージ編集出口ユーザデータ	< 1~32 バイトの文字列> 《32 バイトのスペース文字》
			scyexdata	セキュリティ出口ユーザデータ	< 1~32 バイトの文字列> 《32 バイトのスペース文字》
			sendexit	送信出口名	< 1~128 バイトの文字列> 《128 バイトのスペース文字》
			rcvexit	受信出口名	< 1~128 バイトの文字列> 《128 バイトのスペース文字》
			msgexit	メッセージ出口名	< 1~128 バイトの文字列> 《128 バイトのスペース文字》
			scyexit	セキュリティ出口名	< 1~128 バイトの文字列> 《128 バイトのスペース文字》
		-t	tcpsndbuf	TCP/IP 出力用バッファサイズ	<符号なし整数> ((1024~262144)) (単位：バイト)
			tcprcvbuf	TCP/IP 入力用バッファサイズ	<符号なし整数> ((1024~262144)) (単位：バイト)
			tretrycnt	TCP/IP 確立確認回数	<符号なし整数> ((1~65535)) 《10》 (単位：回)

定義名	定義コマンド	オプション	オペランド	定義内容	指定値
TCP 定義		-t	tretryint	TCP/IP 確立確認 間隔	<符号なし整数> ((1~65535)) 《1》 (単位：秒)
		-d	cnvccsid	変換 CCSID	<符号なし整数> ((1~65535))
		-s	npmspeed	非永続メッセージ転 送速度	normal 《fast》
		-z	なし	転送キュー監視方式	time 《event》
		-a	cluster	クラスタ名	< 1~48 文字の MQ 文字列 >
		-p	netprty	ネットワーク接続優 先順位	<符号なし整数> ((0~9)) 《0》
		-f	adoptchk	強制確立確認条件	off name 《qmgr》 ipaddr all
		-h	なし	注釈	< 1~64 バイトの文字列 > 《64 バイトのスペース文字》
		-mc	mcauser	MCA ユーザ ID	< 1~64 バイトの文字列 > 《64 バイトのスペース文字》
			mcatype	MCA タイプ	《process》 thread
-jn	sndjnl	送信チャネルジャー ナル出力条件	《flush》 noflush		
	mqtalced (チャネル終了定義) 指定数：mqtalcca と 同数	なし	なし	なし	なし

注※

TCP 定義に mqttcp 定義コマンドを指定する場合、チャンネルタイプが sender, server, receiver, または requester のどれかを指定したとき、mqtalcca 定義コマンドは最大 255 個指定できます (デフォルトチャンネル定義は含みません)。

TCP 定義に mqttcps 定義コマンドを指定する場合、チャンネルタイプが clussdr の mqtalcca コマンドを最大 10 個指定できます (デフォルトチャンネル定義は含みません)。

TCP 定義に mqttcpcr 定義コマンドを指定する場合、チャンネルタイプが clusrcvr の mqtalcca 定義コマンドを 1 個指定できます。

MQT 共通定義の指定順序

MQT 通信構成定義のうち、MQT 共通定義のコマンドの詳細を説明します。

MQT 共通定義のコマンドの指定順序を次に示します。

```
mqttenv          (MQT環境定義)
[mqtttim]        (タイム定義)
[ { mqttbuf } ]  (バッファグループ定義)
```

TCP 定義の指定順序

MQT 通信構成定義のうち、TP1/Message Queue の次の TCP 定義コマンドを指定する順序を説明します。

- TCP 定義コマンドの指定順序
- TCP 定義コマンド（デフォルトチャンネル定義）の指定順序
- クラスタセンダ TCP 定義コマンドの指定順序
- クラスタセンダ TCP 定義コマンド（デフォルトチャンネル定義）の指定順序
- クラスタレシーバ TCP 定義コマンドの指定順序

TCP 定義コマンドの指定順序

TCP 定義コマンドの指定順序について、次に示します。

```
mqttcp          (TCP構成定義)
[ { mqtalccha } ] (チャンネル定義)
[ { mqtalced } ] (チャンネル終了定義)
```

TCP 定義コマンド（デフォルトチャンネル定義）の指定順序

TCP 定義コマンド（デフォルトチャンネル定義）の指定順序について、次に示します。

```
[ mqtalccha -y defsnd ]* (デフォルトセンダチャンネル定義)
[ mqtalced ]*           (デフォルトセンダチャンネル終了定義)

[ mqtalccha -y defsrv ]* (デフォルトサーバチャンネル定義)
[ mqtalced ]*           (デフォルトサーバチャンネル終了定義)

[ mqtalccha -y defrcv ]* (デフォルトレシーバチャンネル定義)
[ mqtalced ]*           (デフォルトレシーバチャンネル終了定義)

[ mqtalccha -y defreq ]* (デフォルトリクエストチャンネル定義)
[ mqtalced ]*           (デフォルトリクエストチャンネル終了定義)
```

注※ デフォルトチャンネル定義は、順不同で1~4個の定義を指定できます。
同一チャンネルタイプのデフォルトチャンネル定義は複数指定できません。
デフォルトチャンネルタイプと同一のチャンネルタイプのチャンネル定義は省略できます。

クラスタセンダ TCP 定義コマンドの指定順序

クラスタセンダ TCP 定義コマンドの指定順序について、次に示します。

```
mqttcps        (クラスタセンダTCP構成定義)
[ { mqtalccha } ] (チャンネル定義)
[ { mqtalced } ] (チャンネル終了定義)
```

クラスタセンダ TCP 定義コマンド（デフォルトチャンネル定義）の指定順序

クラスタセンダ TCP 定義コマンド（デフォルトチャンネル定義）の指定順序について、次に示します。

$\left[\begin{array}{l} \text{mqtalccha -y defcsn} \\ \text{mqtalced} \end{array} \right]^*$ (デフォルトクラスタセンダチャンネル定義)
(デフォルトクラスタセンダチャンネル終了定義)

注※ デフォルトチャンネル定義は、1個の定義だけ指定できます。

クラスタレシーバTCP 定義コマンドの指定順序

クラスタレシーバ TCP 定義コマンドの指定順序について、次に示します。

`mqttcpcr` (クラスタレシーバTCP構成定義)
 $\left\{ \left\{ \begin{array}{l} \text{mqtalccha} \\ \text{mqtalced} \end{array} \right\} \right\}$ (チャンネル定義)
(チャンネル終了定義)

mqttenv (MQT 環境定義)

形式

```
mqttenv  -s MQT通信プロセス識別子  
         [-p ポート番号]
```

機能

MQT についての環境を定義します。

オプション

● -s MQT 通信プロセス識別子

～< 16 進数字> ((01～ff))

MQT 通信プロセス識別子を指定します。この指定値は、ほかの MQT 環境定義で指定する MQT 通信プロセス識別子と重複して指定できません。

● -p ポート番号

～< 符号なし整数> ((1024～65535))

このオプションは、使用している TP1/Message Queue のバージョンでは必要ありません。ただし、指定してもエラーにはなりません。このオプションは過去のバージョンで作成された定義との互換性のためだけに指定できるようになっています。

mqtttim (タイマ定義)

形式

```
[mqtttim [-t " [btim = 基本タイマ値]
          [bmtim = 基本タイマ値 (ミリ秒) ] " ] ]
```

機能

MQT で使用されるタイマ値を定義します。

オプション

● -t

(オペランド)

・ btim=基本タイマ値

～<符号なし整数>((0~60))《1》(単位：秒)

MQT で使用される基本タイマ値 (秒) を指定します。

btim オペランド指定値と bmtim オペランド指定値の和の時間が MQT のタイマ制御の精度となります。

・ bmtim=基本タイマ値 (ミリ秒)

～<符号なし整数>((0~999))《0》(単位：ミリ秒)

MQT で使用される基本タイマ値 (ミリ秒) を指定します。

btim オペランド指定値と bmtim オペランド指定値の和の時間が MQT のタイマ制御の精度となります。

btim オペランドと bmtim オペランドには同時に 0 を指定できません。btim オペランドに 0 を指定した場合、bmtim オペランドには 15 以上の値を指定してください。btim オペランド指定値と bmtim オペランド指定値の和が 15ms 未満になる場合、すべての動作を保証できません。

btim オペランド指定値と bmtim オペランド指定値の和は、次のオペランドで指定する各種タイマ値以下かつ 15ms 以上の値にしてください。

- ・ mqttcp -v オプションの itim, etim, rtim オペランド
- ・ mqttcps -v オプションの stim, etim オペランド
- ・ mqttcpsr -v オプションの stim, etim, rtim オペランド
- ・ mqtalccha -b オプションの bretryint, bretrylgint オペランド
- ・ mqtalccha -v オプションの tim1, tim2, tim3, tim4, dtim, htim, btim, bmtim, mtim オペランド
- ・ mqtalccha -t オプションの tretryint オペランド

正しく設定しなかった場合、自システムまたは相手システムで各種タイムアウトが発生したり、タイマ値に指定した時間どおりに動作しなかったりします。

mqttbuf (バッファグループ定義)

形式

```
{ { mqttbuf -g "groupno = バッファグループ番号
                length = バッファ長
                count = バッファ数
                [extend = 拡張バッファ数] " } }
```

機能

メッセージ送受信のバッファについての環境を定義します。

オプション

● -g

(オペランド)

- ・ groupno=バッファグループ番号

～<符号なし整数>((1~512))

メッセージ送受信のバッファグループ番号を指定します。ほかの mqttbuf 定義コマンドのバッファグループ番号と重複して指定できません。

自動定義クラスタセンダチャンネルは、送信用バッファグループ番号にデフォルトで 1 を仮定します。クラスタセンダプロセスではバッファグループ番号に 1 を指定してください。この値以外を指定するには、デフォルトクラスタセンダチャンネル定義または UOC を使用してバッファグループ番号を変更してください。

クラスタセンダプロセスで、このオペランドに 1 を指定した定義が存在しない、かつ MQT 定義オブジェクト生成ユーティリティの mqttcps コマンドの -d オプションにデフォルトチャンネル定義を指定しない場合は、KFCA18021-E のメッセージが出力されます。

- ・ length=バッファ長

～<符号なし整数>((1024~1073741824)) (単位: バイト)

メッセージ送受信に使用するバッファの長さを指定します。

groupno オペランドで指定されたバッファグループ番号によって、該当する値を指定してください。

バッファグループ			用途	指定するバッファ長
定義コマンド	オプション	オペランド		
mqttcp	-g	rcvbuf	開始要求受信用	1024 バイト
mqttcpcr				
mqtalcccha ・ バッファ方式がセグメント方式の場合	-g	sndbuf	メッセージ送信用	送信最大セグメントサイズ
		rcvbuf	メッセージ受信用	受信最大セグメントサイズ

バッファグループ			用途	指定するバッファ長
定義コマンド	オプション	オペランド		
(-g buftype=seg)	-g	rcvbuf	メッセージ受信用	受信最大セグメントサイズ
mqtalcccha <ul style="list-style-type: none"> バッファ方式がメッセージ方式の場合 (-g buftype=msg)	-g	sndbuf	メッセージ送信用	送信最大メッセージ長
		rcvbuf	メッセージ受信用	受信最大メッセージ長
	-e	medbuf	メッセージ編集出口編集用	ユーザ任意

自動定義されるクラスタセンダチャンネルでは、送信用バッファグループ番号にデフォルトで 1 を仮定します。この値を変更するには UOC を作成して属性を変更してください。

mqtalcccha 定義コマンドの -y type オペランドで、receiver または requester を指定しない場合は、mqttcp 定義コマンドの -g オプションを省略できます。したがって、開始要求を受信するための mqttbuf 定義コマンドも省略できます。

・ count=バッファ数

～<符号なし整数> ((1～65535))

メッセージ送受信バッファの個数を指定します。

groupno オペランドで指定されたバッファグループ番号によって、該当する値を指定してください。

バッファグループ			用途	指定するバッファ数
定義コマンド	オプション	オペランド		
mqttcp	-g	rcvbuf	開始要求受信用	バッファ数：1
mqttcpcr				
mqtalcccha <ul style="list-style-type: none"> バッファ方式がセグメント方式の場合 (-g buftype=seg)	-g	sndbuf	メッセージ送信用	バッファ数：送信側 MCA 数
		rcvbuf	メッセージ受信用	バッファ数：受信側 MCA 数
mqtalcccha <ul style="list-style-type: none"> バッファ方式がメッセージ方式の場合 (-g buftype=msg)	-g	sndbuf	メッセージ送信用	バッファ数：送信側 MCA 数
		rcvbuf	メッセージ受信用	バッファ数：受信側 MCA 数
	-e	medbuf	メッセージ編集出口編集用	バッファ数：総 medcnt 値

自動定義されるクラスタセンダチャンネルでは、送信用バッファグループ番号にデフォルトで 1 を仮定します。この値を変更するには UOC を作成して属性を変更してください。

mqtalcccha 定義コマンドの -y type オペランドで、receiver または requester を指定しない場合は、mqttcp 定義コマンドの -g オプションを省略できます。したがって、開始要求を受信するための mqttbuf 定義コマンドも省略できます。

・ extend=拡張バッファ数

～<符号なし整数>((0~65535))《0》

count オペランドで指定したバッファが満杯になった場合に拡張するバッファの数を指定します。

注意事項

チャンネル開始時に、バッファグループ番号 (mqtalcccha 定義コマンドの-g オプションで指定) のバッファ (mqttbuf 定義コマンドの-g オプションの count オペランドおよび extend オペランドで指定) に空きがない場合、KFCA16330-E メッセージを出力し、チャンネルを開始しません。

mqttcp (TCP 構成定義)

形式

```
mqttcp [-i イニシエーションキュー名]
        -p tcp
        -g "rcvbuf = 受信バッファグループ番号"
[-r "portno = 自システムのポート番号
     | servname = 自システムのサービス名 }"]
[-v " [itim = イニシエーションキュー監視タイマ値]
     [etim = 終了処理監視タイマ値]
     [rtim = チャネル確立完了監視タイマ値] "]
[-t " [disk = yes | no]
     [bufcnt = バッファ格納個数]
     [trccnt = トレースファイル個数]
     [bufsize = トレースバッファサイズ] "]
[-z time | event]
```

機能

MQT サーバの TCP/IP プロトコルについての構成を定義します。

オプション

● -i イニシエーションキュー名

～< 1～48 文字の MQ 文字列>

イニシエーションキュー名を指定します。この文字列の先頭または途中で空白を指定できません。また、ほかの mqttcp 定義コマンドで指定するイニシエーションキュー名と重複して指定できません。

イニシエーションキューの詳細については、「[2.3.4 MQT サーバのトリガ起動によるチャネル開始](#)」を参照してください。

● -p tcp

プロトコル種別を指定します。

tcp : TCP/IP プロトコル

● -g

(オペランド)

・rcvbuf=受信バッファグループ番号

～<符号なし整数>((1～512))

チャネル確立属性がレスポンドの場合に開始要求を受信する、受信バッファグループ番号を指定します。

mqttbuf 定義コマンドの-g groupno オペランドで指定したバッファグループ番号を指定してください。

mqtalcccha 定義コマンドの-y type オペランドで、receiver または requester を指定しない場合は、-g オプションを省略できます。また、開始要求を受信するための mqttbuf 定義コマンドも省略できます。

● -r

(オペランド)

- ・ portno=自システムのポート番号

～<符号なし整数>((1024～65535))

開始要求を受信する自システムのポート番号を指定します。この指定値は、相手システムからのネットワークリクエストを受けるために使用されます。

このオプションの servname オペランドを指定した場合、この指定値は省略できます。省略した場合、あらかじめサービス名ファイルにサービス名が登録されている必要があります。

portno オペランドと servname オペランドを両方とも指定した場合は、portno オペランドの指定値が優先されます。

- ・ servname=自システムのサービス名

～<1～32文字の識別子(ハイフン(-)を含む)>

自システムのサービス名を指定します。この指定値は、相手システムからのネットワークリクエストを受けるために使用されます。

このオプションの portno オペランドに指定がある場合、この指定値は省略できます。

サービス名は、サービス名ファイルに次に示す形式で登録します。

サービス名△ポート番号/TCP

portno オペランドと servname オペランドを両方とも指定した場合は、portno オペランドの指定値が優先されます。

■ 注意事項

- ・ mqtalcccha 定義コマンドの-y type オペランドで、receiver または requester を指定しない場合は、-r オプションは省略できます。
- ・ portno オペランドには MQTT サーバが動作するシステム内で使用しないポート番号 (OS が自動割り当てするポート番号の範囲外で、かつほかのプログラムが使用しない番号) を指定してください。
- ・ servname オペランドには MQTT サーバが動作するシステム内で使用しないサービス名 (サービス名に関連づけられたポート番号を考慮) を指定してください。
- ・ -r オプションで指定したポート番号またはサービス名に関連づけられたポート番号が、ほかの通信プログラムで使われている状態で MQTT サーバを起動すると、通信障害 (TCP/IP エラー、不正データ受信) になることがあります。なお、ポート番号には、OS が自動割り当てする番号があります。OS が自動割り当てするポート番号は、OS の種別や OS のバージョンなどによって異なります。詳細は、OS のマニュアルを参照してください。

● -v

(オペランド)

- ・ itim=イニシエーションキュー監視タイマ値

～<符号なし整数>((0~65535))《60》(単位:秒)

イニシエーションキュー監視タイマ値を指定します。

mqttcp 定義コマンドの-z オプションに time を指定した場合にこのオペランドに 0 を指定するとき、イニシエーションキュー監視は実行しません。mqttcp 定義コマンドの-z オプションに event を指定した場合でも、このオペランドの指定値によって、イニシエーションキューの監視を実行します。イベント方式では、通常この機能は必要ありません。これは何らかの予期しない障害によって、イベント連絡に失敗した場合にチャンネル起動がされなくなるのを防ぐ機能です。

- ・ etim=終了処理監視タイマ値

～<符号なし整数>((0, 10~65535))《360》(単位:秒)

終了処理監視タイマ値を指定します。0 を指定した場合、終了処理は監視されません。

■ 注意事項

etim オペランドを指定する場合は、次に示す条件を考慮してください。

- ・ すべての MQTT サーバの TCP 定義 (mqttcp 定義コマンド, mqttcps 定義コマンド, mqttcpcr 定義コマンド) で指定する etim オペランドの指定値の合計が、OpenTP1 のシステム環境定義の system_terminate_watch_time オペランドの指定値よりも小さくなるように 0 以外の値を指定してください。この条件を満たさない場合、dcstop コマンドを入力したあとに、OpenTP1 が強制停止することがあります。
- ・ 「チャンネル動作中」でバッチ処理を行っている場合に dcstop コマンドを入力するときは、バッチ処理時間よりも大きい値を etim オペランドに指定してください。バッチ処理時間よりも小さい値を etim オペランドに指定した場合、送達未確認状態や転送メッセージの消滅 (ファーストメッセージの場合) が発生することがあります。バッチサイズ、転送メッセージ長、およびバッチインターバルを考慮してください。
- ・ 「チャンネル動作中」でバッチ処理を行っていない場合に dcstop コマンドを入力するときは、ハートビート間隔よりも大きい値を etim オペランドに指定してください。ハートビート間隔よりも小さい値を etim オペランドに指定した場合は、相手システムが異常 (TCP/IP エラーやタイムアウト) を検出することがあります。
- ・ 「チャンネル動作中」のチャンネルが存在しない場合に dcstop コマンドを入力するときは、MQTT サーバの終了を監視する時間を etim オペランドに指定してください。

- ・ rtim=チャンネル確立完了監視タイマ値

～<符号なし整数>((0~65535))《60》(単位:秒)

チャンネル確立完了監視タイマ値を指定します。0 を指定した場合、チャンネル確立完了は監視されません。

● -t

(オペランド)

・ disk=yes | no

～ 《yes》

MQT トレースのディスク出力機能を使用するかどうかを指定します。

yes：ディスク出力機能を使用します（メモリおよびディスクに出力）。

no：ディスク出力機能を使用しません（メモリだけに出力）。

no を指定した場合、トレースファイルは作成されないでメモリ上の二つのバッファが交互に使用されます。また、このオプションの bufcnt および trccnt オペランドの指定値は無効になります。

yes を指定、または省略した場合、MQT トレースは、MQT サーバが提供する次に示すファイルに格納されます。

- ・ MQT サービス定義の mqt_trace_file_path オペランドを指定している場合
ファイル名：mqt_trace_file_path オペランドの指定値/mqttrcXXYY
- ・ MQT サービス定義の mqt_trace_file_path オペランドを指定していない場合
ファイル名：\$DCDIR/spool/mqttrcXXYY

(凡例)

XX：MQT 通信プロセス識別子

YY：トレーススワップファイル識別子

なお、MQT サーバで障害が発生し、保守員に調査を依頼する場合、このトレースが必要になります。そのため、常にトレースをディスクに出力して運用することをお勧めします。

・ bufcnt=バッファ格納個数

～<符号なし整数>((10~2147483647)) 《2500》

MQT トレースファイルのサイズを指定します。バッファを何面分格納できる容量にするかを個数単位で指定します。トレースファイルのサイズの算出式を次に示します。

トレースファイルサイズ = バッファ格納個数 ×

トレースバッファサイズ (-t bufsize オペランド指定値)

・ trccnt=トレースファイル個数

～<符号なし整数>((3~99)) 《3》

MQT トレースファイルの個数を指定します。トレースファイルは、必要に応じて作成されますが、この指定値を超えると、最も古いトレースファイルが上書きされます。

・ bufsize=トレースバッファサイズ

～<符号なし整数>((20480~128000000)) 《20480》 (単位：バイト)

MQT トレースを格納するバッファの大きさを指定します。4 の倍数でない値を指定した場合は、4 の倍数に切り上げられます。

注意事項

MQT トレースのディスク出力はトレースバッファサイズ単位に行われ、ディスク出力が完了するまではチャンネルの動作が一時的に停止します。そのため、バッファサイズを設定するときは次に示す点を配慮してください。

- 値が大きい場合：
ディスク出力回数は少なくなりますが、ディスク出力時間（チャンネルの停止時間）は長くなります。
- 値が小さい場合：
ディスク出力時間（チャンネルの停止時間）は短くなりますが、ディスク出力回数は多くなります。

● -z time | event

～ 《time》

MCA をトリガ起動する場合に使用するイニシエーションキューの監視方式を指定します。

- time：タイマ方式
-v itim オペランドで指定したタイマの間隔でトリガメッセージを取り出します。
- event：イベント方式
イニシエーションキューにトリガメッセージが登録されると同時にイベントによってトリガメッセージを取り出します。

mqttcps (クラスタセンダ TCP 構成定義)

形式

```
mqttcps -p tcp
[-v " [stim = チャンネル要求監視間隔]
      [etim = 終了処理監視タイマ値] "]
[-t " [disk = yes | no]
      [bufcnt = バッファ格納個数]
      [trccnt = トレースファイル個数]
      [bufsize = トレースバッファサイズ] "]
```

機能

クラスタセンダプロセスの TCP/IP プロトコルについての構成を定義します。

オプション

● -p tcp

プロトコル種別を指定します。

tcp : TCP/IP プロトコル

● -v

(オペランド)

- ・ stim=チャンネル要求監視間隔

~<符号なし整数> ((1~65535)) 《5》 (単位: 秒)

チャンネル要求監視間隔を指定します。チャンネル要求監視の詳細については、「[2.5.3 チャンネル要求監視](#)」を参照してください。

■ 注意事項

stim オペランドは、次の動作に影響を与えます。このため stim オペランドの値を大きくした場合、クラスタセンダチャンネルに対する処理が遅れることがあります。

- ・ 「チャンネル停止」でメッセージを登録した場合、stim オペランドの指定値の間、チャンネルが開始されないことがあります。
- ・ チャンネル確立再試行間隔が、mqtalcca 定義コマンドの -b bretryint および bretrylgint オペランドの指定値より大きくなる場合があります。
- ・ メッセージの送信経路再設定処理が、stim オペランドの指定値の間、完了しないことがあります。
- ・ 次に示す場合、リポジトリ管理サーバが、stim オペランドの指定値の間、一時的に動作できないことがあります。

- ・リポジトリ管理サーバを再開始した場合
- ・リポジトリ情報を再作成した場合

・ etim=終了処理監視タイマ値

～<符号なし整数>((0, 10~65535))《360》(単位：秒)

終了処理監視タイマ値を指定します。0を指定した場合、終了処理は監視されません。

注意事項

etim オペランドを指定する場合は、次に示す条件を考慮してください。

- ・すべてのMQTサーバのTCP定義(mqttcp定義コマンド, mqttcpes定義コマンド, mqttcpesr定義コマンド)で指定するetimオペランドの指定値の合計が、OpenTP1のシステム環境定義のsystem_terminate_watch_timeオペランドの指定値よりも小さくなるように0以外の値を指定してください。この条件を満たさない場合、dcstopコマンドを入力したあとに、OpenTP1が強制停止することがあります。
- ・「チャンネル動作中」でバッチ処理を行っている場合にdcstopコマンドを入力するときは、バッチ処理時間よりも大きい値をetimオペランドに指定してください。バッチ処理時間よりも小さい値をetimオペランドに指定した場合、送達未確認状態や転送メッセージの消滅(ファーストメッセージの場合)が発生することがあります。バッチサイズ、転送メッセージ長、およびバッチインターバルを考慮してください。
- ・「チャンネル動作中」でバッチ処理を行っていない場合にdcstopコマンドを入力するときは、ハートビート間隔よりも大きい値をetimオペランドに指定してください。ハートビート間隔よりも小さい値をetimオペランドに指定した場合は、相手システムが異常(TCP/IPエラーやタイムアウト)を検出することがあります。
- ・「チャンネル動作中」のチャンネルが存在しない場合にdcstopコマンドを入力するときは、MQTサーバの終了を監視する時間をetimオペランドに指定してください。

● -t

(オペランド)

・ disk=yes | no

～《yes》

MQTトレースのディスク出力機能を使用するかどうかを指定します。

yes：ディスク出力機能を使用します(メモリおよびディスクに出力)。

no：ディスク出力機能を使用しません(メモリだけに出力)。

noを指定した場合、トレースファイルは作成されないでメモリ上の二つのバッファが交互に使用されます。また、このオプションのbufcntおよびtrccntオペランドの指定値は無効になります。

yesを指定、または省略した場合、MQTトレースは、MQTサーバが提供する次に示すファイルに格納されます。

- MQT サービス定義の `mqt_trace_file_path` オペランドを指定している場合
ファイル名：`mqt_trace_file_path` オペランドの指定値/`mqttrcXXYY`
- MQT サービス定義の `mqt_trace_file_path` オペランドを指定していない場合
ファイル名：`$DCDIR/spool/mqttrcXXYY`

(凡例)

XX：MQT 通信プロセス識別子

YY：トレーススワップファイル識別子

なお、MQT サーバで障害が発生し、保守員に調査を依頼する場合、このトレースが必要になります。そのため、常にトレースをディスクに出力して運用することをお勧めします。

- `bufcnt`=バッファ格納個数

～<符号なし整数>((10~2147483647)) 《2500》

MQT トレースファイルのサイズを指定します。バッファを何面分格納できる容量にするかを個数単位で指定します。トレースファイルのサイズの算出式を次に示します。

トレースファイルサイズ = バッファ格納個数 ×

トレースバッファサイズ (-t bufsize オペランド指定値)

- `trcnt`=トレースファイル個数

～<符号なし整数>((3~99)) 《3》

MQT トレースファイルの個数を指定します。トレースファイルは、必要に応じて作成されますが、この指定値を超えると、最も古いトレースファイルが上書きされます。

- `bufsize`=トレースバッファサイズ

～<符号なし整数>((20480~128000000)) 《20480》 (単位：バイト)

MQT トレースを格納するバッファの大きさを指定します。4 の倍数でない値を指定した場合は、4 の倍数に切り上げられます。

注意事項

MQT トレースのディスク出力はトレースバッファサイズ単位に行われ、ディスク出力が完了するまではチャンネルの動作が一時的に停止します。そのため、バッファサイズを設定するときには次に示す点を配慮してください。

- 値が大きい場合：
ディスク出力回数は少なくなりますが、ディスク出力時間（チャンネルの停止時間）は長くなります。
- 値が小さい場合：
ディスク出力時間（チャンネルの停止時間）は短くなりますが、ディスク出力回数は多くなります。

mqttcpcr (クラスタレシーバ TCP 構成定義)

形式

```
mqttcpcr -p tcp
         -g "rcvbuf = 受信バッファグループ番号"
         -r "[portno = 自システムのポート番号]
            [servname = 自システムのサービス名]"
[-v "[stim = チャンネル要求監視間隔]
     [etim = 終了処理監視タイマ値]
     [rtim = 開始要求受信監視タイマ値]" ]
[-t "[disk = yes | no]
     [bufcnt = バッファ格納個数]
     [trcCnt = トレースファイル個数]
     [bufsize = トレースバッファサイズ]" ]
```

機能

クラスタレシーバプロセスの TCP/IP プロトコルについての構成を定義します。

オプション

● -p tcp

プロトコル種別を指定します。

tcp : TCP/IP プロトコル

● -g

(オペランド)

- ・ rcvbuf=受信バッファグループ番号

～<符号なし整数>((1~512))

チャンネル確立属性がレスポンドの場合に開始要求を受信する、受信バッファグループ番号を指定します。

mqttbuf 定義コマンドの-g groupno オペランドで指定したバッファグループ番号を指定してください。

● -r

(オペランド)

- ・ portno=自システムのポート番号

～<符号なし整数>((1024~65535))

開始要求を受信する自システムのポート番号を指定します。この指定値は、相手システムからのネットワークリクエストを受けるために使用されます。

このオプションの servname オペランドを指定した場合、この指定値は省略できます。省略した場合、あらかじめサービス名ファイルにサービス名が登録されている必要があります。

portno オペランドと servname オペランドを両方とも指定した場合は、portno オペランドの指定値が優先されます。

- ・ servname=自システムのサービス名

～< 1～32 文字の識別子 (ハイフン (-) を含む) >

自システムのサービス名を指定します。この指定値は、相手システムからのネットワークリクエストを受け取るために使用されます。

このオプションの portno オペランドに指定がある場合、この指定値は省略できます。

サービス名は、サービス名ファイルに次に示す形式で登録します。

サービス名△ポート番号/TCP

portno オペランドと servname オペランドを両方とも指定した場合は、portno オペランドの指定値が優先されます。

注意事項

- ・ portno オペランドには MQTT サーバが動作するシステム内で使用しないポート番号 (OS が自動割り当てするポート番号の範囲外で、かつほかのプログラムが使用しない番号) を指定してください。
- ・ servname オペランドには MQTT サーバが動作するシステム内で使用しないサービス名 (サービス名に関連づけられたポート番号を考慮) を指定してください。
- ・ -r オプションで指定したポート番号またはサービス名に関連づけられたポート番号が、ほかの通信プログラムで使われている状態で MQTT サーバを起動すると、通信障害 (TCP/IP エラー, 不正データ受信) になることがあります。なお、ポート番号には、OS が自動割り当てする番号があります。OS が自動割り当てするポート番号は、OS の種別や OS のバージョンなどによって異なります。詳細は、OS のマニュアルを参照してください。

● -v

(オペランド)

- ・ stim=チャンネル要求監視間隔

～< 符号なし整数 > ((1～65535)) 《5》 (単位: 秒)

チャンネル要求監視間隔を指定します。チャンネル要求監視の詳細については、「[2.5.3 チャンネル要求監視](#)」を参照してください。

注意事項

stim オペランドは、次の動作に影響を与えます。このため stim オペランドの値を大きくした場合、クラスタレシーバチャンネルに対する処理が遅れることがあります。

- ・ 次に示す場合、リポジトリ管理サーバが、stim オペランドの指定値の間、一時的に動作できないことがあります。

- ・リポジトリ管理サーバを再起動した場合
- ・リポジトリ情報を再作成した場合

・etim=終了処理監視タイマ値

～<符号なし整数>((0, 10~65535))《360》(単位:秒)

終了処理監視タイマ値を指定します。0を指定した場合、終了処理は監視されません。

注意事項

etim オペランドを指定する場合は、次に示す条件を考慮してください。

- ・すべてのMQTサーバのTCP定義(mqttcp定義コマンド, mqttcpes定義コマンド, mqttcpesr定義コマンド)で指定するetimオペランドの指定値の合計が、OpenTP1のシステム環境定義のsystem_terminate_watch_timeオペランドの指定値よりも小さくなるように0以外の値を指定してください。この条件を満たさない場合、dcstopコマンドを入力したあとに、OpenTP1が強制停止することがあります。
- ・「チャンネル動作中」でバッチ処理を行っている場合にdcstopコマンドを入力するときは、バッチ処理時間よりも大きい値をetimオペランドに指定してください。バッチ処理時間よりも小さい値をetimオペランドに指定した場合、送達未確認状態や転送メッセージの消滅(ファーストメッセージの場合)が発生することがあります。バッチサイズ、転送メッセージ長、およびバッチインターバルを考慮してください。
- ・「チャンネル動作中」でバッチ処理を行っていない場合にdcstopコマンドを入力するときは、ハートビート間隔よりも大きい値をetimオペランドに指定してください。ハートビート間隔よりも小さい値をetimオペランドに指定した場合は、相手システムが異常(TCP/IPエラーやタイムアウト)を検出することがあります。
- ・「チャンネル動作中」のチャンネルが存在しない場合にdcstopコマンドを入力するときは、MQTサーバの終了を監視する時間をetimオペランドに指定してください。

・rtim=開始要求受信監視タイマ値

～<符号なし整数>((0~65535))《60》(単位:秒)

開始要求受信監視タイマ値を指定します。このオペランドに0を指定した場合、開始要求受信監視はしません。

● -t

(オペランド)

・disk=yes | no

～《yes》

MQTトレースのディスク出力機能を使用するかどうかを指定します。

yes: ディスク出力機能を使用します(メモリおよびディスクに出力)。

no: ディスク出力機能を使用しません(メモリだけに出力)。

no を指定した場合、トレースファイルは作成されないでメモリ上の二つのバッファが交互に使用されます。また、このオプションの bufcnt および trccnt オペランドの指定値は無効になります。

yes を指定、または省略した場合、MQT トレースは、MQT サーバが提供する次に示すファイルに格納されます。

- MQT サービス定義の mqt_trace_file_path オペランドを指定している場合
ファイル名：mqt_trace_file_path オペランドの指定値/mqttrcXXYY
- MQT サービス定義の mqt_trace_file_path オペランドを指定していない場合
ファイル名：\$DCDIR/spool/mqttrcXXYY

(凡例)

XX：MQT 通信プロセス識別子

YY：トレーススワップファイル識別子

なお、MQT サーバで障害が発生し、保守員に調査を依頼する場合、このトレースが必要になります。そのため、常にトレースをディスクに出力して運用することをお勧めします。

• bufcnt=バッファ格納個数

～<符号なし整数>((10~2147483647)) 《2500》

MQT トレースファイルのサイズを指定します。バッファを何面分格納できる容量にするかを個数単位で指定します。トレースファイルのサイズの算出式を次に示します。

トレースファイルサイズ = バッファ格納個数 ×

トレースバッファサイズ (-t bufsize オペランド指定値)

• trccnt=トレースファイル個数

～<符号なし整数>((3~99)) 《3》

MQT トレースファイルの個数を指定します。トレースファイルは、必要に応じて作成されますが、この指定値を超えると、最も古いトレースファイルが上書きされます。

• bufsize=トレースバッファサイズ

～<符号なし整数>((20480~128000000)) 《20480》(単位：バイト)

MQT トレースを格納するバッファの大きさを指定します。4 の倍数でない値を指定した場合は、4 の倍数に切り上げられます。

注意事項

MQT トレースのディスク出力はトレースバッファサイズ単位に行われ、ディスク出力が完了するまではチャンネルの動作が一時的に停止します。そのため、バッファサイズを設定するときは次に示す点を配慮してください。

- 値が大きい場合：
ディスク出力回数は少なくなりますが、ディスク出力時間（チャンネルの停止時間）は長くなります。
- 値が小さい場合：

ディスク出力時間（チャンネルの停止時間）は短くなりますが、ディスク出力回数は多くなります。

mqtalccha (チャンネル定義)

形式

```
mqtalccha -c チャンネル名
-y "type = sender | server | receiver | requester |
      clussdr | clusrchr | defsnd | defsrv |
      defrcv | defreq | defcsn"
[-j バッチサイズ]
[-q 転送キュー名]
[-m " [maxmsg = 最大メッセージ長]
      [maxseg = 最大セグメントサイズ] "]
[-w メッセージシーケンス番号の最大値]
-g " [sndbuf = 送信バッファグループ番号]
      [rcvbuf = 受信バッファグループ番号]
      [buftype = seg | msg] "
[-e " [medbuf = メッセージ編集出口編集用バッファグループ番号]
      [medcnt = メッセージ編集出口編集用バッファ数] "]
[-i auto | manual | auto2]
[-b " [bretry = yes | no]
      [bretrycnt = 短期確立再試行の回数]
      [bretryint = 短期確立再試行の間隔]
      [bretrylg = yes | no]
      [bretrylgcnt = 長期確立再試行の回数]
      [bretrylgint = 長期確立再試行の間隔]
      [bretrymcp = yes | no] "]
[-r " [{ ipaddr = 自システムのIPアドレス
        | hostname = 自システムのホスト名 }]
      [{ portno = 自システムのポート番号
        | servname = 自システムのサービス名 }]
      [portnum = 自システムのポート番号使用数] "]
[-o " { oipaddr = 相手システムのホストのIPアドレス
        | ohostname = 相手システムのホスト名 }
      { oportno = 相手システムのホストのポート番号
        | oservname = 相手システムのホストのサービス名 }"]
[-v " [tim1 = チャンネル確立応答受信監視タイマ値]
      [tim2 = 確認メッセージ受信監視タイマ値]
      [tim3 = 継続セグメント受信監視タイマ値]
      [tim4 = 転送キュー監視間隔]
      [tim4cnt = 転送キュー監視回数]
      [dtimfct = yes | no]
      [dtim = 切断時間間隔]
      [vretry = yes | no]
      [htim = ハートビート間隔]
      [btim = バッチ終了待ちタイマ値]
      [bmtim = バッチ終了待ちタイマ値 (ミリ秒)]
      [mtim = 継続メッセージ受信監視タイマ値] "]
[-x " [sndexdata = '送信出口ユーザデータ']
      [rcvexdata = '受信出口ユーザデータ']
      [metexdata = 'メッセージ編集出口ユーザデータ']
      [scyexdata = 'セキュリティ出口ユーザデータ']
      [sendexit = '送信出口名']
      [rcvexit = '受信出口名']
      [msgexit = 'メッセージ出口名']
      [scyexit = 'セキュリティ出口名'] "]
[-t " [tcpsndbuf = TCP/IP出力用バッファサイズ]
```

```

        [tcprcvbuf = TCP/IP入力用バッファサイズ]
        [tretrycnt = TCP/IP確立確認の回数]
        [tretryint = TCP/IP確立確認の間隔] ”]
[-d ” [cnvccsid = 変換後CCSID] ”]
[-s ” [npmspeed = normal | fast] ”]
[-z time | event]
[-a ” [cluster = クラスタ名] ”]
[-p ” [netprty = ネットワーク接続優先順位] ”]
[-f ”adoptchk = off | name | qmgr | ipaddr | all”]
[-h ’注釈’]
[-mc ” [mcauser = ’MCAユーザID’]
        [mcatype = process | thread] ”]
[-jn ” [sndjnl = flush | noflush] ”]

```

機能

MQT サーバのチャンネルについての情報を定義します。

オプション

● -c チャンネル名

～< 1～20 文字の MQ 文字列>

チャンネル名を指定します。

ほかの mqtalcccha 定義コマンドの -c オプションで指定するチャンネル名と重複して指定できません。また、相手システムの対応するチャンネル定義に指定されたチャンネル名と同じ名称を指定してください。ほかの MQT プロセスの mqtalcccha 定義コマンドの -c オプションで指定したチャンネル名と重複した場合には、動作を保証しません。

● -y

(オペランド)

・ type=sender | server | receiver | requester | clussdr | clusrcvr | defsnd | defsrv | defrcv
| defreq | defcsn

チャンネルタイプを指定します。

sender：センダ

server：サーバ

receiver：レシーバ

requester：リクエスタ

clusdr：クラスタセンダ

clusrcvr：クラスタレシーバ

defsnd：デフォルトセンダ

defsrv：デフォルトサーバ

defrcv：デフォルトレシーバ

defreq：デフォルトリクエスト

defcsn：デフォルトクラスタセンダ

TCP 定義で mqttcp 定義コマンドを指定している場合は、sender, server, receiver, requester, defsnd, defsrv, defrcv, および defreq を指定できます。TCP 定義で mqttcpcs 定義コマンドを指定している場合は、clusdr および defcsn を指定できます。TCP 定義で mqttcpcr 定義コマンドを指定している場合は、clusrcvr を指定できます。

● -j バッチサイズ

～<符号なし整数>((1～9999))

チャンネルタイプが sender, server, receiver, または requester の場合《1》

チャンネルタイプが clusdr または clusrcvr の場合《50》

バッチサイズを指定します。バッチサイズとして扱うユーザアプリケーションメッセージの個数を指定します。チャンネル確立時の相手システムの MCA とのネゴシエーションによって、この指定値より小さくなる場合があります。

■ 注意事項

- ネゴシエーション時の自動定義クラスタセンダチャンネルのバッチサイズは mqamqtnam 定義コマンド-b オプション指定値および事前定義クラスタセンダチャンネルに指定したバッチサイズに依存します。詳細については、この章の「[チャンネルのネゴシエーション](#)」を参照してください。
- 接続先のクラスタレシーバチャンネルのバッチサイズが 50 よりも大きく、かつ自システムの自動定義クラスタセンダチャンネルのバッチサイズを 50 よりも大きな値で動作させる場合は、クラスタセンダチャンネルのデフォルトチャンネル定義でバッチサイズを指定してください。デフォルトチャンネル定義を使用しない場合は、自動定義クラスタセンダチャンネルのバッチサイズの上限は 50 です。

● -q 転送キュー名

～<1～48 文字の MQ 文字列>

転送キュー名を指定します。mqaremque 定義コマンドの-x オプション（転送キュー名）で指定した転送キューを指定してください。この名称の先頭または途中で空白を指定できません。また、ほかの mqtalcccha 定義コマンドで指定する転送キュー名と重複して指定できません。

-y type オペランドが sender または server の場合、この指定値は省略できません。また、ほかの mqtalcccha 定義コマンドの-q オプション（転送キュー名）で指定した名称と重複して指定できません。

-y type オペランドが clusdr または clusrcvr の場合、このオプションは指定できません。転送キュー名として SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して動作します。

● -m

(オペランド)

・ maxmsg=最大メッセージ長

~<符号なし整数>((1024~127996000)) (単位: バイト)

最大メッセージ長を指定します。

これは、ユーザアプリケーションメッセージの最大値です。チャンネル確立時の相手システムの MCA とのネゴシエーションによって、この指定値より小さくなることがあります。

省略した場合、次に示す値が設定されます。

- ・ -y type=sender, server, または clussdr の場合
-g sndbuf オペランドで指定したバッファグループのバッファサイズ (ただし, -g buftype オペランドが seg の場合は, 127996000)
- ・ -y type=receiver, requester, または clusrvr の場合
-g rcvbuf オペランドで指定したバッファグループのバッファサイズ (ただし, -g buftype オペランドが seg の場合は, 127996000)

・ maxseg=最大セグメントサイズ

~<符号なし整数>((1024~1073741824)) (単位: バイト)

最大セグメントサイズを指定します。

mqttbuf 定義コマンドの -g length オペランドで指定したバッファ長以下の値を指定してください。

省略した場合、次に示す値が設定されます。

- ・ -y type=sender, または server の場合
-g sndbuf オペランドで指定したバッファグループのバッファサイズ
- ・ -y type=clussdr の場合
8192 (ただし, -g sndbuf オペランドで指定したバッファグループのバッファサイズが 8192 よりも小さい場合は, バッファグループのバッファサイズが設定されます)
- ・ -y type=receiver, または requester の場合
-g rcvbuf オペランドで指定したバッファグループのバッファサイズ
- ・ -y type=clusrvr の場合
8192 (ただし, -g rcvbuf オペランドで指定したバッファグループのバッファサイズが 8192 よりも小さい場合は, バッファグループのバッファサイズが設定されます)
- ・ -g buftype オペランドで, msg を指定した場合は, このオペランドに指定した値のローカルメモリを必要とします。

● -w メッセージシーケンス番号の最大値

~<符号なし整数>((100~999999999)) 《999999999》

メッセージシーケンス番号の最大値を指定します。

メッセージシーケンス番号は、この指定値を超えると、1に戻ります。

メッセージシーケンス番号の最大値は、相手システムと同じ値を指定してください。

● -g

(オペランド)

- sndbuf=送信バッファグループ番号

~<符号なし整数>((1~512))

メッセージ送信用のバッファグループ番号を指定します。

mqttbuf 定義コマンドの-g groupno オペランドで指定したバッファグループ番号を指定します。

- rcvbuf=受信バッファグループ番号

~<符号なし整数>((1~512))

メッセージ受信用のバッファグループ番号を指定します。

mqttbuf 定義コマンドの-g groupno オペランドで指定したバッファグループ番号を指定します。

-y type オペランドで clusrcvr を指定した場合、mqttbuf 定義コマンドに clussdr と接続するチャンネル数分のバッファ数を用意してください。

- buftype=msg | seg

~ 《msg》

バッファ方式を指定します。

seg: セグメント方式

msg: メッセージ方式

セグメント方式を指定した場合、転送セグメント単位でメッセージが格納され、取り出し/送信、受信/登録が実行されます。この方式では、MCA で使用するメモリ量を抑えられるのが利点です。しかし、マルチキャスト機能、およびメッセージ編集出口 UOC によるファイル転送機能を使用できません。また、sndbuf, rcvbuf オペランドに指定するバッファグループのバッファ長は、-m maxseg オペランドに指定する値以上にする必要があります。

メッセージ方式を指定した場合、メッセージ全体で取り出し/送信、受信/登録が実行されます。この方式を選んだ場合、マルチキャスト機能、およびメッセージ編集出口 UOC によるファイル転送機能を使用できます。しかし、MCA で使用するメモリ量がセグメント方式より大きくなります。また、sndbuf, rcvbuf オペランドに指定するバッファグループのバッファ長は、-m maxmsg オペランドに指定する値以上にする必要があります。

● -e

(オペランド)

- medbuf=メッセージ編集出口編集用バッファグループ番号

~<符号なし整数>((1~512))

メッセージ編集出口 UOC で使用する編集用バッファグループ番号を指定します。

このオペランドには、mqttbuf 定義コマンドの-g groupno オペランドに指定したバッファグループ番号を指定してください。

- medcnt=メッセージ編集出口編集用バッファ数

～<符号なし整数>((1~65535))

メッセージ編集出口 UOC に引き渡すバッファ数を指定します。

このオペランドには、mqttbuf 定義コマンドの-g count オペランドに指定したバッファ数以下の値を指定してください。

また、複数のチャンネルが同時に同一バッファグループを使用する場合、mqttbuf 定義コマンドで、同時に動作するチャンネル数分のバッファ数を用意する必要があります。

● -i auto | manual | auto2

チャンネルタイプが sender, server, receiver, または requester の場合 《manual》

チャンネルタイプが clussdr, または clusrvr の場合 《auto》

MQT サーバの開始時, 再開時, またはクラスタ環境で自動的にチャンネルを作成する場合, チャンネルを自動的に確立するかどうかを指定します。

- auto

チャンネル状態は「チャンネル停止」から始まります。トリガ起動またはネットワークリクエストによって、チャンネルを自動的に開始します。

- manual

チャンネル状態は「チャンネル使用不可」から始まります。トリガ起動またはネットワークリクエストによるチャンネルの自動開始は行いません。チャンネルを開始するには mqstacha コマンドを入力する必要があります。

システムの運用上、チャンネルを自動的に開始してはいけない場合、必ず manual を指定してください。

- auto2

MQT サーバのサービス開始直後に、自動的にチャンネル確立処理が行われます。この値は、センダ、サーバ、リクエストおよびクラスタセンダに指定できます。

■ 注意事項

自動定義クラスタセンダチャンネルの場合は、次に示す値が設定されます。

- デフォルトチャンネル定義を使用しない場合
すべての自動定義クラスタセンダチャンネルに auto が設定されます。
- デフォルトチャンネル定義を使用する場合
すべての自動定義クラスタセンダチャンネルに、事前定義クラスタセンダチャンネルのデフォルトチャンネル定義の-i オプションに指定した値が設定されます。

● -b

(オペランド)

・ bretry=yes | no

～ 《yes》

チャンネルに通信障害が発生した場合に短期確立再試行を実行するかどうかを指定します。

yes：短期確立再試行を実行します。

no：短期確立再試行を実行しません。

ただし、-o オプションを省略した場合、このオペランドの指定は無効になります。

・ bretrycnt=短期確立再試行の回数

～<符号なし整数> ((1～999999999)) 《10》 (単位：回)

短期確立再試行を実行する回数を指定します。

なお、このオプションの bretry オペランドに no を指定した場合、このオペランドの指定は無効になります。

・ bretryint=短期確立再試行の間隔

～<符号なし整数> ((0～999999999)) 《60》 (単位：秒)

短期確立再試行の時間間隔を指定します。0 を指定した場合、障害が発生したつどに短期確立再試行を実行します。

なお、このオプションの bretry オペランドに no を指定した場合、このオペランドの指定は無効になります。

・ bretrylg=yes | no

～ 《yes》

チャンネルに通信障害が発生した場合に長期確立再試行を実行するかどうかを指定します。

yes：長期確立再試行を実行します。

no：長期確立再試行を実行しません。

ただし、-o オプションを省略した場合、このオペランドの指定は無効になります。

・ bretrylgcnt=長期確立再試行の回数

～<符号なし整数> ((1～999999999)) 《999999999》 (単位：回)

長期確立再試行を実行する回数を指定します。

なお、このオプションの bretrylg オペランドに no を指定した場合、このオペランドの指定は無効になります。

・ bretrylgint=長期確立再試行の間隔

～<符号なし整数> ((0～999999999)) 《1200》 (単位：秒)

長期確立再試行の時間間隔を指定します。0 を指定した場合、障害が発生したつどに長期確立再試行を実行します。

なお、このオプションの bretrylg オペランドに no を指定した場合、このオペランドの指定は無効になります。

チャンネルトリガを起動する場合、通信相手システムの起動状態によってイニシエーションキューが満杯になることがあります。そこで余分なトリガメッセージを発生させないために、次に示す計算式に従って時間間隔を指定してください。

チャンネル確立再試行間隔
(mqtalcccha定義コマンドの-b bretryintオペランド指定値)
≧トリガ起動チャンネル数 × 10

• bretrymcp=yes | no

～《yes》

MQ プロトコルで検出したエラーに対して、確立再試行を実行するかどうかを指定します。この指定は、-y オプションに sender, server, requester, または clussdr を指定し、-b オプションの bretry オペランドまたは bretrylg オペランドに yes を指定した場合に有効です。指定値が yes の場合、および no の場合の詳細について次に説明します。

• yes

MQ プロトコルで検出した障害に対して、確立再試行を実行します。

• no

MQ プロトコルで検出した障害に対して、確立再試行を実行しません。

注意事項

-b オプションに bretrymcp = yes を指定し、次に示す条件が重なった場合、相手チャンネルが接続を切断して再接続します（自チャンネルは「リトライ中」になります）。このため、チャンネルの状態の不一致が発生してチャンネルを確立できません。この現象を回避するには、-f オプションの adoptchk オペランドに off 以外を指定し、相手チャンネルからの再接続要求を受け入れるようにしてください。

- 条件 1：-y オプションの type オペランドに requester が指定された
- 条件 2：接続相手チャンネルが sender である
- 条件 3：自チャンネル (requester) がコーラになる

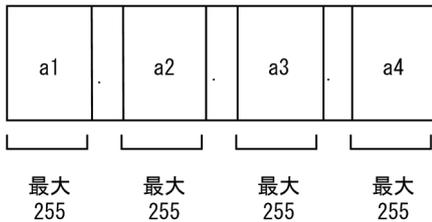
● -r

(オペランド)

• ipaddr=自システムの IP アドレス

～ (nnn.nnn.nnn.nnn) <符号なし整数> ((0~255))

複数の IP アドレスを持つシステムで、IP アドレスを特定する場合に自システムの IP アドレスを指定します。指定形式は、nnn.nnn.nnn.nnn です。nnn の値は、0~255 まで指定できます。nnn の値として先頭に 0 を指定した場合は、8 進数値として扱われます。IP アドレスの形式を次に示します。



このオペランドを省略して、かつ IP アドレスを特定する場合は、hostname オペランドを指定してください。また、ホスト名ファイル、またはネームサーバにホスト名を登録してください。

- hostname=自システムのホスト名
～< 1～255 バイトの文字列 >

複数の IP アドレスを持つシステムで、IP アドレスを特定する場合に自システムのホスト名を指定します。自ホストのホスト名は、ホスト名ファイル、またはネームサーバなどに登録してください。

ipaddr オペランドを指定する場合、または IP アドレスを特定する必要がない場合は省略できます。

注意事項

ipaddr オペランドと hostname オペランドの両方を指定する場合は、ipaddr オペランドの指定値が優先されます。

次に示す場合、どちらかのオペランドを指定してください。

- -y type オペランドに requester, sender, server, または clussdr を指定し、かつ相手ホストに対して IP アドレスを固定に持つ場合
- -y type オペランドが clusrvr の場合

ipaddr オペランドと hostname オペランドの指定の有無には関係なく、ipaddr オペランドまたは hostname オペランドのどちらかを指定できます。

- portno=自システムのポート番号
～< 符号なし整数 > ((1024～65535))

自システムのポート番号を指定します。

servname オペランドを指定した場合、この指定値は省略できます。

- servname=自システムのサービス名
～< 1～32 文字の識別子 (ハイフン (-) 含む) >

自システムのサービス名を指定します。このサービス名は、サービス名ファイルで登録されている必要があります。

サービス名は、次に示す形式で指定します。

サービス名△ポート番号/TCP

portno オペランドを指定した場合、この指定値は省略できます。

・ portnum=自システムのポート番号使用数

～<符号なし整数>((1~64512))《1》

自システムのポート番号使用数を指定します。

自ポートの割り当てが行われる範囲を指定する場合に、ポート番号の使用数を指定します。

注意事項

- ・ portno オペランドと servname オペランドの両方を指定する場合は、portno オペランドの指定値が優先されます。
- ・ -y type オペランドに requester, sender, server, または clussdr を指定し、かつ相手ホストに対して IP アドレスを固定に持つ場合、portno オペランドまたは servname オペランドを指定してください。
- ・ -y type オペランドに clusrcvr を指定する場合、mqttcpcr 定義コマンドの-r オプションに指定したサービス名またはポート番号を使用するため指定する必要はありません。
- ・ portno オペランドと servname オペランドの指定の有無には関係なく、ipaddr オペランドまたは hostname オペランドのどちらかを指定できます。
- ・ portno オペランドには MQTT サーバが動作するシステム内で使用しないポート番号（OS が自動割り当てするポート番号の範囲外で、かつほかのプログラムが使用しない番号）を指定してください。

servname オペランドには MQTT サーバが動作するシステム内で使用しないサービス名（サービス名に関連づけられたポート番号を考慮）を指定してください。

-r オプションで指定したポート番号またはサービス名に関連づけられたポート番号が、ほかの通信プログラムで使われている状態で MQTT サーバを起動すると、通信障害（TCP/IP エラー、不正データ受信）になることがあります。なお、ポート番号には、OS が自動割り当てする番号があります。OS が自動割り当てするポート番号は、OS の種別や OS のバージョンなどによって異なります。詳細は、OS のマニュアルを参照してください。

- ・ 割り当てを試行するポート番号の範囲を指定する場合は、servname オペランドまたは portno オペランドのどちらか一方とともに portnum オペランドを指定してください。servname オペランド指定値から変換したポート番号または portno オペランド指定値のポート番号から自ポートの割り当てが開始され、成功するまで最大 portnum オペランド指定値分の処理が試行されます。割り当てを試行するポート番号が 65535 を超える場合は、65535 で試行は中止されます。
- ・ コーラ側チャネルの自システムのポート番号を portno オペランドに指定（または servname オペランドを指定）すると、チャネル解放時に一定時間、TCP/IP の資源が解放されないことがあります（TCP/IP の仕様）。このためチャネル解放直後にチャネルを確立すると、TCP/IP とのインタフェースで障害を検出します。この場合は、チャネルの確立再試行によってチャネルを再確立するか、portnum オペランドを設定してください。

● -0

(オペランド)

このオペランドの指定値は、チャンネル属性の変更コマンド（mqaltcha コマンドの-l temp オペランド指定）を使用すると TP1/Message Queue の終了まで変更属性を保持します。チャンネル属性の変更コマンド（mqaltcha コマンドの-l perm オペランド指定）を使用するとチャンネル属性の変更コマンド（mqaltcha コマンドの-l temp オペランド指定）を実行するまで、変更属性を保持します。なお、チャンネル管理情報ファイルを使用しない場合は、TP1/Message Queue の終了まで変更属性を保持します。

注意事項

事前定義クラスタセンダチャンネルの場合、参加するクラスタのフルリポジトリキューマネージャに接続するように-o オプションを指定してください。参加するクラスタ内に複数のフルリポジトリキューマネージャがある場合は、最低一つの事前定義クラスタセンダチャンネル定義があれば十分です（必要に応じて自動定義クラスタセンダチャンネルをシステムが自動的に作成します）。どのフルリポジトリキューマネージャに接続してもかまいません。

- oipaddr=相手システムのホストの IP アドレス

～ (nnn.nnn.nnn.nnn) <符号なし整数> ((0~255))

相手システムの IP アドレスを指定します。nnn の値は、0~255 まで指定できます。nnn の値として先頭に 0 を指定した場合は、8 進数値として扱われます。

ホスト名ファイルに相手システムのホスト名を登録し、ohostname オペランドを指定した場合、このオペランドは省略できます。

- ohostname=相手システムのホスト名

～ <1~255 バイトの文字列>

相手システムのホスト名を指定します。このホスト名は、ホスト名ファイル、ネームサーバなどに登録されている必要があります。

oipaddr オペランドを指定する場合、このオペランドは省略できます。

注意事項

oipaddr オペランドと ohostname オペランドの両方を指定する場合は、oipaddr オペランドの指定値が優先されます。次に示す場合、どちらかのオペランドを指定してください。

- -y type オペランドが sender または clussdr の場合
- -y type オペランドが server または requester の場合で、かつ、チャンネル確立属性がコーラの場合

-y type オペランドに clussdr を指定する場合、このオプションで指定する接続先のキューマネージャはリポジトリキューマネージャである必要があります。

- oportno=相手システムのホストのポート番号

～ <符号なし整数> ((1024~65535))

相手システムのホストのポート番号を指定します。

oservname オペランドを指定した場合、この指定値は省略できます。

- ・ oservname=相手システムのホストのサービス名

～< 1～32文字の識別子 (ハイフン (-) 含む) >

相手システムのホストのサービス名を指定します。このサービス名は、サービス名ファイルで登録されている必要があります。

oportno オペランドを指定した場合、この指定値は省略できます。

注意事項

oportno オペランドと oservname オペランドの両方を指定する場合は、oportno オペランドの指定値が優先されます。次に示す場合、どちらかのオペランドを指定してください。

- ・ -y type オペランドが sender または clussdr の場合
- ・ -y type オペランドが server または requester の場合で、かつ、チャンネル確立属性がコーラの場合

-y type オペランドに clussdr を指定する場合、このオプションで指定する接続先のキューマネージャはリポジトリキューマネージャにする必要があります。

● -v

(オペランド)

- ・ tim1=チャンネル確立応答受信監視タイマ値

～<符号なし整数>((0～8191))《30》(単位：秒)

開始要求応答受信監視タイマ値を指定します。0を指定した場合、監視されません。

- ・ tim2=確認メッセージ受信監視タイマ値

～<符号なし整数>((0～8191))

チャンネルタイプが sender または server の場合《10》

チャンネルタイプが clussdr の場合《30》(単位：秒)

確認メッセージ受信監視タイマ値を指定します。0を指定した場合、監視されません。

- ・ tim3=継続セグメント受信監視タイマ値

～<符号なし整数>((0～8191))

チャンネルタイプが receiver または requester の場合《10》

チャンネルタイプが clusrchr の場合《30》(単位：秒)

継続セグメント受信監視タイマ値を指定します。0を指定した場合、監視されません。

- ・ tim4=転送キュー監視間隔

～<符号なし整数>((0～8191))《60》(単位：秒)

転送キュー監視間隔を指定します。0を指定した場合、転送キューは監視されません。

- ・ tim4cnt=転送キュー監視回数

～<符号なし整数>((0～2147483647))《1》(単位：回)

転送キューを監視する回数を指定します。

このオペランドに 0 を指定した場合、チャンネルを確立したまま、転送キューは無限に監視されます。

tim4 オペランドで 0 が指定された場合、転送キューは監視されないですぐに終了します。

• dtimefct=yes | no

～《yes》

切断時間間隔 (dtim オペランド指定値) の間、イベント方式による転送キュー監視をするかどうかを指定します。このオペランドを省略した場合は転送キュー監視をします。no を指定した場合、転送キューにメッセージがないことを検知するとチャンネルを終了します。この指定値は送信側 MCA がイベント方式 (mqtalcccha 定義コマンドの-z オプションに event を指定) による転送キュー監視をする場合に有効です。

-y type オペランドが clussdr または clusrvdr の場合、このオペランドは指定できません。切断時間間隔の間、イベント方式による転送キュー監視を使用して動作します。

• dtim=切断時間間隔

～<符号なし整数> ((0~999999))

チャンネルタイプが sender, または server の場合《60》

チャンネルタイプが clussdr, または clusrvdr の場合《6000》(単位: 秒)

切断時間間隔を指定します。バッチで送達確認を送信したあと、この時間の間、転送キューへのメッセージ登録を待ちます。タイムアウトするとチャンネルを終了します。このオペランドに 0 を指定した場合、転送キューにメッセージが登録されるまで無限に待ちます。この指定値は送信側 MCA がイベント方式 (mqtalcccha 定義コマンドの-z オプションに event を指定) による転送キュー監視をする場合に、切断時間監視の有無指定 (dtimefct オペランド) が yes のとき有効です。

• vretry=yes | no

～《yes》

コーラ側 MCA で時間監視障害が発生した場合に、チャンネル確立の再試行を実行するかどうかを指定します。

yes: 時間監視障害時にチャンネル確立の再試行を実行します。

no: 時間監視障害時にチャンネル確立の再試行を実行しません。

チャンネル確立の再試行は、-b オプションの指定値に従います。-b オプションの bretry オペランドまたは bretrylg オペランドに yes を指定した場合に有効です。

時間監視障害については、「[2.3.7 チャンネル確立再試行](#)」を参照してください。

• htim=ハートビート間隔

～<符号なし整数> ((0~999999)) 《300》(単位: 秒)

送信側 MCA の切断時間間隔中のハートビートメッセージによる確認動作を実行する場合のハートビート間隔を指定します。

このオペランドに 0 を指定した場合、送信側チャンネルの切断時間間隔 (MQT では、tim4×tim4cnt または dtim) 中のハートビートメッセージによる確認動作は実行されません。

• btim=バッチ終了待ちタイマ値

～<符号なし整数> ((0~999999)) 《0》(単位: 秒)

バッチ終了待ちタイマ (バッチインターバル) の秒以上の値を指定します。

btim オペランド指定値と bmtim オペランド指定値の和の時間がバッチ終了待ちタイマとなります。

この指定値は、送信側 MCA がイベント方式 (-z オプションに event を指定) による転送キュー監視をする場合に有効です。

• bmtim=バッチ終了待ちタイマ値 (ミリ秒)

～<符号なし整数> ((0~999)) 《0》 (単位: ミリ秒)

バッチ終了待ちタイマ (バッチインターバル) のミリ秒の値を指定します。btim オペランド指定値と bmtim オペランド指定値の和の時間がバッチ終了待ちタイマとなります。

この指定値は、送信側 MCA がイベント方式 (-z オプションに event を指定) による転送キュー監視をする場合に有効です。

送信中のバッチがバッチサイズに満たない場合、btim オペランド指定値と bmtim オペランド指定値の和の時間まで転送キューへのメッセージ登録を待ちます。

btim オペランドと bmtim オペランドに同時に 0 を指定した場合、または省略した場合、バッチサイズに満たないバッチを即座に終了します。

15ms 未満の値を指定する場合は、一つの MQT サーバに一つのチャンネルを定義することをお勧めします。複数のチャンネルを定義したとき、定義した値で動作できないことがあります。

• mtim=継続メッセージ受信監視タイマ値

～<符号なし整数> ((0~999999)) 《tim3 オペランドの指定値》 (単位: 秒)

継続メッセージ受信監視タイマを指定します。受信中のバッチ内のメッセージ受信後、次のメッセージを受信するまでを監視します。このオペランドを省略した場合、tim3 オペランドの指定値を使用します。送信側 MCA のバッチ終了待ちタイマ (バッチインターバル) より大きな値を指定してください。なお、このオペランドに 0 を指定した場合、継続メッセージの受信監視はされません。

● -x

(オペランド)

• sndexdata='送信出口ユーザデータ'

～<1~32 バイトの文字列> 《32 バイトのスペース文字》

送信出口に渡すユーザデータを指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時に送信出口ユーザデータ (SENDDATA) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を使用しません。

• rcvexdata='受信出口ユーザデータ'

～<1~32 バイトの文字列> 《32 バイトのスペース文字》

受信出口に渡すユーザデータを指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時に受信出口ユーザデータ (RCVDATA) として扱われます。接続する相手が TP1/

Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を使用しません。

• metexdata='メッセージ編集出口ユーザデータ'

～< 1～32 バイトの文字列> 《32 バイトのスペース文字》

メッセージ編集出口 UOC に渡すユーザデータを指定します。

すべてのチャンネルタイプで指定できます。クラスタレシーバで、この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時にメッセージ出口ユーザデータ (MSGDATA) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、クラスタセンダチャンネル動作時にメッセージ編集出口ユーザデータ (metexdata) として扱われます。自システムでは、チャンネル動作時にメッセージ編集出口ユーザデータ (metexdata) として扱われます。

• scyexdata='セキュリティ出口ユーザデータ'

～< 1～32 バイトの文字列> 《32 バイトのスペース文字》

セキュリティ出口に渡すユーザデータを指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時にセキュリティ出口ユーザデータ (SCYDATA) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を使用しません。

• sendexit='送信出口名'

～< 1～128 バイトの文字列> 《128 バイトのスペース文字》

チャンネル送信出口名を指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時に送信出口名 (SENDEXIT) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を使用しません。

• rcvexit='受信出口名'

～< 1～128 バイトの文字列> 《128 バイトのスペース文字》

チャンネル受信出口名を指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時に受信出口名 (RCVEXIT) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を使用しません。

• msgexit='メッセージ出口名'

～< 1～128 バイトの文字列> 《128 バイトのスペース文字》

チャンネルメッセージ出口名を指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時にメッセージ出口名 (MSGEXIT) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムでは、この定義を使用しません。

• scyexit='セキュリティ出口名'

～< 1～128 バイトの文字列> 《128 バイトのスペース文字》

チャンネルセキュリティ出口名を指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時にセキュリティ出口名 (SCYEXIT) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を使用しません。

これらのオペランドの指定値は、アポストロフィ (') で囲んで指定します。アポストロフィ (') で囲まれた文字列にアポストロフィ (') を指定する場合は、必ず 2 文字ペアで指定してください。

設定例 : metexdata=' abc' defg'

出口ユーザデータには ' abc' defg を渡します。

アポストロフィ (') で囲まれた文字列にアポストロフィ (') を指定する場合、出口ユーザデータ内に次に示す文字は指定できません。

タブ (x'09'), スペース (x'20'), 引用符 (x'22'), コンマ (x'2c'), およびイコール (x'3d')

● -t

(オペランド)

・ tcpsndbuf=TCP/IP 出力用バッファサイズ

~<符号なし整数>((1024~262144)) (単位: バイト)

TCP/IP 出力用バッファのサイズを指定します。このオペランドが指定されている場合、このオペランドの指定値で、SO_SNDBUF を設定した TCP/IP の setsockopt 関数を発行します。

省略時にはシステムの既定値を使用します。

■ 注意事項

- 通常はこの定義は省略してください。
- システムの既定値未満の値を設定した場合、MTU の 2 倍以下の値は設定しないでください。MTU の 2 倍以下の値を設定した場合、転送効率に影響が出るおそれがあります。
- 高速な通信媒体や MTU の大きな通信媒体を使用している場合、この値を大きくすると、通信処理に掛かる時間が改善される可能性があります。
- システムによっては、TCP/IP の setsockopt 関数が指定バッファサイズをサポートしていない場合があります。この場合、チャンネル開始時に MQT がエラーを検出し、チャンネルの開始を中断します。システムの指定可能範囲を調査し、指定を変更してください。
- TCP/IP のバッファや MTU の詳細については、TCP/IP の文献や OS のマニュアルを参照してください。

・ tcprcvbuf=TCP/IP 入力用バッファサイズ

~<符号なし整数>((1024~262144)) (単位: バイト)

TCP/IP 入力用バッファのサイズを指定します。このオペランドが指定されている場合、このオペランドの指定値で、SO_RCVBUF を設定した TCP/IP の setsockopt 関数を発行します。

省略時にはシステムの既定値を使用します。

注意事項

- 通常はこの定義は省略してください。
- システムの既定値未満の値を設定した場合、MTU の 2 倍以下の値は設定しないでください。MTU の 2 倍以下の値を設定した場合、転送効率に影響が出るおそれがあります。
- 高速な通信媒体や MTU の大きな通信媒体を使用している場合、この値を大きくすると、通信処理に掛かる時間が改善される可能性があります。
- システムによっては、TCP/IP の `setsockopt` 関数が指定バッファサイズをサポートしていない場合があります。この場合、チャンネル開始時に MQT がエラーを検出し、チャンネルの開始を中断します。システムの指定可能範囲を調査し、指定を変更してください。
- TCP/IP のバッファや MTU の詳細については、TCP/IP の文献や OS のマニュアルを参照してください。

• `tretrycnt` = TCP/IP 確立確認回数

～<符号なし整数> ((1～65535)) 《10》 (単位：回)

TCP/IP コネクションが確立したかどうかを確認する最大回数を指定します。この回数以内に TCP/IP コネクションが確立したと判断できた場合は、チャンネルの確立処理を実行します。判断できなかった場合は、TCP/IP インタフェースエラーになりチャンネル確立処理は失敗します。

• `tretryint` = TCP/IP 確立確認間隔

～<符号なし整数> ((1～65535)) 《1》 (単位：秒)

TCP/IP コネクションが確立したかどうかを確認する間隔を指定します。タイマ値の誤差が大きくなり、指定間隔とおりに動作できないことがあるため、基本タイマ値 (`mqttim` 定義コマンドの `-t btim` オペランド指定値と `bmtim` オペランド指定値の和) より小さい値を指定しないでください。

● `-d`

(オペランド)

• `cnvccsid`=変換後 CCSID

～<符号なし整数> ((1～65535))

アプリケーションから登録されたメッセージを文字コード変換する場合、変換後の CCSID を指定します。この指定値は、`-y type` オペランドが `sender`, `server`, `clusdr`, および `cluscvr` の場合に有効です。

注意事項

- `-d` オプションでの文字コード変換が失敗した場合は、メッセージ送信時に KFCA16334-E メッセージを出力してチャンネルを強制解放します。ただし、デッドレターキューが定義されている場合、メッセージは送信側のデッドレターキューに登録され、以降のメッセージ転送は続行されます。

-d オプションでの文字コード変換が失敗する場合、KFCA16334-E メッセージの理由コードを参照してください。なお、次の理由コードについては、注意事項があります。

< MQRC_CONVERTED_MSG_TOO_BIG >の注意事項

アプリケーションから登録するメッセージの長さは、変換前・変換後共に、次に示す計算式を満たす必要があります。

・バッファ方式がメッセージ (-g buftype=msg) の場合

アプリケーションメッセージ長 ≤ 最大メッセージ長
-MQXQH 構造体長

・バッファ方式がセグメント (-g buftype=seg) の場合

アプリケーションメッセージ長 ≤ 最大セグメント長
-MQXQH 構造体長 - プロトコルヘッダ長 (48 バイト)

最大メッセージ長は、-m オプションの maxmsg オペランドに指定した値と、相手システムのメッセージ長の、小さい方の値です。メッセージ長は、MQ システムによって呼称が異なりますので注意してください。

最大セグメント長は、-m オプションの maxseg オペランドに指定した値と、相手システムのセグメント長で、どちらか小さい方の値です。セグメント長 (送信データの最大長) は、MQ システムによって呼称が異なりますので注意してください。ただし、-g オプションの buftype オペランドに msg を指定した場合は、この注意事項は該当しません。

< MQRC_TARGET_CCSID_ERROR >の注意事項

変換後の CCSID には、コード変換対象となっている文字セット識別子を指定する必要があります。コード変換対象となっている文字セット識別子については、マニュアル「TP1/Message Queue プログラム作成リファレンス」のコード変換対象の文字セット識別子一覧についての記載を参照してください。

- cnvccsid オペランドの指定方法が誤っている場合、-y オプションの指定値に関係なくエラーメッセージが出力されます。
- クラスタレシーバに cnvccsid オペランドを指定する場合、相手システムが IBM MQ のときクラスタセンダの CONVERT 属性は YES になります。相手システムが TP1/Message Queue のときクラスタセンダの属性は変わりません。
- PCF (プログラマブルコマンドフォーマット) データの転送時は、コード変換 (mqtalcha 定義コマンドの -d オプションの cnvccsid オペランドに変換後の文字セット識別子を指定) は有効になりません。

● -s

(オペランド)

・ npmspeed=normal | fast

~ 《fast》

非永続メッセージ転送速度を指定します。

normal：非永続メッセージを永続メッセージと同様に転送します。

fast：非永続メッセージを高速で転送します。

注意事項

次に示す条件をすべて満たす場合は、受信側でキューマネージャによるセグメント分割ができません。KFCA16333-E（キューファイル障害発生。命令名=MQPUT，理由コード=2030）が出力されます。

- リクエスト，レシーバまたはクラスタレシーバに `npmspeed=fast` を指定しています。また，接続先の送信側チャンネルも `npmspeed=fast` 相当のチャンネル属性を持ちます。
- MQMDE 構造体の `MsgFlags` フィールドに `MQMF_SEGMENTATION_ALLOWED` が指定された非永続メッセージを受信しました。

● -z time | event

～《event》

転送キュー監視方式を指定します。

- time：タイマ方式

-v tim4 および tim4cnt オペランドによってメッセージの有無を確認して取り出します。システム構築する場合に旧バージョンとの互換性を持たせたいとき，この方式を採用することをお勧めします。

- event：イベント方式

転送キューにメッセージが登録されイベント連絡されるのを dtim オペランドで指定した時間だけ待ちます。また，バッチ終了待ちタイマ（バッチインターバル）の機能を使用できます。なお，新たにシステム構築する場合，この方式を採用することをお勧めします。

● -a

(オペランド)

- cluster=クラスタ名

～<1~48 文字の MQ 文字列>

チャンネルが所属するクラスタ名を指定します。

● -p

(オペランド)

- netprty=ネットワーク接続優先順位

～<符号なし整数>((0~9))《0》

ネットワーク接続優先順位を指定します。複数の経路が選択できる場合は，優先順位の最高の経路が選択されます。優先順は 0 が最低，9 が最高です。

この指定値は，-y オプションの type オペランドが clusrcvr の場合だけ有効です。

● -f

(オペランド)

- adoptchk=off | name | qmgr | ipaddr | all
 ~ 《qmgr》

強制確立確認条件を指定します。

off を指定した場合は、チャンネル開始要求を受信したレスポンドチャンネルでチャンネル状態不一致が発生したときに、開始要求を受け付けません。off 以外の値を指定すると、チャンネル開始要求を受信したレスポンドチャンネルでチャンネル状態不一致が発生した場合に、動作しているチャンネルを強制終了して開始要求を受け付けます。off 以外の値を指定した場合でチャンネル開始要求を受信したチャンネルがチャンネル動作中 (RUNNING) またはチャンネル解放中 (STOPPING) のときは、次のように動作します。

- name
 チャンネルを強制終了して開始要求を受け付けます。
- qmgr
 チャンネルが現在、確立中の相手と同じキューマネージャからのチャンネル開始要求を受けた場合にチャンネルを強制終了して開始要求を受け付けます。この指定は、相手システムがMQ プロトコルレベル 4 (FAP4) 以降をサポートしている場合に有効です。FAP3 以下ではチェックしません。
- ipaddr
 チャンネルが現在、確立中の相手と同じ IP アドレスからのチャンネル開始要求を受けた場合にチャンネルを強制終了して開始要求を受け付けます。
- all
 チャンネルが現在、確立中の相手と同じキューマネージャおよび IP アドレスからのチャンネル開始要求を受けた場合にチャンネルを強制終了して開始要求を受け付けます。この指定は、相手システムがMQ プロトコルレベル 4 (FAP4) 以降をサポートしている場合にキューマネージャと IP アドレスの両方をチェックします。FAP3 以下では IP アドレスだけをチェックします。

クラスタレシーバでは、マルチインスタンスをサポートしていて、該当インスタンスを特定するためにキューマネージャ名を必ずチェックする必要があります。このため、確立中での各指定は次の意味を持ちます。

name または qmgr を指定した場合

キューマネージャ名チェックを実行します。

ipaddr または all を指定した場合

キューマネージャ名チェック、および IP アドレスチェックを実行します。

チャンネル開始要求の受け付け条件について次に示します。

チェック内容	FAP レベル 3 以下		FAP4 レベル以降 (クラスタレシーバ以外)				クラスタレシーバ	
	name, qmgr	ipaddr, all	name	qmgr	ipaddr	all	name, qmgr	ipaddr, all
チャンネル名	○	○	○	○	○	○	○	○
キューマネージャ名	—	—	—	○	—	○	○	○
IP アドレス	—	○	—	—	○	○	—	○

(凡例)

- ：チェックします。すべて一致した場合だけ、開始要求を受け付けます。
- ：チェックしません。

● -h '注釈'

～< 1～64 バイトの文字列 > 《64 バイトのスペース文字》

チャンネルの注釈を指定します。

このオプションの指定値は、アポストロフィ (') で囲んで指定します。ただし、アポストロフィ (') で囲まれた文字列にアポストロフィ (') は指定できません。

● -mc

(オペランド)

・ mcauser='MCA ユーザ ID'

～< 1～64 バイトの文字列 > 《64 バイトのスペース文字》

IBM MQ の資源へのアクセスに使用するユーザ ID を指定します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時にメッセージチャンネルエージェントユーザ ID (MCAUSER) として扱われます。接続する相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を無視します。詳細については IBM MQ のマニュアルを参照してください。

このオペランドの指定値は、アポストロフィ (') で囲んで指定します。ただし、アポストロフィ (') で囲まれた文字列にアポストロフィ (') は指定できません。

・ mcatype=process | thread

～ 《process》

IBM MQ の MCA プログラムを、スレッドとプロセスのどちらで実行するかを指定します。

process：MCA は、独立のプロセスとして動作します。

thread：MCA は、独立のスレッドとして動作します。

クラスタレシーバだけ指定できます。この指定で接続する相手が IBM MQ の場合は、クラスタセンダチャンネル動作時にメッセージチャンネルエージェントタイプ (MCATYPE) として扱われます。接続す

る相手が TP1/Message Queue (クラスタセンダ) の場合、および自システムのクラスタレシーバでは、この定義を無視します。詳細については IBM MQ のマニュアルを参照してください。

● -jn

(オペランド)

・ `sndjnl=flush | noflush`

～ 《flush》

送信チャンネルジャーナル出力条件を指定します。

送信チャンネルはバッチ転送完了時 (送達確認受信時) にジャーナルを取得します。このジャーナルを即時フラッシュ (ディスク出力) するかどうかを指定します。

`flush` : 即時フラッシュ (ディスク出力) します。

`noflush` : 即時フラッシュ (ディスク出力) しません。

このオペランドは、センダ、サーバ、およびクラスタセンダに指定できます。

`noflush` を指定した場合、ジャーナルをジャーナルバッファにだけ書き込み、ディスク出力は行いません。この場合、ディスク出力は TP1/Server Base の制御下で実行されます。

ファーストメッセージ機能を使用しない場合、または通常メッセージを送信する場合は、`noflush` を指定することでディスク出力回数が少なくなり、チャンネルの送信時間を短縮できます。

■ 注意事項

`noflush` を指定した場合、受信側チャンネルで受信完了している状態で送達未確認状態のチャンネルに `mqtrlvcha -a backout` コマンドを実行すると、不正なメッセージが送信されることがあります。チャンネルを再度開始するか、または受信側チャンネルの状態を確認してから `mqtrlvcha` コマンドを使用することで送達未確認状態を解決してください。

デフォルトチャンネル定義

デフォルトチャンネル定義を使用すると、mqtalcccha 定義コマンドのチャンネル定義が複数あって、それぞれの定義に同一オプションの指定が複数ある場合に、定義項目の指定数を削減できます。デフォルトチャンネル定義では、mqtalcccha 定義コマンドのオプションが省略された場合の解釈値を任意の値に変更できます。ただし、デフォルトチャンネル定義の構文要素や指定範囲は、mqtalcccha 定義コマンドの各オプションで決められた形式に従って指定する必要があります。

チャンネルの属性として採用する値を次の表に示します。

表 4-12 チャンネル属性として採用する値

デフォルトチャンネル定義	mqtalcccha 定義	
	あり	なし
あり	mqtalcccha 定義の値	デフォルトチャンネル定義の値
なし	mqtalcccha 定義の値	システムで定義されている値

なお、mqtalcccha 定義コマンドのオプションに値を定義していない場合で、デフォルト定義に該当オプションの定義があるときはその値を採用します。しかし、デフォルト定義にないときは、システムで定義されている値を採用します。

mqtalcccha 定義コマンドのオプションで値を定義している場合は、その値を採用します。ただし、mqtalcccha 定義コマンドの次に示すオプションには、オペランドの優先順位があります。優先順位の高いオペランドをデフォルトチャンネルで定義すると、デフォルトチャンネルの値が採用されますので注意してください。

- -r オプションの ipaddr オペランドおよび hostname オペランド (ipaddr 優先)
- -r オプションの portno オペランドおよび servname オペランド (portno 優先)
- -o オプションの oipaddr オペランドおよび ohostname オペランド (oipaddr 優先)
- -o オプションの oportno オペランドおよび oservname オペランド (oportno 優先)

次にデフォルトチャンネル定義の使用例を示します。なお、デフォルトチャンネル定義は、別ファイルで定義します。

<デフォルトチャンネル定義を使用しない定義>

```

mqtal ccha -c CHA01          ¥
           -y "type=sender"  ¥
           -j 50             ¥
           -q trnq1          ¥
           -m "maxmsg=8192   ¥
              maxseg=4096"  ¥
           -w 100            ¥
           -g "sndbuf=1     ¥
              buftype=msg"  ¥
           -b "bretrycnt=0   ¥
              bretryint=10" ¥
           -o "ohostname=mqhost1 ¥
              oportno=1414" ¥
           -z event          ¥
mqtal ced
mqtal ccha -c CHA02          ¥
           -y "type=sender"  ¥
           -j 50             ¥
           -q trnq2          ¥
           -m "maxmsg=8192   ¥
              maxseg=2048"  ¥
           -w 100            ¥
           -g "sndbuf=1     ¥
              buftype=msg"  ¥
           -b "bretrycnt=0   ¥
              bretryint=10" ¥
           -o "ohostname=mqhost1 ¥
              oportno=1414" ¥
           -z event          ¥
mqtal ced
mqtal ccha -c CHA03          ¥
           -y "type=sender"  ¥
           -j 50             ¥
           -q trnq3          ¥
           -m "maxmsg=8192   ¥
              maxseg=4096"  ¥
           -w 100            ¥
           -g "sndbuf=1     ¥
              buftype=msg"  ¥
           -b "bretrycnt=0   ¥
              bretryint=10" ¥
           -o "ohostname=mqhost1 ¥
              oportno=1414" ¥
           -z event          ¥
mqtal ced
    
```

<デフォルトチャンネル定義を使用した定義>

```

mqtal ccha -y "type=defsnd"  ¥
           -j 50             ¥ ※1
           -m "maxmsg=8192   ¥
              maxseg=4096"  ¥ ※2
           -w 100            ¥
           -g "sndbuf=1     ¥
              buftype=msg"  ¥
           -b "bretrycnt=0   ¥
              bretryint=10" ¥
           -o "ohostname=mqhost1 ¥
              oportno=1414" ¥
           -z event          ¥ ※1
mqtal ced
mqtal ccha -c CHA01          ¥ ※3
           -y "type=sender"  ¥
           -q trnq1          ¥
mqtal ced
mqtal ccha -c CHA02          ¥ ※3
           -y "type=sender"  ¥
           -q trnq2          ¥
           -m "maxseg=2048"  ¥
mqtal ced
mqtal ccha -c CHA03          ¥ ※3
           -y "type=sender"  ¥
           -q trnq3          ¥
mqtal ced
    
```

注※1 すべてのmqtal ccha定義で共通であるためデフォルトチャンネル定義で定義する

注※2 CHA01とCHA03で共通であるためデフォルトチャンネル定義で定義する

注※3 mqtal ccha定義で必須のオプションとデフォルトチャンネル定義と異なるオプションを定義する

各チャンネルタイプに指定できるオプション

mqtalcccha 定義コマンドの各オプションは、チャンネルタイプごとに指定できるものとできないものがあります。各チャンネルタイプに指定できるオプションについて説明します。

通常チャンネルおよびクラスタチャンネルに指定できるオプション

通常チャンネルおよびクラスタチャンネルに指定できるオプションについて次の表に示します。

表 4-13 通常チャンネルおよびクラスタチャンネルに指定できるオプション

オプション	オペランド	sender	server	receiver	requester	clusdr	clusrcvr
-c	—	●	●	●	●	●	●
-y	type	●	●	●	●	●	●
-a	cluster	—	—	—	—	◎	●
-j	—	○	○	○	○	○	○
-q	—	◎	◎	—	—	—	—
-m	maxmsg	○	○	○	○	○	○
	maxseg	○	○	○	○	○	○
-w	—	○	○	○	○	○	○
-g	sndbuf	◎	◎	—	—	◎	—
	rcvbuf	—	—	◎	◎	—	●
	buftype	○	○	○	○	○	○
-e	medbuf	○	○	○	○	○	○
	medcnt	○	○	○	○	○	○
-i	—	○	○	○ (auto2 指定 は—)	○	○	○ (auto2 指定 は—)
-b	bretry	○	△	—	△	○	○
	bretrycnt	○	△	—	△	○	○
	bretryint	○	△	—	△	○	○
	bretrylg	○	△	—	△	○	○
	bretrylgcnt	○	△	—	△	○	○
	bretrylgint	○	△	—	△	○	○
	bretrymcp	○	△	—	△	○	—

オプション	オペランド	sender	server	receiver	requester	clusdr	clusrcvr
-r	ipaddr	○	○	—	○	○	●
	hostname			—			
	portno	○	○	—	○	○	—
	servname			—			—
	portnum	○	○	—	○	○	—
-o	oipaddr	◎	△	—	△	◎	—
	ohostname			—			—
	oportno	◎	△	—	△	◎	—
	oservname			—			—
-v	tim1	○	△	—	△	○	—
	tim2	○	○	—	—	○	—
	tim3	—	—	○	○	—	○
	tim4	○	○	—	—	—	—
	tim4cnt	○	○	—	—	—	—
	dtimefct	○	○	—	—	—	—
	dtim	○	○	—	—	○	○
	vretry	○	△	—	△	○	—
	htim	○	○	○	○	○	○
	btim	○	○	—	—	○	○
	bmtim	○	○	—	—	○	○
	mtim	—	—	○	○	—	○
-x	sndexdata	—	—	—	—	—	□
	rcvexdata	—	—	—	—	—	□
	metexdata	○	○	○	○	○	○
	scyexdata	—	—	—	—	—	□
	sendexit	—	—	—	—	—	□
	rcvexit	—	—	—	—	—	□
	msgexit	—	—	—	—	—	□
	scyexit	—	—	—	—	—	□
-t	tcpsndbuf	○	○	○	○	○	○

オプション	オペランド	sender	server	receiver	requester	clusdr	clusrcvr
-t	tcprcvbuf	○	○	○	○	○	○
	tretrycnt	○	△	—	△	○	—
	tretryint	○	△	—	△	○	—
-d	cnvccsid	○	○	—	—	○	○
-s	npmspeed	○	○	○	○	○	○
-z	—	○	○	—	—	—	—
-p	netprty	—	—	—	—	—	○
-f	adoptchk	○	○	○	○	—	○
-h	—	○	○	○	○	○	○
-mc	mcauser	—	—	—	—	—	□
	mcatype	—	—	—	—	—	□
-jn	sndjnl	○	○	—	—	○	—

(凡例)

- ：指定が必要です。
- ◎：指定が必要です。ただし、デフォルトチャンネル定義で定義している場合は、省略できます。
- ：省略できます。
- △：コーラだけ指定できます。
- ：TP1/Message Queue では使用しません。
- ：該当しません。

デフォルトチャンネルに指定できるオプション

デフォルトチャンネルに指定できるオプションについて次の表に示します。

表 4-14 デフォルトチャンネルに指定できるオプション

オプション	オペランド	defsnd	defsrv	defrcv	defreq	defcsn
-c	—	—	—	—	—	—
-y	type	●	●	●	●	●
-a	cluster	—	—	—	—	△
-j	—	○	○	○	○	△
-q	—	○	○	—	—	—
-m	maxmsg	○	○	○	○	△
	maxseg	○	○	○	○	△
-w	—	○	○	○	○	△

オプション	オペランド	defsnd	defsrv	defrcv	defreq	defcsn
-g	sndbuf	○	○	—	—	○
	rcvbuf	—	—	○	○	—
	buftype	○	○	○	○	○
-e	medbuf	○	○	○	○	○
	medcnt	○	○	○	○	○
-i	—	○	○	○ (auto2 指定 は—)	○	○
-b	bretry	○	○	—	○	△
	bretrycnt	○	○	—	○	△
	bretryint	○	○	—	○	△
	bretrylg	○	○	—	○	△
	bretrylgcnt	○	○	—	○	△
	bretrylgint	○	○	—	○	△
	bretrymcp	○	○	—	○	○
-r	ipaddr	○	○	—	○	○
	hostname			—		
	portno	○	○	—	○	○
	servname			—		
	portnum	○	○	—	○	○
-o	oipaddr	○	○	—	○	△
	ohostname			—		
	oportno	○	○	—	○	△
	oservname			—		
-v	tim1	○	○	—	○	○
	tim2	○	○	—	—	○
	tim3	—	—	○	○	—
	tim4	○	○	—	—	—
	tim4cnt	○	○	—	—	—
	dtimefct	○	○	—	—	—
	dtim	○	○	—	—	△

オプション	オペランド	defsnd	defsrv	defrcv	defreq	defcsn
-v	vretry	○	○	—	○	○
	htim	○	○	○	○	△
	btim	○	○	—	—	△
	bmtim	○	○	—	—	△
	mtim	—	—	○	○	—
-x	sndexdata	—	—	—	—	—
	rcvexdata	—	—	—	—	—
	metexdata	○	○	○	○	△
	scyexdata	—	—	—	—	—
	sendexit	—	—	—	—	—
	rcvexit	—	—	—	—	—
	msgexit	—	—	—	—	—
	scyexit	—	—	—	—	—
-t	tcpsndbuf	○	○	○	○	○
	tcprcvbuf	○	○	○	○	○
	tretrycnt	○	○	—	○	○
	tretryint	○	○	—	○	○
-d	cnvccsid	○	○	—	—	○
-s	npmspeed	○	○	○	○	△
-z	—	○	○	—	—	—
-p	netprty	—	—	—	—	—
-f	adoptchk	○	○	○	○	—
-h	—	○	○	○	○	△
-mc	mcauser	—	—	—	—	—
	mcatype	—	—	—	—	—
-jn	sndjnl	○	○	—	—	○

(凡例)

●：指定が必要です。

○：省略できます。

△：クラスタに参加するまでの事前定義クラスタセンダチャンネルだけ指定できます。

—：該当しません。

チャネルのネゴシエーション

mqtalccha 定義コマンドで指定するチャネル属性には、チャネル確立時に自システムのチャネルと相手システムのチャネルとの間で、適切な値に調整したり同じ値かどうかをチェックしたりする属性があります。この処理をネゴシエーションといいます。

ネゴシエーションを行うチャネル属性と、ネゴシエーションの結果について次の表に示します。

表 4-15 ネゴシエーションを行うチャネル属性と、ネゴシエーションの結果

チャネル属性	TP1/Message Queue の設定	ネゴシエーションの結果
メッセージチャネルのプロトコルレベル (FAP レベル)	4	相手システムが FAP レベル 4 をサポートしている場合は、FAP レベルは 4 になります。 相手システムが FAP レベル 4 をサポートしていない場合は、FAP レベルは 1 になります。 FAP レベルが 1 の場合、マルチキャスト機能、ファーストメッセージ機能、およびハートビート機能は使用できません。
メッセージシーケンス番号の使用	使用	相手システムが、メッセージシーケンス番号を使用していない場合は、チャネル接続できません。
メッセージ転送時のセグメント分割の使用	使用	相手システムに合わせます。ただし、次に示す二つの条件が同時に成立する場合は、チャネル接続できません。 <ul style="list-style-type: none"> 相手システムがセグメント分割を使用しないとき 次の計算式を満たすとき ネゴシエーション後の最大セグメント長 < ネゴシエーション後の最大メッセージ長 + 476
セキュリティ要求	機能なし	そのまま正常扱いで処理されます。

mqtalccha 定義コマンドのオプションおよびオペランドで、ネゴシエーションを行うチャネル属性について次の表に示します。

表 4-16 ネゴシエーションを行うチャネル属性 (mqtalccha 定義コマンドのオプションおよびオペランド)

オプション	オペランド	ネゴシエーション方法	備考
-c	-	●	-
-y	type	□	適切なチャネルタイプを組み合わせてください。チャネルタイプの組み合わせについては、「2.3.2 MCA のチャネルタイプ」を参照してください。

オプション	オペランド	ネゴシエーション方法	備考
-j	—	○*	—
-m	maxmsg	○	—
-m	maxseg	○	—
-w	—	□	相手システムと同じメッセージシーケンス番号の最大値を指定してください。異なる場合、チャンネル接続できません。
-g	buftype	□	自システムのチャンネル属性がセグメント方式の場合は、次に示す制限があります。 <ul style="list-style-type: none"> 配布リストの送受信ができません。 MQMDE.MsgFlags に MQMF_SEGMENTATION_ALLOWED が指定されたメッセージを受信してもキューマネージャによるセグメント分割ができません。 メッセージ編集出口 UOC が使用できません (UOC デーモンを作成しても UOC は呼び出されません)。
-v	htim	△	—
		□	自システムのチャンネルの属性、または相手システムのチャンネルの属性が 0 (ハートビート機能なし) の場合は 0 となります。
-s	npmspeed	□	自システムのチャンネルの属性、または相手システムのチャンネルの属性が normal の場合は normal になります。 ネゴシエーションの結果、fast になった場合は、MQMDE.MsgFlags に MQMF_SEGMENTATION_ALLOWED が指定された非永続メッセージを受信しても、キューマネージャでは、セグメント分割できません。

(凡例)

- ：自システムのチャンネルの属性と相手システムのチャンネルの属性が、同じ値かどうかチェックします。
- ：自システムのチャンネルの属性と相手システムのチャンネルの属性を比較して、小さい方の値を採用します。
- △：自システムのチャンネルの属性と相手システムのチャンネルの属性を比較して、大きい方の値を採用します。
- ：備考欄を参照してください。
- ：該当しません。

注※

自動定義クラスタセンダチャンネルのバッチサイズは、次に示す計算式の値です。

$$\frac{\text{mqamqtnam定義コマンド-bオプションに指定した総バッチメッセージ数} \times \text{事前定義クラスタセンダチャンネルに指定したバッチサイズの総数}}{\text{mqamqtnam定義コマンドに指定したMCA数} \times \text{事前定義クラスタセンダチャンネル定義総数}}$$

(凡例) ↓ ↓ : 小数点以下を切り下げます。

mqtalced (チャンネル終了定義)

形式

```
mqtalced
```

機能

チャンネル定義の終了を示します。

オプション

なし

MQT 定義オブジェクトファイルの作成手順

MQT 定義オブジェクトファイルの作成手順および旧製品の定義を使用する場合の注意事項について説明します。

MQT 定義オブジェクトファイルの作成手順

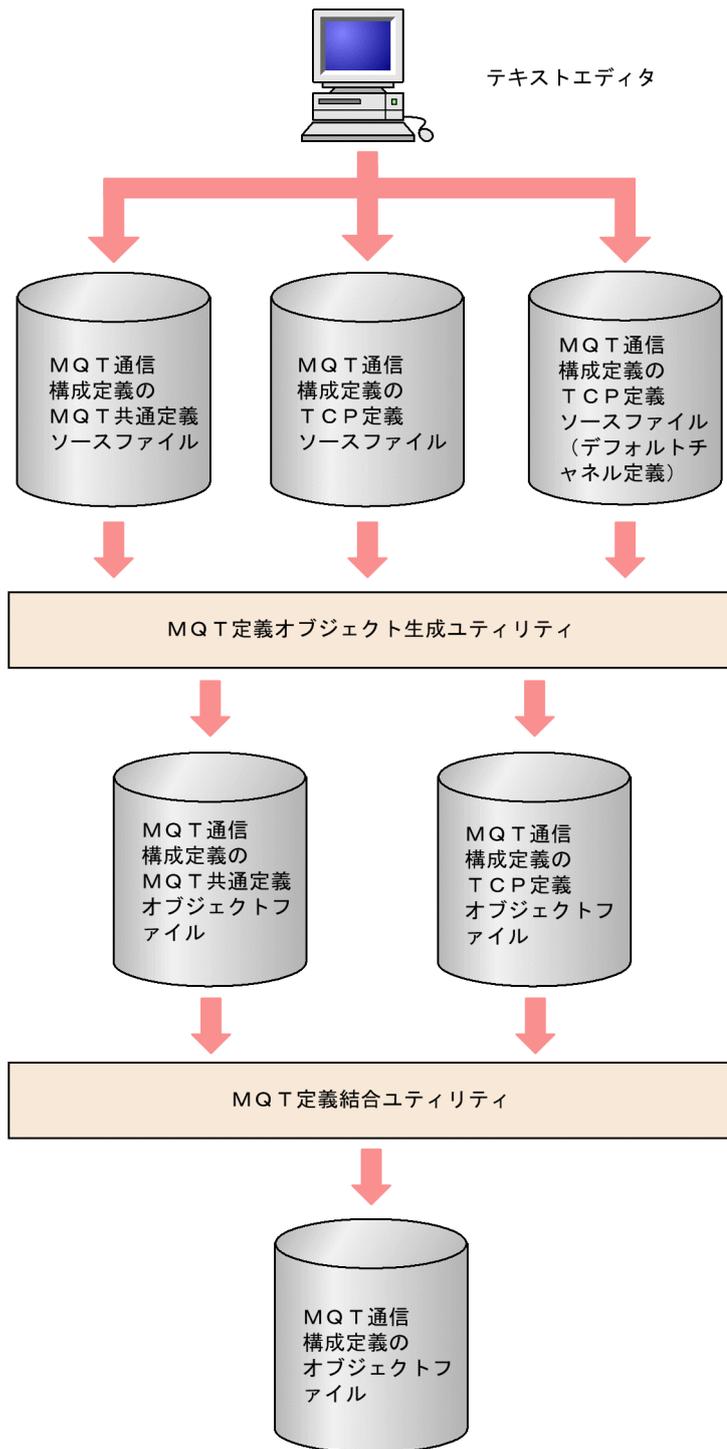
MQT 定義オブジェクトファイルは次に示す手順で作成します。

定義オブジェクトファイルは、開始から再開始の間に変更しないでください。変更した場合、再開始の動作は保証されません。

1. オペレーティングシステムのテキストエディタを使用して、MQT の定義ファイルから、次に示す定義ソースファイルを作成します。
 - MQT 通信構成定義の MQT 共通定義ソースファイル
 - MQT 通信構成定義の TCP 定義ソースファイル
 - MQT 通信構成定義の TCP 定義ソースファイル（デフォルトチャンネル定義）。デフォルトチャンネル定義は、省略できます。
2. MQT 定義オブジェクト生成ユーティリティを使用して、定義ソースファイルから、次に示すオブジェクトファイルを作成します。
 - MQT 通信構成定義の MQT 共通定義オブジェクトファイル
 - MQT 通信構成定義の TCP 定義オブジェクトファイル
3. MQT 定義結合ユーティリティを使用して、MQT 通信構成定義の、MQT 共通定義と TCP 定義のオブジェクトファイルを結合します。
4. 結合した MQT 通信構成定義のオブジェクトファイルを、UNIX の場合は\$DCCONFPATH、Windows の場合は%DCCONFPATH%に登録します。

MQT 定義オブジェクトファイルの作成方法の概要について、次の図に示します。

図 4-3 MQTT 定義オブジェクトファイルの作成方法の概要



MQTT 通信構成定義でホスト名ファイル，およびサービス名ファイルを更新する場合，次に示す点に注意してください。

相手システムのホスト名（mqtalcca 定義コマンドの-o ohostname オペランド）を指定した場合，ホスト名ファイル，またはネームサーバに相手システムのホスト名を登録しなければなりません。

また、相手システムのホストのサービス名（mqtalcccha 定義コマンドの-o oservername オペランド）および自システムのサービス名（mqtalcccha 定義コマンドの-r servname オペランド）を指定して、特定のポート番号を使用する場合、サービス名ファイルにサービス名を登録しなければなりません。

旧製品の MQT 定義を使用する場合の注意事項

TP1/Message Queue 07-50 以降、TP1/Message Queue 05-15 以降および 05-13 以降では、MQT 定義の指定を省略した場合の解釈値を変更しました。TP1/Message Queue 07-01 以前、TP1/Message Queue 05-14 以前または 05-12 以前の定義を使用する場合、動作が異なることがありますので注意してください。変更内容をそれぞれ表 4-17、表 4-18 および表 4-19 に示します。

表 4-17 TP1/Message Queue 07-50 以降で変更した MQT 定義の省略時の解釈値

定義コマンド名, オプション名, およびオペランド名	TP1/Message Queue 07-01 以前で省略した場合の解釈値	TP1/Message Queue 07-50 以降で省略した場合の解釈値	チャンネルタイプ
mqttcp -t disk	no	yes	センダ, サーバ, リクエスト, レシーバ
mqttcps -t disk	no	yes	クラスタセンダ
mqttcpcr -t disk	no	yes	クラスタレシーバ
mqttcp -t bufcnt	100	2500	センダ, サーバ, リクエスト, レシーバ
mqttcps -t bufcnt	100	2500	クラスタセンダ
mqttcpcr -t bufcnt	100	2500	クラスタレシーバ

表 4-18 TP1/Message Queue 05-15 以降で変更した MQT 定義の省略時の解釈値

定義コマンド名, オプション名, およびオペランド名	TP1/Message Queue 05-14 以前で省略した場合の解釈値	TP1/Message Queue 05-15 以降で省略した場合の解釈値	チャンネルタイプ
mqttcp -v etim	0	360	センダ, サーバ, リクエスト, レシーバ
mqttcps -v etim	0	360	クラスタセンダ
mqttcpcr -v etim	0	360	クラスタレシーバ

表 4-19 TP1/Message Queue 05-13 以降で変更した MQT 定義の省略時の解釈値

定義コマンド名, オプション名, およびオペランド名	TP1/Message Queue 05-12 で省略した場合の解釈値	TP1/Message Queue 05-13 以降で省略した場合の解釈値	チャンネルタイプ
mqtttim -t btim	5	1	すべてのチャンネル
mqtalcccha -m maxmsg	127996000	バッファサイズ	センダ, サーバ, リクエスト, レシーバ

定義コマンド名, オプション名, およびオペランド名	TP1/Message Queue 05-12で省略した場合の解釈値	TP1/Message Queue 05-13以降で省略した場合の解釈値	チャンネルタイプ
mqtalcccha -g buftype	seg	msg	センダ, サーバ, リクエスト, レシーバ
mqtalcccha -b bretrycnt	0	10	センダ, サーバ, リクエスト
	65535	10	クラスタセンダ, クラスタレシーバ
mqtalcccha -b bretrymcp	no	yes	センダ, サーバ, リクエスト
mqtalcccha -v vretry	no	yes	センダ, サーバ, リクエスト
mqtalcccha -v htim	0	300	センダ, サーバ, リクエスト, レシーバ
mqtalcccha -s nprmspeed	normal	fast	センダ, サーバ, リクエスト, レシーバ
mqtalcccha -z	time	event	センダ, サーバ
mqtalcccha -f adoptchk	off	qmgr	センダ, サーバ, リクエスト, レシーバ

注意事項

TP1/Message Queue 05-12 以前の動作を継続する場合の注意事項を次に示します。

- mqtttim 定義コマンドの -t btim オペランドは、デフォルトチャンネル定義で指定できないため、TP1/Message Queue 05-12 以前の動作を継続したい場合は、定義値を指定する必要があります。
- mqtalcccha 定義コマンドの -b bretrycnt オペランドは、0（無限回）を指定できなくなり、指定値の範囲を、"0~65535"から"1~999999999"に変更しました。
- mqtalcccha 定義コマンドの -b bretryint オペランドの指定値の範囲を、"0~2550"から"0~999999999"に変更しました。

サンプルデフォルト定義

既存の定義で、表 4-18、および表 4-19 に示されている項目にある定義のどれかを省略していて、今後も今までの動作を継続したい場合、定義値を指定するか、システム提供のサンプルデフォルト定義を使用してください。サンプルデフォルト定義は、通常チャンネル用とクラスタセンダチャンネル用に分かれています。次に通常チャンネル用とクラスタセンダチャンネル用のサンプルデフォルト定義の格納先を示します。

通常チャンネル用のサンプルデフォルト定義の格納場所

`examples/mq/conf/def_cha`

クラスタセンダチャンネル用のサンプルデフォルト定義の格納場所

`examples/mq/conf/def_chac`

指定方法は、この章の「[MQT 定義オブジェクト生成ユティリティ](#)」を参照してください。

MQT 定義オブジェクト生成ユーティリティ

概要

MQT 定義オブジェクト生成ユーティリティは、MQT の定義ファイルの構文のチェックと、MQT 定義オブジェクトファイルへの変換をします。

MQT 定義オブジェクト生成ユーティリティの詳細を説明します。

形式

```
mqt×××× -i [パス名] 入力ファイル名
          [-d [パス名] 入力ファイル名2]
          -o [パス名] 出力オブジェクトファイル名
```

機能

MQT の定義ファイルの構文のチェック、および MQT 定義オブジェクトファイルの作成をします。

ただし、開始から再開始の間は MQT 定義オブジェクトファイルを変更できません。変更した場合、再開始の動作は保証されません。

オプション

● mqt××××

ユーティリティを起動するコマンドを指定します。

- mqtcomn
MQT 共通定義のユーティリティを起動します。
- mqttcpip
TCP 定義のユーティリティを起動します。
- mqttcpcs
クラスタセンダプロセス用 TCP 定義のユーティリティを起動します。
- mqttcpcr
クラスタレシーバプロセス用 TCP 定義のユーティリティを起動します。

● -i [パス名] 入力ファイル名

～<パス名>< 1～8 文字の識別子>

定義ソースが格納されているファイル名を指定します。

● -d [パス名] 入力ファイル名 2

～<パス名>< 1～8 文字の識別子>

デフォルトチャンネル定義ソースが格納されているファイル名を指定します。

mqttcpip コマンド、および mqttcps コマンドで指定できます。

● -o [パス名] 出力オブジェクトファイル名

～<パス名><1～8文字の英数字>

定義オブジェクトを格納するファイル名を指定します。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA18000-E	定義コマンドの指定順序に誤りがあります。	標準エラー出力
KFCA18001-E	定義コマンドは省略できません。	標準エラー出力
KFCA18002-E	定義コマンドの数が上限値を超えています。	標準エラー出力
KFCA18003-E	不当な文字列があります。	標準エラー出力
KFCA18004-E	二重引用符で囲まれた文字数が、1024 けたを超えています。	標準エラー出力
KFCA18005-E	定義コマンドのオプションは省略できません。	標準エラー出力
KFCA18006-E	定義コマンドのオプションの指定に誤りがあります。	標準エラー出力
KFCA18007-E	定義コマンドに不当な文字列があります。	標準エラー出力
KFCA18010-E	定義コマンドのオプションの指定値が、他の定義コマンドのオプションの指定値と重複しています。	標準エラー出力
KFCA18015-E	定義コマンドのオペランドは省略できません。	標準エラー出力
KFCA18016-E	定義コマンドのオペランドの指定に誤りがあります。	標準エラー出力
KFCA18017-E	定義コマンドのオプションに不当な文字列があります。	標準エラー出力
KFCA18018-E	定義コマンドのオペランドは省略できません。	標準エラー出力
KFCA18020-E	他の定義コマンドのオペランドの指定値と重複しています。	標準エラー出力
KFCA18029-E	定義コマンドに対応する定義コマンドが指定されていません。	標準エラー出力
KFCA18030-E	指定ファイルのオープン処理に失敗しました。	標準エラー出力
KFCA18031-E	メモリの確保に失敗しました。	標準エラー出力
KFCA18032-E	指定された定義オブジェクトファイル名は既に存在しています。	標準エラー出力
KFCA18033-E	指定ファイルの入出力処理に失敗しました。	標準エラー出力
KFCA18046-I	定義オブジェクト生成ユティリティ処理が異常終了しました。	標準出力
KFCA18047-I	定義オブジェクト生成ユティリティ処理で定義エラーを検知して終了しました。	標準出力
KFCA18048-I	定義オブジェクト生成ユティリティ処理を開始しました。	標準出力
KFCA18049-I	定義オブジェクト生成ユティリティ処理が正常終了しました。	標準出力

出力メッセージID	内容	出力先
KFCA18107-E	定義コマンドのオプションは指定できません。	標準エラー出力
KFCA18108-E	定義コマンドのオペランドは指定できません。	標準エラー出力
KFCA18113-E	定義コマンドのオペランドには指定できません。	標準エラー出力
KFCA18114-E	定義コマンドのオプションは省略できません。	標準エラー出力
KFCA18119-E	定義コマンドのオペランドは省略できません。	標準エラー出力
KFCA18121-E	定義コマンドのオペランドは省略できません。	標準エラー出力
KFCA18122-E	定義コマンドのオプションのオペランド数が、システム上指定できる上限値を超えています。	標準エラー出力
KFCA18143-E	定義コマンドのオペランドは同時に省略できません。	標準エラー出力

MQT 定義結合ユーティリティ

概要

MQT 定義結合ユーティリティでは、MQT 定義ファイル（MQT 共通定義オブジェクトファイル、および TCP 定義オブジェクトファイル）のコマンド間の関連をチェックし、二つのオブジェクトファイルを結合します。コマンド間の関連が整合しない場合、MQT 定義オブジェクトファイルは生成されず、TP1/Message Queue は処理を終了します。

MQT 定義結合ユーティリティの詳細を説明します。

形式

```
mqmlink -i [パス名] MQT共通定義オブジェクトファイル名
         [パス名] TCP定義オブジェクトファイル名
         -o [パス名] 出力オブジェクトファイル名
```

機能

MQT の定義ファイルのコマンド間の整合性のチェック、および MQT 定義オブジェクトファイルの結合をします。

オプション

- **-i [パス名] MQT 共通定義オブジェクトファイル名 [パス名] TCP 定義オブジェクトファイル名**
～< 1～8 文字の英数字 >

MQT 共通定義オブジェクト、および TCP 定義オブジェクトのファイル名を指定します。ファイル名は、どちらを先に指定してもかまいません。

- **-o [パス名] 出力オブジェクトファイル名**
～< _ (アンダスコア) で始まる 2～8 文字の英数字 >

MQT 定義オブジェクトを格納するファイル名を指定します。ファイル名は、先頭のアンダスコアを含めて 8 文字までの英数字で指定してください。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA18005-E	定義コマンドのオプションは省略できません。	標準エラー出力
KFCA18006-E	定義コマンドのオプションの指定に誤りがあります。	標準エラー出力
KFCA18007-E	定義コマンドに不当な文字列があります。	標準エラー出力
KFCA18021-E	定義コマンドのオペランドに定義されていません。	標準エラー出力

出力メッセージID	内容	出力先
KFCA18030-E	指定ファイルのオープン処理に失敗しました。	標準エラー出力
KFCA18031-E	メモリの確保に失敗しました。	標準エラー出力
KFCA18032-E	指定された定義オブジェクトファイル名は既に存在しています。	標準エラー出力
KFCA18033-E	指定ファイルの入出力処理に失敗しました。	標準エラー出力
KFCA18034-E	定義コマンドのオプションのオペランドの指定値が、対応する定義コマンドのバッファ数の合計値を超えています。	標準エラー出力
KFCA18100-E	定義コマンドのオプションで指定されたファイルは結合できません。	標準エラー出力
KFCA18101-I	定義オブジェクト結合ユーティリティ処理を開始しました。	標準出力
KFCA18102-I	定義オブジェクト結合ユーティリティ処理で定義エラーを検知して終了しました。	標準出力
KFCA18104-I	定義オブジェクト結合ユーティリティ処理が正常終了しました。	標準出力
KFCA18159-E	定義コマンドのオプションのオペランドの指定値が、対応する定義コマンドのオプションのオペランドの指定値を超えています。	標準エラー出力

MQC サービス定義

MQC サーバ機能の定義のうち、MQC サービス定義について説明します。

関連製品である TP1/Message Queue Access を使用する場合に作成します。

概要

MQC サービス定義は、OpenTP1 のシステムサービス定義の一つです。

MQC サービス定義は、OS のテキストエディタを使用して、テキストファイルとして作成します。MQC サービス定義の定義ファイル名（完全パス名）を次に示します。

```
$DCCONFPATH/mqc
```

機能

MQC サーバ機能の動作を定義します。

種類

MQC サービス定義の種類について、次の表に示します。

表 4-20 MQC サービス定義の種類

形式	オプション	定義内容	指定値
mqcenv (MQC リスナサーバ通信環境定義) 指定数：1	-p	自システムのポート番号	～<符号なし整数> ((5001～65535))
	-s	自システムのサービス名	～<1～32 文字の識別子 (ハイフン (-) を含む) >
	-e	終了処理監視タイマ値	～<符号なし整数> ((10～65535)) 《180》 (単位：秒)
	-r	要求受信監視タイマ値	～<符号なし整数> ((0, 10～8191)) 《180》 (単位：秒)
	-n	継続セグメント受信監視タイマ値	～<符号なし整数> ((10～8191)) 《30》 (単位：秒)
	-g	MQC ゲートウェイサーバ監視タイマ値	～<符号なし整数> ((10～8191)) 《30》 (単位：秒)

形式	オプション	定義内容	指定値
mqcenv (MQC リスナサーバ通信環境定義) 指定数：1	-c	MQC コネクション確立完了監視 タイマ値	～<符号なし整数> ((10～8191))《30》 (単位：秒)
	-k	回線監視	yes no ～《no》
	-i	トランザクションブランチ終了監視 タイマ値	～<符号なし整数> ((0～65535))《180》 (単位：秒)
mqcgwpsnam (MQC ゲートウェイサーバ名定義) 指定数：1～256*	-n	MQC ゲートウェイサーバ名	～<1～8文字の識別子>
	-p	自システムのポート番号	～<符号なし整数> ((5001～65535))
	-s	自システムのサービス名	～<1～32文字の識別子(ハイフン (-)を含む)>

注※

次に示す値の合計数を指定してください。

- MQC クライアント機能が動作するマシンの環境変数に DCMQCEXPTRN=N を設定しているか、または非 XA インタフェースの場合、キューマネージャに同時接続するクライアントアプリケーション数
- MQC クライアント機能が動作するマシンの環境変数に DCMQCEXPTRN=Y を設定している場合、トランザクションオブジェクトを結合したアプリケーションの数とトランザクションマネージャの数の合計数
- MQC クライアントが TP1/Server Base と連携して XA インタフェース接続を行っている場合、MQC クライアントが連携している TP1/Server Base のトランザクションサービス定義の trn_recovery_process_count オペランドの指定値に 1 を加算した値
- MQC クライアントが JMS インタフェース機能を使用している場合、次の条件で該当するものの合計値

条件	値
コネクションプーリング機能を使用する	リソースアダプタのプロパティ定義に指定した [property-name] の [MaxPoolSize] の値+ 1
コネクションプーリング機能を使用しない	全アプリケーション数+ 1
MDB キュー監視機能を使用する	各 MDB アプリケーションのプロパティ定義に指定した [pooled-instance] の [maximum] の値の総和

mqcenv (MQC リスナサーバ通信環境定義)

形式

```
mqcenv  {-p 自システムのポート番号 | -s 自システムのサービス名}
        [-e 終了処理監視タイマ値]
        [-r 要求受信監視タイマ値]
        [-n 継続セグメント受信監視タイマ値]
        [-g MQCゲートウェイサーバ監視タイマ値]
        [-c MQCコネクション確立完了監視タイマ値]
        [-k 回線監視]
        [-i トランザクションブランチ終了監視タイマ値]
```

機能

MQC クライアント機能と MQC リスナサーバ間の通信に関する環境を定義します。

mqcenv 定義コマンドは、MQC サービス定義に一つだけ指定してください。省略した場合は、MQC サーバ機能を使用できません。重複して指定した場合は、MQC リスナサーバの開始処理でエラーになります。

オプション

● -p 自システムのポート番号

～<符号なし整数> ((5001～65535))

リスナ要求を受信する自システムのポート番号を指定します。この指定値は、MQC クライアント機能とのコネクション確立のために使用されます。ポート番号は、OS が自動割り当てするポート番号を含めて、自ノード内で重複して指定できません。自システムのポート番号が、ほかの通信プログラムで使用されているポート番号と重複した場合、KFCA30831-E メッセージを出力して MQC リスナサーバが異常終了することがあります。OS が自動割り当てするポート番号については、OS のマニュアルを参照してください。

-s オプションを指定した場合、この指定値は省略できます。ただし、省略する場合は、あらかじめサービス名ファイルにサービス名を登録しておく必要があります。

● -s 自システムのサービス名

～<1～32 文字の識別子 (ハイフン(-)を含む) >

自システムのサービス名を指定します。この指定値は、MQC クライアント機能とのコネクション確立のために使用されます。サービス名は自ノード内で重複して指定できません。サービス名は、サービス名ファイルに次の形式で登録します。

サービス名	ポート番号/TCP
-------	-----------

また、-p オプションの指定がある場合、この指定値は省略できます。

● -e 終了処理監視タイマ値

～<符号なし整数>((10~65535))《180》(単位：秒)

正常終了、計画停止 A、または計画停止 B の場合、MQC クライアント機能の切断を時間監視します。タイムアウトが発生した場合、送受信中の処理を中断し、MQC クライアント機能との接続は強制解放され、MQC リスナサーバが KFCA30830-E メッセージを出力して終了するか、または MQC ゲートウェイサーバが KFCA30840-E メッセージを出力して終了します。

● -r 要求受信監視タイマ値

～<符号なし整数>((0, 10~8191))《180》(単位：秒)

次の範囲について時間監視を行います。

- MQC コネクション確立応答送信から、API 実行要求受信まで
- API 結果応答（またはトランザクション決着結果応答）から、API 実行要求（またはトランザクション決着要求）まで

0 を指定した場合は、時間監視をしません。

タイムアウトが発生した場合、MQC リスナサーバが KFCA30835-E メッセージを出力して待機状態になるか、または MQC ゲートウェイサーバが KFCA30845-E メッセージを出力して待機状態になります。

● -n 継続セグメント受信監視タイマ値

～<符号なし整数>((10~8191))《30》(単位：秒)

MQC クライアント機能からの API 実行要求がセグメント分割される場合に、継続セグメントを受信する時間を監視します。

タイムアウトが発生した場合、MQC リスナサーバが KFCA30834-E メッセージを出力して待機状態になるか、または MQC ゲートウェイサーバが KFCA30844-E メッセージを出力して待機状態になります。

● -g MQC ゲートウェイサーバ監視タイマ値

～<符号なし整数>((10~8191))《30》(単位：秒)

クライアントアプリケーションが MQCONN 命令で呼び出してから、MQC ゲートウェイサーバに接続するまでの時間を監視します。

タイムアウトが発生した場合、MQC ゲートウェイサーバが KFCA30852-E メッセージを出力して待機状態になります。

● -c MQC コネクション確立完了監視タイマ値

～<符号なし整数>((10~8191))《30》(単位：秒)

TCP/IP コネクションの確立から、MQC コネクションを確立するまでの時間を監視します。

タイムアウトが発生した場合、MQC リスナサーバが KFCA30836-E メッセージを出力して待機状態になるか、または MQC ゲートウェイサーバが KFCA30853-E メッセージを出力して待機状態になります。

● -k 回線監視

~ {yes | no} 《no》

TCP/IP ソケットの回線監視 (keepalive) 機能を使用するかどうかを指定します。

- yes

TCP/IP ソケットの回線監視 (keepalive) 機能を使用します。

- no

TCP/IP ソケットの回線監視 (keepalive) 機能を使用しません。

回線監視 (keepalive) 機能とは、接続している TCP/IP コネクションに一定間隔メッセージの転送が発生しない場合に TCP/IP が MQC との接続を確認するデータを自動で転送する機能のことです。この機能を使用すると MQC との回線障害を検出しやすくなります。ただし、TCP/IP レベルで回線状態確認のためのデータが転送されるため、回線の使用量が増加します。

● -i トランザクションブランチ終了監視タイマ値

~ <符号なし整数> ((0~65535)) 《180》 (単位：秒)

開始中のトランザクションに対して xa_end 処理が完了してから xa_prepare, xa_commit または xa_rollback が要求されるまでの時間を指定します。指定した時間を過ぎても xa_prepare, xa_commit または xa_rollback が要求されない場合は、同期点処理へ移行状態 (xa_end 済み状態) のトランザクションを強制的にロールバックします。0 を指定した場合はトランザクションブランチの終了時間監視をしません。

mqcgwpsnam (MQC ゲートウェイサーバ名定義)

形式

```
mqcgwpsnam  -n MQCゲートウェイサーバ名  
             {-p 自システムのポート番号 | -s 自システムのサービス名}
```

機能

MQC ゲートウェイサーバ名に対応する API 実行要求受信用のポート番号を定義します。mqcgwpsnam 定義コマンドは、MQC サービス定義に必ず 1 個以上指定してください。

コマンド引数

● -n MQC ゲートウェイサーバ名

～< 1～8 文字の識別子 >

MQC ゲートウェイサーバ名を指定します。

● -p 自システムのポート番号

～< 符号なし整数 > ((5001～65535))

MQC ゲートウェイサーバ名に対応する API 実行要求受信用の自システムのポート番号を指定します。ポート番号は、OS が自動割り当てするポート番号を含めて、自ノード内で重複して指定できません。自システムのポート番号が、ほかの通信プログラムで使用されているポート番号と重複した場合、KFCA30841-E メッセージを出力して MQC ゲートウェイサーバが異常終了することがあります。OS が自動割り当てするポート番号については、OS のマニュアルを参照してください。

-s オプションを指定した場合、この指定値は省略できます。ただし、省略する場合は、あらかじめサービス名ファイルにサービス名を登録しておく必要があります。

● -s 自システムのサービス名

～< 1～32 文字の識別子 (ハイフン(-)を含む) >

MQC ゲートウェイサーバ名に対応する API 実行要求受信用の自システムのサービス名を指定します。サービス名は自ノード内で重複して指定できません。サービス名は、サービス名ファイルに次の形式で登録します。

サービス名	ポート番号/TCP
-------	-----------

-p オプションを指定した場合、この指定値は省略できます。

注意事項

mqcgwpsnam 定義コマンドの数を変更した場合は、必ず MQA サービス定義の mqa_mqc_clientnum の値を見直してください。

MQC ゲートウェイサーバユーザサービス定義

MQC サーバ機能の定義のうち、MQC ゲートウェイサーバユーザサービス定義について説明します。

関連製品である TP1/Message Queue Access を使用する場合に作成します。

概要

mqcgwp を OpenTP1 のユーザサーバとして開始するため、開始する MQC ゲートウェイサーバごとに MQC ゲートウェイサーバユーザサービス定義が必要です。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

MQC ゲートウェイサーバユーザサービス定義は、OS のテキストエディタを使用して、テキストファイルとして作成します。MQC ゲートウェイサーバユーザサービス定義の定義ファイル名を次に示します。

```
$DCCONFPATH/MQCゲートウェイサーバ名
```

MQC ゲートウェイサーバ名には、mqcgwpsnam 定義コマンドの -n オプションの指定値を指定します。

形式

```
set module = "実行形式プログラム名"  
set atomic_update = Y | N  
set receive_from = none  
set auto_restart = Y  
set critical = N  
[set trn_rm_open_close_scope = process]  
[set nice = プロセスの優先順位]
```

機能

MQC ゲートウェイサーバをユーザサーバとして開始するための情報を定義します。

オプション

● module = "実行形式プログラム名"

～< 1～14 文字の識別子>

実行形式プログラム名には mqcgwp を指定してください。

● atomic_update=Y | N

MQC ゲートウェイサーバプロセスをトランザクションの範囲に入れるかどうかを指定します。

このオペランドの指定値には、必ず N を指定してください。

N を指定すると、MQC ゲートウェイサーバプロセスはトランザクションの範囲に入れます。トランザクションとして動作することを考慮し、トランザクションサービス定義の trn_tran_process_count オペラ

ンドなどの関連する定義を見積もってください。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

なお、N 以外を指定すると、トランザクションの範囲に入れません。このため、MQC ゲートウェイサーバの起動時にアポードコード Mqcgw03 を出力し、MQC ゲートウェイサーバが異常終了します。

● receive_from = none

- none

スケジューラサービスのスケジューラキュー、UNIX ドメイン、およびインターネットドメインのどれも使用しません。

● auto_restart = Y

- Y

UAP が異常終了した場合、プロセスを再起動します。

● critical = N

- N

UAP が異常終了した場合、OpenTP1 システムを続行します。

● trn_rm_open_close_scope = process

\$DCCONFPATH/trn (トランザクションサービス定義) に次のように設定されている場合、指定してください。

```
set trn_rm_open_close_scope = transaction
```

● nice = プロセスの優先順位

~<符号なし整数> ((0~39))

サービスグループプロセスの優先順位を指定します。この値が大きくなると CPU の優先順位は低くなります。

注意事項

MQC ゲートウェイサーバユーザサービス定義は SUP の形式で設定してください。

トランザクションサービス定義

MQC サーバ機能の定義のうち、トランザクションサービス定義について説明します。

トランザクションサービス定義については、マニュアル「OpenTP1 システム定義」を参照してください。

概要

MQC サーバ機能で、TP1/Message Queue Access からグローバルトランザクションを使用するには、トランザクションサービス定義に MQC サーバ機能の定義を追加する必要があります。トランザクションサービス定義の定義ファイル名を次に示します。

```
$DCCONFPATH/trn
```

形式

```
[set trn_crm_use = Y | N]
```

機能

MQC サーバ機能を開始するための情報を定義します。

オペランド

● trn_crm_use=Y | N

～《N》

このノードの MQC サーバ機能で、TP1/Message Queue Access からグローバルトランザクションを使用するかどうかを指定します。TP1/Message Queue Access からグローバルトランザクションを使用する場合、必ず Y を指定してください。

- Y
TP1/Message Queue Access からグローバルトランザクションを使用します。
- N
TP1/Message Queue Access からグローバルトランザクションを使用しません。

注意事項

trn_crm_use オペランドに Y を指定しないで TP1/Message Queue Access からグローバルトランザクションを使用した場合の処理の流れを次に示します。

1. TP1/Message Queue Access からグローバルトランザクションを開始します。
2. MQPMO_MQSYNCPOINT, または MQGMO_SYNCPOINT を指定して, MQPUT 命令, MQPUT1 命令, および MQGET 命令を発行します。

3. 理由コード MQRC_UNEXPECTED_ERROR が TP1/Message Queue Access へ返却されます。また、MQC サーバ側に KFCA00910-E メッセージが出力されます。

システム構成例

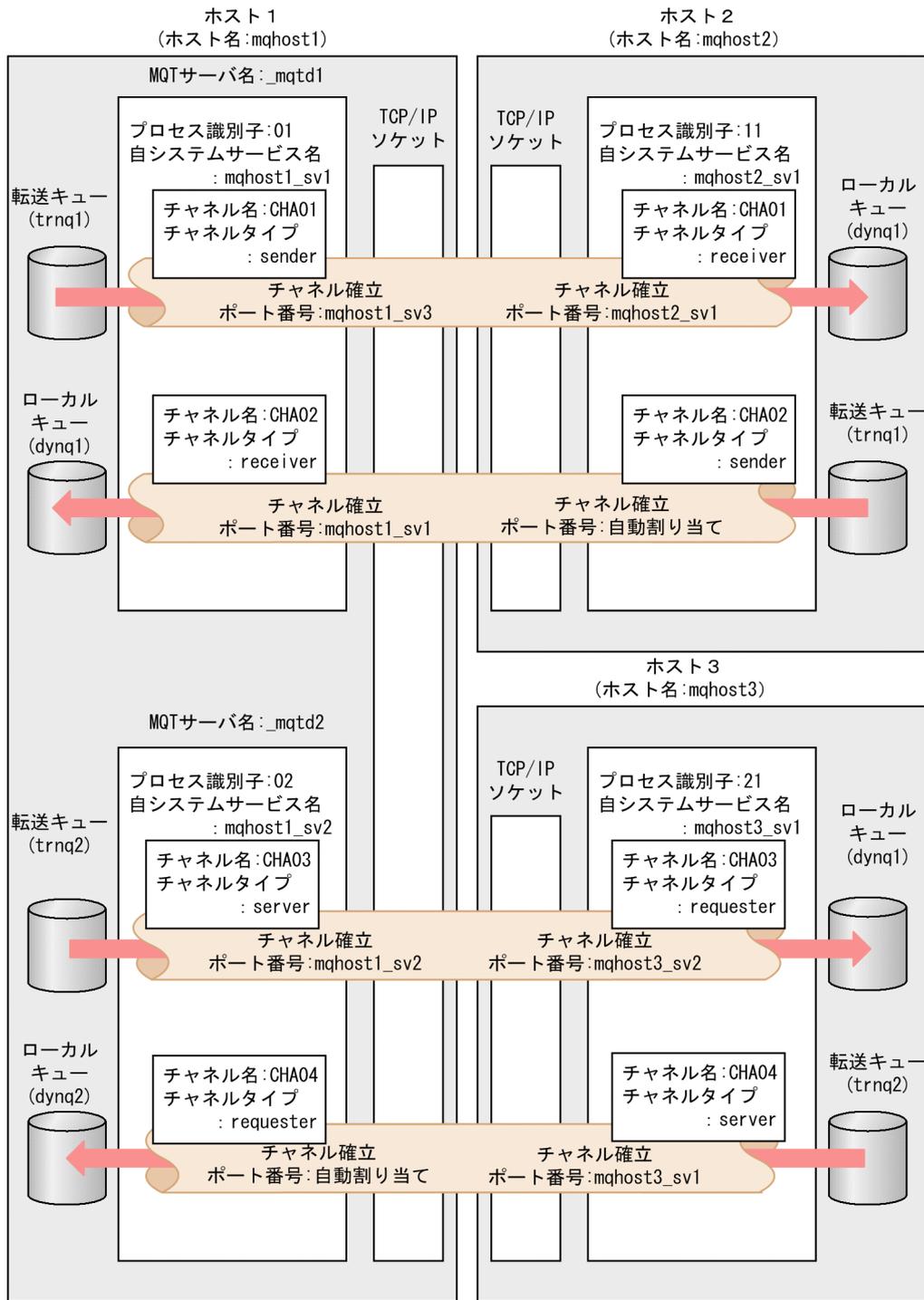
製品が提供するサンプルコードを使用したシステム構成例について説明します。

サンプルコードのシステム構成

製品が提供するサンプルコードは、ホスト 1 とホスト 2、およびホスト 1 とホスト 3 とでメッセージを送受信するシステム構成です。

サンプルコードのシステム構成について、次の図に示します。

図 4-4 サンプルコードのシステム構成



このシステム構成でのホスト 1, ホスト 2, およびホスト 3 の定義コーディング例を次に示します。

- MQA サービス定義のコーディング例

```
#
# Message Queue Access service definition(mqa)
#

set mqa_max_quenum = 20
set mqa_max_grpnum = 10
```

```

set mqa_max_filnum = 10
set mqa_max_msg_recnum = 5
set mqa_ioproc_num = 3

# queue file group definition
mqaquegrp -g sendgroup1 -n 256 ¥
           /BeTRAN/examples/mq/mqfile1/sendqf1
mqaquegrp -g sendgroup2 -n 256 ¥
           /BeTRAN/examples/mq/mqfile1/sendqf2
mqaquegrp -g recvgroup1 -n 256 ¥
           /BeTRAN/examples/mq/mqfile2/recvqf1
mqaquegrp -g recvgroup2 -n 256 ¥
           /BeTRAN/examples/mq/mqfile2/recvqf2
mqaquegrp -g initgroup1 -n 128 ¥
           /BeTRAN/examples/mq/mqfile3/initqf1

# local definition of remote queue
mqaremque -r dynq1 -m qmgr2 -x trnq1 rmtq1
mqaremque -r dynq1 -m qmgr3 -x trnq2 rmtq2

# process definition
mqaprcdef -p PRC1 -t 15 -e ENV1 -u USER1 prc1

# queue manager definition
mqaquemgr -n qmgr1 -q QMGR1 -t 0 -p 9 ¥
           -h 128 -l 5000000 -s sync

#mqt definition
mqamqtnam _mqtd1 mqttcp 2
mqamqtnam _mqtd2 mqttcp 2

```

- ホスト 1 (_mqtd1) の MQT 通信構成定義のコーディング例

```

#
# MQT communication configuration definition, tcp definition
#

# tcp configuration definition
mqttcp -i iniq1 ¥
        -p tcp ¥
        -g "rcvbuf = 3" ¥
        -r "servname = mqhost1_sv1" ¥
        -v "itim = 60 ¥
            etim = 40 ¥
            rtim = 60" ¥
        -t "disk = yes ¥
            bufcnt = 100 ¥
            trcnt = 3" ¥
        -z event

# sender channel definition
mqtalccha -c CHA01 ¥
          -y "type = sender" ¥
          -j 50 ¥
          -q trnq1 ¥
          -m "maxmsg = 8192 ¥
              maxseg = 4096" ¥
          -w 999999999 ¥

```

```

-g "sndbuf = 1                                     ¥
   buftype=msg"                                   ¥
-i auto                                           ¥
-b "bretry = yes                                   ¥
   bretrycnt = 10                                 ¥
   bretryint = 60                                 ¥
   bretrylg = yes                                 ¥
   bretrylgcnt = 99999999                        ¥
   bretrylgint = 1200                             ¥
   bretrymcp = yes"                               ¥
-r "servname = mqhost1_sv3"                       ¥
-o "ohostname = mqhost2                           ¥
   oservname = mqhost2_sv1"                       ¥
-v "tim1 = 30                                     ¥
   tim2 = 30                                     ¥
   dtimefct = yes                                 ¥
   dtim = 60                                     ¥
   vretry = yes                                   ¥
   htim = 15                                     ¥
   btim = 0"                                       ¥
-t "tretrycnt = 10                                ¥
   tretryint = 1"                                 ¥
-s "npmspeed = fast"                              ¥
-z event                                           ¥
-h 'to mqhost2'
mqtalced

# receiver channel definition
mqtalcccha -c CHA02                                ¥
           -y "type = receiver"                  ¥
           -j 50                                  ¥
           -m "maxmsg = 8192                      ¥
              maxseg = 4096"                     ¥
           -w 99999999                            ¥
           -g "rcvbuf = 2                         ¥
              buftype = msg"                      ¥
           -i auto                                 ¥
           -v "tim3 = 10                          ¥
              htim = 15                           ¥
              mtim = 10"                          ¥
           -s "npmspeed = fast"                   ¥
           -f "adoptchk = qmgr"                   ¥
           -h 'from mqhost2'
mqtalced

```

- ホスト 1 (_mqtd2) の MQT 通信構成定義のコーディング例

```

#
# MQT communication configuration definition, tcp definition
#

# tcp configuration definition
mqttcp -p tcp                                     ¥
       -g "rcvbuf = 3"                             ¥
       -r "servname = mqhost1_sv2"                 ¥
       -v "etim = 40                                ¥
          rtim = 60"                                ¥
       -t "disk = yes"                              ¥

```

```

        bufcnt = 100
        trccnt = 3"
¥

# server channel definition
mqtalcccha -c CHA03
¥
        -y "type = server"
¥
        -j 50
¥
        -q trnq2
¥
        -m "maxmsg = 8192
¥
            maxseg = 4096"
¥
        -w 999999999
¥
        -g "sndbuf = 1
¥
            buftype=msg"
¥
        -i auto
¥
        -v "tim2 = 30
¥
            dtimefct = yes
¥
            dtim = 60
¥
            htim = 15
¥
            btim = 0"
¥
        -s "npmspeed = fast"
¥
        -z event
¥
        -f "adoptchk = qmgr"
¥
        -h 'from mqhost3'
¥

mqtalced

# requester channel definition
mqtalcccha -c CHA04
¥
        -y "type = requester"
¥
        -j 50
¥
        -m "maxmsg = 8192
¥
            maxseg = 4096"
¥
        -w 999999999
¥
        -g "rcvbuf = 2
¥
            buftype = msg"
¥
        -i auto
¥
        -b "bretry = yes
¥
            bretrycnt = 10
¥
            bretryint = 60
¥
            bretrylg = yes
¥
            bretrylgcnt = 999999999
¥
            bretrylgint = 1200
¥
            bretrymcp = yes"
¥
        -o "ohostname = mqhost3
¥
            oservername = mqhost3_sv1"
¥
        -v "tim1 = 30
¥
            tim3 = 10
¥
            vretry = yes
¥
            htim = 15
¥
            mtim = 10"
¥
        -t "tretrycnt = 10
¥
            tretryint = 1"
¥
        -s "npmspeed = fast"
¥
        -h 'to mqhost3'
¥

mqtalced

```

- ホスト 2 の MQT 通信構成定義のコーディング例

```

#
# MQT communication configuration definition, tcp definition
#

# tcp configuration definition
mqttcp -i iniq1                ¥
      -p tcp                    ¥
      -g "rcvbuf = 3"          ¥
      -r "servname = mqhost2_sv1" ¥
      -v "itim = 60            ¥
          etim = 40            ¥
          rtim = 60"          ¥
      -t "disk = yes           ¥
          bufcnt = 100         ¥
          trccnt = 3"         ¥
      -z event

# receiver channel definition
mqtalcccha -c CHA01            ¥
          -y "type = receiver" ¥
          -j 50                 ¥
          -m "maxmsg = 8192     ¥
              maxseg = 4096"   ¥
          -w 999999999         ¥
          -g "rcvbuf = 2       ¥
              buftype = msg"   ¥
          -i auto              ¥
          -v "tim3 = 10        ¥
              htim = 15        ¥
              mtim = 10"       ¥
          -s "npmspeed = fast" ¥
          -f "adoptchk = qmgr" ¥
          -h 'from mqhost1'

mqtalced

# sender channel definition
mqtalcccha -c CHA02            ¥
          -y "type = sender"   ¥
          -j 50                 ¥
          -q trnq1             ¥
          -m "maxmsg = 8192     ¥
              maxseg = 4096"   ¥
          -w 999999999         ¥
          -g "sndbuf = 1       ¥
              buftype=msg"     ¥
          -i auto              ¥
          -b "bretry = yes     ¥
              bretrycnt = 10   ¥
              bretryint = 60   ¥
              bretrylg = yes   ¥
              bretrylgcnt = 999999999 ¥
              bretrylgint = 1200 ¥
              bretrymcp = yes" ¥
          -o "ohostname = mqhost1 ¥
              oservname = mqhost1_sv1" ¥
          -v "tim1 = 30        ¥
              tim2 = 30        ¥

```

```

        dtimefct = yes           ¥
        dtim = 60                ¥
        vretry = yes             ¥
        htim = 15                 ¥
        btim = 0"                ¥
    -t "tretrycnt = 10           ¥
        tretryint = 1"          ¥
    -s "npmspeed = fast"        ¥
    -z event                      ¥
    -h 'to mqhost1'             ¥
mqtalced

```

- ホスト3のMQT通信構成定義のコーディング例

```

#
# MQT communication configuration definition, tcp definition
#

# tcp configuration definition

mqttcp -p tcp                    ¥
    -g "rcvbuf = 3"              ¥
    -r "servname = mqhost3_sv1"  ¥
    -v "etim = 40                ¥
        rtim = 60"              ¥
    -t "disk = yes               ¥
        bufcnt = 100            ¥
        trccnt = 3"            ¥

# requester channel definition
mqtalcccha -c CHA03              ¥
    -y "type = requester"        ¥
    -j 50                         ¥
    -m "maxmsg = 8192            ¥
        maxseg = 4096"          ¥
    -w 999999999¥                 ¥
    -g "rcvbuf = 2               ¥
        buftype = msg"          ¥
    -i auto                       ¥
    -b "bretry = yes              ¥
        bretrycnt = 10           ¥
        bretryint = 60           ¥
        bretrylg = yes           ¥
        bretrylgcnt = 999999999 ¥
        bretrylgint = 1200       ¥
        bretrymcp = yes"         ¥
    -r "servname = mqhost3_sv2"  ¥
    -o "ohostname = mqhost1      ¥
        oservname = mqhost1_sv2" ¥
    -v "tim1 = 30                ¥
        tim3 = 10                ¥
        vretry = yes             ¥
        htim = 15                 ¥
        mtim = 10"              ¥
    -t "tretrycnt = 10           ¥
        tretryint = 1"          ¥
    -s "npmspeed = fast"        ¥
    -h 'to mqhost1'             ¥

```

```

mqtalced

# server channel definition
mqtalcccha -c CHA04                ¥
           -y "type = server"      ¥
           -j 50                    ¥
           -q trnq1                 ¥
           -m "maxmsg = 8192        ¥
              maxseg = 4096"       ¥
           -w 999999999             ¥
           -g "sndbuf = 1           ¥
              buftype=msg"         ¥
           -i auto                   ¥
           -v "tim2 = 30            ¥
              dtimfct = yes        ¥
              dtim = 60           ¥
              htim = 15           ¥
              btim = 0"           ¥
           -s "npmspeed = fast"     ¥
           -z event                  ¥
           -f "adoptchk = qmgr"     ¥
           -h 'from mqhost1'       ¥

mqtalced

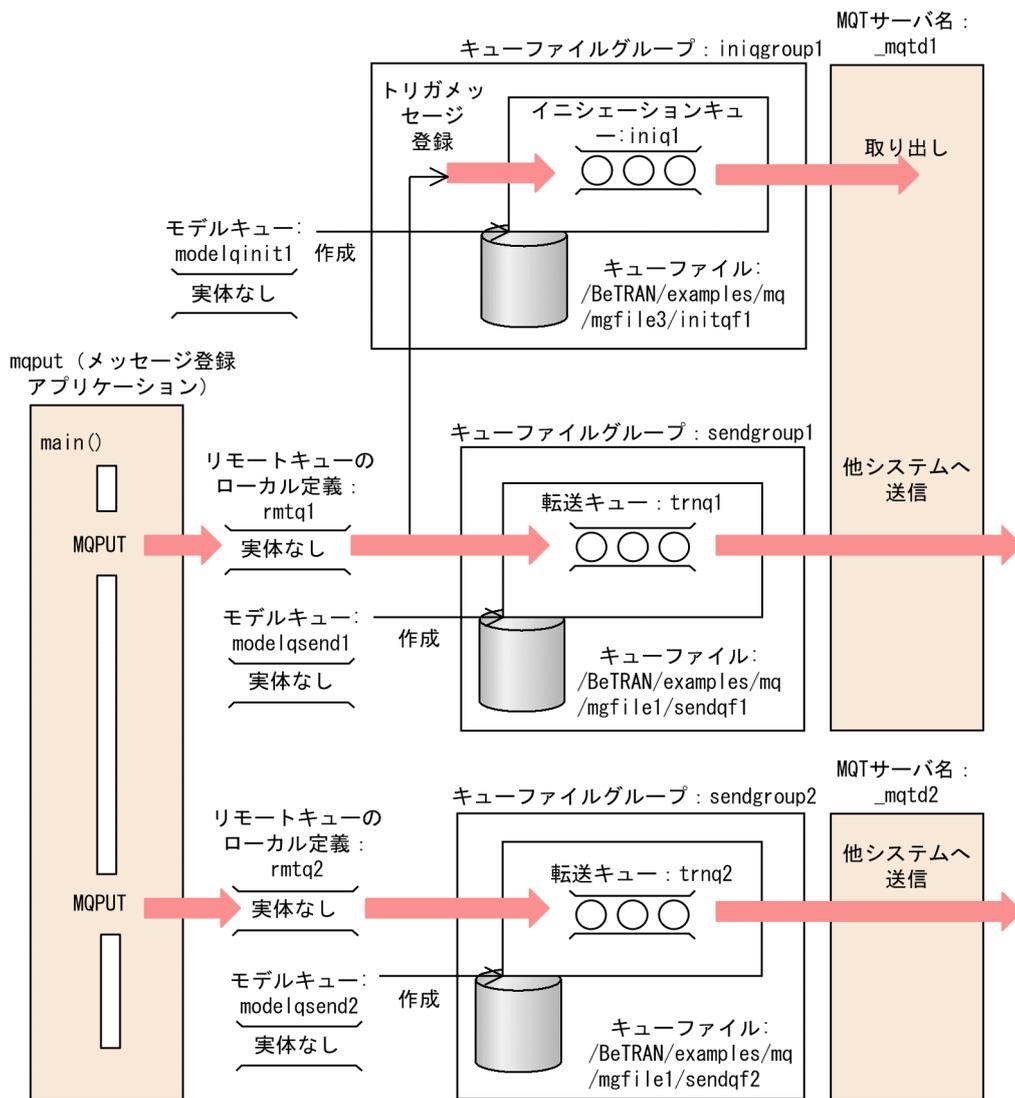
```

メッセージを登録するアプリケーションの例

ホスト 1 上で動作するアプリケーション mqput は、転送キュー trnq1 および trnq2 にメッセージを登録します。転送キュー trnq1 に登録されたメッセージはチャンネル CHA01 を経由してホスト 2 に送信されます。転送キュー trnq2 に登録されたメッセージはチャンネル CHA03 を経由してホスト 3 に送信されます。

メッセージを登録するアプリケーションの例について、次の図に示します。

図 4-5 メッセージを登録するアプリケーションの例



アプリケーション `mqput` のコーディング例 (C 言語) を次に示します。

```

0010 /*****
0020 **  mqput.c
0030 **  specification
0040 **  functions = main()
0050 *****/
0060
0070 /*****
0080 **  name = main()
0090 **  func = main function of UAP
0100 **  (1)start UAP(output start message)
0110 **  (2)MQCONN(Connect queue manager)
0120 **  (3)MQOPEN(Open queue)
0130 **  (4)MQPUT(Put message)
0140 **  (5)MQCLOSE(Close queue)
0150 **  (6)MQDISC(Disconnect queue manager)
0160 **  (7)finish UAP(output end message)
0170 **  arg = nothing
0180 **  return = void

```

```

0190 ** note =
0200 *****/
0210 #ifdef TP1MQ_NT
0220 #include <windows.h>
0230 #endif
0240 #include <stdio.h>
0250 #include <string.h>
0260 #include <stdlib.h>
0270 #include <cmqc.h>
0280 #include <dcrpc.h>
0290 #include <dcadm.h>
0300 #include <dctrn.h>
0310
0320 static char *Mid = "%Z% %M%(%E% %U%)-%I% " ;
0330
0340 #define PUT_DATA "***** sample put data0 *****"
0350
0360 void main()
0370 {
0380     MQCHAR48 QueueManager=" "; /* queue manager name */
0390     MQCHAR message_data[100]; /* message data */
0400     MQHCONN qm_handle; /* connection handle */
0410     MQHOBJ que_handle[2]; /* object handle */
0420     MQLONG comp_code; /* completion code */
0430     MQLONG reason; /* reason code */
0440     MQLONG buffer_length; /* buffer length */
0450     MQLONG rcd; /* return code */
0460     MQOD obj_desc0 = { MQOD_DEFAULT };
0470 /* message descriptor */
0480     MQOD obj_desc1 = { MQOD_DEFAULT };
0490 /* message descriptor */
0500     MQMD msg_desc = { MQMD_DEFAULT };
0510 /* object descriptor */
0520     MQPMO put_options = { MQPMO_DEFAULT };
0530 /* put message options */
0540     int flags = 0;
0550     int loop_cnt;
0560     char que_name[][49] =
0570         {"rmtq1", "rmtq2"}; /* que name */
0580
0590 /* Start of UAP */
0600     rcd = dc_rpc_open(DCNOFLAGS);
0610     if (rcd != DC_OK) {
0620         printf(
0630             "mqput_c : Failed at dc_rpc_open. CODE = %d ¥n",
0640             rcd);
0650         goto PROG_END;
0660     }
0670
0680 /* post the completion of user process start
0690                                     processing */
0700     rcd = dc_adm_complete(DCNOFLAGS);
0710     if (rcd != DC_OK) {
0720         printf(
0730             "mqput_c : Failed at dc_adm_complete. CODE = %d ¥n",
0740             rcd);
0750         goto PROG_END;
0760     }

```

```

0770
0780 /* Connect queue manager */
0790 MQCONN(QueueManager, &qm_handle, &comp_code, &reason);
0800 if (comp_code != MQCC_OK) {
0810     printf("mqput_c : Failed at MQCONN. CODE = %ld ¥n",
0820         reason);
0830     rcd = reason;
0840     goto PROG_END;
0850 }
0860
0870 /* Open object */
0880 strcpy(obj_desc0.ObjectName, que_name[0]);
0890 MQOPEN(qm_handle, &obj_desc0, MQOO_OUTPUT,
0900     &que_handle[0], &comp_code, &reason);
0910 if (comp_code != MQCC_OK) {
0920     printf("mqput_c : Failed at MQOPEN. CODE = %ld¥n",
0930         reason);
0940     rcd = reason;
0950     goto MQ_END1;
0960 }
0970
0980 strcpy(obj_desc1.ObjectName, que_name[1]);
0990 MQOPEN(qm_handle, &obj_desc1, MQOO_OUTPUT,
1000     &que_handle[1], &comp_code, &reason);
1010 if (comp_code != MQCC_OK) {
1020     printf("mqput_c : Failed at MQOPEN. CODE = %ld¥n",
1030         reason);
1040     rcd = reason;
1050     flags = 1;
1060 }
1070
1080 loop_cnt = 0;
1090 do {
1100     /* begin transaction */
1110     rcd = dc_trn_begin();
1120     if(rcd != DC_OK) {
1130         printf(
1140             "mqput_c : Failed at dc_trn_begin. CODE = %d ¥n",
1150             rcd);
1160         goto MQ_END2;
1170     }
1180
1190     /* Put message */
1200     put_options.Options = MQPMO_SYNCPOINT;
1210     msg_desc.Persistence = MQPER_PERSISTENT;
1220     strcpy(message_data, PUT_DATA);
1230     buffer_length = strlen(message_data);
1240     MQPUT(qm_handle, que_handle[loop_cnt], &msg_desc,
1250         &put_options, buffer_length,
1260         (PMQBYTE)message_data, &comp_code, &reason);
1270     if (comp_code == MQCC_FAILED) {
1280         printf("mqput_c : Failed at MQPUT. CODE = %ld¥n",
1290             reason);
1300         rcd = dc_trn_unchained_rollback();
1310         if (rcd != DC_OK) {
1320             printf("mqput_c : Failed at dc_trn_unchained_rollback.
1330                 CODE = %d ¥n", rcd);
1340         }

```

```

1350     goto MQ_END2;
1360     }
1370
1380     /* commit transaction */
1390     rcd = dc_trn_unchained_commit();
1400     if (rcd != DC_OK) {
1410         printf(
1420 mqput_c : Failed at dc_trn_unchained_commit. CODE = %d ¥n,
1430         rcd);
1440     }
1450     loop_cnt++;
1460 } while(flags == 0 && loop_cnt < 2);
1470
1480 MQ_END2:
1490 /* Close object */
1500 loop_cnt = 0;
1510 do {
1520     MQCLOSE(qm_handle, &que_handle[loop_cnt], MQCO_NONE,
1530             &comp_code, &reason);
1540     if (comp_code != MQCC_OK) {
1550         printf("mqput_c : Failed at MQCLOSE. CODE = %ld¥n",
1560             reason);
1570     }
1580     loop_cnt++;
1590 } while (flags == 0 && loop_cnt < 2);
1600
1610 MQ_END1:
1620 /* Disconnect queue manager */
1630 MQDISC(&qm_handle, &comp_code, &reason);
1640 if (comp_code != MQCC_OK) {
1650     printf("mqput_c : Failed at MQDISC. CODE = %ld ¥n",
1660         reason);
1670 }
1680 /* Termination of UAP */
1690 dc_rpc_close(DCNOFLAGS);
1700
1710 PROG_END:
1720 return;
1730 }

```

アプリケーション mqput のコーディング例 (COBOL 言語) を次に示します。

```

0010 *
0020 *****
0030 *      mqput_cb      *
0040 *****
0050 ** name = main()
0060 ** func = main function of UAP
0070 ** (1)start UAP(output start message)
0080 ** (2)MQCONN(Connect queue manager)
0090 ** (3)MQOPEN(Open queue)
0100 ** (4)MQPUT(Put message)
0110 ** (5)MQCLOSE(Close queue)
0120 ** (6)MQDISC(Disconnect queue manager)
0130 ** (7)finish UAP(output end message)
0140 *****
0150 *

```

```

0160 IDENTIFICATION DIVISION.
0170 *
0180 PROGRAM-ID. MAIN.
0190 *
0200 *****
0210 *      set data area      *
0220 *****
0230 *
0240 DATA DIVISION.
0250 WORKING-STORAGE SECTION.
0260 01 RPC-OP-ARG.
0270 02 REQUEST      PIC      X(8) VALUE 'OPEN      '.
0280 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0290 02 FILLER       PIC      X(3).
0300 02 FLAGS        PIC      S9(9) COMP VALUE ZERO.
0310 *
0320 01 RPC-CL-ARG.
0330 02 REQUEST      PIC      X(8) VALUE 'CLOSE     '.
0340 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0350 02 FILLER       PIC      X(3).
0360 *
0370 01 ADM-COMP.
0380 02 REQUEST      PIC      X(8) VALUE 'COMPLETE'.
0390 02 RET-VALUE    PIC      X(5).
0400 02 FILLER       PIC      X(3).
0410 02 DTZRADM     PIC      S9(9) COMP VALUE ZERO.
0420 *
0430 01 TRN-BG-ARG.
0440 02 REQUEST      PIC      X(8) VALUE 'BEGIN     '.
0450 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0460 01 TRN-CL-ARG.
0470 02 REQUEST      PIC      X(8) VALUE 'U-COMMIT'.
0480 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0490 01 TRN-RB-ARG.
0500 02 REQUEST      PIC      X(8) VALUE 'U-ROLL    '.
0510 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0520 *
0530 01 MQ-OBJECT    PIC      X(48) VALUE SPACE.
0540 01 MQ-MGRNAME   PIC      X(48) VALUE SPACE.
0550 01 MQ-HCONN     PIC      S9(9) BINARY.
0560 01 MQ-COMPCODE  PIC      S9(9) BINARY.
0570 01 MQ-REASON    PIC      S9(9) BINARY.
0580 01 MQ-HOBJ1     PIC      S9(9) BINARY.
0590 01 MQ-HOBJ2     PIC      S9(9) BINARY.
0600 01 MQ-OPTIONS  PIC      S9(9) BINARY.
0610 01 MQ-BUFFLEN  PIC      S9(9) BINARY.
0620 01 MQ-DATALEN  PIC      S9(9) BINARY.
0630 01 MQ-PUT-BUFFER PIC      X(200) VALUE SPACE.
0640 01 MQ-FLAGS    PIC      S9(5) BINARY VALUE 0.
0650 *
0660 01 PUT-DATA     PIC      X(33) VALUE
0670      '***** sample put data *****'.
0680 *
0690 01 MQ-OBJECT-DESC.
0700 COPY CMQODV.
0710 *
0720 01 MQ-MESSAGE-DESCRIPTOR.
0730 COPY CMQMDV.

```

```

0740 *
0750 01 MQ-PUT-MESSAGE-OPTIONS.
0760 COPY CMQPMOV.
0770 *
0780 01 MQ-CONSTANTS.
0790 COPY CMQV SUPPRESS.
0800 *
0810 PROCEDURE DIVISION.
0820 *
0830 *****
0840 *          RPC-OPEN(Start of UAP)          *
0850 *****
0860 *
0870 *****
0880 *          RPC-OPEN(Start of UAP)          *
0890 *****
0900 *
0910 CALL 'CBLDCRPC' USING RPC-OP-ARG.
0920 IF RET-VALUE OF RPC-OP-ARG NOT = '00000' THEN
0930   DISPLAY 'MQPUT_CB : RPC-OPEN FAILED. CODE = '
0940   RET-VALUE OF RPC-OP-ARG
0950   GO TO PROG-END
0960 END-IF.
0970 *
0980 *****
0990 *          ADM-COMPLETE(Report of termination of start *
1000 *          Processing of user server)          *
1010 *****
1020 *
1030 CALL 'CBLDCADM' USING ADM-COMP
1040 IF RET-VALUE OF ADM-COMP NOT = '00000' THEN
1050   DISPLAY 'MQPUT_CB : ADM-COMPLETE FAILED.CODE = '
1060   RET-VALUE OF ADM-COMP
1070   GO TO PROG-END
1080 END-IF.
1090 *
1100 *****
1110 *          Connect queue manager          *
1120 *****
1130 *
1140 CALL 'MQCONN' USING MQ-MGRNAME
1150 MQ-HCONN
1160 MQ-COMPCODE
1170 MQ-REASON
1180 IF MQ-COMPCODE NOT = MQCC-OK THEN
1190   DISPLAY 'MQPUT_CB : MQCONN FAILED. REASON CODE = '
1200   MQ-REASON
1210   GO TO PROG-END
1220 END-IF.
1230 *
1240 *****
1250 *          Opening of object          *
1260 *****
1270 *
1280 MOVE 'rmtq1' TO MQ-OBJECT
1290 MOVE MQ-OBJECT TO MQOD-OBJECTNAME
1300 *
1310 MOVE MQOO-OUTPUT TO MQ-OPTIONS

```

```

1320 *
1330 CALL 'MQOPEN' USING MQ-HCONN
1340 MQOD
1350 MQ-OPTIONS
1360 MQ-HOBJ1
1370 MQ-COMPCODE
1380 MQ-REASON
1390 IF MQ-COMPCODE NOT = MQCC-OK THEN
1400     DISPLAY 'MQPUT_CB : MQOPEN FAILED. REASON CODE = '
1410     MQ-REASON
1420     GO TO MQ-END1
1430 END-IF
1440 *
1450 MOVE 'rmtq2' TO MQ-OBJECT.
1460 MOVE MQ-OBJECT TO MQOD-OBJECTNAME
1470 *
1480 MOVE MQOO-OUTPUT TO MQ-OPTIONS
1490 *
1500 CALL 'MQOPEN' USING MQ-HCONN
1510 MQOD
1520 MQ-OPTIONS
1530 MQ-HOBJ2
1540 MQ-COMPCODE
1550 MQ-REASON
1560 IF MQ-COMPCODE NOT = MQCC-OK THEN
1570     DISPLAY 'MQPUT_CB : MQOPEN FAILED. REASON CODE = '
1580     MQ-REASON
1590     MOVE 1 TO MQ-FLAGS
1600 END-IF.
1610 *
1620 *****
1630 *     begin transaction *
1640 *****
1650 *
1660 CALL 'CBLDCTRN' USING TRN-BG-ARG
1670 IF RET-VALUE OF TRN-BG-ARG NOT = '00000' THEN
1680     DISPLAY 'MQPUT_CB : TRN-BEGIN FAILED. CODE = '
1690     RET-VALUE OF TRN-BG-ARG
1700     GO TO MQ-END2
1710 END-IF.
1720 *
1730 *****
1740 *     Definition of putting of message *
1750 *****
1760 *
1770 MOVE MQPMO-SYNCPOINT TO MQPMO-OPTIONS
1780 MOVE MQPER-PERSISTENT TO MQMD-PERSISTENCE
1790 MOVE 33 TO MQ-BUFFLEN
1800 MOVE PUT-DATA TO MQ-PUT-BUFFER
1810 *
1820 *****
1830 *     Putting of message *
1840 *****
1850 *
1860 CALL 'MQPUT' USING MQ-HCONN
1870 MQ-HOBJ1
1880 MQMD
1890 MQPMO

```

```

1900 MQ-BUFFLEN
1910 MQ-PUT-BUFFER
1920 MQ-COMPCODE
1930 MQ-REASON
1940 IF MQ-COMPCODE NOT = MQCC-OK THEN
1950   DISPLAY 'MQPUT_CB : MQPUT FAILED. REASON CODE = '
1960 MQ-REASON
1970 CALL 'CBLDCTRN' USING TRN-RB-ARG
1980 IF RET-VALUE OF TRN-RB-ARG NOT = '00000'
1990   DISPLAY 'MQPUT_CB : TRN-ROLLBK FAILED. CODE = '
2000 RET-VALUE OF TRN-RB-ARG
2010 END-IF
2020   GO TO MQ-END2
2030 END-IF
2040 IF MQ-FLAGS = 0 THEN
2050   CALL 'MQPUT' USING MQ-HCONN
2060 MQ-HOBJ2
2070 MQMD
2080 MQPMO
2090 MQ-BUFFLEN
2100 MQ-PUT-BUFFER
2110 MQ-COMPCODE
2120 MQ-REASON
2130   IF MQ-COMPCODE NOT = MQCC-OK THEN
2140     DISPLAY 'MQPUT_CB : MQPUT FAILED. REASON CODE = '
2150     MQ-REASON
2160     CALL 'CBLDCTRN' USING TRN-RB-ARG
2170     IF RET-VALUE OF TRN-RB-ARG NOT = '00000'
2180       DISPLAY 'MQPUT_CB : TRN-ROLLBK FAILED. CODE = '
2190       RET-VALUE OF TRN-RB-ARG
2200     END-IF
2210     GO TO MQ-END2
2220   END-IF
2230 END-IF.
2240 *
2250 *****
2260 *       commit transaction                               *
2270 *****
2280 *
2290 CALL 'CBLDCTRN' USING TRN-CL-ARG
2300 IF RET-VALUE OF TRN-CL-ARG NOT = '00000' THEN
2310   DISPLAY 'MQPUT_CB : TRN-COMMIT FAILED.CODE = '
2320   RET-VALUE OF TRN-CL-ARG
2330   END-IF.
2340 *
2350 *****
2360 *       Closing of object                               *
2370 *****
2380 *
2390 MQ-END2.
2400 *
2410 MOVE MQCO-NONE TO MQ-OPTIONS
2420 *
2430 CALL 'MQCLOSE' USING MQ-HCONN
2440 MQ-HOBJ1
2450 MQ-OPTIONS
2460 MQ-COMPCODE
2470 MQ-REASON

```

```

2480 IF MQ-COMPCODE NOT = MQCC-OK THEN
2490 DISPLAY 'MQPUT_CB : MQCLOSE FAILED. REASON CODE = '
2500 MQ-REASON
2510 END-IF
2520 *
2530 IF MQ-FLAGS = 0 THEN
2540 MOVE MQCO-NONE TO MQ-OPTIONS
2550 CALL 'MQCLOSE' USING MQ-HCONN
2560 MQ-HOBJ2
2570 MQ-OPTIONS
2580 MQ-COMPCODE
2590 MQ-REASON
2600 IF MQ-COMPCODE NOT = MQCC-OK THEN
2610 DISPLAY 'MQPUT_CB : MQCLOSE FAILED. REASON CODE = '
2620 MQ-REASON
2630 END-IF
2640 END-IF.
2650 *
2660 *****
2670 * Disconnect queue manager *
2680 *****
2690 *
2700 MQ-END1.
2710 *
2720 CALL 'MQDISC' USING MQ-HCONN
2730 MQ-COMPCODE
2740 MQ-REASON
2750 IF MQ-COMPCODE NOT = MQCC-OK THEN
2760 DISPLAY 'MQPUT_CB : MQDISC FAILED. REASON CODE = '
2770 MQ-REASON
2780 END-IF.
2790 *
2800 PROG-END.
2810 *
2820 *****
2830 * RPC-CLOSE(Termination of UAP) *
2840 *****
2850 CALL 'CBLDCRPC' USING RPC-CL-ARG.
2860 *
2870 STOP RUN.
2880 END PROGRAM MAIN.

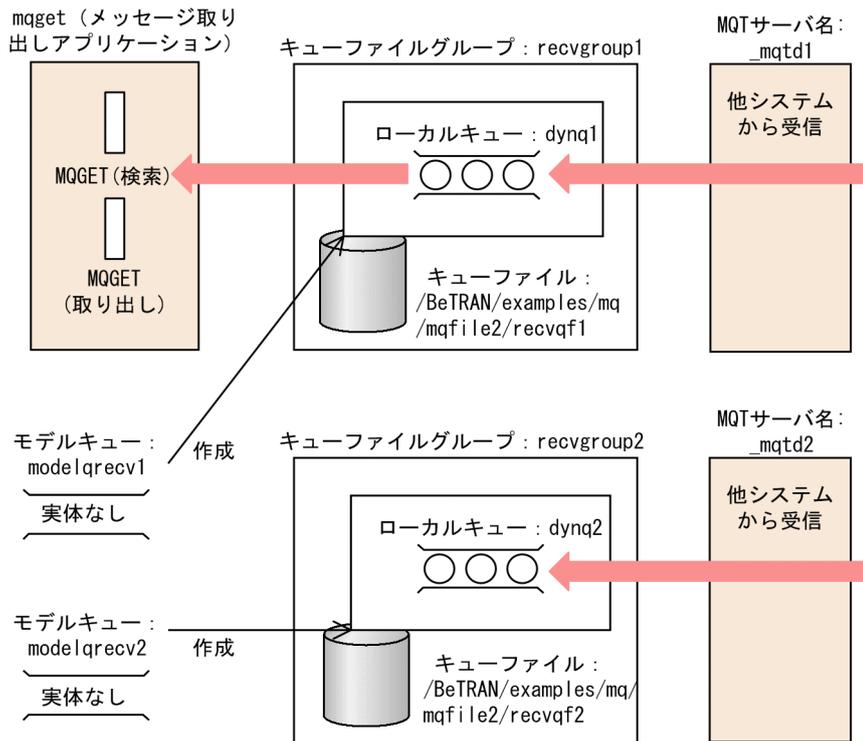
```

メッセージを取り出すアプリケーションの例

ホスト 1 上で動作するアプリケーション mqget は、ローカルキュー dynq1 からメッセージを取り出します。ローカルキュー dynq1 に登録されているメッセージはチャンネル CHA02 を経由してホスト 2 から送信されたものです。

メッセージを取り出すアプリケーションの例について、次の図に示します。

図 4-6 メッセージを取り出すアプリケーションの例



アプリケーション mqget のコーディング例 (C 言語) を次に示します。

```

0010 /*****
0020 **  mqget_c.c
0030 **  specification
0040 **  functions = main()
0050 *****/
0060
0070 /*****
0080 **  name = main()
0090 **  func = main function of UAP
0100 **      (1)start UAP(output start message)
0110 **      (2)MQCONN(Connect queue manager)
0120 **      (3)MQOPEN(Open queue)
0130 **      (4)MQGET(Get message)
0140 **      (5)MQCLOSE(Close queue)
0150 **      (6)MQDISC(Disconnect queue manager)
0160 **      (7)finish UAP(output end message)
0170 **      arg = nothing
0180 **      return = void
0190 **      note =
0200 *****/
0210 #ifdef TP1MQ_NT
0220 #include <windows.h>
0230 #endif
0240 #include <stdio.h>
0250 #include <string.h>
0260 #include <stdlib.h>
0270 #include <cmqc.h>
0280 #include <dcrpc.h>
0290 #include <dcadm.h>

```

```

0300 #include <dctrn.h>
0310
0320 static char *Mid = "%Z% %M%(%E% %U%)-%I% " ;
0330
0340 void main()
0350 {
0360     MQCHAR48    QueueManager=" "; /* queue manager name */
0370     MQCHAR      message_data[100]; /* message data */
0380     MQHCONN     qm_handle; /* connection handle */
0390     MQHOBJ      que_handle; /* object handle */
0400     MQLONG      comp_code; /* completion code */
0410     MQLONG      reason; /* reason code */
0420     MQLONG      buffer_length; /* buffer length */
0430     MQLONG      rcd; /* return code */
0440     MQOD        obj_desc = { MQOD_DEFAULT };
0450                                     /* message descriptor */
0460     MQMD        msg_desc = { MQMD_DEFAULT };
0470                                     /* object descriptor */
0480     MQGMO       get_options = { MQGMO_DEFAULT };
0490                                     /* put message options */
0500     int         flags = 1; /* loop flag */
0510
0520     /* Start of UAP */
0530     rcd = dc_rpc_open(DCNOFLAGS);
0540     if (rcd != DC_OK) {
0550         printf(
0560             "mqget_c : Failed at dc_rpc_open. CODE = %d %n",
0570             rcd);
0580         goto PROG_END;
0590     }
0600
0610     /* post the completion of user process start
0620     processing */
0630     rcd = dc_adm_complete(DCNOFLAGS);
0640     if (rcd != DC_OK) {
0650         printf(
0660             "mqget_c : Failed at dc_adm_complete. CODE = %d %n",
0670             rcd);
0680         goto PROG_END;
0690     }
0700
0710     /* Connect queue manager */
0720     MQCONN(QueueManager, &qm_handle, &comp_code, &reason);
0730     if (comp_code != MQCC_OK) {
0740         printf("mqget_c : Failed at MQCONN. CODE = %ld %n",
0750             reason);
0760         rcd = reason;
0770         goto PROG_END;
0780     }
0790
0800     /* Open object */
0810     strcpy(obj_desc.ObjectName, "dynq1");
0820     MQOPEN(qm_handle, &obj_desc,
0830         MQOO_BROWSE | MQOO_INPUT_SHARED, &que_handle,
0840         &comp_code, &reason);
0850     if (comp_code != MQCC_OK) {
0860         printf("mqget_c : Failed at MQOPEN. CODE = %ld%n",
0870             reason);

```

```

0880     rcd = reason;
0890     goto MQ_END1;
0900 }
0910
0920 /* Get Message */
0930 BROWSE_LOOP:
0940 get_options.Options =
0950     MQGMO_NO_SYNCPOINT | MQGMO_BROWSE_FIRST |
0960     MQGMO_ACCEPT_TRUNCATED_MSG | MQGMO_WAIT;
0970 get_options.WaitInterval = 5000;
0980 memcpy(msg_desc.MsgId, MQMI_NONE,
0990     sizeof(msg_desc.MsgId));
1000 memcpy(msg_desc.CorrelId, MQCI_NONE,
1010     sizeof(msg_desc.CorrelId));
1020 MQGET(qm_handle, que_handle, &msg_desc, &get_options,
1030     0, (PMQBYTE)message_data, &buffer_length,
1040     &comp_code, &reason);
1050 if (comp_code == MQCC_WARNING ||
1060     comp_code == MQCC_FAILED) {
1070     if (reason == MQRC_NO_MSG_AVAILABLE) {
1080         printf("mqget_c : No message.¥n");
1090         goto BROWSE_LOOP;
1100     }
1110     else if (reason == MQRC_TRUNCATED_MSG_ACCEPTED){
1120         while (flags == 1) {
1130             /* begin transaction */
1140             rcd = dc_trn_begin();
1150             if(rcd != DC_OK) {
1160                 printf(
1170                     "mqget_c : Failed at dc_trn_begin. CODE = %d ¥n",
1180                     rcd);
1190                 goto MQ_END2;
1200             }
1210
1220             /* Get message */
1230             memcpy(msg_desc.MsgId, MQMI_NONE,
1240                 sizeof(msg_desc.MsgId));
1250             memcpy(msg_desc.CorrelId, MQCI_NONE,
1260                 sizeof(msg_desc.CorrelId));
1270             get_options.Options = MQGMO_SYNCPOINT;
1280             MQGET(qm_handle, que_handle, &msg_desc,
1290                 &get_options, buffer_length,
1300                 (PMQBYTE)message_data, &buffer_length,
1310                 &comp_code, &reason);
1320             if (comp_code != MQCC_OK) {
1330                 printf("mqget_c : Failed at MQGET. CODE = %ld¥n",
1340                     reason);
1350                 rcd = dc_trn_unchained_rollback();
1360                 if (rcd != DC_OK) {
1370                     printf("mqget_c : Failed at dc_trn_unchained_rollback.
1380                         CODE = %d ¥n",rcd);
1390                 }
1400                 goto MQ_END2;
1410             }
1420             printf("=> %-75.75s¥n", message_data);
1430
1440             /* commit transaction */
1450             rcd = dc_trn_unchained_commit());

```

```

1460         if (rcd != DC_OK) {
1470             printf(
1480 mqget_c : Failed at dc_trn_unchained_commit. CODE = %d %n,
1490             rcd);
1500             goto MQ_END2;
1510         }
1520     }
1530 }
1540 printf(
1550     "mqget_c : Failed at MQGET. CODE = %ld REASON = %ld%n",
1560     comp_code, reason);
1570 }
1580
1590 MQ_END2:
1600 /* Close object */
1610 MQCLOSE(qm_handle, &que_handle, MQCO_NONE, &comp_code,
1620         &reason);
1630 if (comp_code != MQCC_OK) {
1640     printf("mqget_c : Failed at MQCLOSE. CODE = %ld%n",
1650         reason);
1660 }
1670
1680 MQ_END1:
1690 /* Disconnect queue manager */
1700 MQDISC(&qm_handle, &comp_code, &reason);
1710 if (comp_code != MQCC_OK) {
1720     printf("mqget_c : Failed at MQDISC. CODE = %ld %n",
1730         reason);
1740 }
1750
1760 /* Termination of UAP */
1770 dc_rpc_close(DCNOFLAGS);
1780
1790 PROG_END:
1800 return;
1810 }

```

アプリケーション mqget のコーディング例（COBOL 言語）を次に示します。

```

0010 *
0020 *****
0030 *      mqget_cb      *
0040 *****
0050 **      name = main()
0060 **      func = main function of UAP
0070 **      (1)start UAP(output start message)
0080 **      (2)MQCONN(Connect queue manager)
0090 **      (3)MQOPEN(Open queue)
0100 **      (4)MQGET(Get message)
0110 **      (5)MQCLOSE(Close queue)
0120 **      (6)MQDISC(Disconnect queue manager)
0130 **      (7)finish UAP(output end message)
0140 *****
0150 *
0160 IDENTIFICATION DIVISION.
0170 *
0180 PROGRAM-ID. MAIN.

```

```

0190 *
0200 *****
0210 *          set data area                                     *
0220 *****
0230 *
0240 DATA DIVISION.
0250 WORKING-STORAGE SECTION.
0260 01 RPC-OP-ARG.
0270 02 REQUEST      PIC      X(8) VALUE 'OPEN  ' .
0280 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0290 02 FILLER       PIC      X(3).
0300 02 FLAGS        PIC      S9(9) COMP VALUE ZERO.
0310 *
0320 01 RPC-CL-ARG.
0330 02 REQUEST      PIC      X(8) VALUE 'CLOSE  ' .
0340 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0350 02 FILLER       PIC      X(3).
0360 02 DTZRRPC     PIC      S9(9) COMP VALUE ZERO.
0370 *
0380 01 ADM-COMP.
0390 02 REQUEST      PIC      X(8) VALUE 'COMPLETE' .
0400 02 RET-VALUE    PIC      X(5).
0410 02 FILLER       PIC      X(3).
0420 02 DTZRADM     PIC      S9(9) COMP VALUE ZERO.
0430 *
0440 01 TRN-BG-ARG.
0450 02 REQUEST      PIC      X(8) VALUE 'BEGIN  ' .
0460 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0470 *
0480 01 TRN-CL-ARG.
0490 02 REQUEST      PIC      X(8) VALUE 'U-COMMIT' .
0500 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0510 01 TRN-RB-ARG.
0520 02 REQUEST      PIC      X(8) VALUE 'U-ROLL  ' .
0530 02 RET-VALUE    PIC      X(5) VALUE SPACE.
0540 *
0550 01 MQ-OBJECT     PIC X(48) VALUE SPACE.
0560 01 MQ-MGRNAME    PIC X(48) VALUE SPACE.
0570 01 MQ-HCONN     PIC S9(9) BINARY.
0580 01 MQ-COMPCODE  PIC S9(9) BINARY.
0590 01 MQ-REASON    PIC S9(9) BINARY.
0600 01 MQ-HOBJ      PIC S9(9) BINARY.
0610 01 MQ-OPTIONS   PIC S9(9) BINARY.
0620 01 MQ-BUFFLEN   PIC S9(9) BINARY.
0630 01 MQ-DATALEN   PIC S9(9) BINARY.
0640 01 MQ-GET-BUFFER PIC X(128).
0650 *
0660 01 MQ-OBJECT-DESC.
0670 COPY CMQODV.
0680 *
0690 01 MQ-MESSAGE-DESCRIPTOR.
0700 COPY CMQMDV.
0710 *
0720 01 MQ-GET-MESSAGE-OPTIONS.
0730 COPY CMQGMV.
0740 *
0750 01 MQ-CONSTANTS.
0760 COPY CMQV SUPPRESS.

```

```

0770 *
0780 PROCEDURE DIVISION.
0790 *
0800 *****
0810 *         RPC-OPEN(Start of UAP) *
0820 *****
0830 *
0840 CALL 'CBLDCRPC' USING RPC-OP-ARG.
0850     IF RET-VALUE OF RPC-OP-ARG NOT = '00000'
0860     DISPLAY 'MQGET_CB : RPC-OPEN FAILED. CODE = '
0870     RET-VALUE OF RPC-OP-ARG
0880     GO TO PROG-END
0890     END-IF.
0900 *
0910 *****
0920 *         ADM-COMPLETE(post the completion of *
0930 *         user process start processing) *
0940 *****
0950 *
0960 CALL 'CBLDCADM' USING ADM-COMP.
0970     IF RET-VALUE OF ADM-COMP NOT = '00000'
0980     DISPLAY 'MQGET_CB : ADM-COMPLETE FAILED.CODE = '
0990     RET-VALUE OF ADM-COMP
1000     GO TO PROG-END
1010     END-IF.
1020 *
1030 *****
1040 *         Connect queue manager *
1050 *****
1060 *
1070 CALL 'MQCONN' USING MQ-MGRNAME
1080     MQ-HCONN
1090     MQ-COMPCODE
1100     MQ-REASON.
1110     IF MQ-COMPCODE NOT = MQCC-OK
1120     DISPLAY 'MQGET_CB : MQCONN FAILED. REASON CODE = '
1130     MQ-REASON
1140     GO TO PROG-END
1150     END-IF.
1160 *
1170 *****
1180 *         Opening of object *
1190 *****
1200 *
1210 MOVE 'dynq1'          TO MQ-OBJECT.
1220 MOVE MQ-OBJECT       TO MQOD-OBJECTNAME.
1230 *
1240 COMPUTE MQ-OPTIONS = MQOO-INPUT-SHARED +
1250     MQOO-BROWSE.
1260 *
1270 CALL 'MQOPEN' USING MQ-HCONN
1280     MQOD
1290     MQ-OPTIONS
1300     MQ-HOBJ
1310     MQ-COMPCODE
1320     MQ-REASON.
1330     IF MQ-COMPCODE NOT = MQCC-OK
1340     DISPLAY 'MQGET_CB : MQOPEN FAILED. REASON CODE = '

```

```

1350     MQ-REASON
1360     GO TO MQ-END1
1370     END-IF.
1380     *
1390     *****
1400     *         Getting of browse message             *
1410     *****
1420     *
1430     BROWSE-LOOP.
1440     COMPUTE MQGMO-OPTIONS = MQGMO-NO-SYNCPPOINT +
1450     MQGMO-BROWSE-FIRST +
1460     MQGMO-ACCEPT-TRUNCATED-MSG +
1470     MQGMO-WAIT.
1480     MOVE MQMI-NONE TO MQMD-MSGID.
1490     MOVE MQCI-NONE TO MQMD-CORRELID.
1500     MOVE 5000 TO MQGMO-WAITINTERVAL.
1510     MOVE 0     TO MQ-BUFFLEN.
1520     CALL 'MQGET' USING MQ-HCONN
1530     MQ-HOBJ
1540     MQMD
1550     MQGMO
1560     MQ-BUFFLEN
1570     MQ-GET-BUFFER
1580     MQ-DATALEN
1590     MQ-COMPCODE
1600     MQ-REASON
1610     IF (MQ-COMPCODE = MQCC-WARNING) OR
1620     (MQ-COMPCODE = MQCC-FAILED)
1630     IF MQ-REASON = MQRC-NO-MSG-AVAILABLE
1640     DISPLAY 'MQGET_CB : NO MESSAGE'
1650     GO TO BROWSE-LOOP
1660     ELSE
1670     IF MQ-REASON = MQRC-TRUNCATED-MSG-ACCEPTED
1680     PERFORM GET-MSG-PROC
1690     END-IF
1700     ELSE
1710     DISPLAY 'MQGET_CB : MQGET FAILED. CODE = ' MQ-COMPCODE
1720     DISPLAY '                               REASON CODE = '
1730     MQ-REASON
1740     END-IF.
1750     *
1760     *****
1770     *         Closing of object                     *
1780     *****
1790     *
1800     MQ-END2.
1810     *
1820     MOVE MQCO-NONE TO MQ-OPTIONS
1830     *
1840     CALL 'MQCLOSE' USING MQ-HCONN
1850     MQ-HOBJ
1860     MQ-OPTIONS
1870     MQ-COMPCODE
1880     MQ-REASON
1890     IF MQ-COMPCODE NOT = MQCC-OK THEN
1900     DISPLAY 'MQGET_CB : MQCLOSE FAILED. REASON CODE = '
1910     MQ-REASON
1920     END-IF.

```

```

1930 *
1940 *****
1950 *      Disconnect queue manager      *
1960 *****
1970 *
1980 MQ-END1.
1990 *
2000 CALL 'MQDISC' USING MQ-HCONN
2010     MQ-COMPCODE
2020     MQ-REASON
2030     IF MQ-COMPCODE NOT = MQCC-OK
2040         DISPLAY 'MQGET_CB : MQDISC FAILED. REASON CODE = '
2050     MQ-REASON
2060     END-IF.
2070 *
2080 PROG-END.
2090 *
2100 *****
2110 *      RPC-CLOSE(Termination of UAP)  *
2120 *****
2130 CALL 'CBLDCRPC' USING RPC-CL-ARG.
2140 *
2150 *
2160 STOP RUN.
2170 GET-MSG-PROC.
2180 *
2190 *****
2200 *      begin transaction              *
2210 *****
2220 *
2230 CALL 'CBLDCTRN' USING TRN-BG-ARG
2240     IF RET-VALUE OF TRN-BG-ARG NOT = '00000'
2250         DISPLAY 'MQGET_CB : TRN-BEGIN FAILED. CODE = '
2260     RET-VALUE OF TRN-BG-ARG
2270     GO TO MQ-END2
2280     END-IF
2290     MOVE 0 TO MQGMO-OPTIONS
2300     MOVE MQMI-NONE                TO MQMD-MSGID.
2310     MOVE MQCI-NONE                TO MQMD-CORRELID.
2320     COMPUTE MQGMO-OPTIONS = MQGMO-SYNCPOINT
2330     COMPUTE MQ-BUFFLEN = MQ-DATALEN
2340     CALL 'MQGET' USING MQ-HCONN
2350     MQ-HOBJ
2360     MQMD
2370     MQGMO
2380     MQ-BUFFLEN
2390     MQ-GET-BUFFER
2400     MQ-DATALEN
2410     MQ-COMPCODE
2420     MQ-REASON
2430     IF MQ-COMPCODE NOT = MQCC-OK
2440         DISPLAY 'MQGET_CB : MQGET FAILED. CODE = '
2450     MQ-COMPCODE
2460     DISPLAY '                REASON CODE = '
2470     MQ-REASON
2480     CALL 'CBLDCTRN' USING TRN-RB-ARG
2490     IF RET-VALUE OF TRN-RB-ARG NOT = '00000'
2500         DISPLAY 'MQGET_CB : TRN-ROLLBK FAILED. CODE = '

```

```
2510     RET-VALUE OF TRN-RB-ARG
2520     END-IF
2530     GO TO MQ-END2
2540     END-IF
2550     DISPLAY 'MQGET_CB MESSAGE : ' MQ-GET-BUFFER
2560 *
2570 *****
2580 *     commit transaction *
2590 *****
2600 *
2610     CALL 'CBLDCTR' USING TRN-CL-ARG
2620     IF RET-VALUE OF TRN-CL-ARG NOT = '00000'
2630     DISPLAY 'MQGET_CB : TRN-COMMIT FAILED. CODE = '
2640     RET-VALUE OF TRN-CL-ARG
2650     GO TO MQ-END2
2660     END-IF
2670     GO TO GET-MSG-PROC.
2680
2690     END PROGRAM MAIN.
```

5

開始と終了

TP1/Message Queue の開始と終了は、OpenTP1 システムの開始と終了のモードに合わせたモードになります。この章では、TP1/Message Queue の開始と終了、およびチャネル情報の引き継ぎについて説明します。

5.1 TP1/Message Queue の開始

OpenTP1 システムが開始すると同時に、TP1/Message Queue の MQA サーバが開始します。MQA サーバと同時に MQT サーバを開始させるためには、MQA サービス定義の `mqamqtnam` 定義コマンドに MQT サーバ名を指定しておく必要があります。ここで指定した MQT サーバは、MQA サーバと同時に開始されます。

開始モードには、OpenTP1 の開始モードに従って、正常開始および再開があります。

5.1.1 正常開始

TP1/Message Queue は、OpenTP1 システムの正常開始の延長で開始します。

注意事項

- OpenTP1 がオンライン状態になるまでは、オンラインコマンドを実行したり、相手システムからのチャンネルの確立要求を受け付けたりできません。
- MQA サービス定義でチャンネル管理情報格納ファイルが指定されている場合は、チャンネル情報を引き継ぎ、その情報を基にサービスを開始します。チャンネル情報の引き継ぎについては、「[5.3 開始時のチャンネル情報引き継ぎ](#)」を参照してください。
- 前回のオンラインでキューファイル二重化構成の片系障害が発生した場合は、障害キューファイルを回復してから TP1/Message Queue を正常開始してください。障害キューファイルを回復しないで TP1/Message Queue を正常開始した場合、キューファイルの内容は保証されません。
障害キューファイルの回復については、「[7.1.4 キューファイルの入出力エラー](#)」を参照してください。

5.1.2 再開

OpenTP1 システムに障害が発生した場合、TP1/Message Queue は OpenTP1 システムの再開の延長で再開し、障害発生直前の状態に回復されます。

OpenTP1 システムが前回のオンラインの終了を引き継いで開始すると同時に開始します。メッセージ転送中またはアプリケーションがキューアクセス中に強制停止をしたあとは、キューファイルの内容が矛盾するおそれがあるため、必ず再開してください。

開始モードによって、メッセージシーケンス番号の取り扱いが異なります。メッセージシーケンス番号については、「[2.4.4 メッセージシーケンス番号でのメッセージの管理](#)」を参照してください。

OpenTP1 の強制正常開始をする場合、必ず、すべてのキューファイルを再作成してから強制正常開始してください。キューファイルの再作成をしないで、強制正常開始をした場合、不正なメッセージが残る問題が発生することがあります。

5.2 TP1/Message Queue の終了

TP1/Message Queue の終了モードには、次に示す四つがあります。

1. 正常終了

OpenTP1 システムが正常終了した場合の終了モードです。

2. 計画停止 A

OpenTP1 システムが計画停止 A で終了した場合の終了モードです。

3. 計画停止 B

OpenTP1 システムが計画停止 B で終了した場合の終了モードです。

4. 強制停止

OpenTP1 システムが強制停止した場合の終了モードです。

正常終了、計画停止 A、および計画停止 B の場合、メッセージ転送の終了時間の監視ができます。終了処理監視は、TCP 定義の `mqttcp`、`mqttcps`、および `mqttcpcr` 定義コマンドで指定します。終了処理監視でタイムアウトが発生した場合、メッセージの送受信は中断されます。このタイマを指定しない場合、受信側 MCA が動作しているシステムでは、送信側 MCA からアプリケーションが登録したメッセージまたはハートビートメッセージを受信するか、送信側 MCA を終了しないかぎり、チャンネルを終了しませんので注意してください。ハートビートメッセージについては、「[2.5.7 ハートビート機能](#)」を参照してください。

終了モード別のメッセージの取り扱いについて、次の表に示します。

表 5-1 終了モード別のメッセージの取り扱い

終了モード	メッセージの取り扱い	
	送信時	受信時
正常終了 計画停止 A 計画停止 B	送信中のメッセージは送信を完了します。 転送キューのメッセージはキューに残ります。	受信中のメッセージは受信を完了します。 完了したメッセージに対するトリガメッセージは発生しません。
強制停止	メッセージの送信処理を中断します。送信中のメッセージはキューに残り、次回のチャンネル開始要求で、再度メッセージの先頭から送信します。	メッセージの受信処理を中断します。受信中のメッセージは相手システムに残り、次回のチャンネル開始要求で、再度メッセージの先頭から受信します。

また、開始と終了のモードによって、キューの属性およびメッセージの内容を引き継ぐかどうか異なります。開始モード、終了モード別のキューおよびメッセージの取り扱いについて、次の表に示します。なお、メッセージ転送中またはアプリケーションがキューアクセス中に強制停止をしたあとは、キューファイルの内容が矛盾するおそれがあるため、必ず再開始してください。

表 5-2 開始モード，終了モード別のキューおよびメッセージの取り扱い

前回のオンラインモード	開始モード	一時的動的キュー	永続的動的キュー	キューの属性※1	永続性メッセージ	非永続性メッセージ
正常終了	正常開始	×	○	○	○	×
計画停止 A 計画停止 B	再開	×	○	○	○	×
強制停止 システム障害による 全面停止	再開	×	○	○	○※2	×

(凡例)

- ：状態を引き継ぎます。
- ×

注※1

MQSET 命令で変更された属性値です。

注※2

メッセージの登録および取り出しを OpenTP1 のトランザクションの一部として操作していた場合，トランザクション回復処理でそのトランザクションがコミットするか，ロールバックするかによってメッセージ操作が有効になるかどうかが決まります。

5.3 開始時のチャンネル情報引き継ぎ

TP1/Message Queue は、正常開始時にチャンネル情報を引き継ぐために、正常開始時のチャンネル情報引き継ぎ機能を提供します。

正常開始時のチャンネル情報引き継ぎ機能によって、TP1/Message Queue は、OpenTP1 正常開始時にメッセージシーケンス番号、および最終メッセージの送達確認状態などを引き継ぐことができます。正常開始時のチャンネル情報引き継ぎ機能を使用すると再開時と同様のチャンネルの運用が可能になります。引き継ぐことができるチャンネルの情報を、次の表に示します。

表 5-3 正常開始時に引き継ぐことができるチャンネルの情報

情報	引き継ぐ内容	該当するチャンネル
メッセージシーケンス番号	仕掛かりシーケンス番号, 完了シーケンス番号	<ul style="list-style-type: none">すべてのチャンネル
送達確認情報	最終バッチの送達確認状態	<ul style="list-style-type: none">センダチャンネルサーバチャンネルクラスタセンダチャンネル
メッセージシーケンス番号リセット値	mqtrstcha コマンドで指定したリセット値	<ul style="list-style-type: none">すべてのチャンネル
	リセットデータで受信したリセット値	<ul style="list-style-type: none">レシーバチャンネルリクエストチャンネルクラスタレシーバチャンネル
相手システム情報	mqaltcha コマンドに-l perm オペランドを指定して変更した情報	<ul style="list-style-type: none">センダチャンネルリクエストチャンネル (コーラチャンネルだけ)サーバチャンネル (コーラチャンネルだけ)
自動定義チャンネル情報	自動定義クラスタセンダおよびクラスタレシーバインスタンス	<ul style="list-style-type: none">クラスタセンダチャンネルクラスタレシーバチャンネル

5.3.1 チャンネル管理情報格納ファイル

正常開始時のチャンネル情報引き継ぎ機能を使用する場合、チャンネル管理情報格納ファイルを使用します。

チャンネル管理情報格納ファイルは二重化できます。A、および B 系の二つのファイルに同様の内容を格納します。情報を回復する際にはファイル破壊の有無をチェックし、有効なファイルからチャンネル情報を回復します。

チャンネル管理情報格納ファイルのファイル名は、MQA サービス定義の mqa_channel_inf_file_name_a オペランドおよび mqa_channel_inf_file_name_b オペランドで指定します。省略した場合は、正常開始時にチャンネル情報を引き継ぎません。

チャンネル管理情報格納ファイルは、通常ファイル上、または OpenTP1 ファイルシステム上に作成できます。

5.3.2 チャンネル情報を引き継がない条件

正常開始をしても、チャンネル情報を引き継がない場合があります。どのチャンネル情報を引き継がないかは、条件によって異なります。

- 次に示す条件で正常開始した場合は、どの MQT サーバのチャンネル情報も引き継ぎません。
 - 前回のオンラインが強制停止、または OpenTP1 が異常終了し、強制正常開始したとき
 - チャンネル管理情報格納ファイルの A、および B 両系がない、または破壊されているとき
 - 自システムのキューマネージャ名 (mqaquemgr 定義コマンドの -n オプション) を変更したとき
- 次に示す条件で正常開始した場合は、該当する MQT サーバのチャンネル情報を引き継ぎません。
 - MQT 通信プロセス識別子 (mqttenv 定義コマンドの -s オプション) を変更したとき
- 次に示す条件で正常開始した場合は、該当するチャンネルのチャンネル情報を引き継ぎません。
 - TCP 定義の mqtalccha 定義コマンドに指定したチャンネルタイプ (-y オプション) を変更したとき
このとき、該当するチャンネル以外はチャンネル情報を引き継ぎます。
 - 正常開始時に MQT 通信構成定義からチャンネルを削除したとき
 - チャンネルが所属するクラスタ名 (mqtalccha 定義コマンドの -a オプション) を変更したとき
 - SYSTEM.CLUSTER.REPOSITORY.QUEUE のメッセージを削除したとき (クラスタ自動定義チャンネル情報)
- 次に示す条件の送信側チャンネルがある場合は、正常開始時に KFCA31909-E メッセージが出力されます。バッチサイズを変更するまで MQT サーバを開始できません。
 - センダチャンネル、サーバチャンネルまたは事前定義クラスタセンダチャンネルについて、TCP 定義の mqtalccha 定義コマンドに指定したバッチサイズ (-j オプション) が前回のオンラインの値よりも小さく、かつメッセージ送達確認状態=未確認のとき
 - 自動定義クラスタセンダチャンネルについて、バッチサイズが前回のオンラインの値よりも小さく、かつメッセージ送達確認状態=未確認のとき。自動定義クラスタセンダチャンネルのバッチサイズの計算式については、4 章の「チャンネルのネゴシエーション」を参照してください。

チャンネル情報を引き継がない場合、次の表に示す属性値になります。

表 5-4 チャンネルを引き継がない場合の属性値および注意点

情報	属性値	注意点
メッセージシーケンス番号	仕掛かりシーケンス番号=0, 完了シーケンス番号=0 (すべてのチャンネル)	相手システムとの間でメッセージシーケンス番号が不一致になるので、メッセー

情報	属性値	注意点
メッセージシーケンス番号	仕掛かりシーケンス番号=0, 完了シーケンス番号=0 (すべてのチャンネル)	ジシーケンス番号をリセットする必要があります。
送達確認情報	メッセージ送達確認状態 (送信側チャンネル)	送信チャンネルのメッセージ送達確認状態 = 未確認, かつ受信チャンネル側でメッセージの受信処理が完了している場合, 転送メッセージの二重送信が最大バッチサイズ分発生します。
メッセージシーケンス番号リセット値	リセットしたメッセージシーケンス番号	前回のオンライン時にリセットしたメッセージシーケンス番号でメッセージ送受信をする場合は, 再度リセット処理をする必要があります。
相手システム情報	TCP 定義の mqtalccha 定義コマンド-o オプションで指定する属性	前回のオンライン時に mqtaltcha -l perm コマンドによって変更した相手システム情報がある場合は, 再度 mqtaltcha コマンドを入力する必要があります。
自動定義チャンネル情報	自動定義クラスタセンダおよびクラスタレシーバインスタンスなし	-

(凡例)

- : 該当しません。

TCP 定義に追加したチャンネルは TCP 定義を基に初期状態のチャンネル情報で開始します。

5.3.3 注意事項

- 次に示すオンライン開始時には, ログメッセージが出力されます。
 - 正常開始時のチャンネル状態引き継ぎ機能を使用する最初のオンライン
 - チャンネル管理情報格納ファイルを変更したあとのオンライン
 - チャンネル管理情報格納ファイルを意図的に削除したあとのオンライン

出力されるログメッセージを次に示します。

KFCA31905-W 指定したチャンネル管理情報格納ファイルは存在しません。
KFCA31903-W 前回のオンライン中のチャンネル管理情報を引き継ぐ事ができません。

ログメッセージが出力されるのは, チャンネル情報を引き継ぐために必要な管理状態が存在しないためです。ログメッセージを無視して, オンラインを続行してください。

- MQA サービス定義に指定したチャンネル管理情報格納ファイルからの情報読み込みで, ファイル内容に異常が検出された場合, ファイル内容が破壊されていると判断され, その後の書き込み処理が続行されます。このため, MQA サービス定義に誤ったファイル名を指定すると, 既存のデータが失われますので注意してください。

- チャンネル情報引き継ぎ機能を使用しない場合、またはチャンネル情報を引き継がない条件に該当する場合は、メッセージの二重受信と判断し、受信したメッセージを破棄したり、メッセージシーケンス番号不一致になったりすることがあります。正常開始後、チャンネルを開始する前に、メッセージシーケンス番号のリセットを必ず行ってください。二重受信メッセージが破棄される条件については、「[2.4.4\(7\) 二重受信メッセージの破棄](#)」を参照してください。

チャンネル情報引き継ぎ機能を使用しない場合、またはチャンネル情報を引き継がない条件に該当する場合は、メッセージの二重送信が発生することがあります。その場合は、メッセージ送達未確認状態の解決を行ってから TP1/Message Queue を終了してください。メッセージ送達未確認状態の解決については、「[2.4.4\(6\)\(a\) メッセージの送達未確認状態の解決](#)」を参照してください。

6

システムの運用

TP1/Message Queue は、運用コマンドを入力してシステムを運用できます。この章では、TP1/Message Queue の運用コマンドの内容について説明します。

運用コマンドの概要

運用コマンドの入力方法および記述形式について説明します。

入力方法

TP1/Message Queue の運用コマンドの入力方法には、シェルから入力する方法と、アプリケーションの中に組み込んで実行する方法があります。

TP1/Message Queue の運用コマンドは、すべてアプリケーションから実行できます。アプリケーションから実行するには、OpenTP1 の提供する `dc_adm_call_command` 関数を使用して実行します。したがって、コマンドがあるディレクトリを、OpenTP1 のユーザサービス定義の `putenv PATH` であらかじめ定義しておいてください。コマンドの実行結果は、アプリケーションに返されます。返される内容は、標準出力または標準エラー出力に出力される値です。

記述形式

運用コマンドの記述形式を次に示します。

```
コマンド名 オプション コマンド引数
```

コマンド名

コマンド名は、実行するコマンドのファイル名称です。

OpenTP1 の運用コマンドは `$DCDIR/bin/` にありますので、`PATH` に `$DCDIR/bin` を加えてください。

オプション

次に示す説明中に使用する「\$」はシェルのプロンプト、「cmd」はコマンド名称です。

1. オプションはマイナス記号 (-) で始まる文字列で、フラグ引数を取らないか、または 1 個のフラグ引数を取ります。

オプションの記述形式を次に示します。

```
-オプションフラグ  
または  
-オプションフラグ フラグ引数
```

オプションフラグは、1 文字の英数字（英大文字と英小文字は区別されます）です。

フラグ引数は、オプションフラグに対する引数です。

2. フラグ引数を取らないオプションフラグは、一つのマイナス記号 (-) の後ろにまとめて指定できます。
(例) 次に示す二つは同じ意味です。

```
$ cmd -a -b -c  
$ cmd -abc
```

3. フラグ引数を必要とするオプションフラグのフラグ引数は、省略できません。
4. フラグ引数中に空白を含める場合で、シェルから入力するときには、前後を引用符 (") で囲まなければなりません。

(例) 1 2 を引数に持つオプション f は、次に示すとおり記述します。

```
$ cmd -f "1 2"
```

5. 同じオプションフラグを 2 回以上指定すると、最後に指定したものが有効になります。

(例)

```
$ cmd -a 1 -a 2
      無効 有効
```

6. オプションは、コマンド引数より前に指定しなければなりません。

(例)

オプションフラグ a が、フラグ引数を取らない場合、次に示すとおり入力すると、file と -b は、コマンド引数とみなされます。

```
$ cmd -a file -b
```

7. 二つのマイナス記号 (-) は、オプションの終わりを示します。

(例)

次に示すとおり入力すると、-b はコマンド引数とみなされます。

```
$ cmd -a -- -b
```

8. マイナス記号 (-) だけのオプションは入力できません。

(例)

次に示すとおり入力すると、- はコマンド引数とみなされます。

```
$ cmd -
```

コマンド引数

コマンド引数は、コマンド操作の対象になるものを指定します。

TP1/Message Queue の運用コマンド

TP1/Message Queue の運用コマンドについて説明します。

TP1/Message Queue の運用コマンドについて、次の表に示します。各コマンドの詳細については、アルファベット順に掲載します。

表 6-1 TP1/Message Queue の運用コマンド

MQA/ MQT	機能	コマンド名称	定義中組 み込み	オフライン 中に 実行	オンライン 中に 実行	アプリケーション から 実行	アクセス権限
MQA	キューファイルの組み込み	mqaadd	×	×	○	○	—
	キューファイルのバックアップ	mqabkup	×	○	×	○	—
	キューの属性の変更	mqachgque	×	○	×	○	—
	キューファイルのコンバート	mqaconvert	×	○	×	○	—
	キューファイルの削除	mqadel	×	○	×	○	—
	メッセージの削除	mqadelmsg	×	×	○	○	—
	キュー, メッセージの削除	mqadelque	×	○	×	○	—
	キューファイルグループの使用状況表示	mqadf	×	×	○	○	—
	キューファイル内の情報表示	mqafilinf	×	○	×	○	—
	キューファイルの情報表示	mqafills	×	×	○	○	—
	キューファイルの回復	mqafrc	×	○	×	○	—
	キューファイルグループ内のキューファイル状態変更	mqagrpbobs	×	×	○	○	—
	キューファイルの割り当て	mqainit	×	○	×	○	—
	オブジェクトの属性表示	mqainq	×	×	○	○	—
	メッセージの表示	mqalsmsg	×	×	○	○	—
	キューの作成	mqamkque	×	○	×	○	—
	ファイル二重化の回復	mqarcvr	×	×	○	○	—
	キューファイルの閉塞解除	mqarles	×	×	○	○	—
	キューファイルの切り離し	mqarm	×	×	○	○	—
	キューファイルのリストア	mqarstr	×	○	×	○	—
オブジェクト属性の設定	mqaset	×	×	○	○	—	

MQA/ MQT	機能	コマンド名称	定義中組 み込み	オフライン 中に 実行	オンライン 中に 実行	アプリケーションから 実行	アクセス権限
MQC ※1	MQC トランザクション状態 の変更	mqccontrn	×	×	○	○	—
	MQC ゲートウェイサーバの 状態表示	mqcls	×	×	○	○	—
	MQC トランザクション状態 の表示	mqclstrn	×	×	○	○	—
	MQC トレースの強制ス ワップ	mqcswptrc	×	×	○	○	—
MQR	クラスタ情報の表示	mqrcls	×	×	○	○	—
	リポジトリ情報の再作成	mqrrefresh	×	×	○	○	—
	クラスタからの脱退	mqrremove	×	×	○	○	—
	MQR トレースの強制ス ワップ	mqrswptrc	×	×	○	○	—
MQT ※2	チャンネル属性の変更	mqaltcha	×	×	○	○	—
	MQT 通信プロセス情報の 表示	mqtdmped	×	○	○	○	—
	MQT トレースファイルの編 集出力	mqted	×	○	○	○	—
	チャンネルの状態表示	mqtlscha	×	×	○	○	—
	チャンネルのテスト接続	mqtpngcha	×	×	○	○	—
	メッセージ送達未確認の解決	mqtrlvcha	×	×	○	○	—
	メッセージシーケンス番号の リセット	mqtrstcha	×	×	○	○	—
	チャンネルの開始	mqstacha	×	×	○	○	—
	MQT トレースのディスク出 力機能の開始	mqstatrc	×	×	○	○	—
	チャンネルの終了	mqstpcha	×	×	○	○	—
	MQT トレースのディスク出 力機能の終了	mqstptrc	×	×	○	○	—
MQT トレースの強制ス ワップ	mqtswptrc	×	×	○	○	—	

(凡例)

- ：組み込み，または実行ができます。
- ×
- ：任意のユーザが実行できます。

注※1

関連製品である TP1/Message Queue Access の使用時に実行するコマンドです。

注※2

MQT のコマンドは、MQT 通信プロセスと同一ノードの場合だけ実行できます。

MQT のコマンドのうち、オフライン中に実行できないコマンドは、「KFCA16302-I MQT サービスを開始しました」が出力されてから「KFCA16310-I MQT 通信プロセスの終了要求を受け付けました」が出力されるまでの間で実行できます。ただし、クラスタ環境ではリポジトリ管理サーバが開始されるまで「KFCA16302-I MQT サービスを開始しました」は出力されません。

mqaadd (キューファイルの組み込み)

形式

```
mqaadd -g キューファイルグループ名  
        [-b バックアップキューファイル名  
         [ [ , バックアップキューファイル名] ... ] ]  
        キューファイル名 [ [ キューファイル名] ...]
```

機能

指定したキューファイルを組み込み、オンラインで使用できるようにします。

指定したキューファイルグループが登録済みの場合は、登録済みのキューファイルグループに指定したキューファイルだけを組み込みます。

このコマンドで組み込んだキューファイルの状態およびキューファイルグループに登録されているキューファイルの状態は、共に論理閉塞状態になります。

メッセージを操作する場合は、組み込んだキューファイルに対し、mqarles コマンドを入力してください。バックアップキューファイルに対する mqarles コマンドの入力は不要です。

オプション

● -g キューファイルグループ名

～< 1～31 文字の識別子 >

オンラインに組み込むキューファイルグループ名を指定します。

● -b バックアップキューファイル名

～< 1～59 文字の文字列 >

キューファイル二重化構成を行う場合、コマンド引数で指定したキューファイル名 (A 系ファイル名) に対するバックアップキューファイル名 (B 系ファイル名) を指定します。

キューファイル (A 系ファイル)、バックアップキューファイル (B 系ファイル) は、大きさ (mqainit コマンドのオプション)、キュー作成の有無 (mqamkque コマンドのコマンド引数)、指定の順序、および OpenTP1 ファイルシステムのセクタ長を同一にしてください。

コマンド引数

● キューファイル名

～< 1～59 文字の文字列 >

キューに登録したメッセージを格納するためのファイル (キューファイル) の名称を指定します。

一つのキューファイルには、複数のキューを格納できます。

キューファイルを複数指定すると、一つのキューに登録するメッセージは、複数のキューファイルに分散して格納されます。そのためそれぞれのキューファイルを物理的に別ディスクにすると、キューに複数のアプリケーションからの操作が重なった場合に、性能が良くなります。

注意事項

1. キューファイルを組み込む場合、次に示すどれかの数を超えるとエラーになります。
 - MQA サービス定義で指定するキューの最大数
 - キューファイルグループの最大数
 - キューファイルの最大数
2. キューファイルを複数指定した場合、複数ファイルのうち、あるキューファイルでエラーが発生すると、それまでに組み込みが完了したキューファイルは組み込み状態のままになります。
3. このコマンドで組み込んだキューファイルは、次に正常開始をする時に無効（組み込まれていない状態）になります。組み込んだキューファイルを次の正常開始で有効にするには、MQA サービス定義の `mqaquegrp` 定義コマンドに追加する必要があります。ただし、再開始時には、このコマンドで組み込んだキューファイルは有効になります。
4. キューファイル二重化構成を行っているキューファイルグループに組み込む場合は、`-b` オプションを指定してください。指定しない場合はエラーになります。
5. キューファイル二重化構成でアクセスをしたキューファイル名（A 系ファイル名）とバックアップキューファイル名（B 系ファイル名）との定義をオンライン終了後に入れ替えて組み込まないでください。

1回目のオンライン定義

```
mqaquegrp -g GR1
           -b /dev/rdisk/MQA4,/dev/rdisk/MQA5,/dev/rdisk/MQA6
              /dev/rdisk/MQA1,/dev/rdisk/MQA2,/dev/rdisk/MQA3
```

2回目のオンライン定義

```
mqaquegrp -g GR1
           -b /dev/rdisk/MQA1,/dev/rdisk/MQA2,/dev/rdisk/MQA3
              /dev/rdisk/MQA4,/dev/rdisk/MQA5,/dev/rdisk/MQA6
```

上記の指定をした場合、処理は保証できません。

6. `mqaadd` コマンドで、オンライン中にキューファイルを組み込む場合、組み込むキューファイルで使用するサイズの共用メモリを MQA サービス専用共用メモリの拡張領域の空き領域から使用します。
7. MQA サービス定義の `mqaquegrp` 定義コマンドで指定したキューファイル以外の新規キューファイルをオンライン中に組み込むと、組み込むキューファイルで使用するサイズ分の MQA サービス専用共用メモリの拡張領域を使用します。一度、新規キューファイルを組み込むと、`mqarm` コマンドでキューファイルを切り離しても MQA サービス専用共用メモリの拡張領域は予約された状態のままとなります。したがって、同一のキューファイルを組み込む場合だけ MQA サービス専用共用メモリの拡張領域の予約された部分を使用できます。

8.mqaadd コマンドを実行した場合、mqaadd コマンドで指定したキューファイルグループに登録されているキューに対する次の動作は待ち状態となります。mqaadd コマンドの処理が終了した時点で待ち状態は解除されます。

- MQPUT 命令, MQPUT1 命令, MQGET 命令または MQSET 命令の発行
- トランザクションのコミット・ロールバック
- メッセージの送受信
- メッセージの滞留時間の監視
- mqadelsmsg コマンドまたは mqalsmsg コマンドの実行

また、上記の動作中に mqaadd コマンドを実行した場合、mqaadd コマンドの処理は待ち状態になります。上記の動作が終了した時点で mqaadd コマンドの待ち状態は解除されます。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA04200-E	I/O エラーが発生しました。	標準エラー出力
KFCA04203-E	キューファイルグループ名が誤っています。	標準エラー出力
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04211-E	キューファイル名の文字数が 59 文字を超えています。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04217-E	指定したキューファイルでファイルシステム作成時のシステムのバージョンが異なります。	標準エラー出力
KFCA04221-E	キューファイルとキューファイルで重複したキュー名が存在します。	標準エラー出力
KFCA04222-E	キューファイルとキューファイルで重複したメッセージが指定されています。	標準エラー出力
KFCA04223-E	動的共用メモリが足りないため、入出力バッファが確保できません。	標準エラー出力
KFCA04253-E	定義ファイルとキューファイルで重複したキュー名が指定されています。	標準エラー出力
KFCA04294-E	転送キュー名はローカルキュー名ではありません。	メッセージログファイル
KFCA04295-E	イニシエーションキュー名はローカルキュー名ではありません。	メッセージログファイル
KFCA04296-E	転送キュー名として指定されたキューのキュー使用方法が誤っています。	メッセージログファイル
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04313-E	キューファイルに対するアクセス権がありません。	標準エラー出力
KFCA04314-E	指定したキューファイルは他のプロセスで使用中です。	標準エラー出力
KFCA04315-E	キューファイルの V/R が誤っています。	標準エラー出力
KFCA04316-E	スペシャルファイルに対するアクセス権がありません。	標準エラー出力
KFCA04317-E	指定したキューファイルはスペシャルファイル名ではありません。	標準エラー出力
KFCA04318-E	キューファイルを割り当てるディスクパーティションは OpenTP1 ファイルシステムとして初期化されていません。	標準エラー出力
KFCA04319-E	キャラクタ型スペシャルファイルのオープンでシステムから上限値オーバのエラーが報告されました。	標準エラー出力
KFCA04321-E	入出力用バッファ数の指定が誤っています。	標準エラー出力
KFCA04322-E	キューファイルグループ名の文字数が 31 文字を超えています。	標準エラー出力
KFCA04351-E	指定したキューファイルは最大数を超えたため組み込めません。	標準エラー出力
KFCA04352-I	ヘルプメッセージ	標準出力
KFCA04360-E	キュー数をオーバしました。	標準エラー出力
KFCA04364-I	キューファイルの組み込みが完了しました。	メッセージログファイル
KFCA04367-E	指定したキューファイルはキューファイルグループに組み込めません。	標準エラー出力
KFCA04368-E	指定したキューファイルは存在しません。	標準エラー出力
KFCA04375-E	指定されたキューファイル名はすでに存在します。	標準エラー出力
KFCA04387-E	指定したファイルはキューファイルではありません。	標準エラー出力
KFCA26158-E	キューファイルに登録されているメッセージのキュー名が違うキューファイルに存在します。	標準エラー出力
KFCA26164-E	キューファイルに格納されているメッセージのキュー名がキューファイルに存在しません。	標準エラー出力
KFCA26169-E	最大メッセージ長がキューマネージャ定義で指定したメッセージ長の最大値を超えています。	標準エラー出力
KFCA26170-E	トリガのためのメッセージ優先度の下限値がキューマネージャ定義で指定した優先度の最大値を超えています。	標準エラー出力
KFCA26171-E	省略時のメッセージ優先度がキューマネージャ定義で指定した優先度の最大値を超えています。	標準エラー出力
KFCA26174-E	キューファイル名が重複して指定されています。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力

出力メッセージID	内容	出力先
KFCA26190-E	動的共用メモリが足りないため、入出力バッファが確保できません。 要求メモリ量=オーバーフローのため表示できません。	標準エラー出力
KFCA31023-E	キューファイルとバックアップキューファイルの指定数が異なっています。	標準エラー出力
KFCA31024-E	キューファイルとバックアップキューファイルの内容が不一致です。	標準エラー出力
KFCA31035-E	指定したキューファイルグループは、キューファイル二重化構成のため -b オプションを指定してください。	標準エラー出力
KFCA31038-I	キューファイル、およびバックアップキューファイルの組み込みが完了しました。	メッセージログファイル
KFCA31040-E	指定したキューファイルとバックアップキューファイルのキューファイル二重化構成が正しくありません。	標準エラー出力
KFCA31041-E	指定したバックアップキューファイルはキューファイルに対するバックアップキューファイルではありません。	標準エラー出力
KFCA31042-E	指定したキューファイルグループは、キューファイル二重化構成ではないため -b オプションは指定できません。	標準エラー出力
KFCA31050-E	モデルキューの定義に指定したキューファイル名がバックアップキューファイルのため指定できません。	標準エラー出力
KFCA31067-E	キューマネージャが配布リストをサポートしていないため指定したキューファイル内のキューに対して配布リストを指定できません。	標準エラー出力
KFCA31068-E	キューマネージャが配布リストをサポートしていないため指定したキューファイルとバックアップキューファイル内のキューに対して配布リストを指定できません。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力
KFCA31176-E	不正なメッセージがキューファイルに存在します。	標準エラー出力
KFCA31177-E	MQA サービス専用共用メモリの領域が確保できません。	メッセージログファイル
KFCA31179-E	MQA サービス専用共用メモリの拡張領域が不足したため、指定したキューファイルは組み込めません。	メッセージログファイル
KFCA31180-E	MQA サービス専用共用メモリが参照できません。	メッセージログファイル

mqabkup (キューファイルのバックアップ)

形式

```
mqabkup キューファイル名 ファイル名
```

機能

指定したキューファイル (メッセージ情報管理ファイル (キューファイル名.cnt), およびメッセージファイル (キューファイル名.msg)) の内容を, 指定したファイル (メッセージ情報管理ファイル (ファイル名.cnt), およびメッセージファイル (ファイル名.msg)) にバックアップします。

バックアップしたキューファイルの内容は, mqarstr コマンドでリストアできます。

指定したキューファイルが, オンラインで使用中的の場合, mqabkup コマンドはエラーになります。

コマンド引数

● キューファイル名

~< 1~59 文字の文字列 >

バックアップ元のキューファイルの名称を完全パス名で指定します。

● ファイル名

~<パス名>

バックアップ先のファイルの名称を指定します。

注意事項

指定したファイルがすでに存在する場合, 指定したファイルのメッセージ情報管理ファイル (ファイル名.cnt), およびメッセージファイル (ファイル名.msg) に書き込み権があれば上書きします。メッセージ情報管理ファイル (ファイル名.cnt), およびメッセージファイル (ファイル名.msg) のうち, どちらか一つでもファイルに書き込み権がない場合, mqabkup コマンドはエラーになります。

また, キューファイルをバックアップ後, キューファイルを拡張して作成しリストアしても, キューファイルは拡張されません。

mqachgque (キューの属性の変更)

形式

```
mqachgque [-e] [-c クラスタ内共用クラスタ名称] ローカルキュー名  
          モデルキューの定義名 キュー属性定義ファイル名
```

機能

キュー属性定義ファイルに格納されているモデルキューの属性指定定義を基にローカルキューの属性を変更します。

モデルキューの定義の形式は、mqaqueatl 定義コマンドと同様の形式で指定します。詳細については、4章の「mqaqueatl (モデルキューの属性定義)」を参照してください。

キュー属性定義ファイルには、複数のモデルキューの属性指定定義を格納できます。その場合は、モデルキューの属性指定定義に指定するモデルキューの定義名をすべて異なる名称にしてください。

このコマンドでキューの属性を変更する場合、mqaqueatl 定義コマンドに記載してあるオプションのうち、モデルキューの定義名と変更する属性、およびキューファイル名だけを指定してください。そのほかの属性は省略してください。なお、定義タイプは、このコマンドでは変更できません。

また、-e オプション指定時にだけ変更できる属性があります。-e オプションを指定しないで該当する属性を変更したい場合には、キューを再作成してください。

オプション

● -e

変更不可キュー属性の変更許可を指定します。

次に示す情報を変更する場合に指定してください。

- 最大メッセージ登録数 (-d オプション)
- 最大メッセージ長 (-l オプション)
- キューの使用方法 (-u オプション)

ただし、最大メッセージ登録数 (-d オプション) および最大メッセージ長 (-l オプション) は元の値より小さい値に変更できません。

現在の最大メッセージ登録数 (-d オプション) は、次に示すどちらかで表示される値で確認してください。

- mqainq コマンドの出力の"最大メッセージ登録数"
- KFCA31105-E メッセージの"元の最大メッセージ登録数"

現在の最大メッセージ長 (-l オプション) は、KFCA31106-E メッセージの"元の最大メッセージ長"で確認してください。

● -c クラスタ内共用クラスタ名称

～< 1～48 文字の MQ 文字列>

キューが属するクラスタの名称を指定します。

省略された場合は、キューはクラスタには属しません。

コマンド引数

● ローカルキュー名

～< 1～48 文字の MQ 文字列>

属性を変更するキューの名称を指定します。

● モデルキューの定義名

～< 1～48 文字の MQ 文字列>

定義ファイル内に定義されたモデルキューの定義名を指定します。

● キュー属性定義ファイル名

～<パス名>

キュー属性定義（モデルキューの属性指定定義）が格納されているファイルのパス名を指定します。

注意事項

- このコマンドは、次に示すどちらかの場合にエラーになります。
 - 指定されたキューファイルがオンラインで使用されている場合
 - 指定されたキュー名がキューファイル上に存在しない場合
- このコマンドで変更できるキューは、ローカルキューだけです。
- e オプションを指定してキューの使用方法 (-u オプション) を変更したキューについて、次に示す定義を見直してください。
 - リモートキューのローカル定義
転送キュー名に指定したキューの使用方法は transmission（転送キュー）にしてください。
 - モデルキューの属性定義
イニシエーションキュー名に指定したキューの使用方法は normal（通常キュー）にしてください。
- c オプションを指定したキューの使用方法は normal（通常キュー）にしてください。

5. キューファイル二重化構成の場合のバックアップキューファイル名 (B系ファイル名) 用の mqaqueatl 定義は, MQA サービス定義ファイルに作成しないでください。

mqaconvert (キューファイルのコンバート)

形式

```
mqaconvert [-b バッファサイズ]
            コンバート元ファイル名 コンバート先ファイル名
```

機能

旧バージョンの mqabkup コマンドでバックアップしたキューファイル (メッセージ情報管理ファイル (キューファイル名.cnt), およびメッセージファイル (キューファイル名.msg)) を, OpenTP1 ファイルシステム上の最新バージョンのキューファイルへコンバートします。

このコマンドでコンバートできるのは, TP1/Message Queue 05-00 以降で作成したバックアップファイルです。

オプション

● -b バッファサイズ

~((2048~200000000))《65536》(単位:バイト)

キューファイルのコンバート時に使用するバッファサイズ (プロセス領域サイズ) を指定します。

なお, バッファサイズがキューファイルのレコード長の 2 倍未満である場合, KFCA31112-E メッセージを出力して終了します。そのためバッファサイズを指定する場合は, KFCA31112-E メッセージのレコード長の 2 倍以上の値を指定してください。

コマンド引数

● コンバート元ファイル名

~<パス名>

旧バージョンの mqabkup コマンドでバックアップしたキューファイルの名称を完全パス名で指定します。

● コンバート先ファイル名

~<1~59 文字の文字列>

コンバート先のファイル名称を完全パス名で指定します。ファイルは, OpenTP1 ファイルシステム上のファイルを指定してください。同一ファイル名のファイルがある場合はエラーになります。事前に削除してください。

注意事項

1. TP1/Message Queue 05-13 以前のキューファイルから, TP1/Message Queue 05-14 以降のキューファイルへコンバートすると, 次の場合に, キューファイルのサイズが大きくなる可能性があります。

- コンバート元のキューファイルのメッセージ数が1で、メッセージ長が次に示す計算式よりも大きいとき

レコード長-520 (単位: バイト)

- コンバート先のキューファイルシステムのセクタ長が、コンバート元のキューファイルシステムのセクタ長よりも大きいとき
 - コンバート元のキューファイルに長さが0バイトのメッセージが多数格納されているとき
2. TP1/Message Queue 05-13 以前のキューファイルから、TP1/Message Queue 05-14 以降のキューファイルへコンバートすると、次の場合に、コンバート先のキューファイルのメッセージ数 (mqainit コマンドの -m オプションに相当するキューファイルに登録できるメッセージの最大数) が多くなる場合があります。

- コンバート元のキューファイルに長さが0バイトのメッセージが多数格納されているとき
- コンバート元のキューファイルのメッセージ長 (mqainit コマンドの -s オプションに相当するキューファイルに登録できるメッセージの平均メッセージ長) がレコード長-520 バイト以下で指定されていて、次に示す範囲に該当するメッセージ長のメッセージが多数格納されているとき

$(\text{レコード長}-8) \times N^{\ast} \sim (\text{レコード長}-8) \times N^{\ast}-512+1$ (単位: バイト)

注※

N: 正の整数

なお、コンバート先のキューファイルのメッセージ数を確認するには、OpenTP1 の fills コマンドで「コンバート先のキューファイル名.msg」のファイルのレコード数を参照してください。

3. TP1/Message Queue 05-13 以前のキューファイルから、TP1/Message Queue 05-14 以降のキューファイルへコンバートすると、次の場合に、メッセージが使用するレコード数が一つ多くなります。
- コンバート元のキューファイルのメッセージ長 (mqainit コマンドの -s オプションに相当するキューファイルに登録できるメッセージの平均メッセージ長) がレコード長-520 バイト以下で指定されていて、次に示す範囲に該当するメッセージ長のメッセージが格納されているとき

$(\text{レコード長}-8) \times N^{\ast} \sim (\text{レコード長}-8) \times N^{\ast}-512+1$ (単位: バイト)

注※

N: 正の整数

必要に応じて MQA サービス定義の mqa_max_msg_recnum オペランドの値を変更してください。

4. TP1/Message Queue 06-00 以前のキューファイルから、06-01 以降のキューファイルへコンバートすると、メッセージ情報管理ファイルに1レコード追加されます。

mqadel (キューファイルの削除)

形式

```
mqadel キューファイル名
```

機能

mqainit コマンドで、割り当てたキューファイル (メッセージ情報管理ファイル (キューファイル名.cnt) と、メッセージファイル (キューファイル名.msg)) を削除します。

指定したメッセージファイルがオンラインで使用中の場合は、削除できません。この場合、エラーメッセージが出力されます。

コマンド引数

● キューファイル名

～< 1～59 文字の文字列 >

削除するキューファイルの名称を完全パス名で指定します。

mqadelsmsg (メッセージの削除)

形式

```
mqadelsmsg [ {-a | -n メッセージ番号} ] キュー名
```

機能

指定したキュー名に登録されているメッセージを削除します。

オプション

● -a

指定したキュー名に登録されているすべてのメッセージを削除します。

● -n メッセージ番号

~< 1~2147483647 >

指定したキュー名に登録されている、指定したメッセージ番号のメッセージを削除します。

指定したキュー名に登録されているメッセージの先頭からの番号を指定します。

コマンド引数

● キュー名

~< 1~48 文字の MQ 文字列 >

削除するメッセージが登録されているキュー名を指定します。

なお、次に示す種類のキューは指定できません。

- リモートキュー
- モデルキュー

注意事項

1. -a と -n オプションが指定されていない場合、キューに登録されている先頭のメッセージを削除します。
2. mqalsmsg コマンドで参照した内容と同期を取ってメッセージの削除をする場合は、mqalsmsg コマンド実行後にアプリケーションからのメッセージ登録/取り出し、および転送をするとメッセージの順序が異なってしまうため、アプリケーションでのメッセージ登録/取り出し、および転送をしない状態で実行してください。
3. このコマンドは MQOPEN 命令および MQGET 命令を使用してメッセージの削除を実行しているため、指定したキューの属性によってはコマンドが異常になるか、アプリケーションまたはチャンネルが発行する MQOPEN 命令および MQGET 命令が異常になる可能性があります。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA04310-E	コマンドに指定されたパラメタの値が誤っています。	標準エラー出力
KFCA04323-I	登録されているメッセージを削除しました。	メッセージログファイル
KFCA04331-E	キュー名の文字数が 48 文字を超えています。	標準エラー出力
KFCA04337-E	指定したキュー名は存在しません。	標準エラー出力
KFCA04369-E	キュー名に指定できない文字を指定しています。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA26192-I	ヘルプメッセージ	標準出力
KFCA31030-E	コマンド実行中に異常を検知しました。	標準エラー出力
KFCA31044-E	指定したキュー名は指定できない種類のキューです。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqadelque (キュー, メッセージの削除)

形式

```
mqadelque [-m] ローカルキュー名  
          キューファイル名 [[キューファイル名] ...]
```

機能

指定したキューファイルから指定したキューを削除します。

-m オプションを指定した場合は、ローカルキュー名で指定したキューに登録されているメッセージも削除します。指定したキューファイルにキューがない場合、ローカルキュー名で指定したキューに登録されているメッセージだけを削除します。

このコマンドで削除できるキューは、定義済みキュー (mqamkque コマンドで作成したキュー)、および永続的動的キュー (MQOPEN 命令でモデルキューの定義を指定し作成した永続的キュー) の2種類です。

オプション

● -m

指定したローカルキューに登録されているメッセージを削除します。

コマンド引数

● ローカルキュー名

～< 1～48 文字の MQ 文字列>

削除するローカルキューの名称を指定します。

● キューファイル名

～< 1～59 文字の文字列>

削除するキューまたはメッセージが格納されているキューファイルの名称を完全パス名で指定します。

一度オンラインで使用したキューファイルを指定する場合、同一キューファイルグループのすべてのキューファイル名を指定してください。

注意事項

1. このコマンドは、次に示すどれかの場合にエラーになります。
 - 指定したキューにメッセージが登録されている場合に -m オプションを指定しないとき
 - 指定したキューファイルがオンラインで使用されている場合
 - 指定したキューが一時的動的キューの場合

2. 一度オンラインで使用したキューファイルを指定する場合、同一キューファイルグループのすべてのキューファイルを指定する必要があります。指定しないキューファイルがあった場合、指定しないキューファイルに格納されているメッセージが削除されない状態になります。この状態で MQA サーバを開始すると、登録するキューがないメッセージが存在するため、開始処理は失敗します。開始処理が失敗した場合、メッセージが格納されているキューファイルを指定して、このコマンドを入力してください。登録するキューがないメッセージを削除します。

mqadf (キューファイルグループの使用状況表示)

形式

mqadf キューファイルグループ名

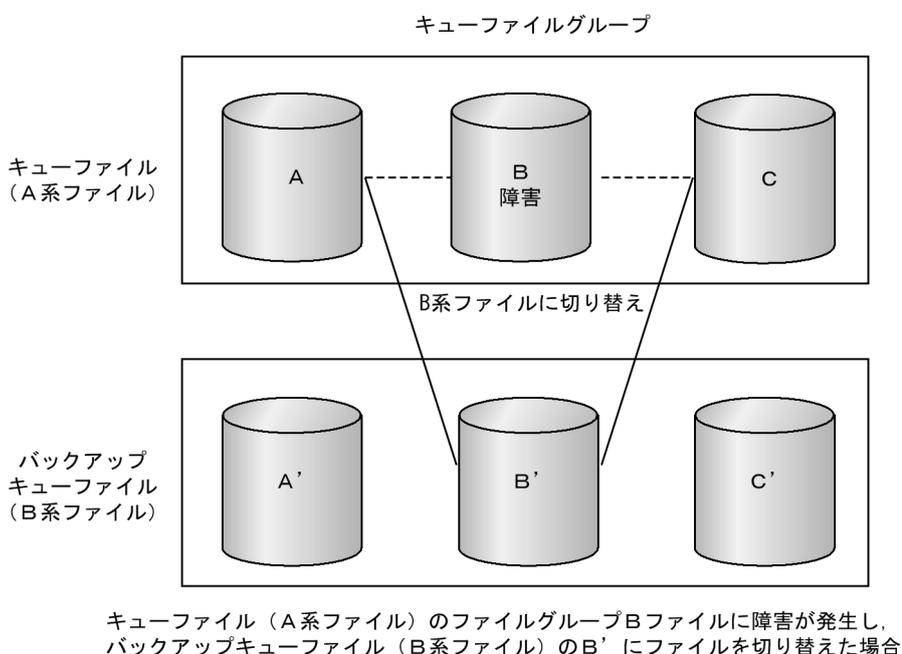
機能

指定したキューファイルグループの使用状況を表示します。

使用状況を表示できるのは、キュー/メッセージの全容量、使用容量、および空き容量です。キューファイルグループの全容量とは、キューファイルグループに属するすべてのキューファイルの容量の合計です。

キューファイル二重化構成を行っているキューファイルグループを指定した場合、キューファイルグループが次に示す図の状態のとき A/B'/C ファイルが対象になります。

図 6-1 ファイルを切り替えた場合



コマンド引数

● キューファイルグループ名

～< 1～31 文字の識別子 >

使用状況を表示するキューファイルグループ名を指定します。

出力形式

全キューの個数 = aa....aa
使用中キューの個数 = bb....bb
未使用キューの個数 = cc....cc

```
グループ形態 = dd....dd
メッセージ格納域全容量 = ee....eeGB + ff....ffB
メッセージ格納域使用容量 = gg....ggGB + hh....hhB
メッセージ格納域空き容量 = ii....iiGB + jj....jjB
```

aa....aa

キューファイルグループ内の全キューの個数

bb....bb

キューファイルグループ内の使用中キューの個数

cc....cc

キューファイルグループ内の未使用キューの個数

dd....dd

キューグループ形態

single : 一重化キューファイルグループ

dual : 二重化キューファイルグループ

ee....ee, ff....ff

キューファイルグループ内のメッセージ格納域全容量 (単位: GB=ギガバイト, B=バイト)

gg....gg, hh....hh

キューファイルグループ内のメッセージ格納域使用容量 (単位: GB=ギガバイト, B=バイト)

ii....ii, jj....jj

キューファイルグループ内のメッセージ格納域空き容量 (単位: GB=ギガバイト, B=バイト)

注意事項

1. このコマンドは、障害閉塞状態または論理閉塞状態になったキューファイルを含むキューファイルグループでも指定できます。ただし、障害または論理閉塞になったキューファイルの容量は、このコマンドで表示する値には含まれません。
2. このコマンドで出力するキューファイルグループ内のメッセージ格納容量は、一つのメッセージで1レコードずつ使用する場合の容量です。MQA サービス定義の `mqa_max_msg_recnm` オペランドに2以上を指定し、キューファイルのレコード長[※]よりも大きいメッセージを格納する場合は、メッセージのレコード分割が行われるため、このコマンドで表示されたメッセージ格納域空き容量よりも実際は大きな領域が使用できます。一つのメッセージを複数レコードに分割して格納する場合、2レコード目からは、空き容量に512バイトずつ加算した領域が使用されます。

注※

レコード長は、`mqainit` コマンドの `-s` オプション (メッセージ長) と、キューファイルのセクタ長で算出した値です。レコード長の計算については、6章の「`mqainit` (キューファイルの割り当て)」を参照してください。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04322-E	キューファイルグループ名の文字数が 31 文字を超えています。	標準エラー出力
KFCA26100-I	ヘルプメッセージ	標準出力
KFCA26151-E	指定したキューファイルグループ名はありません。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqafilinf (キューファイル内の情報表示)

形式

```
mqafilinf [-q キュー名 | -x] キューファイル名
```

機能

オフラインの状態では、指定したキューファイルの管理部の情報を表示します。

指定したキューファイルがオンラインで使用中の場合、mqafilinf コマンドはエラーになります。

オプション

● -q キュー名

～< 1～48 文字の MQ 文字列 >

ローカルキュー情報を出力するキューの名称を指定します。

ローカルキュー情報には、キューの状態に関係なく、キューファイルの管理部がそのまま表示されます。このため、削除したキュー名を指定した場合でも、情報が表示されることがあります。

指定したキュー名が見つからない場合は、次に示すメッセージが出力されます。

```
「KFCA31172-E 指定した情報は存在しません。」
```

● -x

キューファイルについてキュー名一覧を表示する場合に指定します。

削除したキューは、キュー名一覧に表示されません。表示するキュー名がない場合は、次に示すメッセージが出力されます。

```
「KFCA31172-E 指定した情報は存在しません。」
```

コマンド引数

● キューファイル名

～< 1～59 文字の文字列 >

キューファイル情報を出力するキューファイル名を完全パス名で指定します。

出力形式

● 管理部の情報

```
キューファイル名 = aa....aa
全キューの個数 = bb....bb
使用中キューの個数 = cc....cc
未使用キューの個数 = dd....dd
最大メッセージ登録数 = ee....ee
使用中メッセージ数 = ff....ff
残登録可能メッセージ数 = gg....gg
メッセージ長 = hh....hh
メッセージ情報管理ファイルのレコード長 = ii....ii
メッセージ情報管理ファイルのレコード数 = jj....jj
メッセージファイルのレコード長 = kk....kk
メッセージファイルのレコード数 = ll....ll
```

aa....aa

指定したキューファイルの名称 (1~59 文字の文字列)

bb....bb

全キューの個数 (10 進数)

cc....cc

使用中キューの個数 (10 進数)

dd....dd

未使用キューの個数 (10 進数)

ee....ee

mqainit コマンドで指定したメッセージ数 (10 進数)

使用中メッセージ数と残登録可能メッセージ数を足した値ではありません。

ff....ff

使用中メッセージ数 (10 進数)

gg....gg

残登録可能メッセージ数 (10 進数)

登録されているメッセージのメッセージ長が、mqainit コマンドの-s オプションで指定した値より大きい場合、一つのメッセージで複数のレコードが使用されます。そのため、実際に登録できるメッセージ数は残登録可能メッセージ数と異なることがあります。

hh....hh

mqainit コマンドで指定したメッセージ長 (10 進数)

ii....ii

メッセージ情報管理ファイルのレコード長 (10 進数)

jj....jj

メッセージ情報管理ファイルのレコード数 (10 進数)

kk...kk

メッセージファイルのレコード長 (10 進数)

ll...ll

メッセージファイルのレコード数 (10 進数)

● キュー名一覧の情報

```
キュー名 = aa....aa
```

aa...aa

キュー名 (1~48 文字の MQ 文字列)

● ローカルキュー情報

```
キュー名 = aa....aa
キュー記述子 = bb....bb
取り出し許可 = cc....cc
登録許可 = dd....dd
省略時のメッセージ優先度 = ee....ee
省略時のメッセージ永続性 = ff....ff
キューの有効範囲 = gg....gg
プロセス名 = hh....hh
最大メッセージ登録数 = ii....ii
最大メッセージ長 = jj....jj
共用性 = kk....kk
省略時の取り出しオプション = ll....ll
ロールバック回数記録方法 = mm....mm
メッセージ配布順序 = nn....nn
キュー保持時間 = oo....oo
定義タイプ = pp....pp
使用種別 = qq....qq
メッセージ登録数 = rr....rr
作成日付 = YYYY-MM-DD
作成時刻 = HH.MM.SS
イニシエーションキュー名 = ss....ss
トリガ制御 = tt....tt
トリガタイプ = uu....uu
トリガのためのメッセージ優先度の下限値 = vv....vv
トリガのためのメッセージ登録数の下限値 = ww....ww
トリガデータ = xx....xx
配布リスト = yy....yy
デフォルトのバインディング = zz....zz
クラスタ内共用クラスタ名称 = a1....a1
```

aa...aa

キュー名 (1~48 文字の MQ 文字列)

bb...bb

キュー記述子 (1~64 文字の文字列)

cc....cc

取り出し許可（次に示すどちらかを表示）

INHIBITED：取り出しを禁止します。

ALLOWED：取り出しを許可します。

dd....dd

登録許可（次に示すどちらかを表示）

INHIBITED：登録を禁止します。

ALLOWED：登録を許可します。

ee....ee

省略時のメッセージ優先度（10進数）

ff....ff

省略時のメッセージ永続性（次に示すどちらかを表示）

PERSISTENT：永続メッセージにします。

NOT PERSISTENT：永続メッセージにしません。

gg....gg

キューの有効範囲（次に示す値を表示）

QUE MGR：キュー定義はキューマネージャスコープを持ちます。

hh....hh

プロセス名（1～48文字のMQ文字列）

ii....ii

最大メッセージ登録数（10進数）

jj....jj

最大メッセージ長（10進数，単位：バイト）

kk....kk

共用性（次に示すどちらかを表示）

SHAREABLE：キューを共用できます。

NOT SHAREABLE：キューを共用できません。

ll....ll

省略時の取り出しオプション（次に示すどちらかを表示）

INPUT EXCLUSIVE：排他アクセスのメッセージ取り出し用にオープンします。

INPUT SHARED：共有アクセスのメッセージ取り出し用にオープンします。

mm....mm

ロールバック回数記録方法（次に示す値を表示）

NOT HARDENED：ロールバック回数を記録しません。

nn....nn

メッセージ配布順序（次に示すどちらかを表示）

PRIORITY：優先度の順序でメッセージが返されます。

FIFO：FIFO（先入れ先出し）の順序でメッセージが返されます。

oo....oo

キュー保持時間（10 進数，単位：時間）

pp....pp

定義タイプ（次に示すどれかを表示）

PREDEFINED：定義済みキュー

PERMANENT DYNAMIC：永続的動的キュー

TEMPORARY DYNAMIC：一時的動的キュー

qq....qq

使用種別（次に示すどちらかを表示）

NORMAL：通常キュー。通常のアプリケーションがメッセージを登録，または取り出しをする場合に使用します。

TRANSMISSION：転送キュー。リモートキューマネージャあてのメッセージを保持するために使用します。

rr....rr

メッセージ登録数（10 進数）

YYYY-MM-DD

作成日付（年-月-日）

作成日付はシステムの設定時刻で表示されます。

HH.MM.SS

作成時刻（時.分.秒）

作成時刻はシステムの設定時刻で表示されます。

ss....ss

イニシエーションキュー名（1～48 文字の MQ 文字列）

tt....tt

トリガ制御（次に示すどちらかを表示）

OFF：トリガメッセージを要求しません。

ON：トリガメッセージを要求します。

uu....uu

トリガタイプ（次に示すどれかを表示）

NONE：トリガメッセージは通知されません。

FIRST：空でない場合に通知されます。

EVERY：メッセージごとに通知されます。

DEPTH：一定の格納数を超えた場合に通知されます。

VV....VV

トリガのためのメッセージ優先度の下限値（10 進数）

WW....WW

トリガのためのメッセージ登録数の下限値（10 進数）

XX....XX

トリガデータ（1～64 文字の文字列）

yy....yy

配布リスト（次に示すどちらかを表示）

SUPPORTED：配布リストに対応しています。

NOT SUPPORTED：配布リストに対応していません。

ZZ....ZZ

デフォルトのバインディング（次に示すどちらかを表示）

FIXED：キューがオープンされたハンドルにあて先を固定します。

NOT FIXED：キューにメッセージが登録される場合、メッセージ単位にあて先を決定します。

al....al

クラスタ内共用クラスタ名称（1～48 文字の MQ 文字列）

注意事項

1. オンラインで使用したキューファイルにこのコマンドを実行した場合、オンラインの終了状態によっては、終了直前のメッセージ数とコマンド実行結果のメッセージ数は異なることがあります。
2. 存在しないキューファイル名を指定した場合は、次に示すメッセージが出力されます。

「KFCA04339-E 指定されたキューファイル名は存在しません。」

mqafills (キューファイルの情報表示)

形式

```
mqafills {-x | -f キューファイル名}
```

機能

キューファイル名の一覧または指定したキューファイルの情報を表示します。

オプション

● -x

MQA サービスが使用中の、すべてのキューファイル名を一覧で出力する場合に指定します。

● -f キューファイル名

～< 1～59 文字の文字列 >

情報を表示するキューファイルの名称を完全パス名で指定します。

出力形式

● キューファイル名の一覧

```
キューファイル名 = aa....aa  
                  :
```

aa....aa

キューファイルの名称 (1～59 文字の文字列)

使用中のすべてのキューファイル名が出力されます。

● キューファイルの情報

```
キューファイル名 = aa....aa  
キューファイル状態 = bb....bb  
キューファイルグループ名 = cc....cc  
全キューの個数 = dd....dd  
使用中キューの個数 = ee....ee  
未使用キューの個数 = ff....ff  
メッセージ長 = gg....gg  
メッセージ情報管理ファイルのレコード長 = hh....hh  
メッセージ情報管理ファイルのレコード数 = ii....ii  
ファイル形態 = jj....jj  
MQAサービス専用共用メモリ確保要求者識別子 = kk....kk
```

aa....aa

指定したキューファイルの名称 (1～59 文字の文字列)

bb....bb

キューファイルの状態

normal : 正常状態

logical shutdown : 論理閉塞状態

failure shutdown : 障害閉塞状態

temporary failure shutdown (C) : 一時的障害閉塞 (C) 状態

temporary failure shutdown (CR) : 一時的障害閉塞 (CR) 状態

temporary failure shutdown (M) : 一時的障害閉塞 (M) 状態

temporary failure shutdown (C/M) : 一時的障害閉塞 (C/M) 状態

temporary failure shutdown (CR/M) : 一時的障害閉塞 (CR/M) 状態

cc....cc

指定したキューファイルが組み込まれているキューファイルグループの名称 (1~31 文字の識別子)

dd....dd

全キューの個数 (10 進数)

ee....ee

使用中キューの個数 (10 進数)

ff....ff

未使用キューの個数 (10 進数)

gg....gg

mqainit コマンドで指定したメッセージ長 (10 進数)

hh....hh

メッセージ情報管理ファイルのレコード長 (10 進数)

ii....ii

メッセージ情報管理ファイルのレコード数 (10 進数)

jj....jj

ファイル形態

snormal : 一重化キューファイル

dnormal : 二重化キューファイル (A 系キューファイル)

dbackup : 二重化バックアップキューファイル (B 系キューファイル)

kk....kk

MQA サービス専用共用メモリ確保要求者識別子 (00 から ff の 16 進数)

dcshmls -r コマンドで表示される共用メモリプールの確保要求者種別は、MQA サービス専用共用メモリ確保要求者識別子の先頭に 'm' を付加した値になります。dcshmls コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

注意事項

-f オプション指定時に出力されたキューファイルの状態が論理閉塞状態または一時的障害閉塞状態の場合だけ、mqarles コマンドが入力できます。

また、キューファイルの状態が論理閉塞状態、障害閉塞状態、または一時的障害閉塞状態の場合だけ mqarm コマンドが実行可能です。

障害閉塞状態 (failure shutdown) のキューファイルに対してこのコマンドを実行した場合、「全キューの個数」などの情報を正しく表示できないことがあります。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04211-E	キューファイル名の文字数が 59 文字を超えています。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA04339-E	指定したキューファイル名は存在しません。	標準エラー出力
KFCA04357-I	ヘルプメッセージ	標準出力
KFCA04368-E	指定されたキューファイルは存在しません。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqafrc (キューファイルの回復)

形式

```
mqafrc [-s] [-e] [-g] [-k キー] [-m]
        -f 回復対象キューファイル定義ファイル名
        ジャーナルファイル名 [[ ジャーナルファイル名] ...]
```

機能

指定したジャーナルファイルを入力として、キューファイルを回復します。

詳細については、「[2.2.2\(11\) キューファイルの回復](#)」を参照してください。

オプション

● -s

前回の MQA FRC を引き継ぎません。

このオプションを省略すると、前回の MQA FRC が引き継がれます。

このオプションは、jnlcolc コマンドの -f オプションに対応します。

● -e

MQA FRC 終了時に引き継ぎファイルを削除します。

このオプションを指定して MQA FRC を実行した場合、次回の MQA FRC 実行時には、必ず -s オプションを指定してください。

このオプションを省略すると、引き継ぎファイルは削除されません。

このオプションは、jnlcolc コマンドの -l オプションに対応します。

● -g

-s オプションの指定があり、かつジャーナル世代番号が 1 であるアンロードジャーナルファイルの指定がない場合でも、MQA FRC を実行します。その場合、-g オプションを省略すると、処理は中断されます。

-s オプションを省略した場合、このオプションを指定しても無視されます。

● -k キー

～< 3 けたの符号なし整数 > ((001～999)) 《001》

引き継ぎファイルの名称の一部を指定します。

OpenTP1 は、このオプションの指定値に応じてカレントディレクトリに引き継ぎファイルを作成します。引き継ぎファイルの名称は"jnlcolc***"および"mqafrc***"です。"***"にはこのオプションの指定値が設定されます。

複数の MQA FRC を同時に実行する場合は、それぞれ別のキーになるように指定してください。また、damfrc コマンドまたは tamfrc コマンドをこのコマンドと同時または連続して実行する場合にもそれぞれ別のキーになるように指定してください。前回の MQA FRC を引き継ぐ場合は、前回指定したキーを指定してください。

このオプションは、jnlcolc コマンドの-c オプションに対応します。

● -m

キューファイルの回復に必要なジャーナルレコードをファイル上に集積します。

このオプションを省略すると、メモリ上に確保されたバッファにジャーナルレコードが集積されます。

このオプションは、jnlcolc コマンドの-m オプションに対応します。

● -f 回復対象キューファイル定義ファイル名

～<パス名>

回復するキューファイルを定義したファイルの名称を指定します。

回復対象キューファイル定義ファイルは次に示す形式で、テキストエディタなどで作成します。

```
{ { mqafilmap 回復対象キューファイル名 [回復先キューファイル名] }
```

- 回復対象キューファイル名

～<1~59文字の文字列>

回復するキューファイルの名称を完全パス名で指定します。

- 回復先キューファイル名

～<1~59文字の文字列>

回復先キューファイル名を指定する場合は完全パス名で指定します。

このオプションを省略すると、回復対象キューファイル名に指定したファイルが回復先になります。

コマンド引数

● ジャーナルファイル名

～<パス名>

MQA FRC 実行時に使用するアンロードジャーナルファイルの名称を指定します。複数世代のジャーナルを処理する場合は、複数のジャーナルファイルを指定します。

注意事項

1. mqafrfc コマンドの指定が誤っている場合、メッセージ ID が付いていないメッセージ、および mqafrfc コマンドの使用方法が出力されることがあります。
2. mqafrfc コマンドは、jnlcolc コマンドを内部で使用しています。そのため、jnlcolc コマンドについてのエラーメッセージが出力されることがあります。対策については、マニュアル「OpenTP1 メッセージ」を参照してください。
3. mqafrfc コマンドはカレントディレクトリにテンポラリファイルを作成することがあります。そのため、カレントディレクトリには書き込み権を設定してください。テンポラリファイルのディスク容量の計算式を次に示します。

最大 4096+aa...aa (単位: バイト)

- aa...aa: mqafrfc コマンド実行時に指定したアンロードジャーナルファイルの総ディスク容量
- なお、アンロードジャーナルファイルの総ディスク容量は、UNIX の ls コマンドなどで参照できます。コマンド引数に複数のアンロードジャーナルファイルを指定した場合は、その合計になります。
4. -s オプションを指定しないで MQA FRC を実行する場合、つまり 2 回目以降に mqafrfc コマンドを実行する場合の回復対象キューファイル定義は、前回に MQA FRC を実行した時の内容と同じにする必要があります。2 回目以降の mqafrfc コマンドを実行する時に、回復対象キューファイル定義ファイルを変更した場合の結果は保証できません。
 5. 複数キューファイルで構成されたキューファイルグループを使用する場合は、必ずすべてのキューファイルの回復を実行してください。
 6. キューファイルの障害要因を調査し、キューファイル格納媒体損傷の場合は、MQA サービス定義の mqa_jnl_conf オペランドに Y を指定して取得したアンロードジャーナルファイルを使用して MQA FRC を実行してください。mqa_jnl_conf オペランドに N を指定して取得したアンロードジャーナルファイルを使用して MQA FRC を実行すると、キューファイルの情報が不正になります。
 7. キューファイルの障害要因を調査し、キューファイル格納媒体損傷以外の場合は、MQA サービス定義の mqa_jnl_conf オペランドに Y または N を指定して取得したアンロードジャーナルファイルを使用して MQA FRC を実行できます。
 8. mqafrfc コマンドで指定するアンロードジャーナルファイルのバージョンと、mqafrfc コマンドを実行する TP1/Message Queue のバージョンは同じである必要があります。
 9. キューファイルグループが複数のキューファイルで構成されている場合、mqarm コマンドで切り離す前に、mqagrprocs コマンドを実行する必要があります。
 10. mqafrfc コマンドで指定するアンロードジャーナルファイルは、永続的動的キューにアクセスしたことによって取得されたジャーナルレコードを含んだものを使用しないでください。永続的動的キューにアクセスしたことによって取得されたジャーナルレコードを含んだアンロードジャーナルファイルを使用した場合、mqafrfc コマンドによって回復されたキューファイルがオンラインで使用できなくなる場合があります。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA31095-E	コマンド引数が不正です。	標準出力
KFCA31096-E	回復処理中に内部不一致が発生しました。	標準出力
KFCA31097-I	ヘルプメッセージ	標準出力

mqagrps [-r] [-g キューファイルグループ名 | -f キューファイル名]

形式

```
mqagrps [-r]
        {-g キューファイルグループ名 | -f キューファイル名}
```

機能

キューファイルグループ内のすべてのキューファイルを障害閉塞状態に変更するために、キューファイルグループ名、または障害閉塞状態のキューファイル名を指定します。

キューファイル二重化構成を行っている、キューファイルグループ名または障害閉塞状態のキューファイル名を指定する場合は、組み込んだキューファイル（A系ファイル）およびバックアップキューファイル（B系ファイル）の両方のキューファイルを障害閉塞状態にする必要があります。

また、すでに障害閉塞状態のキューファイルがグループ内にある場合は、そのキューファイルについては状態を変更しません。

オプション

● -r

キューファイルの状態をチェックしないで、強制的にキューファイルグループ内のすべてのキューファイルを障害閉塞状態に変更します。

● -g キューファイルグループ名

～< 1～31 文字の識別子 >

キューファイルグループ名を指定します。キューファイルグループ内のすべてのキューファイルが障害閉塞状態になります。ただし、キューファイルグループ内には障害閉塞状態のキューファイルがなければいけません。障害閉塞状態のキューファイルがない場合、コマンドはエラーになります。

● -f キューファイル名

～< 1～59 文字の文字列 >

障害閉塞状態のキューファイル名を指定します。同じキューファイルグループ内のすべてのキューファイルが障害閉塞状態になります。指定したキューファイルが障害閉塞状態でない場合、コマンドはエラーになります。

注意事項

1. 指定したキューファイルグループ名、または指定したキューファイル名が属するキューファイルグループ内のキューファイルに対するアクセスを停止してからこのコマンドを入力してください。キューファイルグループ内のキューファイルに対してアクセス中にこのコマンドを実行すると、キューファイル障害が発生した状態と同じ状態になり、アクセス中のアプリケーションはエラーになります。

2. キューファイルグループ内に論理閉塞状態のキューファイルがある場合、そのキューファイルの状態を障害閉塞状態に変更します。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04211-E	キューファイル名の文字数が 59 文字を超えています。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA04322-E	キューファイルグループ名の文字数が 31 文字を超えています。	標準エラー出力
KFCA04325-E	指定したキューファイルは、二重化構成のため両系閉塞時以外は実行できません。	標準エラー出力
KFCA04339-E	指定されたキューファイル名は存在しません。	標準エラー出力
KFCA26151-E	指定したキューファイルグループ名はありません。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31113-I	ヘルプメッセージ	標準出力
KFCA31114-I	キューファイルグループ内のキューファイルを障害閉塞しました。	メッセージログファイル
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31116-E	指定したキューファイルは障害閉塞ではありません。	標準エラー出力
KFCA31117-E	指定したキューファイルグループ名に障害閉塞のキューファイルがありません。	標準エラー出力
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31121-E	指定したキューファイルグループは、二重化構成のため両系障害閉塞時以外は実行できません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqainit (キューファイルの割り当て)

形式

```
mqainit [-r] [-q キュー数] [-m メッセージ数]
         [-s メッセージ長] キューファイル名
```

機能

キューに登録したメッセージを格納するためのファイル (キューファイル) を割り当てます。

このコマンドで割り当てるキューファイルは、メッセージの管理情報を格納するメッセージ情報管理ファイルとメッセージ本体を格納するメッセージファイルの二つで構成されます。それぞれキューファイル名.cnt とキューファイル名.msg の名前が付けられ、一つの OpenTP1 ファイルシステム上に作成されます。

作成されるキューファイルのファイルの構造について、次の図に示します。

図 6-2 キューファイルの構造

キューファイル名を ABC と指定した場合

ABC.cnt



ABC.msg



オプション

● -r

指定したファイルがすでにあり、再びキューファイルを割り当てる場合に指定します。このオプションの指定を省略すると、指定したファイルがすでにあった場合にエラーになります。

また、このオプションを指定する場合、既存のキューファイルは削除されません。そのため、新しいキューファイルを作成するだけの空き容量が確保されていることを確認してください。空き容量がない場合、KFCA04384-E メッセージが出力されます。

● -q キュー数

~((1~20480))《32》

指定したキューファイルに登録するキューの最大数を指定します。

キュー数に 20480 を指定した場合、メッセージ数とメッセージ長は、それぞれ最大値を指定できません。指定値は、-s オプションの説明で示されている計算式で算出してください。

キュー数には、一時的および永続的なキューを合わせた数を指定してください。

● -m メッセージ数

～((1～1048560))《4096》

指定したキューファイルに登録するメッセージの最大数を指定します。

メッセージ数には、非永続性および永続性のメッセージを合わせた数を指定してください。

登録するメッセージのメッセージ長が-s オプションで指定した値より大きい場合、実際に登録できるメッセージ数は、このオプションで指定した値より少なくなります。

● -s メッセージ長

～((1～128000000))《100000》(単位：バイト)

指定したキューファイルに登録するメッセージの平均メッセージ長を指定します。この場合、指定した値を基にキューファイルのレコード長が決定されます。

レコード長の計算式を次に示します。

$$\left\lceil \frac{-s \text{ オプション指定値} + 8 + 512}{\text{セクタ長}^*} \right\rceil \times \text{セクタ長}^*$$

(凡例) $\left\lceil \right\rceil$: 小数点以下を切り上げます。

注※
セクタ長が1024バイト未満の場合は、1024バイトとして計算してください。

このオプションで指定した値より長いメッセージの登録もできます。ただし、その場合一つのメッセージで複数のレコードを使用するため、-m オプションで指定したメッセージ数より、実際に登録できるメッセージ数は少なくなります。最大で幾つのレコードを使用するかは、MQA サービス定義の mqa_max_msg_reclen オペランドで指定できます。

MQA サービスでは、1 レコードに複数のメッセージは格納しないため、メッセージ長が極端に異なるメッセージを一つのキューに格納すると媒体効率が低下します。このような場合、メッセージを登録するキューを分け、キューを格納するキューファイルグループも分けると媒体効率が向上します。

また、転送キューを作成するキューファイルを割り当てるときには、付加情報 (MQXQH 構造体) の長さを考慮した値を指定してください。

コマンド引数

● キューファイル名

～< 1～59 文字の文字列 >

キューファイルとして割り当てるファイル名を完全パス名で指定します。

キューファイル名に指定している最後の"/"以降の文字列長を 10 文字以内としてください。

キューファイルパス名およびキューファイル名にスペースを指定しないでください。また、ファイルパス名の"/"個所に"/"を追加したキューファイルパス名を指定しないでください。指定した場合、動作は保証されません。

注意事項

1. mqainit コマンドを入力する前に OpenTP1 ファイルシステムの初期設定をしていない場合は、初期設定をしてください。
 2. 複数のキューファイルを一つのキューファイルグループに対応づける場合、キューファイル名で指定する OpenTP1 ファイルシステムを物理的に別ディスクにすることをお勧めします。別ディスクにすると、一つのキューに複数のアプリケーションからの操作が重なった場合に、性能が良くなります。
 3. キューファイル名.cnt またはキューファイル名.msg の片方だけを削除した場合は、処理を保証しません。
 4. -m, -s オプションで指定した値を基に、TP1/Message Queue に必要な共用メモリを、MQA サービス専用共用メモリに確保します。したがって、-m, -s オプションでは、MQA サービス専用の共用メモリの算出式を基に値を指定してください。
 5. mqainit コマンドを入力する前に、キャラクタ型スペシャルファイルまたは UNIX ファイルのアクセス権は、所有者およびグループにリード・ライト権を付与してください。また、OpenTP1 ファイルシステムのアクセス権はユーザ用に設定してください。アクセス権を正しく設定していない場合、KFCA31184-E メッセージが出力されて OpenTP1 の起動に失敗します。
 6. キューファイルの最大格納容量は、メッセージ情報管理ファイルとメッセージファイルの合計で、2 ギガバイト以内です。最大格納容量を超えてキューファイルを作成した場合、動作は保証しません。そのため、キューファイルを作成したあとに、メッセージ情報管理ファイルとメッセージファイルの合計が 2 ギガバイトを超えていないことを確認してください。
 7. キューファイルの最大格納容量を 2 ギガバイト以上にする場合、複数のキューファイルをキューファイルグループにすれば、最大格納容量の上限がキューファイル数×キューファイルの最大格納容量になります。
 8. キューファイルを割り当てる場合は、OpenTP1 ファイルシステム領域の中で、連続した領域を確保する必要があります。連続した空き領域が確保できない場合は、OpenTP1 ファイルシステムのガーベジコレクション（OpenTP1 ファイルシステム中の使用中の領域を集中させ、空き領域を連続させること）が必要となります。
- 詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

mqainq (オブジェクトの属性表示)

形式

```
mqainq { {-m | -x} キューマネージャ名 |  
         {-q | -p} キューマネージャ名 オブジェクト名 }
```

機能

キューマネージャの情報、ローカルキューの情報、モデルキューの情報、リモートキューのローカル定義の情報、別名キューの情報、キュー名の一覧、およびプロセス定義の情報を表示します。

オプション

● -m

キューマネージャの情報を出力する場合に指定します。

● -x

キューの名称を一覧で出力する場合に指定します。

● -q

ローカルキューの情報、モデルキューの情報、リモートキューのローカル定義の情報、または別名キューの情報を出力する場合に指定します。

● -p

プロセス定義の情報を出力する場合に指定します。

コマンド引数

● キューマネージャ名

～< 1～48 文字の MQ 文字列 >

出力する情報のキューマネージャの名称を指定します。

● オブジェクト名

～< 1～48 文字の MQ 文字列 >

出力する情報のオブジェクトのローカル名を指定します。

-q または -p オプションを指定した場合に指定します。

出力形式

● キューマネージャの属性

```
キューマネージャ名 = aa....aa
キューマネージャ記述子 = bb....bb
プラットフォーム = cc....cc
トリガ間隔 = dd....dd
デッドレターキュー名 = ee....ee
最大メッセージ優先度 = ff....ff
省略時の転送キュー名 = gg....gg
文字セット識別子 = hh....hh
最大ハンドル数 = ii....ii
トランザクション完了待ち最大メッセージ数 = jj....jj
最大メッセージ長 = kk....kk
トランザクション有効性 = ll....ll
配布リスト = mm....mm
```

aa....aa

キューマネージャ名 (1~48 文字の MQ 文字列)

bb....bb

キューマネージャ記述子 (1~64 文字の文字列)

cc....cc

プラットフォーム (10 進数) "OpenTP1"を表示

dd....dd

トリガ間隔 (0~999999999 の 10 進数)

ee....ee

デッドレターキュー名 (1~48 文字の MQ 文字列)

ff....ff

最大メッセージ優先度 (10 進数)

gg....gg

省略時の転送キュー名 (1~48 文字の MQ 文字列)

hh....hh

文字セット識別子 (10 進数)

ii....ii

最大ハンドル数 (10 進数)

jj....jj

TP1/Message Queue では、この属性は使用されません。

0 が表示されます。

kk....kk

最大メッセージ長 (10 進数, 単位: バイト)

ll...ll

トランザクション有効性 (次に示すどちらかを表示)

AVAILABLE: トランザクションの同期点処理を利用できます。

NOT AVAILABLE: トランザクションの同期点処理を利用できません。

mm...mm

配布リスト (次に示すどちらかを表示)

SUPPORTED: 配布リストに対応しています。

NOT SUPPORTED: 配布リストに対応していません。

● ローカルキューおよびモデルキューの属性

```
キュー名 = aa....aa
キュータイプ = bb....bb
キュー記述子 = cc....cc
取り出し許可 = dd....dd
登録許可 = ee....ee
省略時のメッセージ優先度 = ff....ff
省略時のメッセージ永続性 = gg....gg
キューの有効範囲 = hh....hh
プロセス名 = ii....ii
最大メッセージ登録数 = jj....jj
最大メッセージ長 = kk....kk
共用性 = ll....ll
省略時の取り出しオプション = mm....mm
ロールバック回数記録方法 = nn....nn
メッセージ配布順序 = oo....oo
キュー保持時間 = pp....pp
定義タイプ = qq....qq
使用種別 = rr....rr
入力オープン数 = ss....ss
出力オープン数 = tt....tt
メッセージ登録数 = uu....uu
作成日付 = YYYY-MM-DD
作成時刻 = HH.MM.SS
イニシエーションキュー名 = vv....vv
トリガ制御 = ww....ww
トリガタイプ = xx....xx
トリガのためのメッセージ優先度の下限値 = yy....yy
トリガのためのメッセージ登録数の下限値 = zz....zz
トリガデータ = a1....a1
キューファイルグループ名 = b1....b1
キューファイル名 = c1....c1
配布リスト = d1....d1
デフォルトのバインディング = e1....e1
クラスタ内共用クラスタ名称 = f1....f1
```

aa....aa

キュー名 (1~48 文字の MQ 文字列)

bb....bb

キュータイプ (次に示すどちらかを表示)

LOCAL QUE：ローカルキュー

MODEL QUE：モデルキュー定義

CC....CC

キュー記述子（1～64文字の文字列）

dd....dd

取り出し許可（次に示すどちらかを表示）

INHIBITED：取り出しを禁止します。

ALLOWED：取り出しを許可します。

ee....ee

登録許可（次に示すどちらかを表示）

INHIBITED：登録を禁止します。

ALLOWED：登録を許可します。

ff....ff

省略時のメッセージ優先度（10進数）

gg....gg

省略時のメッセージ永続性（次に示すどちらかを表示）

PERSISTENT：永続メッセージにします。

NOT PERSISTENT：永続メッセージにしません。

hh....hh

キューの有効範囲（次に示す値を表示）

QUE MGR：キュー定義はキューマネージャスコープを持ちます。

ii....ii

プロセス名（1～48文字のMQ文字列）

jj....jj

最大メッセージ登録数（10進数）

kk....kk

最大メッセージ長（10進数，単位：バイト）

ll....ll

共用性（次に示すどちらかを表示）

SHAREABLE：キューを共用できます。

NOT SHAREABLE：キューを共用できません。

mm....mm

省略時の取り出しオプション（次に示すどちらかを表示）

INPUT EXCLUSIVE：排他アクセスのメッセージ取り出し用にオープンします。

INPUT SHARED：共有アクセスのメッセージ取り出し用にオープンします。

nn....nn

ロールバック回数記録方法（次に示す値を表示）

NOT HARDENED：ロールバック回数を記録しません。

oo....oo

メッセージ配布順序（次に示すどちらかを表示）

PRIORITY：優先度の順序でメッセージが返されます。

FIFO：FIFO（先入れ先出し）の順序でメッセージが返されます。

pp....pp

キュー保持時間（10進数）

qq....qq

定義タイプ（次に示すどれかを表示）

PREDEFINED：定義済みキュー

PERMANENT DYNAMIC：永続的動的キュー

TEMPORARY DYNAMIC：一時的動的キュー

rr....rr

使用種別（次に示すどちらかを表示）

NORMAL：通常キュー。通常のアプリケーションがメッセージを登録、または取り出しをする場合に使用します。

TRANSMISSION：転送キュー。リモートキューマネージャあてのメッセージを保持するために使用します。

ss....ss

入力オープン数（10進数）

tt....tt

出力オープン数（10進数）

uu....uu

メッセージ登録数（10進数）

YYYY-MM-DD

作成日付（年-月-日）

作成日付はシステムの設定時刻で表示されます。

HH.MM.SS

作成時刻（時.分.秒）

作成時刻はシステムの設定時刻で表示されます。

vv....vv

イニシエーションキュー名（1～48文字のMQ文字列）

ww....ww

トリガ制御（次に示すどちらかを表示）

OFF：トリガメッセージを要求しません。

ON：トリガメッセージを要求します。

XX....XX

トリガタイプ (次に示すどれかを表示)

NONE：トリガメッセージは通知されません。

FIRST：空でない場合に通知されます。

EVERY：メッセージごとに通知されます。

DEPTH：一定の格納数を越えた場合に通知されます。

yy....yy

トリガのためのメッセージ優先度の下限値 (10 進数)

ZZ....ZZ

トリガのためのメッセージ登録数の下限値 (10 進数)

a1....a1

トリガデータ (1~64 文字の文字列)

b1....b1

キューファイルグループ名 (1~31 文字の識別子)

c1....c1

キューファイル名 (1~59 文字の文字列)

d1....d1

配布リスト (次に示すどちらかを表示)

SUPPORTED：配布リストに対応しています。

NOT SUPPORTED：配布リストに対応していません。

e1....e1

デフォルトのバインディング (次に示すどちらかを表示)

FIXED：キューがオープンされたハンドルにあて先を固定します。

NOT FIXED：キューにメッセージが登録される場合、メッセージ単位にあて先を決定します。

f1....f1

クラスタ内共用クラスタ名称 (1~48 文字の MQ 文字列)

● リモートキューのローカル定義の属性

```
キュー名      = aa....aa
キュータイプ  = bb....bb
キュー記述子  = cc....cc
登録許可      = dd....dd
省略時のメッセージ優先度 = ee....ee
省略時のメッセージ永続性 = ff....ff
リモートキュー名 = gg....gg
リモートキューマネージャ名 = hh....hh
```

転送キュー名	=	ii....ii
デフォルトのバインディング	=	jj....jj
クラスタ内共用クラスタ名称	=	kk....kk

aa....aa

キュー名 (1~48 文字の MQ 文字列)

bb....bb

キュータイプ

REMOTE QUE：リモートキューのローカル定義

cc....cc

注釈記述用のフィールド (1~64 文字の文字列)

dd....dd

キューに対する登録許可 (次に示すどちらかを表示)

INHIBITED：登録を禁止します。

ALLOWED：登録を許可します。

ee....ee

省略時のメッセージ優先度 (10 進数)

ff....ff

省略時のメッセージ永続性 (次に示すどちらかを表示)

PERSISTENT：永続メッセージにします。

NOT PERSISTENT：永続メッセージにしません。

gg....gg

リモートキュー名 (1~48 文字の MQ 文字列)

hh....hh

リモートキューマネージャ名 (1~48 文字の MQ 文字列)

ii....ii

転送キュー名 (1~48 文字の MQ 文字列)

jj....jj

デフォルトのバインディング (次に示すどちらかを表示)

FIXED：キューがオープンされたハンドルにあて先を固定します。

NOT FIXED：キューにメッセージが登録される場合、メッセージ単位にあて先を決定します。

kk....kk

クラスタ内共用クラスタ名称 (1~48 文字の MQ 文字列)

● 別名キューの属性

キュー名	=	aa....aa
キュータイプ	=	bb....bb

キュー記述子	=	cc....cc
取り出し許可	=	dd....dd
登録許可	=	ee....ee
省略時のメッセージ優先度	=	ff....ff
省略時のメッセージ永続性	=	gg....gg
ベースキュー名	=	hh....hh
デフォルトのバインディング	=	ii....ii
クラスタ内共用クラスタ名称	=	jj....jj

aa....aa

キュー名 (1~48 文字の MQ 文字列)

bb....bb

キュータイプ

ALIAS QUE : 別名キュー定義

cc....cc

注釈記述用のフィールド (1~64 文字の文字列)

dd....dd

キューに対する取り出し許可 (次に示すどちらかを表示)

INHIBITED : 取り出しを禁止します。

ALLOWED : 取り出しを許可します。

ee....ee

キューに対する登録許可 (次に示すどちらかを表示)

INHIBITED : 登録を禁止します。

ALLOWED : 登録を許可します。

ff....ff

省略時のメッセージ優先度 (10 進数)

gg....gg

省略時のメッセージ永続性 (次に示すどちらかを表示)

PERSISTENT : 永続メッセージにします。

NOT PERSISTENT : 永続メッセージにしません。

hh....hh

ベースキュー名 (1~48 文字の MQ 文字列)

ii....ii

デフォルトのバインディング (次に示すどちらかを表示)

FIXED : キューがオープンされたハンドルにあて先を固定します。

NOT FIXED : キューにメッセージが登録される場合、メッセージ単位にあて先を決定します。

jj....jj

クラスタ内共用クラスタ名称 (1~48 文字の MQ 文字列)

● キュー名一覧の情報

```
キュー名 = aa....aa
```

aa....aa

キュー名 (1~48 文字の MQ 文字列)

● プロセス定義の属性

```
プロセス名 = aa....aa  
プロセス記述子 = bb....bb  
アプリケーションタイプ = cc....cc  
アプリケーション記述子 = dd....dd  
環境データ = ee....ee  
ユーザデータ = ff....ff
```

aa....aa

プロセス名 (1~48 文字の MQ 文字列)

bb....bb

プロセス記述子 (1~64 文字の文字列)

cc....cc

アプリケーションタイプ

- OpenTP1 : OpenTP1 アプリケーション
- ユーザ定義のアプリケーションタイプ最低値~ユーザ定義のアプリケーションタイプ最高値

dd....dd

アプリケーション記述子 (1~256 文字の文字列)

ee....ee

環境データ (1~128 文字の文字列)

ff....ff

ユーザデータ (1~128 文字の文字列)

注意事項

1. -q オプションで指定したオブジェクトが次に示すどれかの場合にエラーになります。
 - ローカルキューを指定した際に、格納されているキューファイルが閉塞状態の場合
 - モデルキューを指定した際に、MQA サービス定義の mqaqueatl 定義コマンドで指定したキューファイルにキューを作成できない場合
 - リモートキューのローカル定義を指定した際に、指定されたオブジェクト名がキューマネージャの別名として使用されている場合
2. -x オプションを指定した場合、次に示すキュー名は出力されません。
 - MQA サービス定義にモデルキューが定義されていないキュー名

- MQA サービス定義の mqaqueatl 定義コマンドで指定されたキューファイルに作成できるキュー数が 0 件のキュー名
 - ローカルキューを格納しているキューファイルが閉塞状態のローカルキュー名
 - MQA サービス定義の mqaqueatl 定義コマンドで指定されたキューファイルにキューを作成できないモデルキュー名
 - キューマネージャの別名定義として使用されているリモートキューのローカル定義名
 - MQA サービス定義の mqaqueatl 定義コマンドで指定されたキューファイル名がないモデルキュー名
 - MQA サービス定義の mqaqueatl 定義コマンドで指定されたキューファイル名がオンライン中のキューファイルグループにないモデルキュー名
 - MQA サービス定義の mqaalsque 定義コマンドで指定された BaseQName 属性が、ローカルに定義されたキューの名前でない別名キュー名
 - MQA サービス定義の mqaremque 定義コマンドで指定された XmitQName 属性が空白ではなく、MQUS_TRANSMISSION の Usage 属性を持っていないキューが指定されているリモートキューのローカル定義名
 - MQA サービス定義の mqaremque 定義コマンドで指定された XmitQName 属性が空白で、RemoteQMgrName 属性に、MQUS_TRANSMISSION の Usage 属性を持っていないキューが指定されているリモートキューのローカル定義名
 - MQA サービス定義の mqaremque 定義コマンドで指定された XmitQName 属性が空白ではなく、ローカルに定義されたキューの名前でもないリモートキューのローカル定義名
 - MQA サービス定義の mqaremque 定義コマンドで指定された XmitQName 属性が空白で、あて先キューマネージャと同じ名前前で定義されている転送キューがない場合に、キューマネージャ定義の DefXmitQName 属性によって指定されているキューが、MQUS_TRANSMISSION の Usage 属性を持っていないリモートキューのローカル定義名
 - MQA サービス定義の mqaremque 定義コマンドで指定された XmitQName 属性が空白で、あて先キューマネージャと同じ名前前で定義されている転送キューがない場合に、キューマネージャ定義の DefXmitQName 属性によって指定されている名称が、ローカルに定義されたキューの名前でないリモートキューのローカル定義名
 - MQA サービス定義の mqaremque 定義コマンドで指定された XmitQName 属性が空白で、あて先キューマネージャと同じ名前前で定義されている転送キューがない場合に、キューマネージャ定義の DefXmitQName 属性が空白になっているリモートキューのローカル定義名
3. -q オプションの指定で、コマンド引数のオブジェクト名にモデルキューを指定した場合、指定されたモデルキューの属性で一時的動的キューが作成されます。属性を取得したあと、一時的動的キューは削除されます。
 4. -q オプションの指定で、コマンド引数のオブジェクト名にモデルキューを指定した場合、キューの作成日付およびキューの作成時刻は、このコマンドで一時的動的キューを作成した日付と時刻が出力されます。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA04356-I	ヘルプメッセージ	標準出力
KFCA26162-E	指定したオブジェクト名が誤っています。	標準エラー出力
KFCA26163-E	指定したキューマネージャ名が誤っています。	標準エラー出力
KFCA26172-E	指定したオブジェクト名は有効ではありません。	標準エラー出力
KFCA26176-E	指定したキューをキューファイルに割り当てられません。	標準エラー出力
KFCA26179-E	キュー名の取得に失敗しました。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqalsmsg (メッセージの表示)

形式

```
mqalsmsg [-s メッセージ開始番号] [-e 出力メッセージ数]
          [-o メッセージ内容出力バイト数] [-v] キュー名
```

機能

指定したキュー名に登録されているメッセージ情報を表示します。

オプション

● -s メッセージ開始番号

～< 1～2147483647 >

指定したキュー名に登録されているメッセージの先頭からの開始番号を指定します。

省略した場合、指定したキュー名に登録されているメッセージの先頭が指定されたものと見なします。

● -e 出力メッセージ数

～< 1～100 >

-s オプションで指定されたメッセージ開始番号から、メッセージ情報を表示するメッセージ数を指定します。

省略した場合、1メッセージの情報を表示します。

● -o メッセージ内容出力バイト数

～< 1～1000 > (単位: バイト)

指定したキュー名に登録されているメッセージ内容の先頭からのバイト数を指定します。

省略した場合、メッセージ内容は出力されません。

● -v

MQMD 構造体のすべての情報 (構造体バージョン 2 の情報) を表示します。

省略した場合、構造体バージョン 1 の情報だけを表示します。

コマンド引数

● キュー名

～< 1～48 文字の MQ 文字列 >

出力するメッセージが登録されているキュー名を指定します。

なお、次に示す種類のキューは指定できません。

- リモートキュー
- モデルキュー

出力形式

● キューマネージャの属性

```

メッセージ表示番号      = AA-BB
メッセージ番号 = CC
MQMD
  構造体識別子          = DD...DD
  構造体バージョン番号  = EE...EE
  報告オプション        = [0x FF...FF]
  メッセージタイプ      = [0x GG...GG]
  メッセージ保持時間    = HH...HH
  報告メッセージ返答コード = [0x II...II]
  マシンコード形式      = [0x JJ...JJ]
  文字セット識別子      = [0x KK...KK]
  フォーマット名        = LL...LL
  メッセージ優先度      = MM...MM
  メッセージ永続性      = NN...NN
  メッセージ識別子      = OO...OO
                        [0x PP...PP PP...PP PP...PP PP...PP]
                        [0x PP...PP PP...PP]
  相関識別子            = QQ...QQ
                        [0x RR...RR RR...RR RR...RR RR...RR]
                        [0x RR...RR RR...RR]
  ロールバック回数      = SS...SS
  応答キュー名          = TT...TT
  応答キューマネージャ名 = UU...UU
  ユーザ識別子          = VV...VV[0x WW...WW WW...WW WW...WW]
  課金トークン          = XX...XX
                        [0x YY...YY YY...YY YY...YY YY...YY]
                        [0x YY...YY YY...YY YY...YY YY...YY]
  アプリケーション識別データ = ZZ...ZZ
                        [0x aa...aa aa...aa aa...aa aa...aa]
                        [0x aa...aa aa...aa aa...aa aa...aa]
  登録アプリケーションタイプ = [0x bb...bb]
  登録アプリケーション名    = cc...cc
  登録日付                = dd...dd
  登録時刻                = ee...ee
  登録元データ            = ff...ff ※
  グループID              = gg...gg
                        [0x hh...hh hh...hh hh...hh hh...hh]
                        [0x hh...hh hh...hh]
  論理メッセージの順序番号 = ii...ii
  データのオフセット      = jj...jj
  メッセージフラグ        = [0x kk...kk]
  元のメッセージの長さ    = ll...ll
  メッセージ長            = mm...mm
  メッセージ内容          = 00000000[0x nn...nn nn...nn nn...nn nn...nn] oo...oo
                        .
                        .
                        .
                        000003e0[0x nn...nn ] oo...oo
  
```

注※

バージョン 2 の構造体の情報です。

AA	メッセージ開始番号
BB	メッセージ終了番号
CC	メッセージ番号
DD...DD	構造体識別子 (文字列)
EE...EE	構造体バージョン番号 (10 進数)
FF...FF	報告オプション (16 進数)
GG...GG	メッセージタイプ (16 進数)
HH...HH	メッセージ保持時間 (10 進数)
II...II	報告メッセージ返答コード (16 進数)
JJ...JJ	マシンコード形式 (16 進数)
KK...KK	文字セット識別子 (16 進数)
LL...LL	フォーマット名 (文字列)
MM...MM	メッセージ優先度 (10 進数)
NN...NN	メッセージ永続性 (MQPER_PERSISTENT, または MQPER_NOT_PERSISTENT)
OO...OO	メッセージ識別子 (文字列)
PP...PP	メッセージ識別子 (16 進数)
QQ...QQ	相関識別子 (文字列)

RR....RR

相関識別子 (16 進数)

SS....SS

ロールバック回数 (10 進数)

TT....TT

応答キュー名 (文字列)

UU....UU

応答キューマネージャ名 (文字列)

VV....VV

ユーザ識別子 (文字列)

WW....WW

ユーザ識別子 (16 進数)

XX....XX

課金トークン (文字列)

YY....YY

課金トークン (16 進数)

ZZ....ZZ

アプリケーション識別データ (文字列)

aa....aa

アプリケーション識別データ (16 進数)

bb....bb

登録アプリケーションタイプ (16 進数)

cc....cc

登録アプリケーション名 (文字列)

dd....dd

登録日付 (文字列)

ee....ee

登録時刻 (文字列)

ff....ff

登録元データ (文字列)

gg....gg

グループ ID (文字列)

hh....hh

グループ ID (16 進数)

ii...ii

論理メッセージの順序番号 (10 進数)

jj...jj

データのオフセット (10 進数)

kk...kk

メッセージフラグ (16 進数)

ll...ll

元のメッセージの長さ (10 進数)

mm...mm

メッセージ長 (10 進数)

nn...nn

メッセージ内容 (16 進数)

oo...oo

メッセージ内容 (文字列)

文字列の中に MQ 文字列, および空白以外の文字がある場合は, その文字を. (ピリオド) に変換し出力します。

注意事項

1. アプリケーションからのメッセージ登録/取り出し中, および転送中のメッセージは表示できない場合があります。
2. このコマンドは MQOPEN 命令および MQGET 命令を使用してメッセージを検索して表示するため, 指定したキューの取り出しが禁止されている場合, コマンドが失敗します。
3. メッセージ滞留時間を監視する場合, メッセージ保持時間が終了したメッセージは mqalsmsg コマンドの表示対象になりませんが, mqainq コマンドで表示されるメッセージ登録数に含まれます。
4. 別名キュー (ベースキューがリモートにあるクラスタ共有キュー) を指定してこのコマンドを実行すると, 次に示すエラーメッセージが出力されます。

「KFCA31030-E コマンド実行中に異常を検知しました。関数名=MQOPEN 理由コード=2082」

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04310-E	コマンドに指定されたパラメタの値が誤っています。	標準エラー出力
KFCA04331-E	キュー名の文字数が 48 文字を超えています。	標準エラー出力
KFCA04337-E	指定したキュー名は存在しません。	標準エラー出力
KFCA04369-E	キュー名に指定できない文字を指定しています。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA26189-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA26193-I	ヘルプメッセージ	標準出力
KFCA31025-E	指定したメッセージ開始番号からは、メッセージが存在しません。	標準エラー出力
KFCA31030-E	コマンド実行中に異常を検知しました。	標準エラー出力
KFCA31044-E	指定したキュー名は指定できない種類のキューです。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqamkque (キューの作成)

形式

```
mqamkque [-c クラスタ内共用クラスタ名称] ローカルキュー名  
          モデルキューの定義名 キュー属性定義ファイル名
```

機能

キュー属性定義ファイルに格納されているモデルキューの属性定義を基にローカルキューをキューファイル上に作成します。

モデルキューの定義の形式は、MQA サービス定義の `mqaqueatl` 定義コマンドと同様の形式で指定します。詳細については、4章の「`mqaqueatl` (モデルキューの属性定義)」を参照してください。ただし、このコマンドで作成するキューは定義済みキューだけであるため、`mqaqueatl` 定義コマンドの `-k` オプションで指定する定義タイプは無視します。

キュー属性定義ファイルには、複数のモデルキューの属性定義を格納できます。その場合は、モデルキューの属性定義に指定するモデルキューの定義名を別にしてください。

オプション

● `-c` クラスタ内共用クラスタ名称

～< 1～48 文字の MQ 文字列>

キューが属するクラスタの名称を指定します。

省略された場合は、キューはクラスタには属しません。

MQA サービス定義の `mqaqueatl` 定義コマンドの `-u` オプションで、`transmission` を指定したキューは、指定できません。

コマンド引数

● ローカルキュー名

～< 1～48 文字の MQ 文字列>

作成するキューの名称を指定します。

● モデルキューの定義名

～< 1～48 文字の MQ 文字列>

定義ファイル内に定義されたモデルキューの定義名を指定します。

● キュー属性定義ファイル名

～<パス名>

キュー属性定義（モデルキューの属性定義）が格納されているファイルのパス名を指定します。

注意事項

1. このコマンドは、次に示すどちらかの場合にエラーになります。
 - 指定されたキューファイルがオンラインで使用されている場合
 - 指定されたキュー名がすでにキューファイル上に存在する場合
2. このコマンドで作成するキューは、すべてローカルキューです。
3. このコマンドで作成されたキューにメッセージを登録する場合、メッセージが格納されるキューファイルは、キューが格納されているキューファイルとは異なる場合があります。一つのキューファイルグループに複数のキューファイルが対応づけられている場合、MQA サーバでは、メッセージに対するアクセスの負荷分散を考慮して、キューとメッセージが同一キューファイルグループ上の別々のファイルに格納される場合があります。
4. キューとメッセージを同じキューファイルに格納したい場合、キューファイルグループに対応するキューファイルの一つにしてください。
5. キューファイル二重化構成の場合のバックアップキューファイル名（B系ファイル名）用の `mqaqueatl` 定義は、MQA サービス定義ファイルに作成しないでください。

mqarcvr (ファイル二重化の回復)

形式

```
mqarcvr -f キューファイル名
```

機能

キューファイル二重化構成をしたキューファイルに片系障害が発生した場合、障害ファイルを回復し、再度二重化構成に戻します。指定するキューファイルは、障害が発生しているファイルを指定してください。

オプション

● -f キューファイル名

～< 1～59 文字の文字列 >

キューファイル二重化構成をしたキューファイルで片系障害が発生している障害ファイルを指定します。

注意事項

1. コマンド実行中は、指定したキューファイルに対する正常に動作しているキューファイルのアプリケーションからのメッセージ登録/取り出し、および転送を抑制します。そのためコマンドが終了するまでアプリケーションからのメッセージ登録/取り出し、および転送は実行できません。
2. キューファイル二重化構成を行っていないキューファイルは、指定できません。
3. コマンド実行中にエラーが発生した場合、障害ファイルが削除されている場合があります。その場合、エラー要因を取り除き、再度コマンドを入力してください。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA04200-E	I/O エラーが発生しました。	標準エラー出力
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04211-E	キューファイル名の文字数が 59 文字を超えています。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA04310-E	コマンドに指定されたパラメタの値が誤っています。	標準エラー出力
KFCA04311-E	ファイルシステム作成時のシステムとコマンド実行時のシステムのバージョンが異なります。	標準エラー出力
KFCA04313-E	キューファイルに対するアクセス権がありません。	標準エラー出力
KFCA04314-E	指定したキューファイルは他のプロセスで使用中です。	標準エラー出力
KFCA04315-E	キューファイルの V/R が誤っています。	標準エラー出力
KFCA04316-E	スペシャルファイルに対するアクセス権がありません。	標準エラー出力
KFCA04317-E	指定したキューファイルはスペシャルファイル名ではありません。	標準エラー出力
KFCA04318-E	キューファイルを割り当てるディスクパーティションは OpenTP1 ファイルシステムとして初期化されていません。	標準エラー出力
KFCA04319-E	キャラクタ型スペシャルファイルのオープンでシステムから上限値オーバのエラーが報告されました。	標準エラー出力
KFCA04339-E	指定したキューファイル名は存在しません。	標準エラー出力
KFCA04345-E	指定したキューファイルは障害閉塞されています。	標準エラー出力
KFCA04378-E	OpenTP1 ファイルシステム初期化時に指定したファイル数の上限値を超えたためキューファイルを作成できません。	標準エラー出力
KFCA04384-E	指定したキューファイルを割り当てるだけの空き容量がありません。	標準エラー出力
KFCA04387-E	指定したファイルはキューファイルではありません。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA26189-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA26194-I	ヘルプメッセージ	標準出力
KFCA31032-W	プロセス実行中のため、キューファイルの回復処理を再試行します。	標準エラー出力
KFCA31033-E	指定したキューファイルは、二重化構成を行っていないため指定できません。	標準エラー出力
KFCA31034-E	指定したキューファイルは、両系論理閉塞のため回復する必要がありません。	標準エラー出力
KFCA31037-E	キューファイル二重化構成において、両系障害閉塞のため回復できません。	標準エラー出力
KFCA31039-I	指定したキューファイルを回復しました。	メッセージログファイル
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqarles (キューファイルの閉塞解除)

形式

```
mqarles [-s {l | t}] -f キューファイル名
```

機能

指定したキューファイルの閉塞状態を解除します。

MQA サービスがキューファイルを閉塞状態にする契機は、mqadd コマンドが入力された時（論理閉塞）と、キューファイルの障害を検知した時（障害閉塞または一時的障害閉塞）です。このコマンドは、論理閉塞状態および一時的障害閉塞状態だけを解除できます。障害閉塞状態のキューファイルに対してこのコマンドを入力すると、エラーになります。

- 論理閉塞状態の場合
メッセージを操作する場合は、組み込んだキューファイル（A系ファイル）に対し、mqarles コマンドを入力してください。バックアップキューファイルに対する mqarles コマンドの入力は不要です。
- 一時的障害閉塞状態の場合
一時的障害閉塞が発生しているキューファイルを指定してください。一時的障害閉塞でないキューファイルを指定した場合、コマンドはエラーになります。

オプション

● -s l | t

解除する閉塞状態を指定します。次に示すどちらかを指定します。

- l: 論理閉塞を解除します。
- t: 一時的障害閉塞を解除します。

省略した場合は、論理閉塞が解除されます。

● -f キューファイル名

～< 1～59 文字の文字列>

閉塞状態を解除するキューファイルの名称を指定します。

注意事項

論理閉塞状態の場合、mqadd コマンドでキューファイル二重化構成にキューファイルを組み込んだとき、キューファイル（A系ファイル）に対して入力するとバックアップキューファイル（B系ファイル）の閉塞解除も行います。バックアップキューファイル（B系ファイル）を指定するとエラーになります。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04211-E	キューファイル名の文字数が 59 文字を超えています。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04329-E	指定したキューファイルは閉塞されていません。	標準エラー出力
KFCA04332-I	ヘルプメッセージ	標準出力
KFCA04334-E	指定したキューファイルは閉塞処理中です。	標準エラー出力
KFCA04338-E	指定したキューファイルは障害閉塞状態です。	標準エラー出力
KFCA04339-E	指定したキューファイル名は存在しません。	標準エラー出力
KFCA04343-I	キューファイル、およびバックアップキューファイルの閉塞解除を完了しました。	メッセージログファイル
KFCA04344-I	キューファイルの閉塞解除を完了しました。	メッセージログファイル
KFCA04346-E	指定したキューファイルは、バックアップキューファイルのため指定できません。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力
KFCA31180-E	MQA サービス専用共用メモリが参照できません。	メッセージログファイル
KFCA31188-I	キューファイルを正常状態に変更しました。	メッセージログファイル
KFCA31189-E	キューファイルを正常状態に変更できませんでした。	メッセージログファイル
KFCA31190-E	指定したキューファイルは一時的障害閉塞ではありません。	標準エラー出力

mqarm (キューファイルの切り離し)

形式

```
mqarm [-r] -f キューファイル名
```

機能

指定したキューファイルをオンラインから切り離します。-r オプションを指定すると強制的にオンラインから切り離します。

オプション

● -r

オンライン中にディスク障害が発生し、障害ディスクボリュームを切り離す場合に指定してください。

● -f キューファイル名

～< 1～59 文字の文字列>

オンラインから切り離すキューファイルの名称を指定します。

注意事項

1. キューファイルを切り離す場合、切り離すキューファイルに格納されているキューおよびメッセージは、メモリ上のものだけ削除します。キューファイル上のキューおよびメッセージは削除しません。切り離したキューファイルを再度、そのまま組み込んだ場合、KFCA04323-I メッセージが出力され、削除されたメッセージが再び登録される現象が発生します。したがって、切り離したキューファイルを組み込む場合は、キューファイルの回復または初期化をしてください。
2. キューファイルを切り離す場合、切り離すキューファイルにキューが登録されているときは、このコマンドでキューが削除されます。削除対象のキューにメッセージが登録されている場合は、メッセージも同時に削除します。また、メッセージを削除する場合、切り離すキューファイルとは別のキューファイルに登録されている場合でも、メッセージを削除します。
3. 切り離すキューファイルにメッセージだけが登録されている場合（メッセージを登録したキューが別のキューファイルにある場合）、切り離すキューファイルに登録されているメッセージだけが削除されます。この場合、キューは別ファイルにあるため、キューに対するメッセージの登録および取り出しはできます。メッセージが削除されると、削除されたメッセージのメッセージ識別子は KFCA04323-I メッセージとともに表示されます。メッセージ識別子は、MQA サーバではコードを意識しないで 16 進数で出力されます。
4. このコマンドでエラーが発生した場合でも、KFCA04323-I または KFCA04324-I のメッセージが出力されているときは、キューおよびメッセージは削除されています。エラー発生後にキューファイルをオンラインから切り離すには、エラーの要因を取り除き、再度 mqarm コマンドを入力してください。

5. -r オプションは、障害が発生したディスクボリューム上のキューファイルをオンラインから切り離す場合にだけ使用してください。-r オプションを指定してコマンドを入力すると、キューファイル障害が発生した状態と同じ状態になり、アクセス中のアプリケーションはエラーになります。
6. 正常状態のキューファイルを-r オプションで強制的に切り離す場合に、コマンドがメモリ不足でエラーリターンしたときは、キューファイルは障害閉塞状態になります。
7. 正常状態のキューファイルを-r オプションで強制的に切り離す場合、コマンドの再試行中に `dcstop -a` または `dcstop -b` コマンドで OpenTP1 を終了して再開始するとキューファイルは障害閉塞状態になります。
8. キューファイル二重化構成で片系障害が発生したキューファイルに対して入力できません。入力した場合エラーになります。ただし、-r オプションを指定した場合は入力できますがメッセージの削除はしません。
9. 両系障害が発生したキューファイルに対して入力すると、キューファイル、バックアップキューファイルの両方をオンラインから切り離しメッセージも削除します。
10. 両系障害が発生していないファイルに対して-r オプションを指定した場合、指定したファイルは障害閉塞状態になりメッセージの削除はしません。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04211-E	キューファイル名の文字数が 59 文字を超えています。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04306-W	プロセス実行中のため、キューファイルの切り離し処理を再試行します。	標準エラー出力
KFCA04323-I	キュー名：×××に登録されているメッセージを削除しました。	メッセージログファイル
KFCA04324-I	キュー名：×××を削除しました。	メッセージログファイル
KFCA04325-E	指定したキューファイルは二重化構成のため両系障害閉塞時以外は実行できません。	標準エラー出力
KFCA04329-E	指定したキューファイルは閉塞されていません。	標準エラー出力
KFCA04339-E	指定したキューファイル名は存在しません。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA04354-I	ヘルプメッセージ	標準出力
KFCA04355-E	キューファイル, およびバックアップキューファイルの切り離しが完了しました。	標準エラー出力
KFCA04358-I	キューファイルの切り離しが完了しました。	メッセージログファイル
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31092-E	プロセス固有領域のメモリ不足でクラスタキュー情報をクラスタに通知できません。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31118-E	コマンドが異常を検知しました。	メッセージログファイル
KFCA31119-E	キューファイルの切り離し処理で異常を検知しました。	標準エラー出力
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため, コマンドを実行できません。	標準エラー出力
KFCA31180-E	MQA サービス専用共用メモリが参照できません。	メッセージログファイル

mqarstr (キューファイルのリストア)

形式

```
mqarstr ファイル名 キューファイル名
```

機能

mqabkup コマンドでバックアップしたキューファイル (メッセージ情報管理ファイル (ファイル名.cnt) とメッセージファイル (ファイル名.msg)) を, OpenTP1 ファイルシステム上のキューファイル (メッセージ情報管理ファイル (キューファイル名.cnt) と, メッセージファイル (キューファイル名.msg)) にリストアします。

コマンド引数

● ファイル名

~<パス名>

リストア元のファイルの名称を指定します。

● キューファイル名

~<1~59 文字の文字列>

リストア先のキューファイルの名称を完全パス名で指定します。

注意事項

1. このコマンドは, 次に示すどちらかの場合にエラーになります。

- 指定したキューファイルが, オンラインで使用中的の場合
- ファイルのうち, どちらか一つでもキューファイルのレコード数を越えた場合

2. 次に示す場合, すでに存在するキューファイルにファイルの内容を上書きします。

- 同一キューファイル名が OpenTP1 ファイルシステム上にすでに存在する場合
- ファイルのレコード数がすでに存在するキューファイルと同じか, または小さい場合

上書きしたキューファイルが OpenTP1 ファイルシステム上で使用する領域容量は, 上書き前と変わりません。このため, 上書きするキューファイルを mqadel コマンドで削除後, リストアを実行することをお奨めします。

3. このコマンドは, リストア元のファイルのレコード長と同じレコード長のリストア先キューファイルを開いた OpenTP1 ファイルシステム上に作成します。したがって, リストアを正常に行うには次に示す条件を満たす必要があります。

- リストア元のファイルのレコード長は, リストア先のキューファイルを作成する OpenTP1 ファイルシステムのセクタ長の倍数でなければなりません。

上記の条件を満たさない場合、このコマンドは、KFCA31182-E のメッセージを出力して異常終了します。この場合は、リストア先のキューファイルを作成する OpenTP1 ファイルシステムのセクタ長を見直して再度実行してください。

mqaset (オブジェクト属性の設定)

形式

```
mqaset {{-p {disabled | enabled} |  
        -g {disabled | enabled}} |  
        {-p {disabled | enabled}  
        -g {disabled | enabled}}} オブジェクト名
```

機能

ローカルキュー，リモートキューのローカルキュー定義，または別名キューに登録許可および取り出し許可を設定します。

オプション

● -p disabled | enabled

オブジェクトに対するメッセージの登録操作の禁止または許可を指定します。

次に示すどちらかを指定します。

- disabled：登録操作を禁止します。
- enabled：登録操作を許可します。

● -g disabled | enabled

オブジェクトに対するメッセージの取り出し操作の禁止または許可を指定します。

次に示すどちらかを指定します。

- disabled：取り出し操作を禁止します。
- enabled：取り出し操作を許可します。

コマンド引数

● オブジェクト名

～< 1～48 文字の MQ 文字列>

オブジェクト属性の設定をするオブジェクトのローカル名称を指定します。

注意事項

リモートキューのローカルキュー定義に対しては，-g オプションで取り出し許可を指定できません。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04200-E	I/O エラーが発生しました。	標準エラー出力
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA26162-E	指定したオブジェクト名が誤っています。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31078-I	ヘルプメッセージ	標準出力
KFCA31090-E	リモートキューのローカルキュー定義に対して-g オプションで取り出し許可を指定できません。	標準エラー出力
KFCA31091-E	モデルキューに対して、属性を変更できません。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqccontrn (MQC トランザクション状態の変更)

形式

```
mqccontrn {-c | -r | -f}
           {-n MQCゲートウェイサーバ名 |
            -t OpenTP1トランザクションID |
            -u クライアントトランザクションID | -m MQCID}
```

機能

指定した MQC ゲートウェイサーバ名、トランザクション ID、または MQCID に対応するトランザクションの状態をコミットまたはロールバックに決着したり、無効にしたりします。

オプション

● -c

MQC サーバ機能で管理しているトランザクションの状態をコミットに決着します。ただし、ローカルトランザクション中の場合はトランザクション状態を変更しません。

● -r

MQC サーバ機能で管理しているトランザクションの状態をロールバックに決着します。ただし、ローカルトランザクション中の場合はトランザクション状態を変更しません。

● -f

MQC サーバ機能で管理しているトランザクションの状態を無効にします。

● -n MQC ゲートウェイサーバ名

～< 1～8 文字の識別子 >

状態を変更する MQC ゲートウェイサーバ名を指定してください。MQC ゲートウェイサーバ名は、MQC サービス定義の mqcgwpsnam 定義コマンドで指定した名称を指定します。

● -t OpenTP1 トランザクション ID

～< 2～254 文字の 16 進数 >

状態を変更する OpenTP1 トランザクション ID を指定してください。OpenTP1 トランザクション ID は mqclstrn コマンドの -a オプションまたは -n オプションで出力された ID を指定します。

● -u クライアントトランザクション ID

～< 2～254 文字の 16 進数 >

状態を変更するクライアントトランザクション ID を指定してください。クライアントトランザクション ID は mqclstrn コマンドの -a オプションまたは -n オプションで出力された ID を指定します。ただし、ローカルトランザクション中の場合は指定できません。

● -m MQCID

～< 8 文字の識別子>

状態を変更する MQC サーバ機能で管理しているトランザクションごとの ID を指定してください。MQCID は mqclstrn コマンドの -a オプションまたは -n オプションで出力された ID を指定します。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA30861-E	MQC サービスが異常を検知しました。	メッセージログファイル
KFCA30866-E	メモリ不足が発生しました。	標準エラー出力
KFCA30872-I	ヘルプメッセージ	標準出力
KFCA30873-E	指定した MQC ゲートウェイサーバ名は存在しません。	標準エラー出力
KFCA30874-E	指定したトランザクション ID は存在しません。	標準エラー出力
KFCA30875-E	指定した MQCID は存在しません。	標準エラー出力
KFCA30876-I	該当するトランザクション ID のトランザクションをコミットしました。	メッセージログファイル
KFCA30877-I	該当するトランザクション ID のトランザクションをロールバックしました。	メッセージログファイル
KFCA30880-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA30881-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA30882-E	MQC サービスが終了処理中です。	標準エラー出力
KFCA30883-E	PRC においてネットワーク障害が発生しました。	標準エラー出力
KFCA30884-E	MQC サービスが起動されていません。	標準エラー出力
KFCA30885-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA30887-E	サービスとクライアント間のバージョンが一致しません。	標準エラー出力
KFCA30900-E	タイムアウトとなりました。	標準エラー出力
KFCA30901-E	通信障害が発生しました。	標準エラー出力
KFCA30902-I	該当するトランザクション ID のトランザクションを無効にしました。	メッセージログファイル
KFCA30903-I	MQC ゲートウェイサーバのトランザクションをコミットしました。	メッセージログファイル

出力メッセージ ID	内容	出力先
KFCA30904-I	MQC ゲートウェイサーバのトランザクションをロールバックしました。	メッセージログファイル
KFCA30905-I	MQC ゲートウェイサーバのトランザクションを無効にしました。	メッセージログファイル
KFCA30906-E	指定した MQC ゲートウェイサーバ名のトランザクション状態は変更できません。	標準エラー出力
KFCA30907-E	指定したトランザクション ID のトランザクション状態は変更できません。	標準エラー出力
KFCA30908-I	該当する MQCID のトランザクションをコミットしました。	メッセージログファイル
KFCA30909-I	該当する MQCID のトランザクションをロールバックしました。	メッセージログファイル
KFCA30910-I	該当する MQCID のトランザクションを無効にしました。	メッセージログファイル
KFCA30911-E	指定した MQCID のトランザクション状態は変更できません。	標準エラー出力
KFCA30912-E	指定した MQC ゲートウェイサーバ名の文字数に誤りがあります。	標準エラー出力
KFCA30913-E	MQC ゲートウェイサーバ名に指定できない文字を指定しています。	標準エラー出力
KFCA30914-E	トランザクション ID の文字数に誤りがあります。	標準エラー出力
KFCA30915-E	トランザクション ID に指定できない文字を指定しています。	標準エラー出力
KFCA30916-E	MQCID に誤りがあります。	標準エラー出力
KFCA30918-E	トランザクションは存在しません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqcls (MQC ゲートウェイサーバの状態表示)

形式

```
mqcls
```

機能

MQC ゲートウェイサーバの状態を標準出力に出力します。

出力形式

mqcls コマンドを実行した結果、出力される情報を次に示します。MQC サービス定義の mqcgwpsnam 定義コマンドの定義数分 (1~256) 表示されます。

```
MQCゲートウェイサーバ名 = aa....aa
MQCゲートウェイサーバステータス = bb....bb
クライアントIPアドレス = cc....cc
クライアントポート番号 = dd....dd
クライアントプロセスID = ee....ee
クライアントスレッドID = ff....ff
接続開始時刻 = hhh iii jj kk:ll:mm nnnn [oooooo]
```

aa....aa

MQC ゲートウェイサーバ名 (8 文字以内の識別子)

bb....bb

MQC ゲートウェイサーバのステータス (7 文字以内の文字列)

MQC ゲートウェイサーバには次の四つの状態があります。

なお、ステータスが INIT または STANDBY の場合、cc....cc 以降に表示される情報は無効です。

INIT

MQC サービス定義の mqcgwpsnam 定義コマンドで MQC ゲートウェイサーバは定義されていますが、MQC ゲートウェイサーバが起動していない状態です。

STANDBY

MQC ゲートウェイサーバがクライアントアプリケーションと接続していない状態です。次に示すときにこの状態になります。

- MQC ゲートウェイサーバの起動後、クライアントアプリケーションから、MQCONN 命令^{*1} が発行されるまでの間
- クライアントアプリケーションが MQDISC 命令^{*2} を発行してから、次の MQCONN 命令を発行するまでの間
- MQC ゲートウェイサーバが RESERVE または CONNECT の状態で、タイムアウト (要求受信監視タイマ、MQC ゲートウェイサーバ監視タイマ)、クライアントアプリケーションとの通信エラー、または MQA、トランザクションマネージャとの処理中でエラーが発生したとき

RESERVE

クライアントアプリケーションが MQCONN 命令^{※1} を発行してから、MQCONN 命令が完了するまでの状態^{※3} です。

CONNECT

クライアントアプリケーションからの MQCONN 命令^{※1} が完了してから、MQDISC 命令^{※2} を発行するまでの状態^{※3} です。

注※1

次の条件に該当する用語に読み替えてください。

環境変数に DCMQCEXPTRN=N を設定しているか、または非 XA インタフェースライブラリを使用している場合、MQCONN 命令を発行。

環境変数に DCMQCEXPTRN=Y を設定し、かつ XA インタフェースライブラリを使用している場合、xa_open を発行。

TP1/Message Queue Access の 05-00 未満のバージョンと接続する場合、MQCONN 命令を発行。

注※2

次の条件に該当する用語に読み替えてください。

環境変数に DCMQCEXPTRN=N を設定しているか、または非 XA インタフェースライブラリを使用している場合、MQDISC 命令を発行。

環境変数に DCMQCEXPTRN=Y を設定し、かつ XA インタフェースライブラリを使用している場合、xa_close を発行。

TP1/Message Queue Access の 05-00 未満のバージョンと接続する場合、MQDISC 命令を発行。

注※3

トランザクションのコミット、またはロールバック前に MQDISC 命令を発行した場合は、コミット、またはロールバック時に再度クライアントアプリケーションと接続するため、一時的に RESERVE、または CONNECT の状態になります。

cc....cc

クライアントの IP アドレス (15 文字以内の符号なし整数)

dd....dd

クライアントのポート番号 (10 進数)

ee....ee

クライアントのプロセス ID (10 進数)

ff....ff

クライアントのスレッド ID (10 進数)

hhh iii jj kk:ll:mm nnnn [oooooo]

曜日 月 日 時:分:秒 年 (西暦) [マイクロ秒]

接続開始時刻が表示されます。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04202-E	通信障害が発生しました。	標準エラー出力
KFCA04207-E	MQA が異常を検知しました。	メッセージログファイル
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04215-E	タイムアウトとなりました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA30861-E	MQC サービスが異常を検知しました。	メッセージログファイル
KFCA30865-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA30866-E	メモリ不足が発生しました。	標準エラー出力
KFCA30879-I	ヘルプメッセージ	標準出力
KFCA30880-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA30881-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA30884-E	MQC サービスが起動されていません。	標準エラー出力 メッセージログファイル
KFCA30887-E	サービスとクライアント間のバージョンが一致しません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqclstrn (MQC トランザクション状態の表示)

形式

```
mqclstrn {-a | -n MQCゲートウェイサーバ名}
```

機能

MQC ゲートウェイサーバ名に対応するトランザクションの状態を表示します。

オプション

● -a

MQC サーバ機能で管理しているすべてのトランザクション情報を表示します。

● -n MQC ゲートウェイサーバ名

～< 1～8 文字の識別子 >

表示する MQC ゲートウェイサーバ名を指定してください。MQC ゲートウェイサーバ名は、MQC サービス定義の mqcgwpsnam 定義コマンドで指定した名称を指定します。

出力形式

```
MQCゲートウェイサーバ名 = aa....aa  
MQCID = bb....bb  
トランザクション開始時間 = cc....cc  
トランザクション状態 = dd....dd  
OpenTP1トランザクションID = ee....ee  
クライアントトランザクションID = ff....ff
```

aa....aa

MQC ゲートウェイサーバ名 (8 文字以内の識別子) です。

bb....bb

MQCID (8 文字の識別子) です。

cc....cc

トランザクション開始時間を YYYY/MM/DD hh:mm:ss の形式で表示します。YYYY/MM/DD は年 (西暦) 月日, hh:mm:ss は時分秒を示します。

dd....dd

トランザクション状態を次の値で表示します。

- T: トランザクション中
- P: プリペア中
- R: 決着待ち

- HC：ヒューリスティックコミット
- HR：ヒューリスティックロールバック
- HH：ヒューリスティック不明
- LT：ローカルトランザクション中

ee....ee

OpenTP1 トランザクション ID (254 文字以内の 16 進数) です。

OpenTP1 トランザクション ID の形式

AAAAAAAAAAAAAAAABBBBBBBCCCCCCCCCCCCCCCCDDDDDDDEEEEEEE

- AAAAAAAAAAAAAAAAAA :

OpenTP1 のシステムノード ID (16 けたの 16 進数) です。16 進数を文字列 (すべて英数字) に変換することで, OpenTP1 の trnls コマンドで表示される「TRNGID: OpenTP1 のシステムノード ID (8 文字)」に対応づけることができます。

- BBBBBBBB :

グローバルトランザクション番号 (8 けたの 16 進数) です。OpenTP1 の trnls コマンドで表示される「グローバルトランザクション番号 (8 けた)」に対応づけることができます。

- CCCCCCCCCCCCCCCC :

OpenTP1 のシステムノード ID (16 けたの 16 進数) です。16 進数を文字列 (すべて英数字) に変換することで, OpenTP1 の trnls コマンドで表示される「TRNBID: OpenTP1 のシステムノード ID (8 文字)」に対応づけることができます。

- DDDDDDDD :

トランザクションブランチ番号 (8 けたの 16 進数) です。OpenTP1 の trnls コマンドで表示される「トランザクションブランチ番号 (8 けた)」に対応づけることができます。

- EEEEEEEE :

付加情報 (8 けたの 16 進数) です。

ff....ff

クライアントトランザクション ID (254 文字以内の 16 進数) です。

ローカルトランザクションの場合は***と表示します。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA30861-E	MQC サービスが異常を検知しました。	メッセージログファイル
KFCA30865-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA30866-E	メモリ不足が発生しました。	標準エラー出力
KFCA30870-I	ヘルプメッセージ	標準出力

出力メッセージID	内容	出力先
KFCA30873-E	指定した MQC ゲートウェイサーバ名は存在しません。	標準エラー出力
KFCA30880-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA30881-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA30882-E	MQC サービスが終了処理中です。	標準エラー出力
KFCA30883-E	RPC においてネットワーク障害が発生しました。	標準エラー出力
KFCA30884-E	MQC サービスが起動されていません。	標準エラー出力
KFCA30885-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA30887-E	サービスとクライアント間のバージョンが一致しません。	標準エラー出力
KFCA30900-E	タイムアウトとなりました。	標準エラー出力
KFCA30901-E	通信障害が発生しました。	標準エラー出力
KFCA30912-E	指定した MQC ゲートウェイサーバ名の文字数に誤りがあります。	標準エラー出力
KFCA30913-E	MQC ゲートウェイサーバ名に指定できない文字を指定しています。	標準エラー出力
KFCA30918-E	トランザクションは存在しません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqcswptrc (MQC トレースの強制スワップ)

形式

```
mqcswptrc
```

機能

MQC サーバの MQC トレース情報を強制的に MQC トレースファイルにスワップします。

このコマンドを入力するとメモリ中の MQC トレースが MQC トレースファイルに出力されます。MQC トレースファイルの取得情報については、「[7.4 障害時に取得する情報](#)」を参照してください。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA30861-E	MQC サービスが異常を検知しました。	メッセージログファイル
KFCA30866-E	メモリ不足が発生しました。	標準エラー出力
KFCA30878-I	ヘルプメッセージ	標準出力
KFCA30880-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA30881-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA30882-E	MQC サービスが終了処理中です。	標準エラー出力
KFCA30883-E	RPC においてネットワーク障害が発生しました。	標準エラー出力
KFCA30884-E	MQC サービスが起動されていません。	標準エラー出力
KFCA30887-E	サービスとクライアント間のバージョンが一致しません。	標準エラー出力
KFCA30888-I	運用コマンドが終了しました。	標準出力
KFCA30900-E	タイムアウトとなりました。	標準エラー出力
KFCA30901-E	通信障害が発生しました。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqrsls (クラスタ情報の表示)

形式

```
mqrsls [-x] [-u]
        { -c | -m | -q | -t }
        [-s クラスタ情報開始番号]
        [-e 出力メッセージ数]
        [オブジェクト名]
```

機能

指定したクラスタ情報を表示します。

オプション

● -x

一覧で出力する場合に指定します。

● -c

出力するクラスタ名称を指定します。

● -m

出力するクラスタキューマネージャ名称を指定します。

● -q

出力するクラスタキュー名称を指定します。

● -t

出力する問い合わせ名を指定します。

問い合わせ名は、キューマネージャ名、キューマネージャ識別子、キュー名を指定します。

● -s クラスタ情報開始番号

~< 1~2147483647 >

指定したクラスタ情報の先頭からの開始番号を指定します。

このオプションを省略すると、指定したクラスタ情報の先頭が指定されたものと見なします。

● -e 出力メッセージ数

~< 1~800 >

-s オプションで指定されたクラスタ情報開始番号から、クラスタ情報を表示する数を指定します。

このオプションを省略すると、最大値が指定されたものと見なします。

● -u

クラスタキューマネージャ、クラスタキュー、問い合わせの各情報にオブジェクトの有効期限情報を付加して表示します。-u オプションを有効にするには、-m、-q、-t オプションのどれかに加えて-x オプションと同時に指定してください。-x オプションを指定しない場合、このオプションは無視されます（有効期限情報は表示されません）。

リポジトリ情報の保持期間

各リポジトリ情報の保持期間を次の表 6-2、表 6-3、表 6-4 に示します。

表 6-2 リモートキューマネージャ情報の保持期間

リポジトリ情報名	内容	保持期間※
クラスタキューマネージャ情報	リモートのキューマネージャに存在するクラスタレシーバチャンネル情報	60～90 日
クラスタキュー情報	リモートのキューマネージャに存在するクラスタキュー情報	60～90 日

注※

リモートのキューマネージャの情報を受信したときからローカルキューマネージャ上で削除されるまでの期間。
この期間は、フルリポジトリキューマネージャが設定した有効期間に 60 日の猶予期間を加えた合計値です。

表 6-3 ローカルキューマネージャ情報の保持期間

リポジトリ情報名	内容	保持期間※
クラスタキューマネージャ情報	ローカルキューマネージャに存在するクラスタレシーバチャンネル情報	90 日
クラスタキュー情報	ローカルキューマネージャに存在するクラスタキュー情報	90 日

注※

ローカルキューマネージャの情報を送信したときからフルリポジトリキューマネージャで削除されるまでの期間。
この期間は、クラスタ転送キュー（SYSTEM.CLUSTER.TRANSMIT.QUEUE）に登録したときの有効期間（30 日）とフルリポジトリキューマネージャが受信した際に加算する猶予期間（60 日）の合計値です。

表 6-4 問い合わせ情報の保持期間

リポジトリ情報名	内容	保持期間※
キューマネージャ問い合わせ情報	リモートのキューマネージャの問い合わせ情報	30 日
キューマネージャ識別子問い合わせ情報	リモートのキューマネージャ識別子の問い合わせ情報	30 日
キュー問い合わせ情報	リモートのキューマネージャに存在するクラスタキューの問い合わせ情報	30 日

注※

ローカルキューマネージャが送信した情報がフルリポジトリキューマネージャで削除されるまでの期間。
この期間は、ローカルキューマネージャが問い合わせ情報をクラスタ転送キュー（SYSTEM.CLUSTER.TRANSMIT.QUEUE）に登録したときの有効期間（30 日）です。

コマンド引数

● オブジェクト名

～< 1～48 文字の文字列>

出力する情報のオブジェクト名称を指定します。

出力形式

● クラスタ情報

(a) 一覧表示 (mqrls -x -c)

```
エントリ表示番号 = aa - bb  
エントリ番号 = cc  
クラスタ名 = dd....dd
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1～48 文字の MQ 文字列)

(b) 詳細表示 (mqrls -c クラスタ名称)

```
エントリ表示番号 = aa - bb  
エントリ番号 = cc  
クラスタ名 = dd....dd  
UUID = ee....ee  
参加状態 = ff....ff  
参加時使用チャンネル名1 = gg....gg  
参加時使用チャンネル名2 = hh....hh
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

キューマネージャ識別子 (1~48 文字の文字列)

参加状態が FALSE の場合は、表示されないことがあります。

ff....ff

参加状態。

BINDING : クラスタ参加中です。

FALSE : クラスタ参加失敗です。

INIT : クラスタ未参加です。

JOIN : クラスタ参加済みです。

REMOVING : 脱退依頼中です。

REMOVE : 脱退済みです。

REFRESHING : リポジトリ情報再作成依頼中です。

REFRESH : リポジトリ情報再作成処理中です。

gg....gg

参加時に使用したチャンネル名 1 (1~20 文字の MQ 文字列)

参加時に使用したチャンネル名 1 が設定されていない場合、****を表示します。

hh....hh

参加時に使用したチャンネル名 2 (1~20 文字の MQ 文字列)

参加時に使用したチャンネル名 2 が設定されていない場合、****を表示します。

● クラスタキューマネージャ情報

(a) 一覧表示 (mqrls -x -m)

```
エントリ表示番号 = aa - bb  
エントリ番号 = cc  
クラスタ名 = dd....dd  
キューマネージャ名 = ee....ee  
チャンネル名 = ff....ff
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

キューマネージャ名 (1~48 文字の MQ 文字列)

ff....ff

チャンネル名 (1~20 文字の MQ 文字列)

(b) 詳細表示 (mqrls -m クラスタキューマネージャ名称)

```
エントリ表示番号 = aa - bb
エントリ番号 = cc
クラスタ名 = dd....dd
キューマネージャ名 = ee....ee
チャンネル名 = ff....ff
UUID = gg....gg
定義種別 = hh....hh
キューマネージャタイプ = ii....ii
サスペンド状態 = jj....jj
クラスタ情報更新日付 = YYYY-MM-DD
クラスタ情報更新時刻 = HH.MM.SS
接続名 = kk....kk
注釈 = ll....ll
チャンネル更新日付 = YYYY-MM-DD
チャンネル更新時刻 = HH.MM.SS
転送タイプ = mm....mm
チャンネルタイプ = nn....nn
バッチサイズ = oo....oo
切断時間間隔 = pp....pp
短期確立再試行回数 = qq....qq
短期確立再試行間隔 = rr....rr
長期確立再試行回数 = ss....ss
長期確立再試行間隔 = tt....tt
セキュリティ出口名 = uu....uu
セキュリティ出口ユーザデータ = vv....vv
メッセージシーケンス番号ラップ値 = ww....ww
最大メッセージ長 = xx....xx
MCAユーザID = yy....yy
ハートビートフロー間の時間 = zz....zz
バッチ終了待ちタイマ値 = a1....a1
非永続メッセージが送信される速度 = b1....b1
ネットワーク接続優先順位 = c1....c1
チャンネル状態 = d1....d1
送信出口名 = e1....e1
送信出口ユーザデータ = f1....f1
メッセージ出口名 = g1....g1
メッセージ出口ユーザデータ = h1....h1
受信出口名 = i1....i1
受信出口ユーザデータ = j1....j1
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

キューマネージャ名 (1~48 文字の MQ 文字列)

ff....ff

チャンネル名 (1~20 文字の MQ 文字列)

gg....gg

キューマネージャ識別子 (1~48 文字の文字列)

hh....hh

定義種別

AUTOINST

自動的に作成されたクラスタセンダチャンネルです。

MANINST

事前定義クラスタセンダチャンネルまたはクラスタレシーバチャンネルです。

ii....ii

キューマネージャタイプ

FULL : フルリポジトリです。

PARTIAL : パーシャルリポジトリです。

jj....jj

サスペンド状態

YES : サスペンド状態です。

NO : サスペンド状態ではありません。

YYYY-MM-DD

日付 (年-月-日)

チャンネル更新日付およびクラスタ情報更新日付は、システムの設定日付で表示されます。日付が設定されていない場合、****を表示します。

HH.MM.SS

時刻 (時.分.秒)

チャンネル更新時刻およびクラスタ情報更新時刻は、システムの設定時刻で表示されます。時刻が設定されていない場合、****を表示します。

kk....kk

接続名 (文字列)

ll....ll

注釈記述用のフィールド (1~64文字の文字列)

mm....mm

転送タイプ (文字列)

nn....nn

チャンネルタイプ

crc : クラスタレシーバチャンネルです。

csn : クラスタセンダチャンネルです。

oo....oo

バッチサイズ

pp....pp

切断時間間隔

qq....qq

短期確立再試行回数

rr....rr

短期確立再試行間隔

ss....ss

長期確立再試行回数

tt....tt

長期確立再試行間隔

uu....uu

セキュリティ出口名

vv....vv

セキュリティ出口ユーザデータ

ww....ww

メッセージシーケンス番号ラップ値

xx....xx

最大メッセージ長

yy....yy

MCA ユーザ ID

zz....zz

ハートビートフロー間の時間

al....al

バッチ終了待ちタイマ値

bl....bl

非永続メッセージが送信される速度

FAST：非永続メッセージを高速に転送します。

NORMAL：非永続メッセージを永続メッセージと同様に転送します。

cl....cl

ネットワーク接続優先順位

dl....dl

チャンネル状態

ACTIVE：動作可能です。

BINDING：接続中です。

DISABLE：動作不可能です。

RETRY：リトライ中です。

el....el

送信出口名

fl....fl

送信出口ユーザデータ

gl....gl

メッセージ出口名

hl....hl

メッセージ出口ユーザデータ

il....il

受信出口名

jl....jl

受信出口ユーザデータ

● クラスタキュー情報

(a) 一覧表示 (mqrls -x -q)

```
エントリ表示番号 = aa - bb  
エントリ番号 = cc....cc  
クラスタ名 = dd....dd  
キューマネージャ名 = ee....ee  
キュー名 = ff....ff
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

キューマネージャ名 (1~48 文字の MQ 文字列)

ff....ff

キュー名 (1~48 文字の MQ 文字列)

(b) 詳細表示 (mqrls -q クラスタキュー名称)

```
エントリ表示番号 = aa - bb
エントリ番号 = cc
クラスタ名 = dd....dd
キューマネージャ名 = ee....ee
キュー名 = ff....ff
UUID = gg....gg
クラスタキュータイプ = hh....hh
定義種別 = ii....ii
キュー記述子 = jj....jj
登録許可 = kk....kk
省略時のメッセージ優先度 = ll....ll
省略時のメッセージ永続性 = mm....mm
作成日付 = YYYY-MM-DD
作成時刻 = HH.MM.SS
クラスタ情報更新日付 = YYYY-MM-DD
クラスタ情報更新時刻 = HH.MM.SS
デフォルトのバインディング = nn....nn
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

キューマネージャ名 (1~48 文字の MQ 文字列)

ff....ff

キュー名 (1~48 文字の MQ 文字列)

gg....gg

キューマネージャ識別子 (1~48 文字の文字列)

hh....hh

クラスタキュータイプ

ALIAS QUE : 別名キュー定義です。

ALIAS QUE MGR : 別名キューマネージャです。

LOCAL QUE : ローカルキューです。

REMOTE QUE : リモートキューのローカル定義です。

ii....ii

定義種別

LOCAL : ローカルに存在します。

REMOTE : リモートに存在します。

jj....jj

キュー記述子 (1~64 文字の文字列)

kk....kk

キューに対する登録許可

ALLOWED : 取り出し許可は許可されています。

INHIBITED : 登録操作は禁止されます。

ll....ll

省略時のメッセージ優先度 (10 進文字)

mm....mm

省略時のメッセージ永続性

NOT PERSISTENT : 非永続性です。

PERSISTENT : 永続性です。

YYYY-MM-DD

日付 (年-月-日)

作成日付およびクラスタ情報更新日付は、システムの設定日付で表示されます。日付が設定されていない場合、****を表示します。

HH.MM.SS

時刻 (時.分.秒)

作成時刻およびクラスタ情報更新時刻は、システムの設定時刻で表示されます。時刻が設定されていない場合、****を表示します。

nn....nn

デフォルトのバインディング

FIXED

キューがオープンされたハンドルにあて先を固定します。

NOT FIXED

キューにメッセージが登録される時，メッセージ単位にあて先を決定します。

● クラスタキューマネージャ情報と有効期限

(a) 一覧表示 (mqrls -x -m -u)

```
エントリ表示番号 = aa - bb
エントリ番号 = cc
クラスタ名 = dd....dd
キューマネージャ名 = ee....ee
チャンネル名 = ff....ff
有効期限 = ggg....ggg
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

キューマネージャ名 (1~48 文字の MQ 文字列)

ff....ff

チャンネル名 (1~20 文字の MQ 文字列) ※

注※

ローカルキューマネージャの事前定義クラスタセンダチャンネルの情報も表示されますが，フルリポジトリキューマネージャには情報は通知されません。そのため，ローカルキューマネージャの事前定義クラスタセンダチャンネルの有効期限は無視してください。

ggg....ggg

有効期限 (曜日月日時:分:秒 年 (西暦))

出力例 Wed Mar 19 11:32:56 2014

次に示す情報の保持期間が切れる期限が表示されます。

- リモートのキューマネージャに存在するクラスタレシーバチャンネル情報

- ローカルキューマネージャに存在するクラスタレシーバチャンネル情報

● クラスタキュー情報と有効期限

(a) 一覧表示 (mqrls -x -q -u)

```
エントリ表示番号 = aa - bb  
エントリ番号 = cc  
クラスタ名 = dd....dd  
キューマネージャ名 = ee....ee  
キュー名 = ff....ff  
有効期限 = ggg....ggg
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

キューマネージャ名 (1~48 文字の MQ 文字列)

ff....ff

キュー名 (1~48 文字の MQ 文字列)

ggg....ggg

有効期限 (曜日月日時:分:秒年 (西暦))

出力例 Wed Mar 19 11:32:56 2014

次に示す情報の保持期間が切れる期限が表示されます。

- リモートのキューマネージャに存在するクラスタキュー情報
- ローカルキューマネージャに存在するクラスタキュー情報

● 問い合わせ情報

(a) 一覧表示 (mqrls -x -t)

```
エントリ表示番号 = aa - bb  
エントリ番号 = cc  
クラスタ名 = dd....dd  
問い合わせ名 = ee....ee  
問い合わせ種別 = ff....ff
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd...dd

クラスタ名 (1~48 文字の MQ 文字列)

ee...ee

問い合わせ名 (1~48 文字の文字列)

ff...ff

問い合わせ種別

MGR: キューマネージャです。

UUID: キューマネージャ識別子です。

QUE: キューです。

(b) 詳細表示 (mqrls -t 問い合わせ名)

```
エントリ表示番号 = aa - bb
エントリ番号 = cc
クラスタ名 = dd....dd
問い合わせ名 = ee....ee
問い合わせ種別 = ff....ff
作成日付 = YYYY-MM-DD
作成時刻 = HH.MM.SS
問い合わせチャンネル名1 = gg....gg
問い合わせチャンネル名2 = hh....hh
返却結果 = ii....ii
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd...dd

クラスタ名 (1~48 文字の MQ 文字列)

ee...ee

問い合わせ名 (1~48 文字の文字列)

ff....ff

問い合わせ種別

MGR：キューマネージャです。

UUID：キューマネージャ識別子です。

QUE：キューです。

YYYY-MM-DD

作成日付（年-月-日）

作成日付は、システムの設定日付で表示されます。

HH.MM.SS

作成時刻（時.分.秒）

作成時刻は、システムの設定時刻で表示されます。

gg....gg

問い合わせを実行したチャンネル名 1（1～20 文字の MQ 文字列）

問い合わせを実行したチャンネル名 1 が設定されていない場合、****を表示します。

hh....hh

問い合わせを実行したチャンネル名 2（1～20 文字の MQ 文字列）

問い合わせを実行したチャンネル名 2 が設定されていない場合、****を表示します。

ii....ii

返却結果

YES：返却結果があります。

NO：返却結果がありません。

WAIT：問い合わせ中です。

(c) 一覧表示 (mqrls -x -t -u)

```
エントリ表示番号 = aa - bb
エントリ番号 = cc
クラスタ名 = dd....dd
問い合わせ名 = ee....ee
問い合わせ種別 = ff....ff
有効期限 = ggg...ggg
```

aa

エントリ開始番号

bb

エントリ終了番号

cc

エントリ番号

dd....dd

クラスタ名 (1~48 文字の MQ 文字列)

ee....ee

問い合わせ名 (1~48 文字の文字列)

ff....ff

問い合わせ種別

MGR: キューマネージャです。

UUID: キューマネージャ識別子です。

QUE: キューです。

ggg....ggg

有効期限 (曜日 月 日 時:分:秒 年 (西暦))

出力例 Wed Mar 19 11:32:56 2014

次に示す情報の保持期間が切れる期限が表示されます。

- キューマネージャ問い合わせ情報
- キューマネージャ識別子問い合わせ情報
- キュー問い合わせ情報

注意事項

リポジトリ管理サーバが開始していない場合はエラーになります。また、指定した情報が存在しない場合もエラーになります。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA26162-E	指定したオブジェクト名が誤っています。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31030-E	コマンド実行中に異常を検知しました。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31123-I	ヘルプメッセージ	標準出力
KFCA31124-E	リポジトリ管理サーバが起動されていません。	標準エラー出力
KFCA31172-E	指定した情報は存在しません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力
KFCA31205-I	継続情報が存在します。	標準出力

mqrrefresh (リポジトリ情報の再作成)

形式

```
mqrrefresh [ {-f | [-m] [-t] } ] {-c クラスタ名 | -a}
```

機能

リポジトリ情報を削除し、再作成します。

オプションの指定と削除するリポジトリ情報について、次の表に示します。

表 6-5 オプションの指定と削除するリポジトリ情報

リポジトリ情報	すべて省略	-f	-m	-t	-m -t
フルリポジトリキューマネージャ情報 (フルリポジトリのクラスタレシーバチャンネル情報)	×	○	×	×	×
パーシャルリポジトリキューマネージャ情報 (パーシャルリポジトリのクラスタレシーバチャンネル情報)	×	○	○	×	○
リモートに存在するクラスタキュー情報	○	○	○	○	○
マネージャ問い合わせ情報	×	○	○	×	○
キュー問い合わせ情報	○	○	○	×	×

(凡例)

- ：削除します。
- ×

コマンドを入力する状況と指定するオプションについて、次の表に示します。

表 6-6 コマンドを入力する状況と指定するオプション

コマンドを入力する状況	指定するオプション
相手システムのクラスタキューを使用できない場合	すべて省略
次に示す条件を同時に満たす場合 • 相手システムのクラスタキューを使用できません。 • 相手システム (フルリポジトリキューマネージャ) のクラスタレシーバチャンネルを認識できません。	-f
次に示す条件を同時に満たす場合 • 相手システムのクラスタキューを使用できません。 • 相手システム (パーシャルリポジトリキューマネージャ) のクラスタレシーバチャンネルを認識できません。	-m
次に示す条件を同時に満たす場合 • 相手システムのクラスタキューを使用できません。	-t

コマンドを入力する状況	指定するオプション
<ul style="list-style-type: none"> 以前のアクセス時にフルリポジトリキューマネージャに場所を問い合わせたオブジェクトを、あらかじめ mqrfresh コマンド入力時に問い合わせしておきます。 	-t
次に示す条件を同時に満たす場合 <ul style="list-style-type: none"> 相手システムのクラスタキューを使用できません。 相手システム（パーシャルリポジトリキューマネージャ）のクラスタレシーバチャンネルを認識できません。 以前のアクセス時にフルリポジトリキューマネージャに場所を問い合わせたオブジェクトを、あらかじめ mqrfresh コマンド入力時に問い合わせしておきます。 	-m -t

オプション

● -f

フルリポジトリキューマネージャ情報を再作成します。

このオプションを省略すると、フルリポジトリキューマネージャについての情報は再作成されません。

● -m

リモートのパーシャルリポジトリキューマネージャ情報を削除します。

● -t

過去に問い合わせた情報を削除しないで残します。

アクセスするチャンネル情報やキュー情報を保持していない場合、リポジトリ管理サーバはこれらの情報をフルリポジトリに問い合わせます。このオプションを指定すると、リポジトリ情報の再作成時に情報を削除しないで残します。

● -c クラスタ名

～< 1～48 文字の MQ 文字列>

リポジトリ情報を再作成するクラスタ名を指定します。

● -a

リポジトリ管理サーバが参加しているすべてのクラスタに対し、リポジトリ情報の再作成処理を行います。

実行手順

1. クラスタキューにアクセスするすべてアプリケーションを終了します。
2. mqstpcha コマンドでクラスタレシーバチャンネルを終了します。
3. SYSTEM.CLUSTER.TRANSMIT.QUEUE にメッセージが滞留していないことを確認します。
KFCA16350-I, KFCA16351-I または KFCA16352-W が出力されることを確認してください。

クラスタセンダチャンネルは、mqststpcha コマンドを入力すると、該当チャンネルに割り当てられたメッセージを別のチャンネルで送信するため、メッセージ送信経路の再設定処理を行います。再設定処理では、転送キューに登録されているメッセージを再登録します。

KFCA16350-I, KFCA16351-I または KFCA16352-W が出力されない場合は、mqststpcha -f コマンドを入力してチャンネルを強制解放してください。

4. チャンネル状態を確認して、送達未確認メッセージがないことを確認します。

チャンネルの送達未確認メッセージがないことを確認するには、mqtlscha コマンドで、クラスタセンダチャンネルの状態を確認します。

```
mqtlscha -n クラスタセンダチャンネル名
```

送達未確認メッセージがある場合は、必ずそれを解決してから作業を進めてください。

解決しないで mqrrefresh コマンドを実行するとメッセージが二重に送信されるおそれがあります。

5. mqrrefresh コマンドを実行します。

KFCA31526-I および KFCA31527-I が出力されることを確認してください。

KFCA31526-I が出力されない場合、コマンドが異常終了しているおそれがあります。原因を取り除き、再度 mqrrefresh コマンドを実行してください。KFCA31527-I が出力されない場合、リポジトリ管理サーバが開始していることを確認してください。

6. クラスタレシーバチャンネルを開始します (mqststacha コマンドを実行します)。

7. クラスタセンダチャンネルを開始します (mqststacha コマンドを実行します)。

KFCA31529-I が出力されることを確認してください。

注意事項

1. リポジトリ管理サーバが開始していない場合はエラーとなります。また、指定したクラスタに参加していない場合もエラーとなります。
2. リポジトリ情報の再作成処理中に MQSET 命令を発行したり、mqaset コマンドを入力したりした場合、フルリポジトリに情報が反映されるのは再作成処理終了メッセージが出力されたあととなります。
3. フルリポジトリでキューマネージャをクラスタから強制的に除去したあとに mqrrefresh コマンドを実行する場合は、10 秒以上経過してから実行してください。強制的な除去はフルリポジトリ側だけの処理となりパーシャルリポジトリに通知は来ないため、メッセージを表示できません。このためフルリポジトリ側の処理完了をパーシャルリポジトリ側の運用者が待つ必要があります。
10 秒経過していない場合、mqrrefresh コマンドは無視され、再作成処理完了メッセージは出力されません。この状態になったときは、TP1/Message Queue で、再度 mqrrefresh -f コマンドを実行してください。
4. 再作成処理中に mqrstop または OpenTP1 が異常終了した場合、再作成処理は中断されます。その後、mqrstop 開始時に再作成処理は再開されますが、その際再び KFCA31527-I メッセージが出力されます。
5. KFCA31526-I メッセージ出力後 KFCA31529-I メッセージ出力までの間は、クラスタに未参加となるため、リモートのクラスタキューに対する MQOPEN, MQPUT, および MQPUT1 命令がエラー

となる場合があります。リフレッシュ処理完了後、再度 MQOPEN, MQPUT, および MQPUT1 命令を発行してください。

6. SYSTEM.CLUSTER.TRANSMIT.QUEUE にメッセージが滞留している状態で -m オプションを指定し mqrefresh コマンドを実行した場合、メッセージがあて先キューに送信されないで、クラスタ転送キューに滞留した状態となります。必ずメッセージが滞留していないことを確認してください。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31030-E	コマンド実行中に異常を検知しました。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31124-E	リポジトリ管理サーバが起動されていません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力
KFCA31502-E	MQI でエラーが発生しました	メッセージログファイル
KFCA31526-I	リポジトリ管理サーバに処理を依頼しました。	メッセージログファイル
KFCA31528-I	ヘルプメッセージ	標準出力
KFCA31530-E	リポジトリ管理サーバは指定したクラスタに参加していません。	標準エラー出力
KFCA31531-E	クラスタは再作成処理中です。	標準エラー出力
KFCA31536-E	リポジトリ管理サーバへの処理依頼に失敗しました。	メッセージログファイル
KFCA31538-E	指定したクラスタ名が誤っています。	標準エラー出力

mqrremove (クラスタからの脱退)

形式

```
mqrremove [ {-s | -q} ] {-c クラスタ名 | -a}
```

機能

リポジトリ管理サーバにクラスタからの脱退または脱退解除を依頼します。

リポジトリ管理サーバは参加済みのクラスタに所属するクラスタレシーバチャンネルについてフルリポジトリキューマネージャに削除依頼を送信します。または脱退済みのクラスタに所属するクラスタレシーバチャンネルについてフルリポジトリキューマネージャに追加依頼を送信します。

mqrremove コマンドの使用手順については、「[2.8.13 高度な作業](#)」を参照してください。

オプション

● -s

クラスタからの脱退を解除します。

● -q

フルリポジトリキューマネージャのリポジトリから不要なキュー情報を削除する場合に指定します。ローカルのクラスタキュー情報の削除をフルリポジトリに通知します。

● -c クラスタ名

～< 1～48 文字の MQ 文字列 >

脱退または脱退解除するクラスタ名を指定します。

● -a

リポジトリ管理サーバが参加しているすべてのクラスタから脱退します。または、脱退しているすべてのクラスタから脱退解除します。

注意事項

1. リポジトリ管理サーバが開始していない場合はエラーとなります。また、指定したクラスタ名が存在しない場合またはクラスタの参加状態がオプションと異なる場合もエラーとなります。
2. 脱退が完了するのは、クラスタメッセージがフルリポジトリに送信された時です。OpenTP1 を終了する前に、SYSTEM.CLUSTER.TRANSMIT.QUEUE の中にメッセージが滞留していないことを確認してください。メッセージが滞留している状態で、OpenTP1 を終了しキューファイルの再作成を行った場合は、脱退処理が完了していません。脱退する場合はクラスタを管理しているフルリポジトリから RESET CLUSTER コマンドに自システムのキューマネージャ名を指定して実行してください。

出力メッセージ

出力メッセージID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA04307-E	指定したオプションの数が誤っています。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31030-E	コマンド実行中に異常を検知しました。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31124-E	リポジトリ管理サーバが起動されていません。	標準エラー出力
KFCA31172-E	指定した情報は存在しません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力
KFCA31526-I	リポジトリ管理サーバに処理を依頼しました。	メッセージログファイル
KFCA31530-E	リポジトリ管理サーバは指定したクラスタに参加していません。	標準エラー出力
KFCA31532-I	ヘルプメッセージ	標準出力
KFCA31536-E	リポジトリ管理サーバへの処理依頼に失敗しました。	メッセージログファイル
KFCA31537-E	リポジトリ管理サーバは指定したクラスタから脱退していません。	標準エラー出力
KFCA31538-E	指定したクラスタ名が誤っています。	標準エラー出力

mqrswptrc (MQR トレースの強制スワップ)

形式

```
mqrswptrc
```

機能

MQR トレースを強制的に MQR トレースファイルにスワップします。

このコマンドはリポジトリ管理サーバユーザサービス定義ファイル (\$DCCONFPATH/mqrsup) で、MQR トレース情報を出力することを指定 (DCMQA_TRACE_STATUS に Y を指定) した場合に有効になります。このコマンドを実行するとメモリ中の MQR トレースが MQR トレースファイルに出力されます。

リポジトリ管理サーバユーザサービス定義ファイル (\$DCCONFPATH/mqrsup) で、MQR トレース情報を出力しないことを指定 (DCMQA_TRACE_STATUS に N を指定) した場合は、このコマンドを入力しても MQR トレースは MQR トレースファイルに出力されません。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA04208-E	MQA サーバの V/R が誤っています。	標準エラー出力
KFCA04209-E	メモリ不足で処理を続行できません。	標準エラー出力
KFCA04300-E	コマンドで指定した引数の数が正しくありません。	標準エラー出力
KFCA04301-E	定義解析中にエラーが発生しました。	標準エラー出力
KFCA04302-E	MQA サービスが終了処理中です。	標準エラー出力
KFCA04303-E	RPC でネットワーク障害が発生しました。	標準エラー出力
KFCA04304-E	MQA サービスが起動されていません。	標準エラー出力
KFCA26184-E	プロセス固有領域のメモリ不足で処理を続行できません。	標準エラー出力
KFCA31115-E	コマンドに対して処理結果を送信できません。	メッセージログファイル
KFCA31120-E	RPC インタフェースで異常を検知しました。	標準エラー出力
KFCA31124-E	リポジトリ管理サーバが起動されていません。	標準エラー出力
KFCA31173-E	OpenTP1 がオンライン状態ではないため、コマンドを実行できません。	標準エラー出力

mqtaltcha (チャンネル属性の変更)

形式

```
mqtaltcha [-s MQT通信プロセス識別子] -n チャンネル名
          [-l perm | temp]
          -o " [ {ohostname=相手システムのホスト名
                | oipaddr=相手システムのIPアドレス} ]
              [ {oservname=相手システムのサービス名
                | oportno=相手システムのポート番号} ] "
```

機能

チャンネル定義 (TCP 定義の mqtalcccha 定義コマンド) に指定されたチャンネルの属性 (通信相手システム) を変更します。

このコマンドは、チャンネル状態が「チャンネル使用不可」または「チャンネル停止」の場合に使用できます。

オプション

● -s MQT 通信プロセス識別子

~< 16 進数字 > ((01 ~ ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtaltcha コマンドを実行します。

● -n チャンネル名

~< 1~20 文字の MQ 文字列 >

属性を変更するチャンネルの名称を指定します。ただし、クラスタセンダチャンネルの名称は指定できません。

複数のチャンネル名を指定する場合は、引用符 (") で囲んで、チャンネル名とチャンネル名との間を空白で区切ります。同一チャンネル名は、重複して指定できません。

また、チャンネル名は、アスタリスク (*) を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャンネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

* : すべてのチャンネルの属性を変更します。

先行文字列* : 先行文字列で始まるチャンネルの属性を変更します。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャンネルを指定する場合

```
-n "cha*"
```

● -l perm | temp

～ 《temp》

このコマンドで変更したチャンネル属性をいつまで有効にするか指定します。

・ perm

次回以降の OpenTP1 開始時にもこのコマンドで変更後のチャンネル属性を有効にします。ただし、チャンネル情報を引き継がない状態で OpenTP1 を正常開始すると無効になり、TCP 定義の mqtalcccha 定義コマンドで指定した値に戻ります。

チャンネル状態を引き継がないようにする条件については、「[5.3.2 チャンネル情報を引き継がない条件](#)」を参照してください。

・ temp

このコマンドで変更後のチャンネル属性を、動作中の OpenTP1 を終了するまで有効にします。次回以降の OpenTP1 開始時は、このコマンドで変更後のチャンネル属性は無効になり、TCP 定義の mqtalcccha 定義コマンドで指定した値に戻ります。

また、perm を指定して属性を変更したあと、temp を指定して属性を設定すると、次回以降の OpenTP1 開始時には TCP 定義の mqtalcccha 定義コマンドに指定した値に戻ります。

このオプションを省略すると、temp が設定されます。

● -o

(オペランド)

チャンネルを確立する相手システムを変更する場合に指定します。また、LAN を二重化した構成では、ネットワークを変更できません。

変更するチャンネルがコーラ側の場合だけこのオプションは有効です。また、チャンネルタイプがレシーバ、クラスタセンダ、またはクラスタレシーバの場合は指定できません。

チャンネルタイプがサーバで、リクエストの相手システムの情報を無効にする場合は""だけを指定してください。

・ ohostname=相手システムのホスト名

～< 1～255 バイトの文字列>

ホスト名ファイルに登録した相手システムのホスト名を指定します。

- oipaddr=相手システムの IP アドレス
 ~<符号なし整数>((0~255))
 相手システムの IP アドレスを指定します。指定形式は nnn.nnn.nnn.nnn です。nnn の値は 0~255 まで指定できます。nnn の値として先頭に 0 を指定した場合は、8 進数値として扱われます。
- oservname=相手システムのサービス名
 ~<1~32 文字の識別子 (ハイフン (-) 含む) >
 相手システムのサービス名を指定します。
- oportno=相手システムのポート番号
 ~<符号なし整数>((1024~65535))
 相手システムのポート番号を指定します。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16355-I	コマンドを正常に受け付けました。	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16386-I	mqtalcccha コマンドの入力形式の表示	標準出力
KFCA16387-I	チャンネル属性を変更しました。	標準出力
KFCA16388-I	変更したチャンネル属性の表示	標準出力

出力メッセージ ID	内容	出力先
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力

mqtddmped (MQT 通信プロセス情報の表示)

形式

```
mqtddmped -i スナップダンプファイル名  
          [-n チャンネル名]  
          [-q 相手キューマネージャ名]  
          [-k 出力情報種別]
```

機能

mqttscha コマンドで取得したスナップダンプの情報を編集して、標準出力に出力します。

オプション

● -i スナップダンプファイル名

～< 1～32 文字の英数字 >

mqttscha コマンドで取得したスナップダンプファイル名を指定します。

スナップダンプファイルのパス名には、コマンドを実行した環境のカレントパスが設定されます。

● -n チャンネル名

～< 1～20 文字の MQ 文字列 >

状態を表示するチャンネルの名称を指定します。

-k オプションに mqcd が指定されている場合に有効です。

複数のチャンネル名を指定する場合は、引用符 (") で囲んで、チャンネル名とチャンネル名との間を空白で区切ります。同一チャンネル名は、重複して指定できません。

また、チャンネル名は、アスタリスク (*) を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャンネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャンネルを編集の対象にします。

先行文字列*：先行文字列で始まるチャンネルを編集の対象にします。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャンネルを指定する場合

```
-n "cha*"
```

● -q 相手キューマネージャ名

～< 1～48 文字の MQ 文字列>

状態を表示するチャンネルの相手キューマネージャ名を指定します。

-k オプションに mqcd が指定されている場合に有効です。

複数の相手キューマネージャ名を指定する場合は、引用符 (") で囲んで、相手キューマネージャ名と相手キューマネージャ名との間を空白で区切ります (コマンド入力可能な、けた数を指定できます)。同一相手キューマネージャ名は、重複して指定できません。

また、相手キューマネージャ名は、*を使って一括指定ができます。一括指定は、一つだけ指定できます。一括指定と一括指定以外の相手キューマネージャ名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*:すべてのチャンネルを編集の対象にします。

先行文字列*:先行文字列で始まる相手キューマネージャ名が一致するチャンネルの情報を標準出力に出力します。

<複数指定の例>

相手キューマネージャ名 1, 相手キューマネージャ名 2, および相手キューマネージャ名 3 を指定する場合

```
-q "相手キューマネージャ名1△相手キューマネージャ名2△相手キューマネージャ名3"
```

<一括指定の例>

QM1, QM2, および QM3 を一括して指定する場合

```
-q "QM*"
```

-n オプションと-q オプションを同時に省略すると、すべてのチャンネルを編集の対象にします。

● -k 出力情報種別

～<文字列> 《mqcd》

スナップダンプファイルから編集する情報の種別を指定します。次に示す種別を一つ以上指定できます。

- mqcd
チャンネルデータ定義ブロック (dcmtcq_uoc_mqcd) を編集の対象にします。
- process
MQT 通信プロセスの共通情報を編集の対象にします。

複数の種別を指定する場合は、引用符 (") で囲んで、種別と種別との間を空白で区切ります。

出力形式

```
KFCA16361-I PROCESS状態表示を開始します。
*****
* Snap dumping information edit display Date : yyyy-mm-dd Time : hh.mm.ss *
* Snap dumping information extraction Date : yyyy-mm-dd Time : hh.mm.ss *
*****

*****
*** PROCESS ***
*****
btim(基本タイマ値)
bmtim(基本タイマ値(ミリ秒))
InitQName(イニシエーションキュー名)
InitQMonType(a1...a1イニシエーションキュー監視方式)
PortNum(開始要求受信ポートのポート番号)
ServName(開始要求受信ポートのサービス名)
itim(イニシエーションキュー監視タイマ値)
stim(チャンネル要求監視間隔)
etim(終了処理監視タイマ値)
rtim(チャンネル確立完了監視タイマ値)
TraceDisk(b1...b1MQTトレースのディスク出力要否)
TraceBufCnt(MQTトレースのバッファ格納個数)
TraceFilCnt(MQTトレースのファイル個数)
TraceBufSize(c1...c1MQTトレースのバッファサイズ)
TraceFilePath(MQTトレースのディスク出力先パス名)

*****
*** MQCD ***
*****
ChannelName(チャンネル名)
PartnerQMName(接続先キューマネージャ名)
DefineType(aa...aa)
<Common>
Version(MQCDのバージョン)
ChannelType(bb...bb)
ClusterName(クラスタ名)
BatchSize(バッチサイズ)
XmitQName(転送キュー名)
MaxMsgLength(最大メッセージ長)
MaxSegLength(最大セグメントサイズ)
SeqNumberWrap(メッセージシーケンス番号ラップ値)
BufType(cc...cc)
SendBufGroup(送信バッファグループ番号)
ReceiveBufGroup(受信バッファグループ番号)
MsgEdtBufGroup(メッセージ編集出口編集用バッファグループ番号)
MsgEdtBufCount(メッセージ編集出口編集用バッファ数)
StartType(dd...dd)

<Separate>
LongRetryType(ee...ee)
LongRetryCount(長期確立時再試行の回数)
LongRetryInterval(長期確立時再試行の間隔)
OwnIPAddr(自ホストIPアドレス)
OwnHostName(自ホストのホスト名)
OwnPortNum(自ホストのポート番号)
OwnServName(自ホストのサービス名)
```

PartnerIPAddr(相手ホストのIPアドレス)
 PartnerHostName(相手ホストのホスト名)
 PartnerPortNum(相手ホストのポート番号)
 PartnerServName(相手ホストのサービス名)
 ConRspRecvInterval(チャネル確立応答受信監視タイマ値)
 RspMsgRecvInterval(確認メッセージ受信監視タイマ値)
 NextSegRecvInterval(継続セグメント受信監視タイマ値)
 DiscInterval(転送キュー監視間隔)
 DiscIntervalCount(転送キュー監視回数)
 DiscInterval2(切断時間間隔)
 DiscInterval2Effect(ff.... ff)
 TimeoutConRetry(gg.... gg)
 NextMsgRecvInterval(継続メッセージ受信監視タイマ値)
 HeartbeatInterval(ハートビートフロー間の時間)
 BatchInterval(バッチ終了待ちタイマ値)
 MsgEdtUserData(hh.... hh)
 TcpSendBufLength(ii.... ii)
 TcpRecvBufLength(jj.... jj)
 ConvCCSID(変換後CCSID)
 NonPersistentMsgSpeed(kk.... kk)
 TranQMonType(ll.... ll)
 NetworkPriority(ネットワーク接続優先順位)
 AlterTime(mm.... mm)
 MsgChaProRetry(nn.... nnMQプロトコル障害確立再試行可否)
 Adoptchk(oo.... oo強制再接続条件)
 ShortRetryType(pp.... pp短期確立再試行の可否)
 ShortRetryCount(短期確立再試行の回数)
 ShortRetryInterval(短期確立再試行の間隔)
 SendUserData(qq.... qq送信出口ユーザデータ)
 ReceiveUserData(rr.... rr受信出口ユーザデータ)
 SecurityUserData(ss.... ssセキュリティ出口ユーザデータ)
 SendExit(tt.... tt送信出口名)
 RcvExit(uu.... uu受信出口名)
 MsgExit(vv.... vvメッセージ出口名)
 ScyExit(ww.... wwセキュリティ出口名)
 Descr(xx.... xx注釈)
 MCAUser(yy.... yyMCAユーザID)
 MCAType(zz.... zzMCAタイプ)
 TcpRetryCount(TCP/IP確立確認の回数)
 TcpRetryInterval(TCP/IP確立確認の間隔)
 SndJnlIO(AA.... AA送信チャネルジャーナル出力条件)
 BatchMilInterval(バッチ終了待ちタイマ値(ミリ秒))
 OwnPortAltNum(自システムのポート番号使用数)

詳細説明 (process)

MQT 通信プロセスの共通情報が数値 (10 進数) または文字で表示されます。

ただし、次に示す項目は独自の形式で表示されます。

al....al

イニシエーションキュー監視方式

TIME：タイマ方式

EVENT：イベント方式

bl....bl

MQT トレースのディスク出力要否

NO : MQT トレースのディスク出力抑止中

YES : MQT トレースのディスク出力中

cl....cl

MQT トレースのバッファサイズ

4 の倍数に切り上げて補正した値が表示されます。

詳細説明 (mqcd)

チャンネルデータ定義ブロックに設定されている値が数値 (10 進数) または文字で表示されます。

ただし、次に示す項目は独自の形式で表示されます。

aa....aa

定義種別

MANINST :

MQT 通信構成定義で指定されたクラスタレシーバ以外のチャンネルです。

AUTOINST :

リポジトリ情報によって、自動的に作成されたクラスタセンダチャンネルです。

INSTMODL :

MQT 通信構成定義で指定されたクラスタレシーバチャンネルの属性です。

MULTINST :

INSTMODL によって作成されたクラスタレシーバチャンネルのインスタンスです。

bb....bb

チャンネルタイプ

DCMTCQ_MQCHT_SENDER : センダ

DCMTCQ_MQCHT_SERVER : サーバ

DCMTCQ_MQCHT_RECEIVER : レシーバ

DCMTCQ_MQCHT_REQUESTER : リクエスタ

DCMTCQ_MQCHT_CLUSSDR : クラスタセンダ

DCMTCQ_MQCHT_CLUSRCVR : クラスタレシーバ

cc....cc

バッファ方式

DCMTCQ_MQBUFT_SEG : セグメント方式

DCMTCQ_MQBUFT_MSG : メッセージ方式

dd....dd

チャンネル確立方式

DCMTCQ_MQSTART_AUTO :

トリガ起動またはネットワークリクエストによって、チャンネルを自動確立します。

DCMTCQ_MQSTART_AUTO2 :

MQT サーバのサービス開始直後にチャンネルを自動確立します。

DCMTCQ_MQSTRAT_MANUAL :

チャンネルを mqtstacha コマンド入力によって確立します。

ee....ee

長期確立時再試行の要否

DCMTCQ_MQLRTY_YES : 長期確立再試行を実行します。

DCMTCQ_MQLRTY_NO : 長期確立再試行を実行しません。

ff....ff

切断時間間隔使用の要否

DCMTCQ_DINT2EFC_YES :

切断時間間隔 (DiscInterval2) による監視をします。

DCMTCQ_DINT2EFC_NO :

切断時間間隔 (DiscInterval2) による監視をしません。

gg....gg

タイムアウト時チャンネル確立再試行の要否

DCMTCQ_MQTIMO_YES : チャンネル確立再試行を実行します。

DCMTCQ_MQTIMO_NO : チャンネル確立再試行を実行しません。

hh....hh

メッセージ編集出口ユーザデータ

32 バイトのユーザデータを 16 進数および文字で表示します。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.' を出力します)

ii....ii

TCP/IP 出力用バッファ長

0 : システムのデフォルトバッファ長を使用

0 以外 : TCP/IP 出力用バッファ長

jj....jj

TCP/IP 入力用バッファ長

0 : システムのデフォルトバッファ長を使用

0 以外 : TCP/IP 入力用バッファ長

kk....kk

非永続メッセージが送信される速度

DCMTCQ_MQNPMS_NORMAL：標準速度

DCMTCQ_MQNPMS_FAST：高速度

ll....ll

転送キュー監視方式

DCMTCQ_MQTRNQS_TIMER：タイマ方式

DCMTCQ_MQTRNQS_EVENT：イベント方式

mm....mm

チャンネル定義の更新日時

yyyy-mm-dd hh.mm.ss

年-月-日 時.分.秒

特定できない場合は”****-**-** **.**.**”が表示されます。

nn....nn

MQ プロトコル障害確立再試行要否

DCMTCQ_MQMCP_YES：確立再試行を実行します。

DCMTCQ_MQMCP_NO：確立再試行を実行しません。

oo....oo

強制再接続条件

DCMTCQ_MQADPCK_OFF

強制再接続を実行しません。

DCMTCQ_MQADPCK_NAME

チャンネル名が一致した時、強制再接続を実行します。

DCMTCQ_MQADPCK_QMGR

チャンネル名と相手キューマネージャ名が一致した時、強制再接続を実行します。

DCMTCQ_MQADPCK_IPADDR

チャンネル名と相手 IP アドレスが一致した時、強制再接続を実行します。

DCMTCQ_MQADPCK_ALL

チャンネル名、相手キューマネージャ名、および相手 IP アドレスが一致した時、強制再接続を実行します。

pp....pp

短期確立再試行の要否

DCMTCQ_MQSRTY_YES：短期確立再試行を実行します。

DCMTCQ_MQSRTY_NO：短期確立再試行を実行しません。

qq....qq

送信出口ユーザデータ

32 バイトのユーザデータが 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

rr....rr

受信出口ユーザデータ

32 バイトのユーザデータが 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

ss....ss

セキュリティ出口ユーザデータ

32 バイトのユーザデータが 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

tt....tt

送信出口名

128 バイトの送信出口名が 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

uu....uu

受信出口名

128 バイトの受信出口名が 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

vv....vv

メッセージ出口名

128 バイトのメッセージ出口名が 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

WW....WW

セキュリティ出口名

128 バイトのセキュリティ出口名が 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

XX....XX

注釈

64 バイトの注釈が 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

YY....YY

MCA ユーザ ID

64 バイトの MCA ユーザ ID が 16 進数、および文字で表示されます。

XX...XX CC...CC

XX...XX : 16 進数

CC...CC : 文字 (表示可能な ASCII 文字以外は, '.'を出力します)

ZZ....ZZ

MCA タイプ

DCMTCQ_MQMCATYPE_PROCESS : プロセス

DCMTCQ_MQMCATYPE_THREAD : スレッド

AA....AA

送信チャネルジャーナル出力条件

DCMTCQ_SNDJNL_FLUSH : 即時フラッシュ (ディスク出力) を行います。

DCMTCQ_SNDJNL_NOFLUSH : 即時フラッシュ (ディスク出力) を行いません。

注意事項

チャネルデータ定義ブロック (dcmtcq_uoc_mqcd) の出力形式は、すべてのチャネルタイプで共通です。そのため、チャネルタイプによっては有効でない属性の値も表示されます。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16361-I	状態表示を開始します。	標準出力
KFCA16362-I	状態表示を終了します。	標準出力

出力メッセージ ID	内容	出力先
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA31920-I	mqtddmped コマンドの入力形式の表示	標準出力
KFCA31921-E	ファイルの編集集中に入出力障害が発生しました。	標準エラー出力
KFCA31922-E	ファイルに対するアクセス権がありません。	標準エラー出力
KFCA31923-E	指定したファイルは存在しません。	標準エラー出力
KFCA31924-E	ファイルに異常が発生しました。	標準エラー出力

mqted (MQT トレースファイルの編集出力)

形式

```
mqted -i MQTトレースファイル名
      [{-k 出力情報種別 [, 出力情報種別] ... | -p}]
      [-t 開始日時 [, 終了日時] ]
      [-n チャンネル名 [, チャンネル名] ...]
      [-q 相手キューマネージャ名 [, 相手キューマネージャ名] ...]
      [-c クラスタ名 [, クラスタ名] ...]
```

機能

MQT トレースファイルを入力にして、編集結果を標準出力に出力します。

オプション

● -i MQT トレースファイル名

～<1～64バイトのパス名>

MQT トレースファイルのパス名を指定します。ファイル名だけを指定した場合、カレントディレクトリのMQT トレースファイルが編集されます。絶対パス名で指定した場合、指定したディレクトリのMQT トレースファイルが編集されます。

● -k 出力情報種別

～<文字列>

出力情報の種別を指定します。次に示す種別を一つ以上指定できます。

- mgr：接続先のキューマネージャ名を出力します。
- ip：接続先のIPアドレスを出力します。
- port：接続先のポート番号を出力します。
- clus：接続先のクラスタ名を出力します。
- msg：転送メッセージ情報を出力します。転送メッセージが先頭セグメントまたは単一セグメントの場合に出力します。

複数指定する場合は、種別と種別との間をコンマ (,) で区切ります。複数を指定する場合の指定順序はありませんが、同一種別は重複して指定できません。

なお、mgr、ip、port および clus を同時に指定した場合は、キューマネージャ名、IP アドレス、ポート番号およびクラスタ名の順序で出力されます。

● -p

トレース取得日時情報の出力を指定します。

-k オプションと同時に指定できません。-t, -n, -q または -c オプションと同時に指定した場合、条件に一致する MQT トレース情報のトレース取得日時情報が出力されます。

条件に一致する MQT トレース情報がない場合は、****/**/** **:**:*(*****)が出力されます。

● -t 開始日時 [終了日時]

～<文字列> 《00000001011970,23595912319999》

出力対象の日時を指定します。開始日時および終了日時は次に示す形式で指定します。

hhmmss [MMDD [YYYY]]

- hh：時 (00～23)
- mm：分 (00～59)
- ss：秒 (00～59)
- MM：月 (01～12)
- DD：日 (01～31)
- YYYY：年 (西暦 1970～9999)

開始日時と終了日時を指定する場合は、開始日時 ≤ 終了日時である必要があります。このオプションを省略した場合、または終了日時を省略した場合は、次に示すとおりを設定されます。

- 開始日時の省略値：00000001011970 (00 時 00 分 00 秒 01 月 01 日 1970 年)
- 終了日時の省略値：23595912319999 (23 時 59 分 59 秒 12 月 31 日 9999 年)

● -n チャネル名

～<1～20 文字の MQ 文字列> 《すべてのチャネル名》

出力対象となる、チャネルの名称を指定します。

チャネル名を指定しない場合は、すべてのチャネル名の情報を出力対象とします。

複数指定する場合は、チャネル名とチャネル名との間をコンマ (,) で区切ります。また、複数指定する場合の文字列は 256 文字以内 (コンマを含む) で指定してください。

● -q 相手キューマネージャ名

～<1～48 文字の MQ 文字列> 《すべての相手キューマネージャ名》

出力対象となる、チャネルの相手キューマネージャ名を指定します。

相手キューマネージャ名を指定しない場合は、すべての相手キューマネージャ名の情報を出力対象とします。

複数指定する場合は、相手キューマネージャ名と相手キューマネージャ名との間をコンマ (,) で区切ります。また、複数指定する場合の文字列は 256 文字以内 (コンマを含む) で指定してください。

● -c クラスタ名

～< 1～48 文字の MQ 文字列> 《すべてのクラスタ名》

出力対象となる、チャンネルのクラスタ名を指定します。

クラスタ名を指定しない場合は、すべてのクラスタ名の情報を出力対象とします。

複数指定する場合は、クラスタ名とクラスタ名との間をコンマ (,) で区切ります。また、複数指定する場合の文字列は 256 文字以内 (コンマを含む) で指定してください。

クラスタプロセスではない MQT プロセスが出力した MQT トレースファイルにこのオプションを指定した場合は共通情報だけが出力されます。

出力形式

• 共通情報

32ビットの場合

```
uCosminexus TP1/Message Queue                YYYY/MM/DD HH:MM:SS
      :
      (個別情報が出力されます)
      :
***** End Of File *****
```

64ビットの場合

```
uCosminexus TP1/Message Queue (64)           YYYY/MM/DD HH:MM:SS
      :
      (個別情報が出力されます)
      :
***** End Of File *****
```

• 個別情報 (-k オプションを省略した場合)

No	Date	Time	Channel-Name	Event
NNNNNN	yyyy/mm/dd	hh:mm:ss (mmmmmm)	aa...aa:bbb	cc...cc

• 個別情報 (-k に mgr を指定した場合)

No	Date	Time	Channel-Name	Event	Queue-Manager-Name
NNNNNN	yyyy/mm/dd	hh:mm:ss (mmmmmm)	aa...aa:bbb	cc...cc	dd...dd

• 個別情報 (-k に ip を指定した場合)

No	Date	Time	Channel-Name	Event	IP-Address
NNNNNN	yyyy/mm/dd	hh:mm:ss (mmmmmm)	aa...aa:bbb	cc...cc	ee...ee

• 個別情報 (-k に port を指定した場合)

No	Date	Time	Channel-Name	Event	Port-No
NNNNNN	yyyy/mm/dd	hh:mm:ss (mmmmmm)	aa...aa:bbb	cc...cc	ff...ff

• 個別情報 (-k に clus を指定した場合)

No	Date	Time	Channel-Name	Event	Cluster-Name
NNNNNN	yyyy/mm/dd	hh:mm:ss (mmmmmm)	aa...aa:bbb	cc...cc	gg...gg

• 個別情報 (-k に mgr,ip を指定した場合)

No	Date	Time	Channel-Name	Event	Queue-Manager-Name	IP-Address
NNNNNN	yyyy/mm/dd	hh:mm:ss (mmmmmm)	aa...aa:bbb	cc...cc	dd...dd	ee...ee

• 個別情報 (-k に msg を指定した場合)

No	Date	Time	Channel-Name	Event
NNNNNN	yyyy/mm/dd	hh:mm:ss (mmmmmm)	aa...aa:bbb	cc...cc
メッセージシーケンス番号			= a1...a1	
メッセージ総サイズ			= b1...b1	
セグメント番号			= c1...c1	
セグメントサイズ			= d1...d1	
リモートキュー名			= e1...e1	
リモートキューマネージャ名			= f1...f1	
構造体識別子			= g1...g1	
構造体バージョン番号			= h1...h1	
報告オプション			= [0x i1...i1]	
メッセージタイプ			= [0x j1...j1]	
メッセージ保持時間			= k1...k1	
報告メッセージ返答コード			= [0x l1...l1]	
マシンコード形式			= [0x m1...m1]	
文字セット識別子			= [0x n1...n1]	
フォーマット名			= o1...o1	
メッセージ優先度			= p1...p1	
メッセージ永続性			= q1...q1	
メッセージ識別子			= r1...r1	
			[0x s1...s1 s1...s1 s1...s1 s1...s1]	
			[0x s1...s1 s1...s1]	
関連識別子			= t1...t1	
			[0x u1...u1 u1...u1 u1...u1 u1...u1]	
			[0x u1...u1 u1...u1]	
ロールバック回数			= v1...v1	
応答キュー名			= w1...w1	
応答キューマネージャ名			= x1...x1	
ユーザ識別子			= y1...y1 [0x z1...z1 z1...z1 z1...z1]	
課金トークン			= a2...a2	
			[0x b2...b2 b2...b2 b2...b2 b2...b2]	
			[0x b2...b2 b2...b2 b2...b2 b2...b2]	
アプリケーション識別データ			= c2...c2	
			[0x d2...d2 d2...d2 d2...d2 d2...d2]	
			[0x d2...d2 d2...d2 d2...d2 d2...d2]	
登録アプリケーションタイプ			= [0x e2...e2]	
登録アプリケーション名			= f2...f2	
登録日付			= g2...g2	
登録時刻			= h2...h2	
登録元データ			= i2...i2	

• 個別情報 (-p を指定した場合)

Trace Start Time = j2...j2
Trace End Time = k2...k2

YYYY/MM/DD HH:MM:SS

編集出力の実行時刻 (年/月/日 時:分:秒)

NNNNNN

トレース通番 (6けたの10進数)

999999の次は000000に戻ります。

yyyy/mm/dd

トレースを取得した日付 (年月日)

hh:mm:ss(mmmmmmm)

トレースを取得した時刻 (時:分:秒(マイクロ秒))

Windows の場合はマイクロ秒ではなくミリ秒で出力されます。

aa...aa:bbb

チャンネル名 (1~20 文字の文字列) :チャンネルタイプ

snd : センダ

rcv : レシーバ

srv : サーバ

req : リクエスト

csn : クラスタセンダ

crc : クラスタレシーバ

CC...CC

イベント情報 (1~17 文字の文字列)

出力されるイベント情報と意味について、次の表に示します。

表 6-7 出力されるイベント情報と意味

イベント		意味
種類	情報	
チャンネルの確立・解放	Connect	チャンネル確立
	DisConnect	チャンネル解放
バッチ処理	Commit	バッチ処理成功
	Rollback	バッチ処理失敗
送信チャンネル	Recv(ERR)	送達確認 (失敗) 受信
	Recv(OK)	送達確認 (成功) 受信
	Send(C)	送達確認要求送信
	Send(F)	先頭セグメント送信
	Send(F+L)	単一セグメント送信
	Send(F+L+C)	単一セグメント + 送達確認要求送信
	Send(L)	最終セグメント送信
	Send(L+C)	最終セグメント + 送達確認要求送信
	Send(M)	中間セグメント送信
受信チャンネル	Recv(C)	送達確認要求受信
	Recv(F)	先頭セグメント受信
	Recv(F+L)	単一セグメント受信
	Recv(F+L+C)	単一セグメント + 送達確認要求受信
	Recv(L)	最終セグメント受信

イベント		意味
種類	情報	
受信チャンネル	Recv(L+C)	最終セグメント+送達確認要求受信
	Recv(M)	中間セグメント受信
	Send(ERR)	送達確認（失敗）送信
	Send(OK)	送達確認（成功）送信

dd...dd

相手キューマネージャ名（1～48文字のMQ文字列）
表示する情報がない場合はピリオド（.）が48文字出力されます。

ee...ee

相手システムのIPアドレス（1～15文字の文字列）

ff...ff

相手システムのポート番号（1～5けたの10進数）

gg...gg

クラスタ名（1～48文字のMQ文字列）
表示する情報がない場合はピリオド（.）が48文字出力されます。

al...al

メッセージシーケンス番号（1～9けたの16進数）

bl...bl

メッセージ総サイズ（1～8けたの10進数）
プロトコルヘッダ、および転送キューヘッダ（MQXQH構造体）の長さは含まれません。

cl...cl

セグメント番号（1～9けたの16進数）

dl...dl

セグメントサイズ（1～8けたの10進数）
プロトコルヘッダの長さは含まれません。転送キューヘッダ（MQXQH構造体）の長さは含まれます。

el...el

リモートキュー名（1～48文字の文字列）
配布リストメッセージを送受信する場合は、何も出力しません。

fl...fl

リモートキューマネージャ名（1～48文字の文字列）
配布リストメッセージを送受信する場合は、何も出力しません。

gl...gl

構造体識別子（文字列）

hl...hl
構造体バージョン番号 (10 進数)

il...il
報告オプション (16 進数)

jl...jl
メッセージタイプ (16 進数)

kl...kl
メッセージ保持時間 (10 進数)

ll...ll
報告メッセージ返答コード (16 進数)

ml...ml
マシンコード形式 (16 進数)

nl...nl
文字セット識別子 (16 進数)

ol...ol
フォーマット名 (文字列)

pl...pl
メッセージ優先度 (10 進数)

ql...ql
メッセージ永続性
MQPER_PERSISTENT : 永続メッセージ
MQPER_NOT_PERSISTENT : 非永続メッセージ

rl...rl
メッセージ識別子 (文字列)

sl...sl
メッセージ識別子 (16 進数)

tl...tl
相関識別子 (文字列)

ul...ul
相関識別子 (16 進数)

vl...vl
ロールバック回数 (10 進数)

wl...wl
応答キュー名 (文字列)

x1...x1

応答キューマネージャ名 (文字列)

y1...y1

ユーザ識別子 (文字列)

z1...z1

ユーザ識別子 (16 進数)

a2...a2

課金トークン (文字列)

b2...b2

課金トークン (16 進数)

c2...c2

アプリケーション識別データ (文字列)

d2...d2

アプリケーション識別データ (16 進数)

e2...e2

登録アプリケーションタイプ (16 進数)

f2...f2

登録アプリケーション名 (文字列)

g2...g2

登録日付 (文字列)

h2...h2

登録時刻 (文字列)

i2...i2

登録元データ (文字列)

j2...j2

最初に取得した MQTT トレースのトレース取得日時情報
yyyy/mm/dd hh:mm:ss(mmmmmm)の形式で出力されます。

k2...k2

最後に取得した MQTT トレースのトレース取得日時情報
yyyy/mm/dd hh:mm:ss(mmmmmm)の形式で出力されます。

- 出力例

センダチャネルの確立後に一つのメッセージが送信され、1分後にセンダチャネルを解放した場合の出力例 (-k mgr を指定した場合) を次に示します。

これは製品が提供するサンプルコードにあるホスト 1 の CHA01 チャンネルの動作に相当します。サンプルコードのシステム構成については、4 章の「システム構成例」を参照してください。

```

uCosminexus TP1/Message Queue                               2006/10/15 15:15:11
No      Date      Time      Channel-Name  Event      Queue-Manager-Name
000001  2006/10/07 20:44:05 (774161) CHA01:snd   Connect    qmgr2
000002  2006/10/07 20:44:05 (776437) CHA01:snd   Send (F+L) qmgr2
000003  2006/10/07 20:44:05 (792538) CHA01:snd   Send (C)   qmgr2
000004  2006/10/07 20:44:05 (849525) CHA01:snd   Recv (OK)  qmgr2
000005  2006/10/07 20:44:05 (862444) CHA01:snd   Commit     qmgr2
000006  2006/10/07 20:45:05 (798495) CHA01:snd   DisConnect qmgr2
***** End Of File *****

```

注意事項

バージョンおよび OS が異なる TP1/Message Queue によって出力された MQT トレースファイルを指定した場合の動作は保証されません。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA18383-E	MQT トレース編集に必要なローカルメモリが確保できません。	標準エラー出力
KFCA18393-E	MQT トレースファイルの入出力中に障害が発生しました。	標準エラー出力
KFCA31949-I	mqtcd コマンドの入力形式の表示	標準出力

mqtlscha (チャンネルの状態表示)

形式

```
mqtlscha [-s MQT通信プロセス識別子]
          {-n チャンネル名 | -q 相手キューマネージャ名
           | -n チャンネル名 -q 相手キューマネージャ名 | -D }
```

機能

チャンネルの状態を標準出力に出力します。

オプション

● -s MQT 通信プロセス識別子

～< 16 進数字> ((01 ~ ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して mqtlscha コマンドが実行されます。このとき、MQT サーバが開始された順序で情報が表示されます。

● -n チャンネル名

～< 1～20 文字の MQ 文字列>

状態を表示するチャンネルの名称を指定します。

複数のチャンネル名を指定する場合は、引用符 (") で囲んで、チャンネル名とチャンネル名との間を空白で区切ります。同一チャンネル名は、重複して指定できません。

また、チャンネル名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャンネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャンネルの状態を標準出力に出力します。

先行文字列*：先行文字列で始まるチャンネルの状態を標準出力に出力します。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャンネルを指定する場合

```
-n "cha*"
```

● -q 相手キューマネージャ名

～< 1～48 文字の MQ 文字列>

状態を表示するチャンネルの相手キューマネージャ名を指定します。

複数の相手キューマネージャ名を指定する場合は、引用符 (") で囲んで、相手キューマネージャ名と相手キューマネージャ名との間を空白で区切ります (コマンド入力可能な、けた数を指定できます)。同一相手キューマネージャ名は、重複して指定できません。

また、相手キューマネージャ名は、*を使って一括指定ができます。一括指定は、一つだけ指定できます。一括指定と一括指定以外の相手キューマネージャ名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャンネルの状態を標準出力に出力します。

先行文字列*：先行文字列で始まる相手キューマネージャ名が一致するチャンネルを開始します。

<複数指定の例>

相手キューマネージャ名 1, 相手キューマネージャ名 2, および相手キューマネージャ名 3 を指定する場合

```
-q "相手キューマネージャ名1△相手キューマネージャ名2△相手キューマネージャ名3"
```

<一括指定の例>

QM1, QM2, および QM3 を一括して指定する場合

```
-q "QM*"
```

● -D

MQT 通信プロセスのスナップダンプファイルを取得します。

このオプションは、mqtdmped コマンドの入力情報および MQT 通信プロセスの保守情報を採取する場合に使用します。

スナップダンプファイル名：\$DCDIR/spool/mqtdmpXX

(XX：MQT 通信プロセス識別子)

このオプションは、通信プロセスで動作している MCA の動作を一時的に中断します。そのため、MCA が送受信中の場合には相手システムが通信の監視タイマにタイムアウトを検出することがありますので注意してください。

出力形式

mqtlscha コマンドを実行した結果、出力される情報を次に示します。

KFCA16361-I CHAの状態表示を開始します。
 KFCA16379-I mmm aa...aa:ttt チャンネル状態=cc....cc 転送キュー名=qq....qq
 KFCA31931-I 定義種別=dd....dd
 KFCA16380-I シーケンス番号最大値=rr....rr 最大メッセージ長=ss....ss
 バッチサイズ=bb....bb
 KFCA16383-I 仕掛かりシーケンス番号=tt....tt 完了シーケンス番号=nn....nn
 LUWID=ll....ll メッセージ送達未確認=iii
 KFCA16389-I プロトコルレベル=u メッセージ編集出口=vvv 配付リスト=www
 非永続メッセージ転送速度=xx....xx ハートビート=yyy(zz....zz)
 KFCA31930-I 相手キューマネージャ名=mm....mm
 KFCA16388-I 相手システム=(oo....oo, pp....pp)
 KFCA31936-I 自システム=(00....00, PP....PP)
 KFCA31932-I 経路再設定中=eee
 KFCA16381-I 開始時刻 YYYY/MM/DD-hh:mm:ss 転送メッセージ数=ee....ee
 KFCA31935-I ZZZZ/NN/EE-ii:nn:tt以降の転送メッセージ総数=TT....TT
 KFCA31933-I 短期確立再試行回数の残数=SS....SS 長期確立再試行回数の残数=LL....LL
 KFCA31934-I クラスタ名=CC....CC
 KFCA16362-I CHAの状態表示を終了します。

mmm

MQT 識別子

aa...aa

チャンネル名

ttt

チャンネルタイプ

snd : センダ

rcv : レシーバ

srv : サーバ

req : リクエスト

csn : クラスタセンダ

crc : クラスタレシーバ

CC....CC

チャンネル状態

STOPPED:チャンネル停止です。

STARTING : MCA 起動中です。

BINDING : チャンネル確立中です。

RUNNING : チャンネル動作中です。

STOPPING : チャンネル解放中です。

RETRYING : チャンネル確立リトライ中です。

DISABLED : チャンネル使用不可です。

REQUESTING : チャンネル確立リクエスト中です。

qq....qq

転送キュー名

チャンネルタイプが snd, srv, または csu の場合は転送キュー名

チャンネルタイプが rcv, req, または crc の場合は"****"

dd....dd

定義種別

MANINST :

MQT 通信構成定義で指定されたクラスタレシーバ以外のチャンネルです。

AUTOINST :

リポジトリ情報によって、自動的に作成されたクラスタセンダチャンネルです。

INSTMODL :

MQT 通信構成定義で指定されたクラスタレシーバチャンネルの属性です。

MULTINST :

INSTMODL によって作成されたクラスタレシーバチャンネルのインスタンスです。

rr....rr

メッセージシーケンス番号の最大値 (チャンネル定義の値)

ss....ss

最大メッセージ長

アプリケーションのメッセージの最大長です。MQXQH 構造体の長さは含みません。

- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル確立リトライ中」または「チャンネル使用不可」の場合
チャンネル定義の値が表示されます。
- チャンネル状態が「チャンネル確立中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル確立リクエスト中」の場合
相手チャンネルとネゴシエーションした値が表示されます。

bb....bb

バッチサイズ

バッチサイズは、MCP で規定された 1 回のメッセージ転送単位で転送するメッセージの個数です。

- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル確立リトライ中」または「チャンネル使用不可」の場合
チャンネル定義の値が表示されます。
- チャンネル状態が「チャンネル確立中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル確立リクエスト中」の場合
相手チャンネルとネゴシエーションした値が表示されます。

tt...tt

最終仕掛かりメッセージのメッセージシーケンス番号

処理中メッセージのメッセージシーケンス番号が表示されます。

チャンネル情報を引き継がないで開始し、バッチ処理を始めるまでの間は、"0"が表示されます。バッチ転送中は1ずつ増加します。mqtrstcha コマンドを実行すると、リセットした値が表示されます（ただし、バッチ転送中の場合、およびメッセージの送達が無確認の場合は、リセットした値は表示されません）。

- チャンネルタイプが snd, srv, または csn の場合
バッチ処理中に障害が発生すると、メッセージ送達無確認が"NO"のときはバッチ処理開始前の値に戻ります。メッセージ送達無確認が"YES"のときは戻りません。
- チャンネルタイプが rcv, req, または crc の場合
バッチ処理中に障害が発生すると、バッチ処理開始前の値に戻ります。

nn....nn

最終完了メッセージのメッセージシーケンス番号

処理が完了したメッセージのメッセージシーケンス番号が表示されます。

チャンネル情報を引き継がないで開始し、バッチ処理を始めるまでの間は、"0"が表示されます。バッチ処理中は、1ずつ増加します。

- チャンネルタイプが snd, srv, または csn の場合
バッチ処理中に障害が発生すると、メッセージ送達無確認が"NO"のときはバッチ処理開始前の値に戻ります。メッセージ送達無確認が"YES"のときは戻りません。
- チャンネルタイプが rcv, req, または crc の場合
バッチ処理中に障害が発生すると、バッチ処理開始前の値に戻ります。

ll...ll

LUWID (16進数)

LUWID は、MCP で規定された1回のメッセージ転送単位を管理するためのIDです。

チャンネル情報を引き継がないで開始し、バッチ処理を始めるまでの間は、"00...00"が表示されます。

- チャンネルタイプが snd, srv, または csn の場合
メッセージ送達無確認が"YES"のときは、最終仕掛かりバッチのLUWIDが表示されます。また、メッセージ送達無確認が"NO"のときは、最終完了バッチのLUWIDが表示されます。
- チャンネルタイプが rcv, req, または crc の場合
最終完了バッチのLUWIDが表示されます。

iii

メッセージの送達無確認

YES：未確認です。

NO：確認済みです。

チャンネルタイプが rcv, req の場合は常に"NO"が表示されます。

U

プロトコルレベル

チャンネルで使用するプロトコルレベル

- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル確立リトライ中」または「チャンネル使用不可」の場合
"4"が表示されます。
- チャンネル状態が「チャンネル確立中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル確立リクエスト中」の場合
相手チャンネルとネゴシエーションした値が表示されます。

VVV

メッセージ編集出口呼び出しの有無

メッセージ編集出口 UOC を組み込んで MQTT サーバを作成し、該当チャンネル定義のバッファ方式にメッセージ方式 (msg) を指定した場合、またはチャンネルタイプが csn または crc の場合 "ON" が表示されます。バッファ方式にセグメント方式 (seg) を指定した場合 "OFF" が表示されます。

ON: 出口呼び出しがあります。

OFF: 出口呼び出しがありません。

WWW

配布リスト付きメッセージ転送機能の有無

ON: 配布リスト転送があります。

OFF: 配布リスト転送がありません。

- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル確立リトライ中」または「チャンネル使用不可」の場合
チャンネル定義のバッファ方式にメッセージ方式 (msg) を指定した場合は "ON" になります。セグメント方式 (seg) を指定した場合は "OFF" になります。
- チャンネル状態が「チャンネル確立中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル確立リクエスト中」の場合
相手チャンネルとネゴシエーションした結果が表示されます。プロトコルレベルが 1 の場合は、"OFF" になります。

XX....XX

非永続メッセージ転送速度

NORMAL: 標準速度

FAST: 高速度

- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル確立リトライ中」または「チャンネル使用不可」の場合
チャンネル定義の値が表示されます。
- チャンネル状態が「チャンネル確立中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル確立リクエスト中」の場合

相手チャンネルとネゴシエーションした結果が表示されます。プロトコルレベルが 1 の場合は、"NORMAL"になります。

yyy

ハートビートメッセージ転送機能の有無

ON：ハートビート転送があります。

OFF：ハートビート転送がありません。

- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル確立リトライ中」または「チャンネル使用不可」の場合
チャンネル定義のハートビート間隔が 0 の場合は"OFF", 以外の場合は"ON"になります。
- チャンネル状態が「チャンネル確立中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル確立リクエスト中」の場合
相手チャンネルとネゴシエーションした結果の値が表示されます。プロトコルレベルが 1 の場合は、"OFF"になります。

zz....zz

ハートビート間隔

ハートビート転送機能が"OFF"の場合、"0"が表示されます。

- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル確立リトライ中」または「チャンネル使用不可」の場合
チャンネル定義の値が表示されます。
- チャンネル状態が「チャンネル確立中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル確立リクエスト中」の場合
相手チャンネルとネゴシエーションした結果の値が表示されます。プロトコルレベルが 1 の場合は、"0"になります。

mm....mm

相手システムのキューマネージャ名

不明な場合、またはプロトコルレベル 1 で接続中の場合は"****"が表示されます。

oo....oo

相手システムのホスト名、または IP アドレス

ホスト名は 32 バイトまで表示されます。ホスト名が 33 バイト以上の場合、"ホスト名..."の形式で表示されます。

pp....pp

相手システムのサービス名、またはポート番号

KFCA16388-I メッセージは、相手システムが定義されていない場合、出力されません。KFCA16388-I の出力条件については、次の表を参照してください。

表 6-8 KFCA16388-I の出力条件

チャンネルタイプ	チャンネル状態							
	チャンネル停止	MCA 起動中	チャンネル確立中	チャンネル動作中	チャンネル解放中	チャンネル確立リトライ中	チャンネル使用不可	チャンネル確立リクエスト中
センダ	△	△	○	○	○	△	△	—
レシーバ	×	×	○	○	○	×	×	—
サーバ	△	△	○	○	○	△	△	—
リクエスタ	△	△	○	○	○	△	△	○
クラスタセンダ	△	△	○	○	○	△	△	—
クラスタレシーバ	×	×	○	○	○	×	×	—

(凡例)

○：接続した相手システムの IP アドレスとポート番号をログ出力します。

△：システム定義または mqaltcha コマンドで変更した、相手システムのホスト名または IP アドレスと、サービス名またはポート番号をログ出力します。サーバまたはリクエスタで値がない場合（定義の指定がないまたは mqaltcha コマンドで値が設定されていない）は、ログ出力しません。

×：ログ出力しません。

—：あり得ない状態なのでログ出力しません。

OO....OO

自システムの IP アドレス

PP....PP

自システムのポート番号

KFCA31936-I メッセージは、チャンネルがコーラ側として動作し、かつチャンネル状態が「チャンネル動作中」「チャンネル解放中」の場合に出力されます。

eee

経路再設定状態

YES：メッセージ送信経路の再設定処理を実行しています。

NO：メッセージ送信経路の再設定処理は実行していません。

KFCA31932-I メッセージは、チャンネルタイプが csnn の場合だけ出力されます。

YYYY/MM/DD-hh:mm:ss

年（西暦）/月/日-時:分:秒（MCA 開始時刻）

MCA 開始時刻が出力されます。MCA は運用コマンド、トリガ起動、ネットワークリクエスト、またはシステム起動によって開始されます。

MCA の開始についての詳細は、「2.3.5 チャンネルの開始」を参照してください。なお、確立再試行は MCA の開始に含まれません。そのため、確立再試行によってチャンネルが確立しても MCA 開始時刻は更新されません。

- チャンネル状態が「チャンネル停止」、または「チャンネル使用不可」の場合表示されません。
- チャンネル状態が「MCA 起動中」、「チャンネル確立中」、「チャンネル動作中」、「チャンネル解放中」、「チャンネル確立リトライ中」、または「チャンネル確立リクエスト中」の場合表示されます。

ee....ee

転送完了したメッセージ数

- チャンネル状態が「チャンネル停止」、または「チャンネル使用不可」の場合表示されません。
- チャンネル状態が「MCA 起動中」、「チャンネル確立中」、「チャンネル動作中」、「チャンネル解放中」、「チャンネル確立リトライ中」、または「チャンネル確立リクエスト中」の場合
チャンネル確立後、転送が完了したメッセージ数が出力されます。チャンネルを解放すると、0に戻ります。

ZZZZ/NN/EE-ii:nn:tt

年（西暦）/月/日-時:分:秒（1 回目の MCA 開始時刻）

MQT サービス開始後の 1 回目の MCA 開始時刻が出力されます。

MCA を開始していない場合は、MQT サービス開始時刻が出力されます。

TT....TT

転送メッセージ総数

2147483647 を超えた場合は 1 に戻ります。

MQT サービス開始後にバッチ転送が完了したメッセージ数が出力されます。次に示す状態の場合、あて先キューのメッセージ数と転送メッセージ総数は一致しません。

- 転送メッセージがファーストメッセージの場合で、あて先キューへの登録を失敗した場合
- あて先数が 2 以上の配布リストメッセージを送受信する場合

SS....SS

短期確立再試行回数の残数

- チャンネル状態が「チャンネル確立中」、「チャンネル確立リトライ中」、または「チャンネル確立リクエスト中」の場合
短期確立再試行回数の残数を表示します。
- チャンネル状態が「チャンネル停止」、「MCA 起動中」、「チャンネル動作中」、「チャンネル解放中」、または「チャンネル使用不可」の場合
チャンネル定義の値を表示します。

mqtalccha 定義コマンドの-b bretry オペランドに no を指定している場合は、常に 0 が表示されます。

LL....LL

長期確立再試行回数の残数

- チャンネル状態が「チャンネル確立中」, 「チャンネル確立リトライ中」, または「チャンネル確立リクエスト中」の場合
長期確立再試行回数の残数を表示します。
- チャンネル状態が「チャンネル停止」, 「MCA 起動中」, 「チャンネル動作中」, 「チャンネル解放中」, または「チャンネル使用不可」の場合
チャンネル定義の値を表示します。

mqtalccha 定義コマンドの -b bretry オペランドに no を指定している場合は、常に 0 が表示されます。KFCA31933-I メッセージは、チャンネルタイプが snd, srv, req, csn の場合だけ出力されます。

CC...CC

クラスタ名

KFCA31934-I メッセージは、チャンネルタイプが csn または crc の場合だけ出力されます。

接続先のキューマネージャで、ネームリスト中にクラスタ名を定義している場合は、そのネームリストで定義したクラスタ名のどれか一つが表示されます。参加しているクラスタ名を表示するには、mqrls コマンドを使用してください。

注意事項

1. クラスタレシーバプロセスに、チャンネルインスタンスがない状態では、MQT 通信構成定義に指定した属性だけを表示します。
2. クラスタレシーバプロセスに、複数のチャンネルインスタンスがある場合、-n オプションを指定するとすべてのインスタンスに対して有効になります。チャンネルインスタンスを特定する場合、-q オプションを指定してください。
3. クラスタレシーバチャンネル以外のチャンネルは、-q オプションには、"*"だけ指定できます。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16358-I	mqtlscha コマンドの入力形式の表示	標準出力
KFCA16361-I	状態表示を開始します。	標準出力
KFCA16362-I	状態表示を終了します。	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力

出力メッセージID	内容	出力先
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16379-I	チャンネル状態の表示	標準出力
KFCA16380-I	チャンネル状態の表示	標準出力
KFCA16381-I	チャンネル状態の表示	標準出力
KFCA16383-I	チャンネル状態の表示	標準出力
KFCA16388-I	チャンネル状態の表示	標準出力
KFCA16389-I	チャンネル状態の表示	標準出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力
KFCA31930-I	チャンネル状態の表示	標準出力
KFCA31931-I	チャンネル状態の表示	標準出力
KFCA31932-I	チャンネル状態の表示	標準出力
KFCA31933-I	チャンネル状態の表示	標準出力
KFCA31934-I	チャンネル状態の表示	標準出力
KFCA31935-I	チャンネル状態の表示	標準出力
KFCA31936-I	チャンネル状態の表示	標準出力

mqtpngcha (チャネルのテスト接続)

形式

```
mqtpngcha [-s MQT通信プロセス識別子] -n チャネル名  
          [-d データ長]
```

機能

チャネルを開始し、最大セグメントサイズ以内のデータ送受信ができるかどうかを確認します。

このコマンドはチャネルタイプがセンダ、サーバ（コーラの場合）、またはクラスタセンダの場合だけ有効です。

また、このコマンドはチャネル状態が「チャネル停止」、 「チャネル使用不可」、または「チャネル確立リトライ中」であれば受け付けられますが、それ以外の場合は受け付けられません。

オプション

● -s MQT 通信プロセス識別子

～< 16 進数字 > ((01 ~ ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtpngcha コマンドを実行します。

● -n チャネル名

～< 1～20 文字の MQ 文字列 >

開始できるかどうかを確認したいチャネルの名称を指定します。

複数のチャネル名を指定する場合は、引用符 (") で囲んで、チャネル名とチャネル名との間を空白で区切ります。同一チャネル名は、重複して指定できません。

また、チャネル名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャネルが開始できるかを確認します。

先行文字列*：先行文字列で始まるチャネルが開始できるかを確認します。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャンネルを指定する場合

```
-n "cha*"
```

● -d データ長

～<符号なし整数>((16~32768))《16》(単位：バイト)

送信データ (Ping データ) 長を指定します。

送信データ長は複数指定できません。

送信データ長は、「ネゴシエーション後の最大セグメントサイズ-28」以下の値を指定してください。「ネゴシエーション後の最大セグメントサイズ-28」よりも大きい値が指定された場合は、「ネゴシエーション後の最大セグメントサイズ-28」が指定されたと仮定して処理します。

注意事項

クラスタセンダチャンネルは、該当するチャンネルに割り当てられたメッセージを現状の最適なチャンネルへ割り当て直す機能 (メッセージ送信経路の再設定) を持っています。機能が動作している間は、このコマンドを入力できません。

出力メッセージ

出力メッセージID	内容	出力先
KFCA16355-I	コマンドを正常に受け付けました。	標準出力
KFCA16360-I	mqtpngcha コマンドの入力形式の表示	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力

mqtrlvcha (メッセージ送達未確認の解決)

形式

```
mqtrlvcha [-s MQT通信プロセス識別子] -n チャンネル名  
          -a {commit | backout}
```

機能

送信側 MCA がメッセージの送達を確認できないで終了した場合、メッセージ送達未確認状態が発生します。この状態で何らかの理由によってチャンネルの再開始ができない場合に未確認状態を解決し、チャンネルを確立できるようにします。

このコマンドはチャンネルタイプがセンダ、サーバ、またはクラスタセンダの場合だけ有効です。また、チャンネル状態が「チャンネル停止」または「チャンネル使用不可」で、メッセージの送達が未確認の場合に受け付けられません。

オプション

● -s MQT 通信プロセス識別子

~< 16 進数字 > ((01 ~ ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtrlvcha コマンドを実行します。

● -n チャンネル名

~< 1~20 文字の MQ 文字列 >

メッセージの未確認状態を解決するチャンネルのチャンネル名を指定します。

複数のチャンネル名を指定する場合は、引用符 (") で囲んで、チャンネル名とチャンネル名との間を空白で区切ります。同一チャンネル名は、重複して指定できません。

また、チャンネル名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャンネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*:すべてのチャンネルが開始できるかを確認します。

先行文字列*:先行文字列で始まるチャンネルが開始できるかを確認します。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャンネルを指定する場合

```
-n "cha*"
```

● -a commit | backout

メッセージの送達未確認状態の解決方法を指定します。

- commit：メッセージをコミット（転送キューから削除）します。
- backout：メッセージをロールバック（転送キューに戻す）します。

このオプションの指定がチャンネル再確立時のチャンネルの解決方法と異なる場合、メッセージを二重送信したり、失ったりすることがあります。

注意事項

1. クラスタセンダチャンネルは、該当するチャンネルに割り当てられたメッセージを現状の最適なチャンネルへ割り当て直す機能（メッセージ送信経路の再設定）を持っています。機能が動作している間は、このコマンドを入力できません。
2. 送信チャンネルジャーナル出力条件（mqtalcca 定義コマンド-jn sndjnl オペランド）に noflush を指定した場合、メッセージ送達未確認状態のメッセージが受信側チャンネルで受信を完了している状態で -a オプションに backout を指定すると不正なメッセージが送信されることがあります。必ず受信側チャンネルの状態を確認してからこのコマンドを使用してください。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16355-I	コマンドを正常に受け付けました。	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16384-I	mqtrlvcha コマンドの入力形式の表示	標準出力
KFCA16385-E	コミットを失敗しました。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力

mqtrstcha (メッセージシーケンス番号のリセット)

形式

```
mqtrstcha [-s MQT通信プロセス識別子]
           {-n チャンネル名 | -q 相手キューマネージャ名
            | -n チャンネル名 -q 相手キューマネージャ名 }
           [-m メッセージシーケンス番号]
```

機能

メッセージシーケンス番号をリセットします。

センダ、サーバ、またはクラスタセンダに mqtrstcha コマンドが入力された場合は、コマンドを入力した側の MCA が持つメッセージシーケンス番号をリセットし、さらに、通信相手側のレシーバ、リクエスト、およびクラスタレシーバのメッセージシーケンス番号も同じ値にリセットします。

レシーバ、リクエスト、またはクラスタレシーバに mqtrstcha コマンドが入力された場合は、コマンドを入力した側の MCA が持つメッセージシーケンス番号だけをリセットします。

オプション

● -s MQT 通信プロセス識別子

～< 16 進数字> ((01～ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtrstcha コマンドを実行します。

● -n チャンネル名

～< 1～20 文字の MQ 文字列>

メッセージシーケンス番号をリセットしたいチャンネルの名称を指定します。

複数のチャンネル名を指定する場合は、引用符 (") で囲んで、チャンネル名とチャンネル名との間を空白で区切ります。同一チャンネル名は、重複して指定できません。

また、チャンネル名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャンネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャンネルのメッセージシーケンス番号をリセットします。

先行文字列*：先行文字列で始まるチャンネルのメッセージシーケンス番号をリセットします。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャンネルを指定する場合

```
-n "cha*"
```

● **-q 相手キューマネージャ名**

~<1~48 文字の MQ 文字列>

メッセージシーケンス番号をリセットするチャンネルの相手キューマネージャ名を指定します。

複数の相手キューマネージャ名を指定する場合は、引用符 (") で囲んで、相手キューマネージャ名と相手キューマネージャ名との間を空白で区切ります (コマンド入力可能な、けた数を指定できます)。同一相手キューマネージャ名は、重複して指定できません。

また、相手キューマネージャ名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外の相手キューマネージャ名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*:すべてのチャンネルのメッセージシーケンス番号をリセットします。

先行文字列*:先行文字列で始まる相手キューマネージャ名が一致するチャンネルのメッセージシーケンス番号をリセットします。

<複数指定の例>

相手キューマネージャ名 1, 相手キューマネージャ名 2, および相手キューマネージャ名 3 を指定する場合

```
-q "相手キューマネージャ名1△相手キューマネージャ名2△相手キューマネージャ名3"
```

<一括指定の例>

QM1, QM2, および QM3 を一括して指定する場合

```
-q "QM*"
```

● **-m メッセージシーケンス番号**

~<符号なし整数> ((1~999999999)) 《1》

リセットしたい値を指定します。

メッセージシーケンス番号は複数指定できません。

この値は、TCP 定義の mqtalcccha 定義コマンドの-w オプションで指定したメッセージシーケンス番号の最大値より小さい値を指定してください。

1 を指定、または省略した場合、メッセージシーケンス番号は 1 にリセットされます。

注意事項

1. クラスタレシーバプロセスに、チャンネルインスタンスがない状態では、このコマンドは使用できません。
2. クラスタレシーバプロセスに複数のチャンネルインスタンスが存在する場合、`-n` オプションを指定するとすべてのインスタンスに対して有効になります。チャンネルインスタンスを特定する場合は、`-q` オプションを使用してください。
3. クラスタレシーバチャンネル以外のチャンネルは、`-q` オプションに "*" だけ指定できます。
4. リセットしたシーケンス番号は、チャンネル確立後初めて行うメッセージ送受信で使用します。そのため、チャンネル確立後メッセージ送受信を行ったあとでリセットしたシーケンス番号は、一度チャンネルを解放し、再度チャンネルを確立したあとのメッセージ送受信で使用します。

シーケンス番号をリセットして、チャンネルを確立するまでの間は、`mqtlscha` コマンドの仕掛かりシーケンス番号にはリセットしたシーケンス番号が表示されます。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16355-I	コマンドを正常に受け付けました。	標準出力
KFCA16359-I	<code>mqrstcha</code> コマンドの入力形式の表示	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16382-I	メッセージシーケンス番号が不当です。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力

mqtstacha (チャンネルの開始)

形式

```
mqtstacha [-s MQT通信プロセス識別子]
           {-n チャンネル名 | -q 相手キューマネージャ名
            | -n チャンネル名 -q 相手キューマネージャ名 }
           [-a]
```

機能

チャンネルを開始します。

レシーバチャンネル、またはクラスタレシーバチャンネルに対して入力された場合、チャンネルは開始しませんが、メッセージチャンネルエージェントは起動可能になります。

オプション

● -s MQT 通信プロセス識別子

～< 16 進数字> ((01～ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtstacha コマンドを実行します。

● -n チャンネル名

～< 1～20 文字の MQ 文字列>

開始するチャンネルのチャンネル名を指定します。

複数のチャンネル名を指定する場合は、引用符 (") で囲んで、チャンネル名とチャンネル名との間を空白で区切ります。同一チャンネル名は、重複して指定できません。

また、チャンネル名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャンネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャンネルを開始します。

先行文字列*：先行文字列で始まるチャンネルを開始します。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャンネルを指定する場合

```
-n "cha*"
```

● -q 相手キューマネージャ名

~<1~48文字のMQ文字列>

開始するチャンネルの相手キューマネージャ名を指定します。

複数の相手キューマネージャ名を指定する場合は、引用符 (") で囲んで、相手キューマネージャ名と相手キューマネージャ名との間を空白で区切ります (コマンド入力可能な、けた数を指定できます)。同一相手キューマネージャ名は、重複して指定できません。

また、相手キューマネージャ名は、*を使って一括指定ができます。一括指定は、一つだけ指定できます。一括指定と一括指定以外の相手キューマネージャ名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

* : すべてのチャンネルを開始します。

先行文字列* : 先行文字列で始まる相手キューマネージャ名が一致するチャンネルを開始します。

<複数指定の例>

相手キューマネージャ名 1, 相手キューマネージャ名 2, および相手キューマネージャ名 3 を指定する場合

```
-q "相手キューマネージャ名1△相手キューマネージャ名2△相手キューマネージャ名3"
```

<一括指定の例>

QM1, QM2, および QM3 を一括して指定する場合

```
-q "QM*"
```

● -a

チャンネル開始要求を送信しないで、状態だけを「チャンネル使用不可」から「チャンネル停止」にします。相手システムが指定されたチャンネル (TCP 定義の mqtalcccha 定義コマンドの -o オプションの指定あり) の場合、チャンネル開始要求を送信しません。省略した場合はチャンネルを開始 (チャンネル開始要求の送信) します。

相手システムを指定していない場合は、このオプションの指定は意味を持ちません。このオプションはチャンネル状態が「チャンネル使用不可」の場合だけ指定できます。

注意事項

1. クラスタレシーバプロセスに、チャンネルインスタンスがない状態では、このコマンドは使用できません。

2. クラスタレシーバプロセスに複数のチャンネルインスタンスが存在する場合、-n オプションを指定するとすべてのインスタンスに対して有効になります。チャンネルインスタンスを特定する場合は、-q オプションを使用してください。
3. クラスタレシーバチャンネル以外のチャンネルは、-q オプションに"*"だけ指定できます。
4. クラスタセングダチャンネルは、該当するチャンネルに割り当てられたメッセージを現状の最適なチャンネルへ割り当て直す機能（メッセージ送信経路の再設定）を持っています。機能が動作している間は、このコマンドを入力できません。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16355-I	運用コマンドを正常に受け付けました。	標準出力
KFCA16356-I	mqtstacha コマンドの入力形式の表示	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました（コマンド）。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました（通信）。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました（通信）。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました（コマンド）。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力

mqtstatrc (MQT トレースのディスク出力機能の開始)

形式

```
mqtstatrc [-s MQT通信プロセス識別子]
```

機能

オンライン中に MQT トレースのディスク出力機能を開始します。MQT トレースについては、「[7.4.1 MQT トレースファイルの出力](#)」を参照してください。

このコマンドは、MQT トレースのディスク出力機能を使用していない次のどちらかの場合に、MQT トレースのディスク出力を開始できます。

- mqttcp, mqttcpes または mqttcpesr 定義コマンドの -t disk オペランドに no を指定して OpenTP1 を開始した場合
- mqtstptrc コマンドを入力して MQT トレースのディスク出力機能を終了している場合

オプション

● -s MQT 通信プロセス識別子

~ < 16 進数字 > ((01 ~ ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtstatrc コマンドを実行します。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16355-I	コマンドを正常に受け付けました。	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力
KFCA31947-I	mqtstatrc コマンドの入力形式の表示	標準出力

mqtstpcha (チャネルの終了)

形式

```
mqtstpcha [-s MQT通信プロセス識別子]
           {-n チャネル名 | -q 相手キューマネージャ名
           | -n チャネル名 -q 相手キューマネージャ名 }
           [-f]
```

機能

チャネルを終了します。

オプション

● -s MQT 通信プロセス識別子

～< 16 進数字> ((01～ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtstpcha コマンドを実行します。

● -n チャネル名

～< 1～20 文字の MQ 文字列>

終了するチャネルの名称を指定します。

複数のチャネル名を指定する場合は、引用符 (") で囲んで、チャネル名とチャネル名との間を空白で区切ります。同一チャネル名は、重複して指定できません。

また、チャネル名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のチャネル名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャネルを終了します。

先行文字列*：先行文字列で始まるチャネルを終了します。

<複数指定の例>

cha1, cha2, および cha3 を指定する場合

```
-n "cha1 cha2 cha3"
```

<一括指定の例>

cha で始まるすべてのチャネルを指定する場合

```
-n "cha*"
```

● -q 相手キューマネージャ名

～<1～48 文字の MQ 文字列>

終了するチャンネルの相手キューマネージャ名を指定します。

複数の相手キューマネージャ名を指定する場合は、引用符 (") で囲んで、相手キューマネージャ名と相手キューマネージャ名との間を空白で区切ります (コマンド入力可能な、けた数を指定できます)。同一相手キューマネージャ名は、重複して指定できません。

また、相手キューマネージャ名は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外の相手キューマネージャ名を混在して指定できません。一括名は引用符 (") で囲んで指定します。

*：すべてのチャンネルを終了します。

先行文字列*：先行文字列で始まる相手キューマネージャ名が一致するチャンネルを終了します。

<複数指定の例>

相手キューマネージャ名 1, 相手キューマネージャ名 2, および相手キューマネージャ名 3 を指定する場合

```
-q "相手キューマネージャ名1△相手キューマネージャ名2△相手キューマネージャ名3"
```

<一括指定の例>

QM1, QM2, および QM3 を一括して指定する場合

```
-q "QM*"
```

● -f

該当するチャンネルを強制的に終了します。省略した場合は、処理中のメッセージの処理が完了後、終了します。

注意事項

1. クラスタレシーバプロセスに、チャンネルインスタンスがない状態では、このコマンドは使用できません。
2. クラスタレシーバプロセスに複数のチャンネルインスタンスが存在する場合、-n オプションを指定するとすべてのインスタンスに対して有効になります。チャンネルインスタンスを特定する場合は、-q オプションを使用してください。
3. クラスタレシーバチャンネル以外のチャンネルは、-q オプションに"*"だけ指定できます。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16355-I	運用コマンドを正常に受け付けました。	標準出力

出力メッセージID	内容	出力先
KFCA16357-I	mqtstpcha コマンドの入力形式の表示	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16366-E	該当するチャンネルがありません。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力

mqtstptrc (MQT トレースのディスク出力機能の終了)

形式

```
mqtstptrc [-s MQT通信プロセス識別子]
```

機能

オンライン中に MQT トレースのディスク出力機能を終了します。

このコマンドは、MQT トレースのディスク出力機能を使用している次のどちらかの場合に、MQT トレースのディスク出力を終了できます。

- mqttcp, mqttcpes または mqttcpesr 定義コマンドの -t disk オペランドに yes を指定して OpenTPI を開始した場合
- mqtstatrc コマンドを入力して MQT トレースのディスク出力機能を開始している場合

オプション

● -s MQT 通信プロセス識別子

~< 16 進数字 > ((01 ~ ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると、すべての MQT サーバに対して、mqtstptrc コマンドを実行します。

出力メッセージ

出力メッセージID	内容	出力先
KFCA16355-I	運用コマンドを正常に受け付けました。	標準出力
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16365-E	運用コマンドを無視しました。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力
KFCA31948-I	mqtstptrc コマンドの入力形式の表示	標準出力

mqtswptrc (MQT トレースの強制スワップ)

形式

```
mqtswptrc [-s MQT通信プロセス識別子]
```

機能

MQT トレースを強制的に MQT トレースファイルにスワップします。

TCP 定義でトレースファイルのディスク出力機能を使用することを指定 (TCP 定義の mqttcp 定義コマンド, mqttcps 定義コマンド, または mqttcpcr 定義コマンドの -t disk オペランドに yes を指定) した場合, このコマンドを入力すると, メモリ中の MQT トレースを MQT トレースファイルに出力します。ディスク出力機能を使用しないことを指定 (TCP 定義の mqttcp 定義コマンド, mqttcps 定義コマンド, または mqttcpcr 定義コマンドの -t disk オペランドに no を指定) していても, このコマンドを入力するとメモリ中の MQT トレースを MQT トレースファイルに出力します。

オプション

● -s MQT 通信プロセス識別子

~< 16 進数字 > ((01 ~ ff))

MQT 通信プロセス識別子を指定します。

MQT 通信プロセス識別子は複数指定できません。

このオプションを省略すると, すべての MQT サーバに対して, mqtswptrc コマンドを実行します。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA16363-W	入力形式に誤りがあります。	標準エラー出力
KFCA16364-W	引数の指定に誤りがあります。	標準エラー出力
KFCA16367-E	MQT 開始処理中です。	標準エラー出力
KFCA16368-E	MQT 終了処理中です。	標準エラー出力
KFCA16369-E	コマンド処理中に障害が発生しました。	標準エラー出力
KFCA16370-I	運用コマンドが終了しました。	標準出力
KFCA16371-E	MQT 通信プロセスは起動されていません。	標準エラー出力
KFCA16372-E	コマンド応答の連絡の受信に失敗しました。	標準エラー出力
KFCA16373-E	コマンド応答の連絡に失敗しました。	標準エラー出力
KFCA16374-E	プロセスの検索に失敗しました。	標準エラー出力

出力メッセージID	内容	出力先
KFCA16375-E	RPC 障害が発生しました (コマンド)。	標準エラー出力
KFCA16376-E	RPC 障害が発生しました (通信)。	標準エラー出力
KFCA16377-E	ネットワーク障害が発生しました (通信)。	標準エラー出力
KFCA16378-E	ネットワーク障害が発生しました (コマンド)。	標準エラー出力
KFCA16390-E	ローカルメモリ不足が発生しました。	標準エラー出力
KFCA16391-E	メモリ不足が発生しました。	標準エラー出力
KFCA16392-E	論理矛盾が発生しました。	標準エラー出力
KFCA16393-E	障害が発生しました。	標準エラー出力
KFCA16394-E	障害が発生しました。処理を続行します。	標準エラー出力
KFCA16395-E	共用メモリ不足が発生しました。	標準エラー出力
KFCA18370-I	MQT トレースファイルへの出力を開始しました。	メッセージログファイル
KFCA18371-I	MQT トレースファイルへの出力を終了しました。	メッセージログファイル
KFCA31946-I	mqtswptrc コマンドの入力形式の表示	標準出力

7

障害対策

この章では、TP1/Message Queue の運用中に、ファイル障害または通信障害が発生した場合の、TP1/Message Queue の対応処理およびユーザの対応処理、障害時に取得する情報について説明します。

7.1 ファイル障害

ファイルについての障害が発生した場合の対策について説明します。

7.1.1 キューおよびキューファイルの障害とユーザの処理

キューおよびキューファイルの障害に対するユーザの処理について説明します。

MQPUT 命令で相手システムがメッセージを送信する場合を例にして、キューおよびキューファイルの障害発生個所について、次の図に示します。また、キューおよびキューファイルの障害とユーザの処理について、図のあとの表に示します。図中の障害の発生個所を示す番号は、表の発生個所と対応づけてあります。

図 7-1 メッセージ送信中の障害発生個所

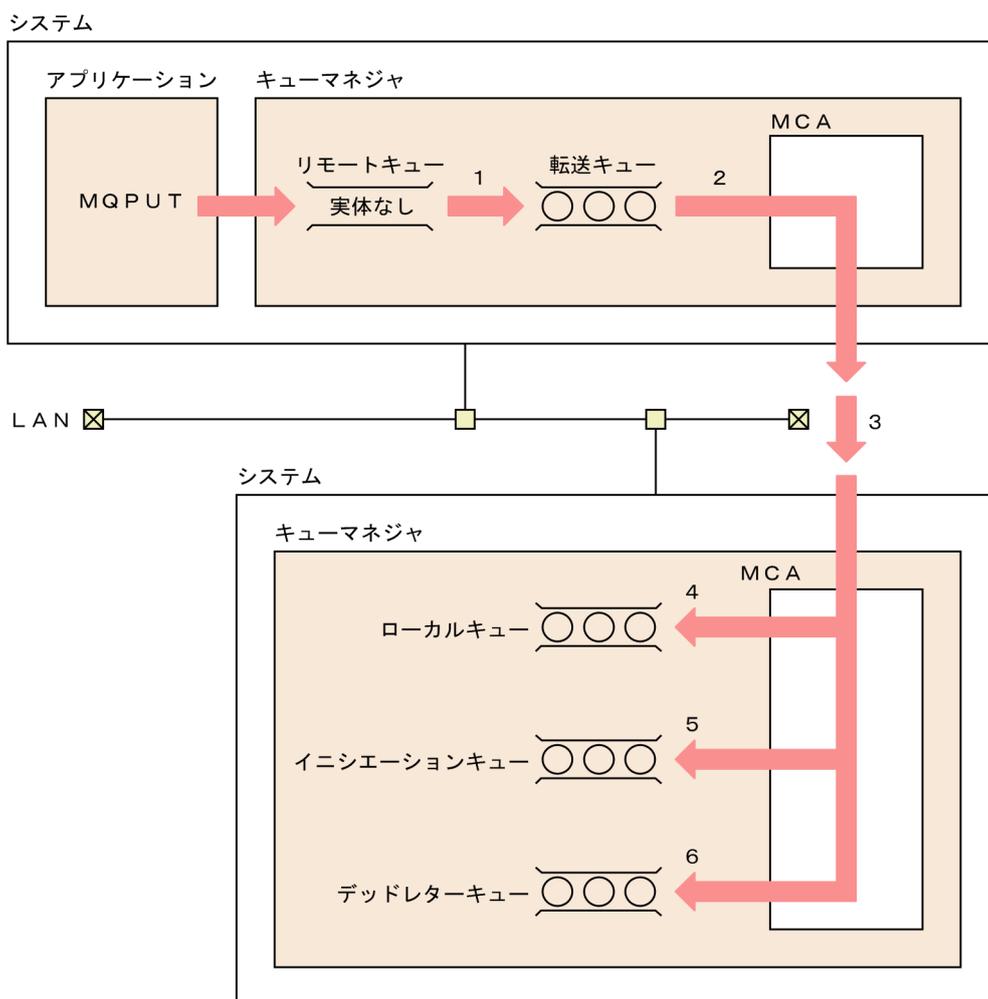


表 7-1 キューおよびキューファイルの障害とユーザの処理

発生箇所	障害の種類	障害内容・原因	MQAの障害処理	ユーザの処理
1	転送キュー障害	転送キューの属性不正（登録禁止）	MQPUT または MQPUT1 エラーリターン	MQSET 命令または mqaset コマンドで登録許可に変更後、アプリケーションを再実行してください。
		転送キューの属性不正（使用方法が転送キュー (MQUS_TRANSMISSION) でない）		転送キューを再作成後、アプリケーションを再起動してください。 転送キューの再作成方法 <ul style="list-style-type: none"> • TP1/Message Queue を終了してください。mqaqueatl 定義コマンドを使用して転送キューを再作成したあと、または mqachgque コマンドで該当するキューの使用方法を転送キュー (MQUS_TRANSMISSION) に変更したあと、TP1/Message Queue を正常開始してください。 • 動的キューの場合、キューを削除 (MQCLOSE 命令) してください。TP1/Message Queue を終了し、モデルキューの定義を修正後、動的キューを作成 (MQOPEN 命令) してください。
		入力用バッファ不足（メモリ不足） < 26188-E > 「入出力バッファが満杯」		MQA サービス定義で該当キューファイルグループの入出力バッファ数を大きくして、TP1/Message Queue を正常開始してください。
2	転送キュー障害	転送ヘッダなし、転送ヘッダ不正（転送キューに対し MQPUT 命令を呼び出し） < 16334-E > 「メッセージ異常検出」	送信側のデッドレターキューに登録、ログヘメッセージ出力	リモートキューのローカル定義をオープンし MQPUT 命令を呼び出すようアプリケーションを修正し、再実行してください。
2 ～ 5	キューファイル障害	入出力エラー 送信側：< 04200-E > 「I/O エラー」 受信側：< 04200-E > 「I/O エラー」	キューファイルを障害閉塞状態とし、アプリケーションからのアクセスはエラー 受信側にデッドレターキューあり <ul style="list-style-type: none"> • デッドレターキューに登録 受信側にデッドレターキューなし <ul style="list-style-type: none"> • 非永続メッセージは損失 	「7.1.4 キューファイルの入出力エラー」を参照してください。

発生箇所	障害の種類	障害内容・原因	MQAの障害処理	ユーザの処理
2 ～ 5	キューファイル障害	入出力エラー 送信側：< 04200-E > 「I/O エラー」 受信側：< 04200-E > 「I/O エラー」	<ul style="list-style-type: none"> 永続メッセージは、送信側の転送キューに戻る 	「7.1.4 キューファイルの入出力エラー」を参照してください。
3	転送キュー障害	転送キューオーバフロー	MQPUT エラーリターン、通信路を回復すると、メッセージを自動送信	回線またはチャネル状態を確認して回復してください。アプリケーションを再実行してください。
4	受信キュー障害	受信キューの属性不正（登録禁止） 送信側：< 16335-W > 「デッドレターキューに登録」 受信側：< 16333-E > 「キューファイル障害発生」	チャネル停止後送信側のデッドレターキューに登録、ログヘメッセージ出力	MQSET 命令または mqaset コマンドで登録許可に変更後、アプリケーションを再実行してください。
		入力用バッファ不足（メモリ不足） 受信側：< 26188-E > 「入出力バッファが満杯」	受信側のデッドレターキューに登録、ログヘメッセージ出力	MQA サービス定義で該当キューファイルグループの入出力バッファ数を大きくして、TP1/Message Queue を正常開始してください。
		受信キュー満杯 <ul style="list-style-type: none"> キューの深さの最大値が小さい キューファイル作成時（mqainit コマンド）のメッセージ数が小さい 受信側：< 16333-E > 「キューファイル障害発生」	受信側のデッドレターキューに登録、ログヘメッセージ出力	受信キューの最大メッセージ数を増やしてください。または mqainit コマンドのメッセージ数を増やしてください。
		受信キューなし <ul style="list-style-type: none"> 受信キュー未作成 送信側の受信キュー指定 < 16333-E > 「キューファイル障害発生」		受信キューを再作成後、アプリケーションを再起動してください。
5	イニシエーションキュー障害	イニシエーションキューの属性不正（使用方法が正常（MQUS_NORMAL）でない） 受信側：< 04229-E > 「イニシエーションキューの使用種別」	トリガメッセージを生成しないでログヘメッセージ出力	イニシエーションキューを再作成後、アプリケーションを再起動してください。
		イニシエーションキューの属性不正（キュータイプがローカルキュー（MQQT_LOCAL）でない）		イニシエーションキューの再作成方法 <ul style="list-style-type: none"> TP1/Message Queue を終了してください。MQA サービス定義の mqaqueueatll 定義コマンドを使用してイ

発生箇所	障害の種類	障害内容・原因	MQAの障害処理	ユーザの処理
5	イニシエーションキュー障害	イニシエーションキューの属性不正（キュータイプがローカルキュー（MQQT_LOCAL）でない）	トリガメッセージを生成しないでログヘメッセージ出力	<p>ニシエーションキューを再作成したあと、または mqachgque コマンドで該当するキューの使用方法をローカルキュー（MQQT_LOCAL）に変更したあと、TP1/Message Queue を正常開始してください。</p> <ul style="list-style-type: none"> 動的キューの場合、キューを削除（MQCLOSE 命令）してください。 <p>TP1/Message Queue を終了し、モデルキューの定義を修正後、動的キューを作成（MQOPEN 命令）してください。</p>
		イニシエーションキューの属性不正（登録禁止）	トリガメッセージを生成しない	MQSET 命令または mqaset コマンドで登録許可に変更後、アプリケーションを再実行してください。
		イニシエーションキューを指しているローカルキューの属性不正（プロセス名の指定なし） 受信側：< 04228-E > 「プロセス定義がない」	トリガメッセージを生成しないでログヘメッセージ出力	イニシエーションキュー、ローカルキュー、プロセス定義のどれかを再作成後、アプリケーションを再起動してください。
		イニシエーションキューを指しているローカルキューの属性不正（プロセス名の指定はあるがプロセス定義なし） 受信側：< 04228-E > 「プロセス定義がない」		<p>イニシエーションキューおよびローカルキューの再作成方法</p> <p>「イニシエーションキューの再作成方法」を参照してください。</p> <p>プロセス定義の再作成方法</p> <p>TP1/Message Queue を終了し、MQA サービス定義を修正後、TP1/Message Queue を正常開始してください。</p>
		イニシエーションキュー満杯 <ul style="list-style-type: none"> キューの深さの最大値が小さい キューファイル作成時（mqainit コマンド）のメッセージ数が小さい 受信側：< 04241-E > 「トリガメッセージの登録に失敗」	イニシエーションキューが存在するキューマネージャに定義されているデッドレターキューにトリガメッセージを登録し、ログヘメッセージ出力	イニシエーションキューの最大メッセージ数を増やしてください。または mqainit コマンドのメッセージ数を増やしてください。
		イニシエーションキューなし <ul style="list-style-type: none"> イニシエーションキュー未作成 送信側のイニシエーションキュー名誤り 	トリガメッセージを生成しないでログヘメッセージ出力	イニシエーションキューを作成後、アプリケーションを再起動してください。または TP1/Message Queue を終了してイニシエーションキュー名を修正（mqachgque コマンド、モデルキュー定義修正）後、

発生箇所	障害の種類	障害内容・原因	MQAの障害処理	ユーザの処理
5	イニシエーションキュー障害	送信側：< 04227-E > 「イニシエーションキューがない」	トリガメッセージを生成しないでログヘメッセージ出力	TP1/Message Queue を正常開始してください。
6	デッドレターキュー障害	デッドレターキュー満杯 <ul style="list-style-type: none"> キューの深さの最大値が小さい キューファイル作成時 (mqainit コマンド) のメッセージ数が小さい 送信側：< 16345-E > 「エラーデータ受信」 受信側：< 16333-E > 「キューファイル障害発生」	ログヘメッセージ出力	デッドレターキューの最大メッセージ数を増やしてください。または mqainit コマンドのメッセージ数を増やしてください。
		デッドレターキューなし <ul style="list-style-type: none"> デッドレターキュー未作成 受信側：< 16333-E > 「キューファイル障害発生」 送信側のデッドレターキュー名誤り 送信側：< 16333-E > 「キューファイル障害発生」 		デッドレターキューを作成後、アプリケーションを再起動してください。または TP1/Message Queue を終了し、デッドレターキュー名を修正 (キューマネージャ定義修正) 後、TP1/Message Queue を正常開始してください。
		デッドレターキューの属性不正 (登録禁止) 送信側：< 16345-E > 「エラーデータ受信」 受信側：< 16333-E > 「キューファイル障害発生」		MQSET 命令または mqaset コマンドで登録許可に変更後、アプリケーションを再実行してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

7.1.2 メッセージ送受信中の障害

(1) メッセージ送信中の障害

メッセージが送信できなくなる事象が発生した場合、MQT サーバは、該当するメッセージを送信側のデッドレターキューに格納します。また、MQPUT 命令で指定した応答キューに、発生した事象の内容を報告メッセージとして登録します。

(2) メッセージ受信中の障害

MQT サーバがメッセージを受信し、送信側が指定したあて先キューへの登録が失敗した場合、該当するメッセージを受信側のデッドレターキューに格納します。また、送信側で指定した応答キューに発生した事象の内容を報告メッセージとして登録します。

7.1.3 イニシエーションキューの障害

イニシエーションキューへのトリガメッセージ登録に失敗した場合、デッドレターキューが定義されているときは、トリガメッセージはデッドレターキューに登録されます。デッドレターキューが定義されていないときは、トリガメッセージは生成されません。次に示す場合はエラーメッセージが出力されます。

1. イニシエーションキューがない。
2. 受信したキューにプロセス名が定義されていない、または該当するプロセス名のプロセス定義がない。
3. イニシエーションキューの使用種別 (Usage 属性) が MQUS_NORMAL でない。
4. イニシエーションキューに登録するキューファイルで入出力エラーが発生した。

メッセージ登録の延長でトリガメッセージの生成が失敗しても、アプリケーションはメッセージの登録に成功します。また、MCA はチャンネルを解放しないで処理を続行します。

7.1.4 キューファイルの入出力エラー

キューファイルに入出力エラーが発生した場合 (入出力エラーのメッセージが出力された場合、または MQPUT 命令もしくは MQGET 命令の理由コードが MQRC_OBJECT_DAMAGED の場合)、障害要因を調査してキューファイル格納媒体損傷による障害かそれ以外による障害か判断し、それぞれに応じて対処します。

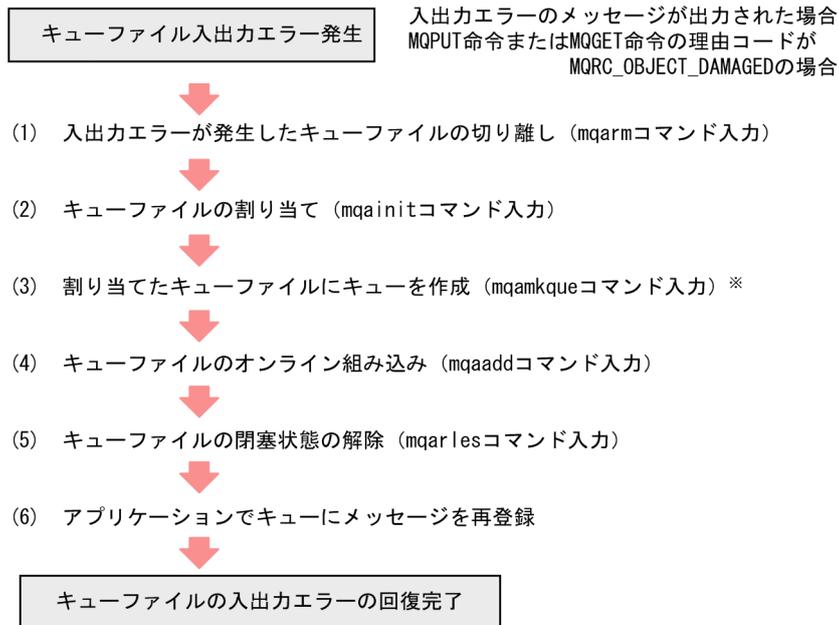
(1) キューファイル格納媒体損傷による障害の場合

キューファイル格納媒体損傷による障害の場合の対処について説明します。

(a) キューファイルの入出力エラーの対処

キューファイルに入出力エラーが発生した場合の対処について、次の図に示します。

図 7-2 キューファイルの入出力エラーの対処

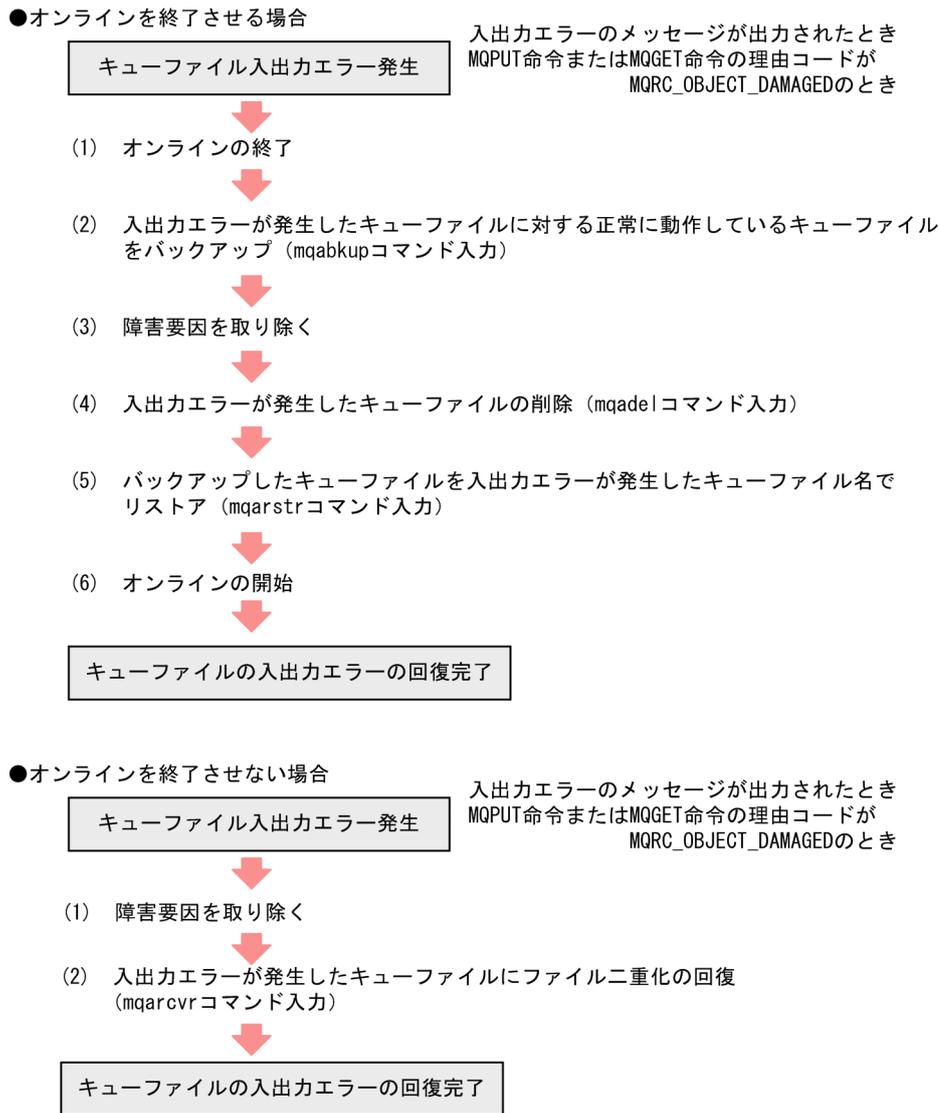


注※ (3)は、動的キューの場合は不要です。また、入出力エラーが発生したキューファイルにキューが登録されている場合だけ実行します。

(b) 二重化構成しているキューファイルの入出力エラーの対処

二重化構成しているキューファイルで片系のキューファイルに入出力エラーが発生した場合の対処について、次の図に示します。

図 7-3 二重化構成しているキューファイルの入出力エラーの対処



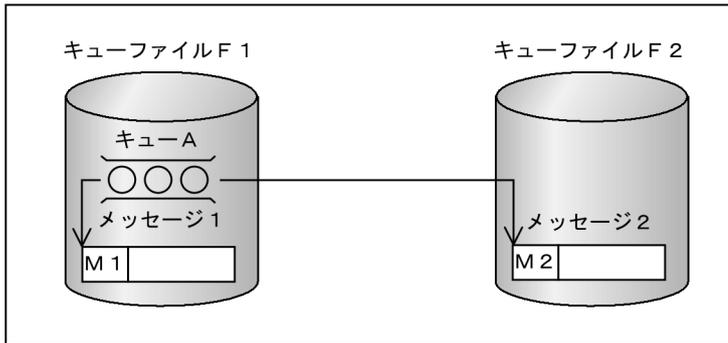
OpenTP1 を正常開始する場合、前回のオンラインで二重化構成しているキューファイルの入出力エラーが発生したときは、該当するキューファイルの回復を必ず実行してください。キューファイルの回復を実行しないで OpenTP1 を正常開始した場合、キューファイルの内容は保証されません。

(c) キューファイル内のキューおよびメッセージの扱い

障害が発生したキューファイルに、キューおよびメッセージが登録されている場合があります。ファイルグループの構成例とその場合の対処について、次の図に示します。

図 7-4 キューファイル内のキューおよびメッセージの扱い

キューファイルグループ



(凡例)

M 1, M 2 : メッセージ識別子

<キューファイル F 1 が入出力エラーとなった場合>

- (1) キューファイル F 1 に対してmqarmコマンドを入力します。
- (2) TP1/Message Queueは、キューA、メッセージ1およびメッセージ2を削除します。
- (3) 次に示すログメッセージが出力されます。
「キューAに格納されているメッセージ識別子M 1のメッセージを削除しました。」
「キューAを削除しました。」 (メッセージ2の削除)

<キューファイル F 2 が入出力エラーとなった場合>

- (1) キューファイル F 2 に対してmqarmコマンドを入力します。
- (2) TP1/Message Queueは、メッセージ2を削除します。
- (3) 次に示すログメッセージが出力されます。
「キューAに格納されているメッセージ識別子M 2のメッセージを削除しました。」
このとき、キューAはアクセス可能で、メッセージ1は残ります。メッセージ1とメッセージ2に関連がない場合は運用を継続できます。

<キューファイルを切り離れたあとの処理>

- ・メッセージ識別子でメッセージが識別できる場合は、ログメッセージに出力されたメッセージ識別子のメッセージだけは再登録できます。
- ・メッセージ識別子でメッセージが識別できない場合は、アプリケーションからのキューAに対するメッセージの登録を抑制したあとメッセージ1を削除します。メッセージ1およびメッセージ2を再登録する必要があります。

(d) キューファイルの入出力エラーの対処 (MQA FRC 使用時)

MQA FRC 使用時のキューファイルに入出力エラーが発生した場合の対処について、次の図に示します。キューファイルのバックアップ方法、リストア方法、およびmqafrcコマンドの実行方法については、「2.2.2(12) mqafrc コマンドの使用例 (キューファイル格納媒体損傷の場合)」を参照してください。

図 7-5 キューファイルの入出力エラーの対処 (MQA FRC 使用時)

●オンラインを終了させる場合

(1) キューファイルのバックアップ取得



(2) オンラインの開始



キューファイル入出力エラー発生

入出力エラーのメッセージが出力されたとき
MQPUT命令またはMQGET命令の理由コードが
MQRC_OBJECT_DAMAGEDのとき



(3) オンラインの終了 (OpenTP1は計画停止または強制停止)



(4) 障害要因を取り除く



(5) 障害が発生したキューファイルのバックアップからリストア
(バックアップ時にメッセージが登録されていないキューは再作成可能)



(6) アンロードジャーナルファイルを使用してキューファイルを回復
(mqafrcコマンド入力)



(7) オンラインの開始 (OpenTP1の再開始)



(8) キューファイルの閉塞状態の解除 (mqarlesコマンド入力)



キューファイルの入出力エラーの回復完了

●オンラインを終了させない場合

(1) キューファイルのバックアップ取得



(2) オンラインの開始



キューファイル入出力エラー発生

入出力エラーのメッセージが出力されたとき
MQPUT命令またはMQGET命令の理由コードが
MQRC_OBJECT_DAMAGEDのとき



(3) 障害が発生したキューファイルへのアクセス中止
(アプリケーションおよびチャネルを終了)



(4) アンロードジャーナルファイルの用意
キューファイルの回復に必要なジャーナルがあるファイルグループをjnlswpfgコマンドに-j sysを指定して実行してスワップし、待機状態にします。そのあと、jnlunlfgコマンドに-j sysを指定して実行し、アンロード済み状態にします。



(5) 障害が発生したキューファイルにキューが作成されている場合は、障害が発生したキューファイルを含むキューファイルグループ内のすべてのキューファイルの状態を障害閉塞に変更 (mqagrprobsコマンド入力) のあと、キューファイルを含むキューファイルグループ内のすべてのキューファイルの切り離し (mqarmコマンド入力)
障害が発生したキューファイルにキューが作成されていない場合は、キューファイルの切り離し (mqarmコマンド入力)



(6) 障害要因を取り除く



(7) 次に示すどちらかからリストア
(バックアップ時にメッセージが登録されていないキューは再作成可能)
・障害が発生したキューファイル (キューあり) を含むキューファイルグループ内のすべてのキューファイルのバックアップ
・障害が発生したキューファイル (キューなし) のバックアップ



(8) アンロードジャーナルファイルを使用してキューファイルを回復 (mqafrcコマンド入力)



(9) キューファイルのオンライン組み込み (mqaaddコマンド入力)



(10) キューファイルの閉塞状態の解除 (mqarlesコマンド入力)

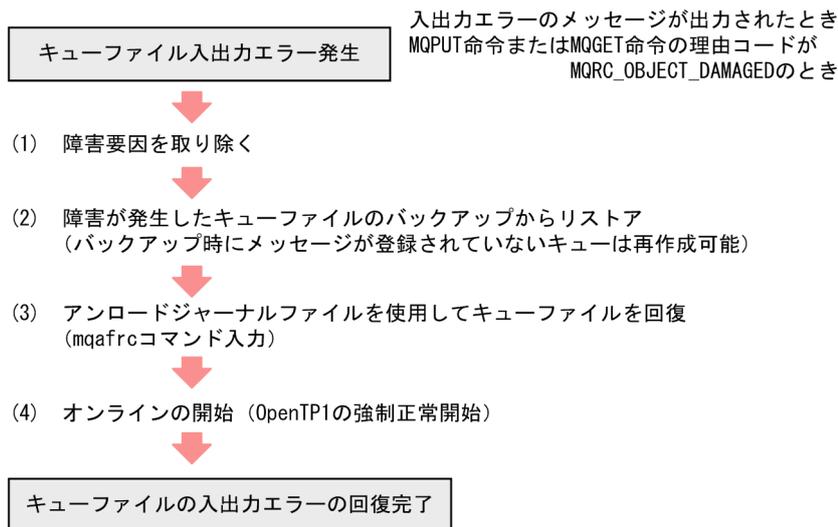


(11) アプリケーションおよびチャネルの開始



キューファイルの入出力エラーの回復完了

●OpenTP1が異常終了したあと、再度開始できない場合



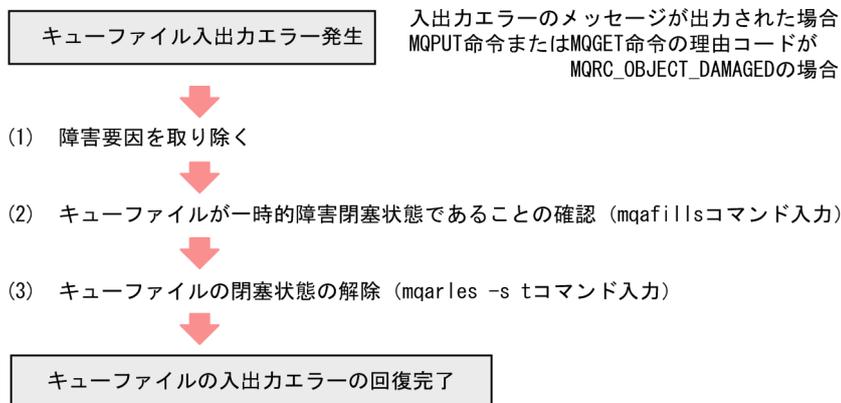
(2) キューファイル格納媒体損傷以外による障害の場合

キューファイル格納媒体損傷以外による障害の場合の対処について説明します。

(a) キューファイルの入出力エラーの対処

キューファイルに入出力エラーが発生した場合の対処について、次の図に示します。

図 7-6 キューファイルの入出力エラーの対処



注 KFCA31187-Eメッセージに“物理ファイル=cnt”が出力されたあと、mqarles -s t コマンドによって一時的障害閉塞状態を解除する前にオンラインが終了（計画停止、強制停止、または異常終了）した場合、mqafrcコマンドでの復旧作業を行う必要があります。詳細については、「(c) キューファイルの入出力エラーの対処 (MQA FRC使用時)」を参照してください。

(b) 二重化構成しているキューファイルの入出力エラーの対処

二重化構成しているキューファイルの場合、両系が正常のときにファイル入出力エラーが発生すると、エラーが発生した系のキューファイルは障害閉塞になります。

また、片系障害の状態でもう一方の系のキューファイルにファイル入出力エラーが発生すると、その系のキューファイルは一時的障害閉塞になります。

二重化構成しているキューファイルの入出力エラーについて、次の表に示します。

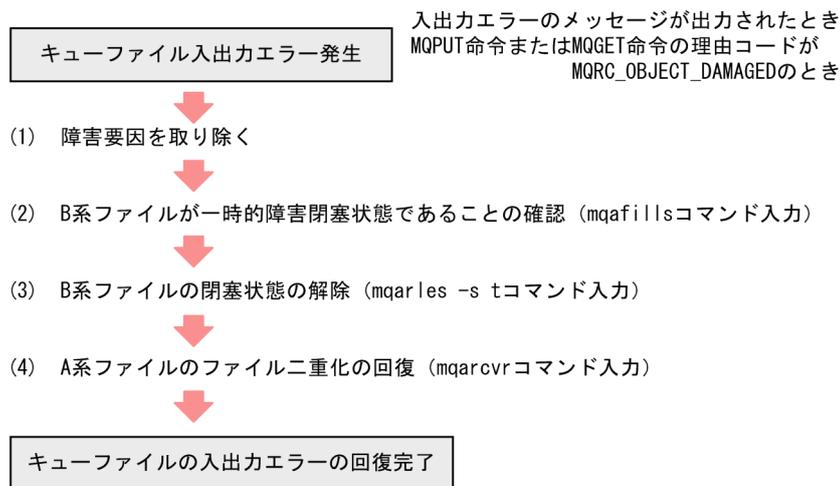
表 7-2 二重化構成しているキューファイルの入出力エラー

障害内容	A系ファイルの状態	B系ファイルの状態
A系ファイルで障害発生	障害閉塞	正常
B系ファイルで障害発生	正常	障害閉塞
A系ファイル障害状態でB系ファイルで障害発生	障害閉塞	一時的障害閉塞
B系ファイル障害状態でA系ファイルで障害発生	一時的障害閉塞	障害閉塞

二重化構成しているキューファイルでA系ファイルが障害閉塞状態になったあと、B系ファイルに入出力エラーが発生した場合の対処について、次の図に示します。

なお、二重化構成しているキューファイルでA系ファイルに入出力エラーが発生した場合の対処については、「(1)(b) 二重化構成しているキューファイルの入出力エラーの対処」を参照してください。

図 7-7 二重化構成しているキューファイルの入出力エラーの対処



注 KFCA31187-Eメッセージに“物理ファイル=cnt”が出力されたあと、mqarles -s t コマンドによって一時的障害閉塞状態を解除する前にオンラインが終了（計画停止、強制停止、または異常終了）した場合、mqafrcコマンドでの復旧作業を行う必要があります。詳細については、「(c) キューファイルの入出力エラーの対処 (MQA FRC使用時)」を参照してください。

(c) キューファイルの入出力エラーの対処 (MQA FRC 使用時)

KFCA31187-Eメッセージに“物理ファイル=cnt”が出力されたあと、mqarles -s t コマンドによって一時的障害閉塞状態を解除する前にオンラインが終了（計画停止、強制停止、または異常終了）した場合の対処について、次の図に示します。

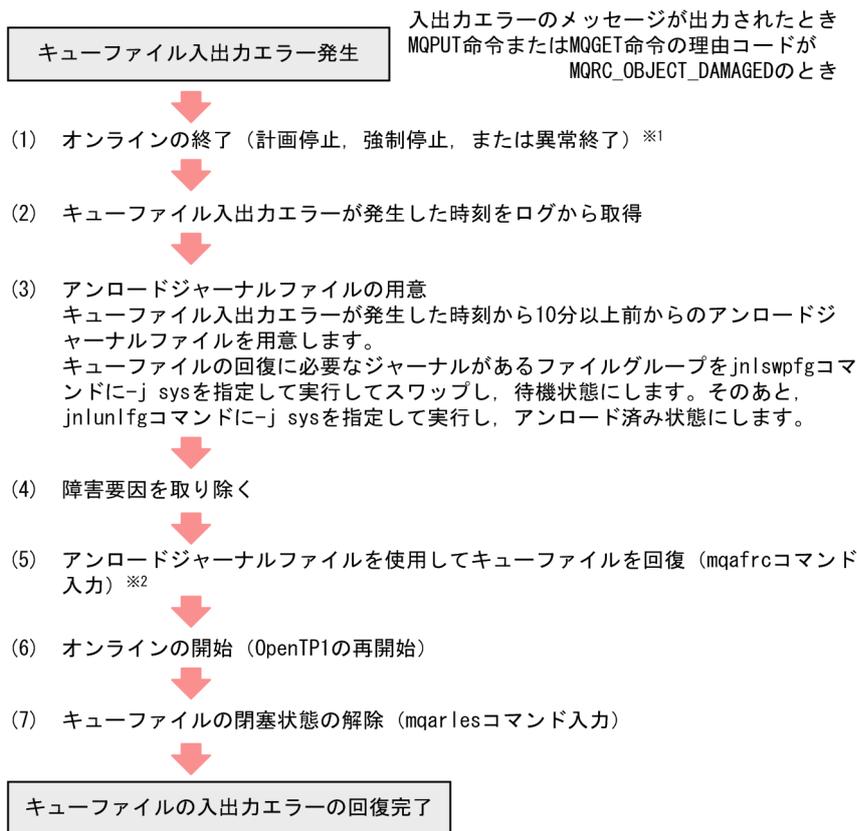
mqafrc コマンドの実行方法については、「2.2.2(13) mqafrc コマンドの使用例 (キューファイル格納媒体損傷以外の場合)」を参照してください。

なお、次に示す場合は、OpenTP1 を再開したあとで「(a) キューファイルの入出力エラーの対処」または「(b) 二重化構成しているキューファイルの入出力エラーの対処」の手順を実行してください。

- KFCA31193-I メッセージまたは KFCA31195-I メッセージが出力された場合
- KFCA31187-E メッセージに"物理ファイル= msg"が出力されたあと、mqarles -s t コマンドで一時的障害閉塞状態を解除する前にオンラインが終了した場合

図 7-8 キューファイルの入出力エラーの対処 (MQA FRC 使用時)

●入出力エラーが発生したあと、オンラインが終了した場合

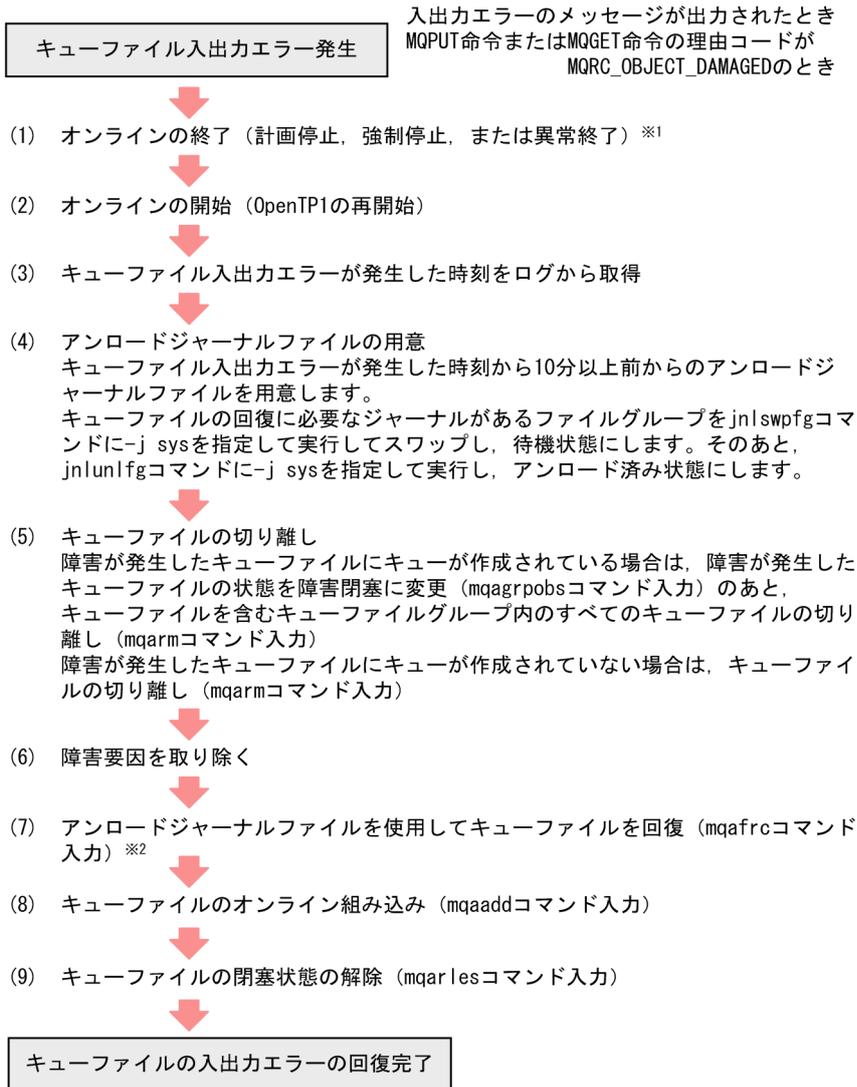


注※1 正常終了した場合でもこの手順でキューファイルを回復できます。しかし、(5)でキューファイルを回復しないで(6)でオンラインを開始するとキューファイルの内容に矛盾が生じるのでご注意ください。

注※2 二重化構成しているキューファイルの場合、次に示す手順でキューファイルを回復してください。

1. mqafrcコマンドで、一時的障害閉塞となった系のキューファイルを回復する。
2. 回復したキューファイルをバックアップ (mqabkupコマンド入力) する。
3. 2. でバックアップしたファイルをもう一方の系のキューファイル名でリストア (mqarstrコマンド入力) する。

- 入出力エラーが発生したあとオンラインが終了し、MQA FRCを使用しないでOpenTP1を再開始した場合



注※1 正常終了した場合、(2)でオンラインを開始するとキューファイルの内容に矛盾が生じるのでご注意ください。

注※2 二重化構成しているキューファイルの場合、次に示す手順でキューファイルを回復してください。

1. mqafrcコマンドで、一時的障害閉塞となった系のキューファイルを回復する。
2. 回復したキューファイルをバックアップ（mqabkupコマンド入力）する。
3. 2.でバックアップしたファイルをもう一方の系のキューファイル名でリストア（mqarstrコマンド入力）する。

7.1.5 クラスタ環境のシステムキューの障害

クラスタ環境で次のシステムキューに障害が発生した場合の対処について説明します。

- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE

障害要因を調査してキューファイル格納媒体損傷による障害かそれ以外による障害か判断し、それぞれに応じて対処します。

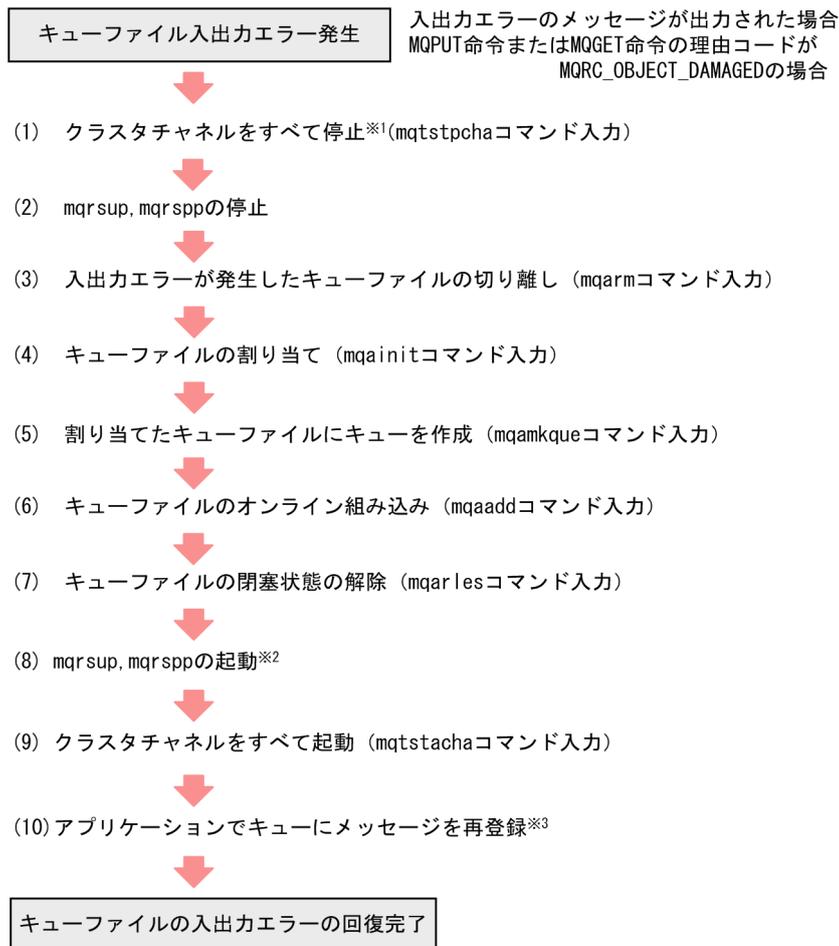
(1) キューファイル格納媒体損傷による障害の場合

キューファイル格納媒体損傷による障害の場合の対処について説明します。

(a) キューファイルの入出力エラーの対処

クラスタ環境のシステムキューファイルに入出力エラーが発生した場合の対処について、次の図に示します。

図 7-9 キューファイルの入出力エラーの対処



注※1 KFCA16350-I, KFCA16351-I, またはKFCA16352-Wの出力を確認してください。
KFCA16351-I または KFCA16352-Wが出力されない場合、mqtstpcha -f コマンドを入力して、チャンネルを強制停止させてください。

注※2 入出力エラーが発生したキューファイルに登録されているキューがSYSTEM.CLUSTER.REPOSITORY.QUEUEの場合、以前のクラスタ参加情報は失われ、再度クラスタ参加を実行します。mqr sup, mqr sppを起動する前にフルリポジトリキューマネージャからRESET CLUSTERコマンドを実行して、自キューマネージャをクラスタから削除してください。

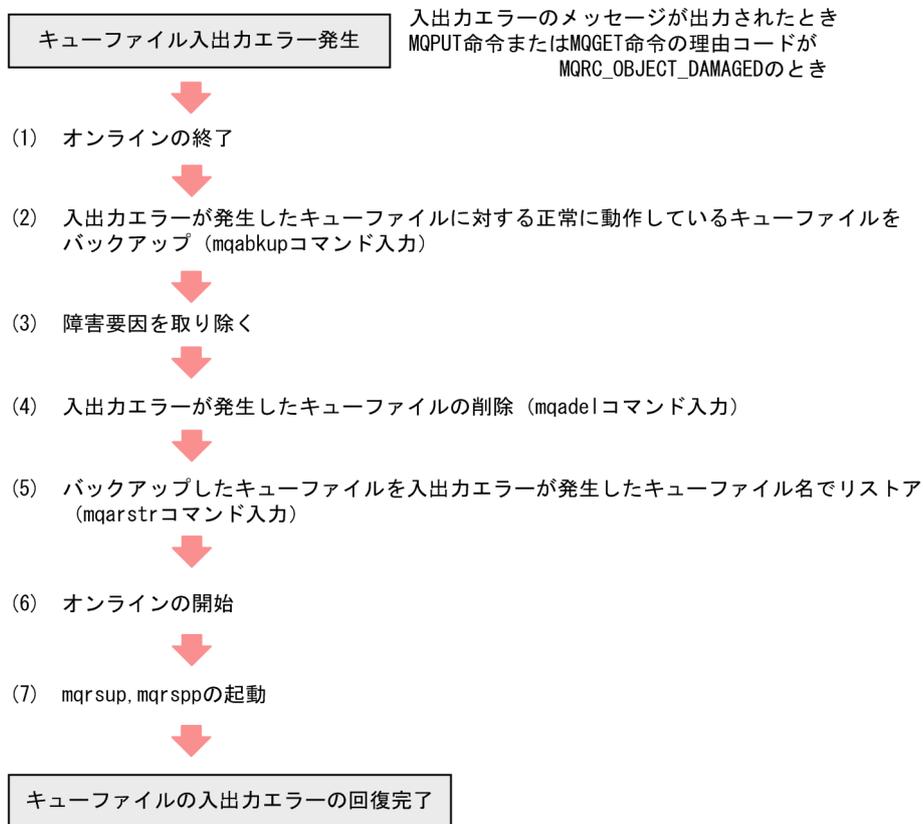
注※3 入出力エラーが発生したキューファイルに登録されているキューがSYSTEM.CLUSTER.TRANSMIT.QUEUEの場合だけ実行します。

(b) 二重化構成しているキューファイルの入出力エラーの対処

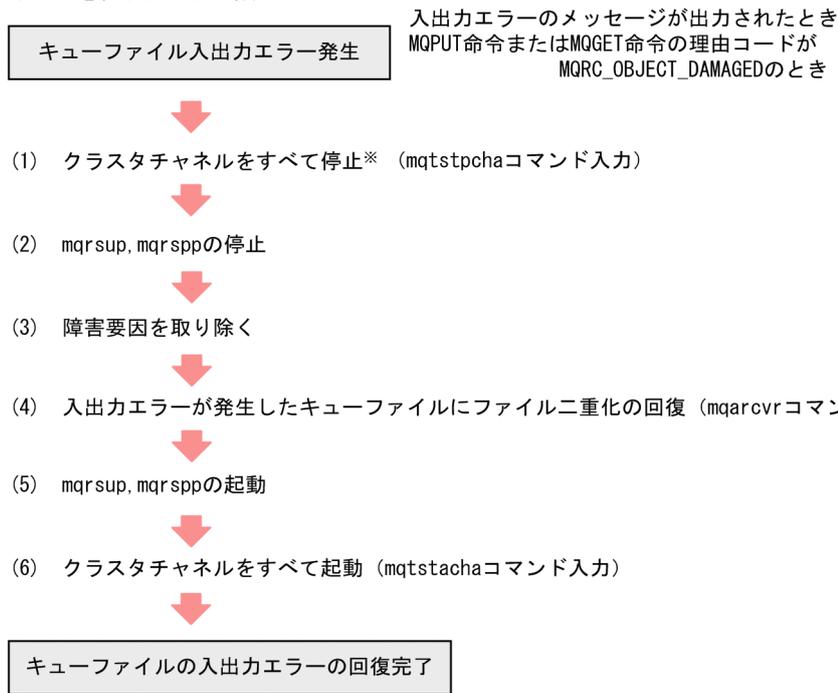
二重化構成しているキューファイルで片系のキューファイルに入出力エラーが発生した場合の対処について、次の図に示します。

図 7-10 二重化構成しているキューファイルの入出力エラーの対処

●オンラインを終了させる場合



●オンラインを終了させない場合



注※ KFCA16350-I, KFCA16351-I, またはKFCA16352-Wの出力を確認してください。
KFCA16351-I または KFCA16352-Wが出力されない場合、mqtstpcha -f コマンドを
入力して、チャネルを強制停止させてください。

OpenTP1 を正常開始する場合、前回のオンラインで二重化構成しているキューファイルの入出力エラーが発生したときは、該当するキューファイルの回復を必ず実行してください。キューファイルの回復を実行しないで OpenTP1 を正常開始した場合、キューファイルの内容は保証されません。

(c) キューファイル内のキューおよびメッセージの扱い

障害が発生したキューファイルに、キューおよびメッセージが登録されている場合があります。障害が発生したキューファイルと同一のキューファイルグループに複数ファイル指定してある場合は、障害が発生したキューファイルを含むすべてのキューファイルの状態を障害閉塞に変更後 (mqagrpbobs コマンド入力)、障害回復を実行してください。

(d) キューファイルの入出力エラーの対処 (MQA FRC 使用時)

MQA FRC 使用時のキューファイルに入出力エラーが発生した場合の対処について、次の図に示します。キューファイルのバックアップ方法、リストア方法、および mqafrc コマンドの実行方法については、「2.2.2(11) キューファイルの回復」を参照してください。

図 7-11 キューファイルの入出力エラーの対処 (MQA FRC 使用時)

●オンラインを終了させる場合

(1) キューファイルのバックアップ取得



(2) オンラインの開始



(3) mqr sup, mqr sppの起動



キューファイル入出力エラー発生

入出力エラーのメッセージが出力されたとき
MQPUT命令またはMQGET命令の理由コードが
MQRC_OBJECT_DAMAGEDのとき



(4) クラスタチャネルをすべて停止※ (mqtstpchaコマンド入力)



(5) mqr sup, mqr sppの停止



(6) オンラインの終了 (OpenTP1の計画停止または強制停止)



(7) 障害要因を取り除く



(8) 障害が発生したキューファイルのバックアップからリストア (バックアップ時に
メッセージが登録されていないキューは再作成可能)



(9) アンロードジャーナルファイルを使用してキューファイルを回復
(mqafrcコマンド入力)



(10) オンラインの開始 (OpenTP1の再開始)



(11) キューファイルの閉塞状態の解除 (mqarlesコマンド入力)



(12) mqr sup, mqr sppの起動



(13) クラスタチャネルをすべて起動 (mqtstachaコマンド入力)



キューファイルの入出力エラーの回復完了

注※ KFCA16350-I, KFCA16351-I, またはKFCA16352-Wの出力を確認してください。
KFCA16351-I または KFCA16352-Wが出力されない場合, mqtstpcha -f コマンドを入力し
て, チャネルを強制停止させてください。

●オンラインを終了させない場合

(1) キューファイルのバックアップ取得



(2) オンラインの開始



(3) mqrsup, mqrsppの起動



キューファイル入出力エラー発生

入出力エラーのメッセージが出力されたとき
MQPUT命令またはMQGET命令の理由コードが
MQRC_OBJECT_DAMAGEDのとき



(4) クラスタチャネルをすべて停止※(mqtstpchaコマンド入力)



(5) mqrsup, mqrsppの停止



(6) アンロードジャーナルファイルの用意

キューファイルの回復に必要なジャーナルファイルがあるファイルグループをjnlswpfgコマンドに-j sysを指定して実行してスワップし、待機状態にします。
その後、jnlnlfgコマンドに-j sysを指定して実行し、アンロード済み状態にします。



(7) 障害が発生したキューファイルを含むすべてのキューファイルの状態を障害閉塞に変更後
(mqagrpbobsコマンド入力)、障害が発生したキューファイルを含むキューファイルグループ内のすべてのキューファイルの切り離し (mqarmコマンド)



(8) 障害原因を取り除く



(9) 次に示すどちらかからリストア

(バックアップ時にメッセージが登録されていないキューは再作成可能)

- ・ 障害が発生したキューファイル (キューあり) を含むキューファイルグループ内のすべてのキューファイルのバックアップ
- ・ 障害が発生したキューファイル (キューなし) のバックアップ



(10) アンロードジャーナルファイルを使用してキューファイルを回復 (mqafrcコマンド入力)



(11) mqrsup, mqrsppの起動



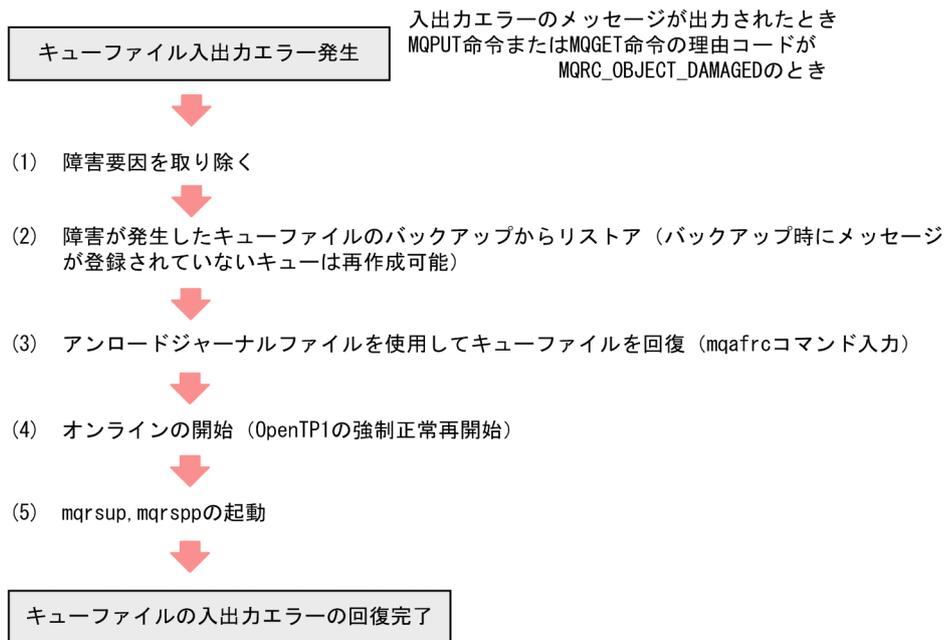
(12) クラスタチャネルをすべて起動 (mqtstachaコマンド入力)



キューファイルの入出力エラーの回復完了

注※ KFCA16350-I, KFCA16351-I, またはKFCA16352-Wの出力を確認してください。
KFCA16351-I, またはKFCA16352-Wが出力されない場合, mqtstpcha -fコマンドを入力してチャネルを強制停止させてください。

●OpenTP1が異常終了したあと、再開始できない場合



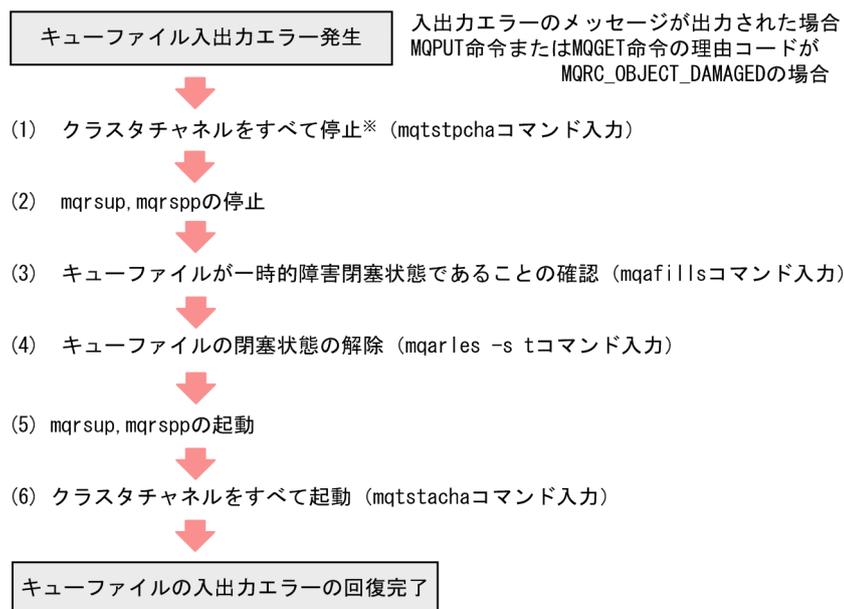
(2) キューファイル格納媒体損傷以外による障害の場合

キューファイル格納媒体損傷以外による障害の場合の対処について説明します。

(a) キューファイルの入出力エラーの対処

クラスタ環境のシステムキューファイルに入出力エラーが発生した場合の対処について、次の図に示します。

図 7-12 キューファイルの入出力エラーの対処



注 KFCA31187-Eメッセージに“物理ファイル=cnt”が出力されたあと、mqarles -s t コマンドによって一時的障害閉塞状態を解除する前にオンラインが終了（計画停止、強制停止、または異常終了）した場合、mqafrcコマンドでの復旧作業を行う必要があります。詳細については、「(c) キューファイルの入出力エラーの対処 (MQA FRC使用時)」を参照してください。

注※ KFCA16350-I, KFCA16351-I, またはKFCA16352-Wの出力を確認してください。
KFCA16351-I または KFCA16352-Wが出力されない場合、mqtstpcha -f コマンドを入力して、チャンネルを強制停止させてください。

(b) 二重化構成しているキューファイルの入出力エラーの対処

二重化構成しているキューファイルの場合、両系が正常のときにファイル入出力エラーが発生すると、エラーが発生した系のキューファイルは障害閉塞になります。

また、片系障害の状態でもう一方の系のキューファイルにファイル入出力エラーが発生すると、その系のキューファイルは一時的障害閉塞になります。

二重化構成しているキューファイルの入出力エラーについて、次の表に示します。

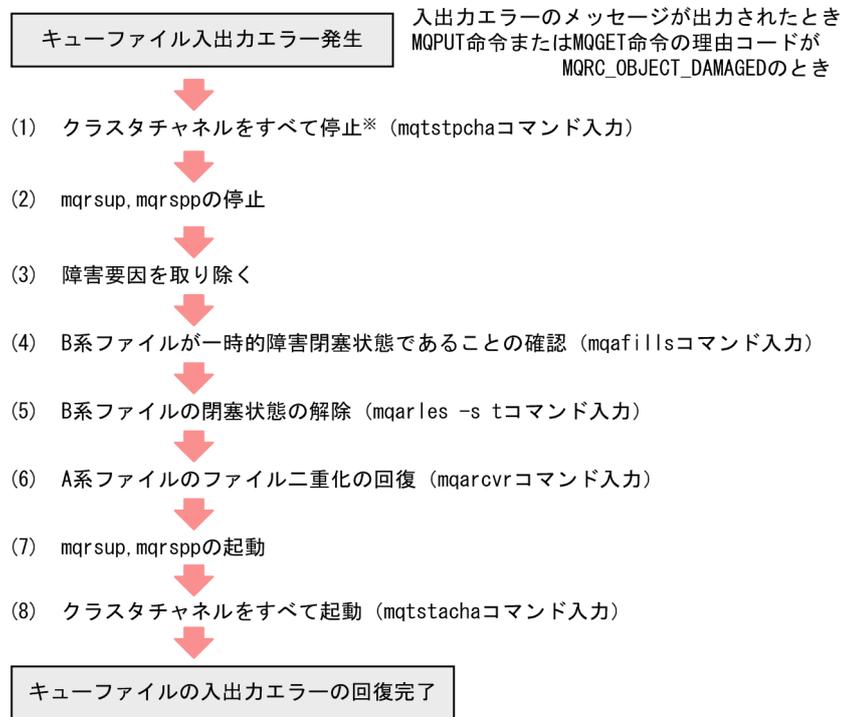
表 7-3 二重化構成しているキューファイルの入出力エラー

障害内容	A系ファイルの状態	B系ファイルの状態
A系ファイルで障害発生	障害閉塞	正常
B系ファイルで障害発生	正常	障害閉塞
A系ファイル障害状態でB系ファイルで障害発生	障害閉塞	一時的障害閉塞
B系ファイル障害状態でA系ファイルで障害発生	一時的障害閉塞	障害閉塞

二重化構成しているキューファイルで A 系ファイルが障害閉塞状態になったあと、B 系ファイルに入出力エラーが発生した場合の対処について、次の図に示します。

なお、二重化構成しているキューファイルで A 系ファイルに入出力エラーが発生した場合の対処については、「(1)(b) 二重化構成しているキューファイルの入出力エラーの対処」を参照してください。

図 7-13 二重化構成しているキューファイルの入出力エラーの対処



注 KFCA31187-Eメッセージに“物理ファイル=cnt”が出力されたあと、mqarles -s t コマンドによって一時的障害閉塞状態を解除する前にオンラインが終了（計画停止、強制停止、または異常終了）した場合、mqafrcコマンドでの復旧作業を行う必要があります。詳細については、「(c) キューファイルの入出力エラーの対処（MQA FRC使用時）」を参照してください。

注※ KFCA16350-I, KFCA16351-I, またはKFCA16352-Wの出力を確認してください。
KFCA16351-I, またはKFCA16352-Wが出力されない場合、mqtstpcha -fコマンドを入力してチャンネルを強制停止させてください。

(c) キューファイルの入出力エラーの対処（MQA FRC 使用時）

KFCA31187-Eメッセージに“物理ファイル=cnt”が出力されたあと、mqarles -s t コマンドによって一時的障害閉塞状態を解除する前にオンラインが終了（計画停止、強制停止、または異常終了）した場合の対処について、次の図に示します。

mqafrc コマンドの実行方法については、「2.2.2(13) mqafrc コマンドの使用例（キューファイル格納媒体損傷以外の場合）」を参照してください。

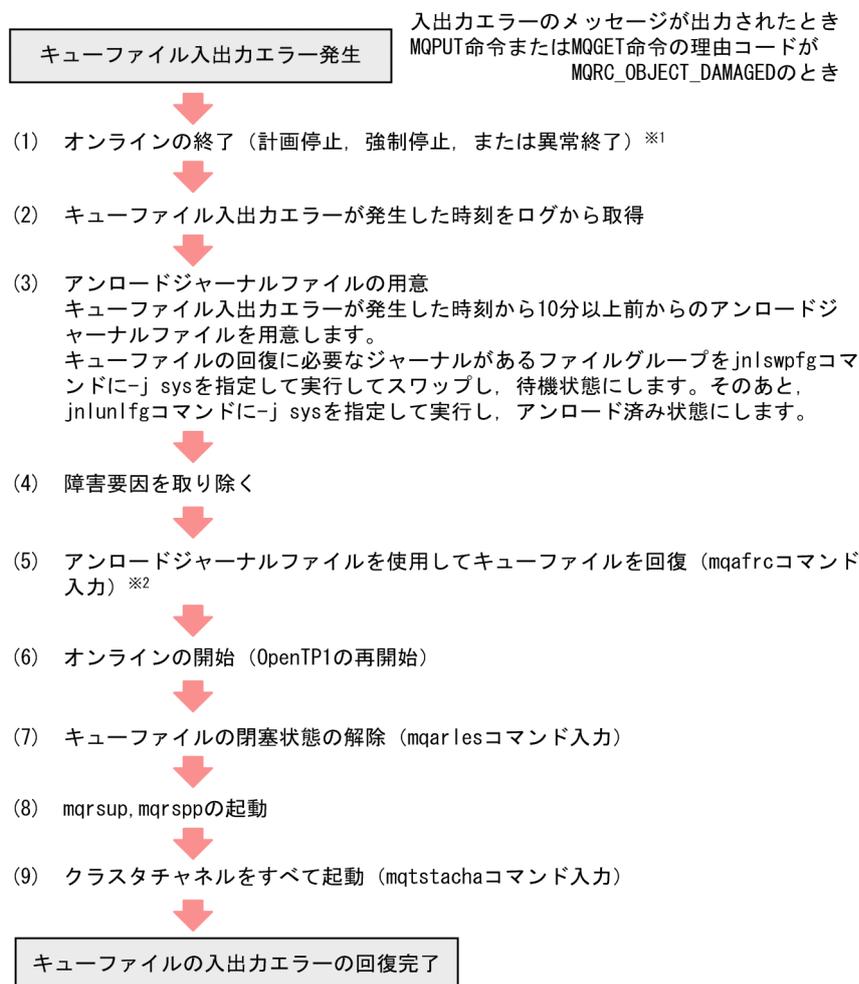
なお、次に示す場合は、OpenTP1 を再開始したあとで「(a) キューファイルの入出力エラーの対処」または「(b) 二重化構成しているキューファイルの入出力エラーの対処」の手順を実行してください。

- KFCA31193-I メッセージまたは KFCA31195-I メッセージが出力された場合

- KFCA31187-E メッセージに"物理ファイル= msg"が出力されたあと、mqarles -s t コマンドで一時的障害閉塞状態を解除する前にオンラインが終了した場合

図 7-14 キューファイルの入出力エラーの対処 (MQA FRC 使用時)

- 入出力エラーが発生したあと、オンラインが終了した場合

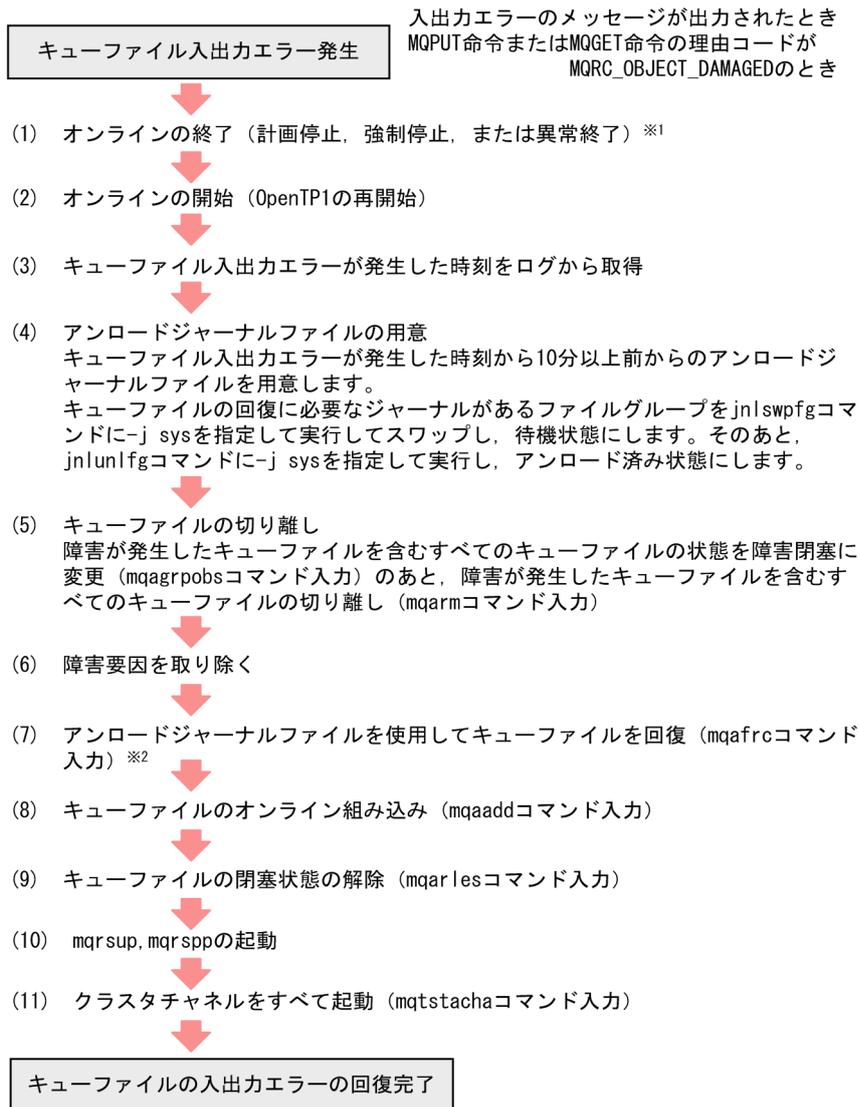


注※1 正常終了した場合でもこの手順でキューファイルを回復できます。しかし、(5)でキューファイルを回復しないで(6)でオンラインを開始するとキューファイルの内容に矛盾が生じるのでご注意ください。

注※2 二重化構成しているキューファイルの場合、次に示す手順でキューファイルを回復してください。

1. mqafrcコマンドで、一時的障害閉塞となった系のキューファイルを回復する。
2. 回復したキューファイルをバックアップ (mqabkupコマンド入力) する。
3. 2. でバックアップしたファイルをもう一方の系のキューファイル名でリストア (mqarstrコマンド入力) する。

- 入出力エラーが発生したあとオンラインが終了し、MQA FRCを使用しないでOpenTP1を再開した場合



注※1 正常終了した場合，(2)でオンラインを開始するとキューファイルの内容に矛盾が生じるのでご注意ください。

注※2 二重化構成しているキューファイルの場合，次に示す手順でキューファイルを回復してください。

1. mqafrcコマンドで，一時的障害閉塞となった系のキューファイルを回復する。
2. 回復したキューファイルをバックアップ（mqabkupコマンド入力）する。
3. 2.でバックアップしたファイルをもう一方の系のキューファイル名でリストア（mqarstrコマンド入力）する。

7.2 通信障害

TP1/Message Queue のメッセージの送受信についての障害と、障害に対する処理について障害の種類ごとに説明します。

7.2.1 コネクション障害（コーラ側）

コネクション障害（コーラ側）の障害処理について、次の表に示します。

表 7-4 コネクション障害（コーラ側）の障害処理

障害内容	MQT の障害処理	ユーザの処理
コネクション確立時の障害 (重度) ※1	<ol style="list-style-type: none">1. < 16331-E > 「TCP インタフェースエラー」または< 16322-E > 「確立に失敗」2. チャンネルを終了します（チャンネル使用不可状態）。	障害要因を取り除いたあと mqtstacha コマンドを入力してください。
コネクション確立時の障害 (軽度) ※2	<ol style="list-style-type: none">1. < 16331-E > 「TCP インタフェースエラー」または< 16322-E > 「確立に失敗」2. チャンネル定義の確立再試行の指定によって、チャンネル確立を再試行します。3. 再試行しない場合※3, または再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。	障害要因を取り除いてください。3.の場合は、障害要因を取り除いたあと mqtstacha コマンドを入力してください。
メッセージ転送時の障害	<ol style="list-style-type: none">1. < 16331-E > 「TCP インタフェースエラー」2. メッセージの転送を中断します。3. 送信中のメッセージ※4 は転送キューに戻し、受信中のメッセージ※4 は破棄されます。4. チャンネル定義の確立再試行の指定によって、チャンネル確立を再試行します。5. 再試行しない場合※3 または再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。	障害要因を取り除いてください。5.の場合は、障害要因を取り除いたあと mqtstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

コネクション確立時の障害（重度）とは、エラーナンバ ENOBUFS（Windows の場合は、WSAENOBUFS）エラーのことであり、メモリ不足によって発生します。

注※2

コネクション確立時の障害（軽度）とは、エラーナンバ ENOBUFS（Windows の場合は、WSAENOBUFS）以外のエラーのことです。

注※3

クラスタチャネルの場合は、チャネル状態は「チャネル停止」になります。

注※4

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

7.2.2 コネクション障害（レスポンド側）

コネクション障害（レスポンド側）の障害処理について、次の表に示します。

表 7-5 コネクション障害（レスポンド側）の障害処理

障害内容	MQT の障害処理	ユーザの処理
コネクション確立障害	<ol style="list-style-type: none"> 1. < 16331-E > 「TCP インタフェースエラー」または< 16332-W > 「TCP インタフェースエラー処理続行」 2. TCP/IP バッファ不足の場合だけ、ネットワークリクエスト機能を縮退します。 	<p>相手システムからの新しいチャネル開始要求を中止してください。</p> <p>ネットワークリクエスト機能を使用する場合は、障害要因を取り除き、MQT サーバを再起動してください。</p>
メッセージ転送時の障害	<ol style="list-style-type: none"> 1. < 16331-E > 「TCP インタフェースエラー」 2. メッセージの転送を中断します。 3. 送信中のメッセージ※は転送キューに戻し、受信中のメッセージ※は破棄されます。 4. チャネルを終了し、再度相手システムからのコネクション確立要求を待ちます（チャネル停止状態）。 	<p>障害要因を取り除いてください。</p>

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

7.2.3 MCP 障害（チャンネル確立時の障害）

MCP 障害（チャンネル確立時の障害）の障害処理について、次の表に示します。

表 7-6 MCP 障害（チャンネル確立時の障害）の障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> チャンネルタイプ対応不可 最大メッセージシーケンス番号不一致 同一チャンネル名未定義 ネゴシエーション不可能 その他 	<ol style="list-style-type: none"> < 16341-E > 「開始要求拒否」（コーラ側だけで出力） チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 再試行しない場合※、または再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。 	3.の場合は、チャンネル定義を見直して、相手システムの属性と一致させたあと、mqststacha コマンドを入力してください。
<ul style="list-style-type: none"> 相手キューマネージャ終了中/未起動 相手システムでチャンネル使用不可状態 その他 	<ol style="list-style-type: none"> < 16341-E > 「開始要求拒否」（コーラ側だけで出力） チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 再試行しない場合はチャンネルを終了します（チャンネル停止状態）。 再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。 	3.または 4.の場合は、相手システムでチャンネル状態を確認し mqststacha コマンドを入力してください。
<ul style="list-style-type: none"> チャンネル状態不一致（相手からの開始要求を受信したが、チャンネルは動作している） 	<ol style="list-style-type: none"> < 16317-E > 「チャンネル状態不一致発生」または< 16354-W > 「チャンネル状態不一致発生開始要求受入」（レスポンド側だけで出力） チャンネル定義の強制確立確認条件の指定によって、強制接続します。 強制接続しない場合または強制接続条件を満たさない場合は、動作中のチャンネルは処理を続行し、チャンネル開始要求は拒否します。 	なし。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

7.2.4 MCP 障害（チャンネル確立後，自システムで検出した MQ プロトコル障害）

MCP 障害（チャンネル確立後，自システムで検出した MQ プロトコル障害）の障害処理について，次の表に示します。

表 7-7 MCP 障害（チャンネル確立後，自システムで検出した MQ プロトコル障害）の障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> メッセージ長不正 バッチサイズ不正 セグメント番号不正 その他 	<ol style="list-style-type: none"> < 16323-E > 「不正データ受信」または < 16324-E > 「プロトコルエラー発生」 メッセージの転送を中断します。 送信中のメッセージ[*]は転送キューに戻し，受信中のメッセージ[*]は破棄されます。 コネクションを強制解放します。 チャンネルを終了します（チャンネル使用不可状態）。 	保守情報を取得し，mqstacha コマンドを入力してください。

（凡例）

< >：出力されるメッセージログ ID です。メッセージログ ID には，先頭に KFCA が付きます。

[]：メッセージログの内容です。

注※

バッチ転送によって複数のメッセージを転送していた場合，そのバッチのすべてのメッセージが該当します。

ファーストチャンネルが転送する非永続メッセージが，転送キューから取り出されて相手システムに登録される前に，該当するバッチに障害が発生した場合，非永続メッセージは破棄されることがあります。

7.2.5 MCP 障害（チャンネル確立後，相手システムで検出した MQ プロトコル障害）

MCP 障害（チャンネル確立後，相手システムで検出した MQ プロトコル障害）の障害処理について，次の表に示します。

表 7-8 MCP 障害（チャンネル確立後，相手システムで検出した MQ プロトコル障害）の障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> リモートキューマネージャ状態不正 セキュリティ不正 チャンネル状態不正 	<ol style="list-style-type: none"> < 16345-E > 「エラーデータ受信」 メッセージの転送を中断します。 送信中のメッセージ[*]は転送キューに戻し，受信中のメッセージ[*]は破棄されます。 コネクションを強制解放します。 	相手システムの状態を確認し，mqstacha コマンドを入力してください。

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> リモートキューマネージャ状態不正 セキュリティ不正 チャンネル状態不正 	<p>5. コーラ側はチャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。</p> <p>6. 再試行しない場合またはレスポンド側ではチャンネルを終了します（チャンネル停止状態）。</p> <p>7. 再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。</p>	相手システムの状態を確認し、mqstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャンネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

7.2.6 メモリ不足

メモリ不足の障害処理について、次の表に示します。

表 7-9 メモリ不足の障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> 開始時のメモリ不足 受信バッファ不足 (TCP 定義の mqttcp 定義コマンドの -g オペランド) 	<p>1. < 16390-E > 「ローカルメモリ不足」、< 16391-E > 「メモリ不足」、または < 16395-E > 「共用メモリ不足」</p> <p>2. OpenTP1 を終了します。</p>	メモリサイズを見直し、再度 OpenTP1 システムを開始してください。
<ul style="list-style-type: none"> メッセージ転送時のメモリ不足 送受信バッファ不足 (TCP 定義の mqtalcca 定義コマンドの -g オペランド) 	<p>1. < 16390-E > 「ローカルメモリ不足」、< 16391-E > 「メモリ不足」、または < 16330-E > 「バッファグループ番号不正」</p> <p>2. チャンネルを終了します (チャンネル停止状態)。</p>	メモリサイズを見直し、再度チャンネル開始要求を出してください。 < 16330-E > の場合はバッファグループ番号に該当するバッファ数 (MQT 共通定義の mqttbuf 定義コマンドの -g count オペランド指定値) を大きくしてください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

7.2.7 MCP 障害（メッセージ送信時の転送障害）

MCP 障害（メッセージ送信時の転送障害）の障害処理について、次の表に示します。

表 7-10 MCP 障害（メッセージ送信時の転送障害）の障害処理

障害内容	MQT の障害処理	ユーザの処理
転送キューのメッセージ不正（チャンネルの転送サイズオーバー含む）	<ol style="list-style-type: none"> 1. < 16334-E > 「メッセージ異常検出」 2. デッドレターキューがある場合：送信中のメッセージを登録し、送信処理を続行します。 3. デッドレターキューがない場合（コーラ側）： <ol style="list-style-type: none"> (a) 送信中のメッセージ^{※1}を転送キューに戻し、コネクションを強制解放してチャンネルを終了します。 (b) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 (c) 再試行しないとき^{※2}、または再試行回数を超えたときはチャンネルを終了します（チャンネル使用不可状態）。 4. デッドレターキューがない場合（レスポンド側）：送信中のメッセージ^{※2}を転送キューに戻し、コネクションを強制解放してチャンネルを終了します（チャンネル使用不可状態）。 	<ol style="list-style-type: none"> 2.の場合は、メッセージを登録したアプリケーションを起動してください。 3.の(c)または4.の場合は、転送キューから該当するメッセージを取り除き、メッセージを登録したアプリケーションを起動したあと、mqststacha コマンドを入力してください。
メッセージ取り出し障害	<ol style="list-style-type: none"> 1. < 16333-E > 「キューファイル障害発生」 2. メッセージ^{※1}の送信を中断します。 3. コネクションを強制解放します。 4. コーラ側： <ol style="list-style-type: none"> (a) チャンネル定義の MCP 障害再試行の指定によって、再試行します。 (b) 再試行しない場合^{※2}、または再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。 5. レスポンド側：チャンネルを終了します（チャンネル使用不可状態）。 	<p>障害要因を取り除いたあと、4.の(b)の場合は、mqststacha コマンドを入力してください。</p>
メッセージシーケンス番号不一致	<ol style="list-style-type: none"> 1. < 16343-E > 「シーケンス番号不一致」 2. メッセージの送信を中断します。 3. 送信中のメッセージ^{※1}は転送キューに戻されます。 4. コーラ側： 	<p>mqtrstcha コマンドを入力し、番号を一致させてください。4.の(c)の場合は、mqststacha コマンドを入力してください。</p>

障害内容	MQT の障害処理	ユーザの処理
メッセージシーケンス番号不一致	<p>(a) チャンネルを終了します。</p> <p>(b) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。</p> <p>(c) 再試行しない場合^{※2}、または再試行回数を超えた場合はチャンネルを終了します (チャンネル使用不可状態)。</p> <p>5. レスポンド側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます (チャンネル停止状態)。</p>	mqtrstcha コマンドを入力し、番号を一致させてください。4.の (c) の場合は、mqstacha コマンドを入力してください。
相手デッドレターキューにメッセージが登録された	<p>1. < 16335-W > 「デッドレターキューに登録された」</p> <p>2. メッセージの送信を続行します。</p>	相手システムのデッドレターキューのメッセージを取り出し、原因を調査してください。
転送キュー重複使用	<p>1. < 16315-E > 「キューが他サーバで使用」</p> <p>2. メッセージ^{※1}の送信を中断します。</p> <p>3. コネクションを強制解放します。</p> <p>4. コーラ側：</p> <p>(a) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。</p> <p>(b) 再試行しない場合^{※2}、または再試行回数を超えた場合はチャンネルを終了します (チャンネル使用不可状態)。</p> <p>5. レスポンド側：チャンネルを終了します (チャンネル使用不可状態)。</p>	自システム内のシステムサーバで該当キューが重複使用されていないか確認し、構成を変更してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャンネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

注※2

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

7.2.8 MCP 障害 (メッセージ受信時の転送障害)

MCP 障害 (メッセージ受信時の転送障害) の障害処理について、次の表に示します。

表 7-11 MCP 障害（メッセージ受信時の転送障害）の障害処理

障害内容	MQT の障害処理	ユーザの処理
メッセージ登録障害（あて先キューがない場合を含む）	<ol style="list-style-type: none"> 1. < 16333-E > 「キューファイル障害発生」 2. デッドレターキューがある場合：受信中のメッセージを登録し、受信処理を続行します。 3. デッドレターキューがない場合（コーラ側）： <ol style="list-style-type: none"> (a) 受信中のメッセージ※¹ を破棄し、チャンネルを終了します。 (b) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 (c) 再試行しないとき※²、または再試行回数を超えたときはチャンネルを終了します（チャンネル使用不可状態）。 4. デッドレターキューがない場合（レスポンド側）：受信中のメッセージ※¹ を破棄し、チャンネルを終了します（チャンネル使用不可状態）。 	<ol style="list-style-type: none"> 2. の場合、障害要因を取り除いてください。 3. の (c) または 4. の場合、障害要因を取り除いたあと、mqststacha コマンドを入力してください。
メッセージシーケンス番号不一致	<ol style="list-style-type: none"> 1. < 16343-E > 「シーケンス番号不一致」 2. メッセージの受信を中断します。 3. 受信中のメッセージ※¹ を破棄します。 4. コーラ側： <ol style="list-style-type: none"> (a) チャンネルを終了します。 (b) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 (c) 再試行しない場合※²、または再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。 5. レスポンド側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます（チャンネル停止状態）。 	<p>mqtrstcha コマンドを入力して番号を一致させてください。</p> <p>4. コーラ側の (c) の場合は mqststacha コマンドを入力してください。</p>

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャンネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

注※2

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

7.2.9 MCP 障害（デッドレターキュー障害）

MCP 障害（デッドレターキュー障害）の障害処理について、次の表に示します。

表 7-12 MCP 障害（デッドレターキュー障害）の障害処理

障害内容	MQT の障害処理	ユーザの処理
デッドレターキュー登録障害	<ol style="list-style-type: none">1. <16333-E> 「キューファイル障害発生」2. 送信中のメッセージ※¹ は転送キューに戻し、接続を強制解放します。3. 受信中のメッセージ※¹ を破棄します。4. コーラ側およびレスポング側の送受信側： コーラ側の送信側 (a) チャンネルを終了します。 (b) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 (c) 再試行しない場合※²、または再試行回数を超えた場合は、チャンネルを終了します（チャンネル使用不可状態）。 レスポング側の送信側 (a) チャンネルを終了します。 コーラ側の受信側、およびレスポング側の受信側 (a) 送信側にエラー応答します。5. 送信側および受信側： 送信側 (a) チャンネルを終了します（チャンネル使用不可の状態※²）。 受信側 (a) 送信側からの要求に従って、チャンネルを終了します（チャンネル停止）。	障害要因を取り除いたあと mqtstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャンネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

注※2

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

7.2.10 イニシエーションキュー障害

イニシエーションキュー障害の障害処理について、次の表に示します。

表 7-13 イニシエーションキュー障害の障害処理

障害内容	MQT の障害処理	ユーザの処理
トリガメッセージ取り出し障害	<ol style="list-style-type: none"> 1. < 16307-E > 「キューファイル障害発生」 2. トリガ起動機能を縮退します。 3. イニシエーションキューにトリガメッセージが残ります。 	<p>トリガメッセージの発生する転送キューへのメッセージの登録を中止してください。</p> <p>トリガ起動機能を使用する場合は、障害要因を取り除き MQT サーバを再度起動してください。</p>

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

7.2.11 時間監視障害

時間監視障害の障害処理について、次の表に示します。

表 7-14 時間監視障害の障害処理

障害内容	MQT の障害処理	ユーザの処理
チャンネル確立完了監視タイムアウト	<ol style="list-style-type: none"> 1. < 16329-E > 「タイムアウト発生」 2. コネクションを強制解放します。 	相手システムの状態と自システムの TCP 構成定義の -v rtim を見直してください。
開始要求応答監視タイムアウト	<ol style="list-style-type: none"> 1. < 16329-E > 「タイムアウト発生」 2. コネクションを強制解放します。 3. チャンネルを終了します*1 (チャンネル使用不可状態*2)。 	相手システムの状態と自システムのチャンネルの定義の -v tim1 を見直し、mqtstacha コマンドを入力してください。
確認メッセージ受信監視タイムアウト	<ol style="list-style-type: none"> 1. < 16329-E > 「タイムアウト発生」 2. メッセージの送信を中断します。 3. 送信中のメッセージ*3 は転送キューに戻します。 4. コネクションを強制解放します。 5. コーラ側：チャンネルを終了します*1 (チャンネル使用不可状態*2)。 レスポンド側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます (チャンネル停止状態)。 	<p>相手システムの状態と自システムのチャンネル定義の -v tim2 を見直し、処置してください。</p> <p>5. のコーラ側の場合は、mqtstacha コマンドを入力してください。</p>
継続セグメント受信監視タイムアウト	<ol style="list-style-type: none"> 1. < 16329-E > 「タイムアウト発生」 2. メッセージの受信を中断します。 	相手システムの状態と自システムのチャンネル定義の -v tim3 を見直し、処置してください。

障害内容	MQT の障害処理	ユーザの処理
継続セグメント受信監視タイムアウト	3. 受信中のメッセージ ^{※3} を破棄します。 4. コネクションを強制解放します。 5. コーラ側：チャンネルを終了します ^{※1} (チャンネル使用不可状態)。 レスポンダ側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます (チャンネル停止状態)。	5. のコーラ側の場合は、mqtstacha コマンドを入力してください。
継続メッセージ受信監視タイムアウト	1. < 16329-E > 「タイムアウト発生」 2. メッセージの受信を中断します。 3. 受信中のメッセージ ^{※3} を破棄します。 4. コネクションを強制解放します。 5. コーラ側：チャンネルを終了します ^{※1} (チャンネル使用不可状態)。 レスポンダ側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます (チャンネル停止状態)。	相手システムの状態と自システムのチャンネル定義の-v mtim を見直し、処置してください。 5. のコーラ側の場合は、mqtstacha コマンドを入力してください。
終了処理監視タイムアウト	1. < 16312-E > 「タイムアウト発生」 2. メッセージの送受信を中断します。 3. 送信中のメッセージ ^{※3} は転送キューに戻し、受信中のメッセージ ^{※3} は破棄します。 4. コネクションを強制解放します。 5. 全チャンネルを終了します。 6. MQT を強制停止します。	TCP 構成定義の-v etim を見直し、修正してください。
ハートビートメッセージ受信監視タイムアウト	1. < 16329-E > 「タイムアウト発生」 2. コネクションを強制解放します。 3. コーラ側：チャンネルを終了します ^{※1} (チャンネル使用不可状態)。 レスポンダ側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます (チャンネル停止状態)。	相手システムの状態と自システムのチャンネル定義の-v htim を見直し、処置してください。 3. のコーラ側の場合は、mqtstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

チャンネル定義のタイムアウト時のチャンネル確立再試行の有無指定 (TCP 定義の mqtalccha 定義コマンドの-v vretry オペランド指定) に yes を指定した場合は、チャンネル確立再試行の指定 (-b オプション) に従ってチャンネル確立再試行を実行します。

注※2

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

注※3

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャンネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

7.2.12 出口障害

出口障害の障害処理について、次の表に示します。

表 7-15 出口障害の障害処理

障害内容	MQT の障害処理	ユーザの処理
出口エラー	<ol style="list-style-type: none">1. < 16346-E > 「UOC インタフェースエラー発生」2. ログメッセージの出口応答に従って処理します。	出口の処理に問題がないか確認してください。
応答パラメタ不正	<ol style="list-style-type: none">1. < 16347-E > 「UOC 応答パラメタ不正」2. メッセージの送受信を中断します。3. 送信中のメッセージ*は転送キューに戻し、受信中のメッセージ*は破棄します。4. コネクションを強制解放します。5. チャンネルを終了します（チャンネル使用不可状態）。	理由コードから出口処理を見直し、出口を修正してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

バッチ転送によって複数のメッセージを転送していた場合、そのバッチのすべてのメッセージが該当します。

ファーストチャンネルが転送する非永続メッセージが、転送キューから取り出されて相手システムに登録される前に、該当するバッチに障害が発生した場合、非永続メッセージは破棄されることがあります。

7.2.13 OpenTP1 システム、および MQ システム障害

OpenTP1 システム、および MQ システム障害の障害処理について、次の表に示します。

表 7-16 OpenTP1 システム、および MQ システム障害の障害処理

障害内容	MQT の障害処理	ユーザの処理
OpenTP1 異常終了	<ol style="list-style-type: none">1. OS によるコアダンプを出力します。2. メッセージ送受信を中断します。	障害要因を取り除いたあと OpenTP1 システムを再開*してください。

注※

MQT サーバを正常開始する場合、送信側 MCA でチャンネル開始要求を送る前に、送達未確認（mqtlscha コマンドで確認）を mqtrlvcha コマンドで解決してください。

7.2.14 チャンネル管理情報格納ファイル障害

チャンネル管理情報格納ファイル障害の障害処理について、次の表に示します。

表 7-17 チャンネル管理情報格納ファイル障害の障害処理

障害内容	MQT の障害処理	ユーザの処理
ファイル名称不正	<ol style="list-style-type: none"> 1. < 31902-E > 「ファイル名文字数オーバー」または< 31906-E > 「ファイル名不正」 2. MQT が異常終了します。 	MQA サービス定義に指定したチャンネル管理情報格納ファイル名を見直し、再度 OpenTP1 システムを正常開始してください。
ファイル入出力形式不正	<ol style="list-style-type: none"> 1. < 31941-E > 「入出力形式の指定誤り」 2. MQT が異常終了します。 	MQA サービス定義に指定したチャンネル管理情報格納ファイルの入出力形式を見直し、再度 OpenTP1 を正常開始してください。
開始時のファイル読み込み障害（重度障害）	<ol style="list-style-type: none"> 1. < 31901-E > 「ファイル異常」、< 31904-E > 「アクセス権不正」、< 31908-E > 「データ入出力エラー」、< 31906-E > 「ファイル名不正」、または< 31943-E > 「OpenTP1 ファイルのアクセス失敗」 2. 処理を続行します。ただし、指定したチャンネル管理情報格納ファイルのすべてに読み込み重度障害が発生した場合は異常終了します。 	障害要因を取り除きます。異常終了している場合、再度 OpenTP1 システムを正常開始してください。
チャンネル管理情報格納ファイルのバージョン不正	<ol style="list-style-type: none"> 1. < 31942-E > 「バージョンが異なる」 2. 処理を続行します。ただし、指定したチャンネル管理情報格納ファイルのすべてに読み込み重度障害が発生した場合は異常終了します。 	該当するファイルを削除するか別のファイル名を指定します。異常終了している場合、再度 OpenTP1 を正常開始してください。
開始時のファイル読み込み障害（軽度障害）	<ol style="list-style-type: none"> 1. < 31900-W > 「終了時の書き込みなし」または< 31905-W > 「指定ファイルなし」 2. 処理を続行します。ただし、指定したチャンネル管理情報格納ファイルのすべてに読み込み障害が発生した場合は、次のメッセージを出力します。 < 31903-W > 「チャンネル情報引き継ぎ不可」 	なし。
開始時のファイル書き込み障害	<ol style="list-style-type: none"> 1. < 31901-E > 「ファイル異常」、< 31904-E > 「アクセス権不正」、< 	障害要因を取り除きます。異常終了している場合、再度 OpenTP1 システムを正常開始してください。

障害内容	MQT の障害処理	ユーザの処理
開始時のファイル書き込み障害	<p>31906-E > 「ファイル名不正」, < 31907-E > 「ファイル容量不足」, < 31908-E > 「データ入出力エラー」, または < 31943-E > 「OpenTP1 ファイルのアクセス失敗」</p> <p>2. 処理を続行します。ただし、指定したチャンネル管理情報格納ファイルのすべてに書き込み障害が発生した場合は異常終了します。</p>	障害要因を取り除きます。異常終了している場合、再度 OpenTP1 システムを正常開始してください。
終了時のファイル書き込み障害	<p>1. < 31901-E > 「ファイル異常」, < 31904-E > 「アクセス権不正」, < 31906-E > 「ファイル名不正」, < 31907-E > 「ファイル容量不足」, < 31908-E > 「データ入出力エラー」, または < 31943-E > 「OpenTP1 ファイルのアクセス失敗」</p> <p>2. 処理を続行します。ただし、指定したチャンネル管理情報格納ファイルのすべてに書き込み障害が発生した場合は異常終了します。</p>	<p>障害要因を取り除いてください。</p> <p>異常終了している場合で、正常開始するときは、いったん OpenTP1 システムを再開して、正常終了（または計画停止）したあと、正常開始してください。</p>

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

7.2.15 トリガ起動処理障害

トリガ起動処理障害の障害処理について、次の表に示します。

表 7-18 トリガ起動処理障害の障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> 転送キュー重複使用 イニシエーションキューファイル障害 トリガメッセージ不正 	<p>1. < 16315-E > 「キューが他サーバで使用」, < 16307-E > 「キューファイル障害発生」, < 16308-E > 「トリガメッセージ不正」, または < 16309-E > 「トリガ起動処理障害」</p> <p>2. トリガ起動処理を縮退します。</p> <p>3. トリガ起動処理以外は処理を続行します。</p>	各メッセージに従って障害要因を取り除き、OpenTP1 を再度、開始してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

7.2.16 MCA 数オーバ

MCA 数オーバの障害処理について、次の表に示します。

表 7-19 MCA 数オーバの障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> クラスタレシーバインスタンス追加障害 クラスタセンダ自動定義障害 	<ol style="list-style-type: none"> < 16349-W > 「MCA 数超過」 インスタンスまたは自動定義チャネルの追加を中止します。 	MQA サービス定義の MQT デーモン構成定義に指定した MCA 数を確認してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

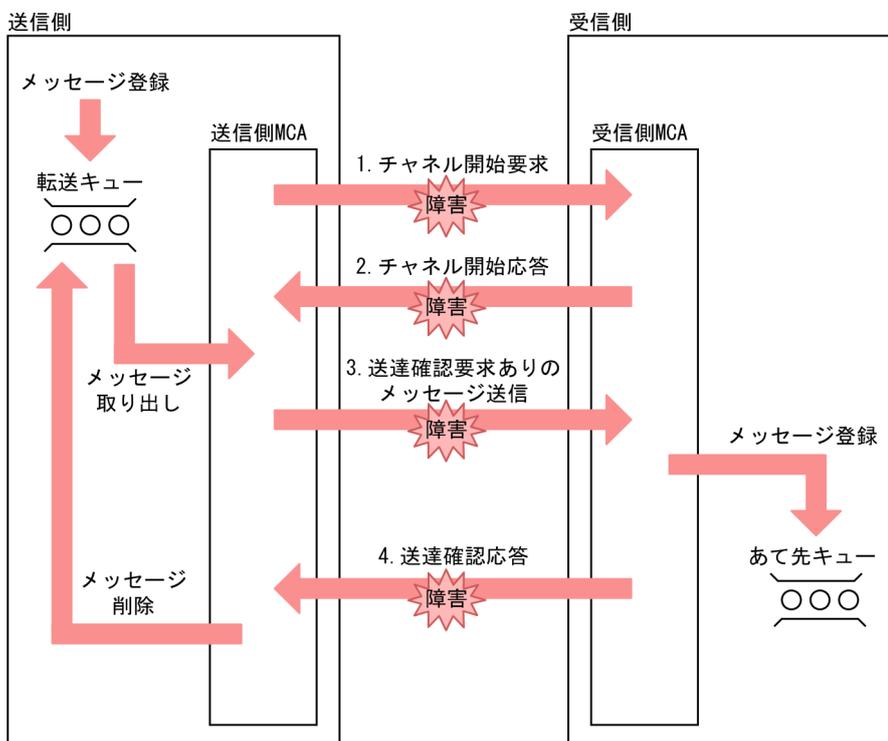
7.2.17 通信障害時のメッセージ送受信の流れ

ここでは、次の場合のメッセージの送受信で通信障害が発生したときの流れを説明します。

- チャネルがファーストメッセージ機能を使用していない。
- チャネルがファーストメッセージ機能を使用し、かつ一つ以上の通常メッセージが含まれている。

メッセージの送受信時の通信障害について次の図に示します。

図 7-15 メッセージの送受信時の通信障害



図中の 1.~4.の通信障害について次に説明します。

1. チャネル開始要求時に通信障害が発生した場合

この場合、チャネルの確立に失敗します。メッセージはそのまま送信側に残ります。

2. チャネル開始応答時に通信障害が発生した場合

この場合、チャネルの確立に失敗します。メッセージはそのまま送信側に残ります。

3. 送達確認要求ありのメッセージ送信時に通信障害が発生した場合

この場合の送信側 MCA, 受信側 MCA の処理を次に示します。

分類	処理内容
送信側 MCA	障害回復後は、LUWID*とメッセージシーケンス番号を基に、送達確認要求を送信してから、転送キュー内のメッセージを再送します。
受信側 MCA	送信側と受信側の LUWID*とメッセージシーケンス番号が一致しないため、バッチ内で受信したメッセージは無効化（削除）されます。 また、受信側の LUWID とメッセージシーケンス番号は、前回バッチ転送完了時の状態に戻され、送信側からメッセージが再送されます。 なお、メッセージシーケンス番号を一致させる運用は必要ありません。

注※

LUWID は、MCP で規定された 1 回のメッセージ転送単位（バッチ単位）を管理するための ID です。

なお、コネクション障害、またはオンラインシステムの停止などでメッセージの送信が中断された場合の、転送キュー内のメッセージの処理については、「2.4.4(3)(a) 送信側 MCA の場合」を参照してください。

4. 送達確認応答時に通信障害が発生した場合

この場合の送信側 MCA, 受信側 MCA の処理を次に示します。

分類	処理内容
送信側 MCA	障害回復後は、LUWID とメッセージシーケンス番号を基に、送達確認要求を送信します。
受信側 MCA	送信側と受信側の LUWID とメッセージシーケンス番号が一致するため、送信側では送信済みのメッセージが削除されます。 送信側から送信される次のメッセージを受信します。 なお、メッセージは有効化されているので、更新した受信側のメッセージシーケンス番号は戻しません。

7.3 ファーストメッセージ障害

チャンネルがファーストメッセージを転送する場合、転送キューから取り出したあと、相手システムのあて先キューへ登録するまでの間に、障害を検出して転送に失敗するとメッセージは破棄されます。

TP1/Message Queue のファーストメッセージの送受信の障害と、障害に対する処理について障害の種類ごとに説明します。

7.3.1 コネクション障害（コーラ側・ファーストメッセージ）

コネクション障害（コーラ側・ファーストメッセージ）の障害処理について、次の表に示します。

表 7-20 コネクション障害（コーラ側・ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
メッセージ転送時の障害	<ol style="list-style-type: none">1. < 16331-E > 「TCP インタフェースエラー」2. メッセージの転送を中断します。3. 送信中のメッセージ※1 は破棄し、受信中のメッセージ※1 も破棄します。4. チャンネル定義の確立再試行の指定によって、チャンネル確立を再試行します。5. 再試行しない場合※2 または再試行回数を越えた場合はチャンネルを終了します（チャンネル使用不可状態）。	<ul style="list-style-type: none">• 障害要因を取り除いてください。• メッセージを再登録してください。• 5.の場合は mqtstacha コマンドを入力してください。

（凡例）

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

バッチ転送によって通常メッセージを同時に転送していた場合、送信中のメッセージは転送キューに戻り、受信中のメッセージは破棄されます。

バッチ転送によって送信済みのファーストメッセージは、ロールバックされません。

注※2

非永続メッセージはすべて破棄されます。

7.3.2 コネクション障害（レスポンド側・ファーストメッセージ）

コネクション障害（レスポンド側・ファーストメッセージ）の障害処理について、次の表に示します。

表 7-21 コネクション障害（レスポンド側・ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
メッセージ転送時の障害	<ol style="list-style-type: none"> 1. < 16331-E > 「TCP インタフェースエラー」 2. メッセージの転送を中断します。 3. 送信中のメッセージ※は破棄し、受信中のメッセージ※も破棄します。 4. チャネルを終了し、再度相手システムからのコネクション確立要求を待ちます（チャネル停止状態）。 	<ul style="list-style-type: none"> • 障害要因を取り除いてください。 • メッセージを再登録してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

バッチ転送によって通常メッセージを同時に転送していた場合、送信中のメッセージは転送キューに戻り、受信中のメッセージは破棄されます。

バッチ転送によって送信済みのファーストメッセージは、ロールバックされません。

7.3.3 MCP 障害（チャネル確立後、自システムで検出した MQ プロトコル障害・ファーストメッセージ）

MCP 障害（チャネル確立後、自システムで検出した MQ プロトコル障害・ファーストメッセージ）の障害処理について、次の表に示します。

表 7-22 MCP 障害（チャネル確立後、自システムで検出した MQ プロトコル障害・ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> • メッセージ長不正 • バッチサイズ不正 • セグメント番号不正 • その他 	<ol style="list-style-type: none"> 1. < 16323-E > 「不正データ受信」または < 16324-E > 「プロトコルエラー発生」 2. メッセージの転送を中断します。 3. 送信中のメッセージ※は破棄し、受信中のメッセージ※も破棄します。 4. コネクションを強制解放します。 5. チャネルを終了します（チャネル使用不可状態）。 	<ul style="list-style-type: none"> • 保守情報を取得してください。 • メッセージを再登録してください。 • mqstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

バッチ転送によって通常メッセージを同時に転送していた場合、送信中のメッセージは転送キューに戻り、受信中のメッセージは破棄されます。

バッチ転送によって送信済みのファーストメッセージは、ロールバックされません。

7.3.4 MCP 障害（チャンネル確立後、相手システムで検出した MQ プロトコル障害・ファーストメッセージ）

MCP 障害（チャンネル確立後、相手システムで検出した MQ プロトコル障害・ファーストメッセージ）の障害処理について、次の表に示します。

表 7-23 MCP 障害（チャンネル確立後、相手システムで検出した MQ プロトコル障害・ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
<ul style="list-style-type: none"> リモートキューマネージャ状態不正 セキュリティ不正 チャンネル状態不正 	<ol style="list-style-type: none"> < 16345-E > 「エラーデータ受信」 メッセージの転送を中断します。 送信中のメッセージ※は破棄し、受信中のメッセージ※も破棄します。 コネクションを強制解放します。 コーラ側はチャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 再試行しない場合またはレスポンド側ではチャンネルを終了します（チャンネル停止状態）。 再試行回数を超えた場合はチャンネルを終了します（チャンネル使用不可状態）。 	<ul style="list-style-type: none"> ログメッセージの詳細と相手システムの状態を確認して対策してください。 メッセージを再登録してください。 mqtstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

バッチ転送によって通常メッセージを同時に転送していた場合、送信中のメッセージは転送キューに戻り、受信中のメッセージは破棄されます。

バッチ転送によって送信済みのファーストメッセージは、ロールバックされません。

7.3.5 MCP 障害（メッセージ送信時の転送障害・ファーストメッセージ）

MCP 障害（メッセージ送信時の転送障害・ファーストメッセージ）の障害処理について、次の表に示します。

表 7-24 MCP 障害（メッセージ送信時の転送障害・ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
転送キューのメッセージ不正 (チャンネルの転送サイズオーバー含む)	<ol style="list-style-type: none"> 1. < 16334-E > 「メッセージ異常検出」 2. デッドレターキューがある場合：送信中のメッセージを登録し、送信処理を続行します。 3. デッドレターキューがない場合：送信中のメッセージを破棄し、送信処理を続行します。 	原因を調査して対策したあと、メッセージを再登録してください。
メッセージシーケンス番号不一致	<ol style="list-style-type: none"> 1. < 16343-E > 「シーケンス番号不一致」 2. メッセージの送信を中断します。 3. 送信中のメッセージを破棄します。 4. コーラ側： <ol style="list-style-type: none"> (a) チャンネルを終了します。 (b) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 (c) 再試行しない場合※、または再試行回数を越えた場合はチャンネルを終了します（チャンネル使用不可状態）。 5. レスポンダ側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます（チャンネル停止状態）。 	<ul style="list-style-type: none"> • mqtrstcha コマンドを入力し、シーケンス番号を一致させてください。 • 4.のコーラ側 MCA の場合は mqtstacha コマンドを入力してください。
相手デッドレターキューにメッセージが登録された	<ol style="list-style-type: none"> 1. < 16335-W > 「デッドレターキューに登録された」 2. メッセージの送信を続行します。 	相手システムのデッドレターキューからメッセージを取り出し、原因を調査してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

7.3.6 MCP 障害（メッセージ受信時の転送障害・ファーストメッセージ）

MCP 障害（メッセージ受信時の転送障害・ファーストメッセージ）の障害処理について、次の表に示します。

表 7-25 MCP 障害（メッセージ受信時の転送障害・ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
メッセージ登録障害（あて先キューがない場合を含む）	<ol style="list-style-type: none"> 1. < 16333-E > 「キューファイル障害発生」 2. デッドレターキューがある場合：受信中のメッセージを登録し、受信処理を続行します※1。 3. デッドレターキューがない場合：受信中のメッセージを破棄し、受信処理を続行します※1。 	<ul style="list-style-type: none"> • 障害要因を取り除いてください。 • メッセージを再登録してください。
メッセージシーケンス番号不一致	<ol style="list-style-type: none"> 1. < 16343-E > 「シーケンス番号不一致」 2. 受信中のメッセージ※2 を登録します。 3. コーラ側： <ol style="list-style-type: none"> (a) チャンネルを終了します。 (b) チャンネル定義の MCP 障害再試行の指定によって、チャンネル確立を再試行します。 (c) 再試行しない場合※3，または再試行回数を越えた場合はチャンネルを終了します（チャンネル使用不可状態）。 4. レスポンダ側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます（チャンネル停止状態）。 	<ul style="list-style-type: none"> • mqtrstcha コマンドを入力してシーケンス番号を一致させてください。 • 3.のコーラ側の場合は mqtstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

チャンネルのバッファ方式がセグメント方式で、メッセージの中間セグメントおよび最終セグメントの登録で障害が発生した場合、コネクションを強制解放することがあります。

注※2

バッチ転送によって通常メッセージを同時に転送していた場合、送信中のメッセージは転送キューに戻り、受信中のメッセージは破棄されます。

バッチ転送によって送信済みのファーストメッセージは、ロールバックされません。

注※3

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

7.3.7 MCP 障害（デッドレターキュー障害・ファーストメッセージ）

MCP 障害（デッドレターキュー障害・ファーストメッセージ）の障害処理について、次の表に示します。

表 7-26 MCP 障害（デッドレターキュー障害・ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
デッドレターキュー登録障害	<ol style="list-style-type: none"> 1. < 16333-E > 「キューファイル障害発生」 2. 送信中のメッセージを破棄し、送信処理続行します※。 3. 受信中のメッセージを破棄し、受信処理続行します※。 	原因を調査して対策したあと、メッセージを再登録してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

チャンネルのバッファ方式がセグメント方式で、メッセージの中間セグメントおよび最終セグメントの登録で障害が発生した場合、コネクションを強制解放することがあります。

7.3.8 時間監視障害（ファーストメッセージ）

時間監視障害（ファーストメッセージ）の障害処理について、次の表に示します。

表 7-27 時間監視障害（ファーストメッセージ）の障害処理

障害内容	MQT の障害処理	ユーザの処理
確認メッセージ受信監視タイムアウト	<ol style="list-style-type: none"> 1. < 16329-E > 「タイムアウト発生」 2. メッセージの送信を中断します。 3. 送信中のメッセージ※1 は破棄します。 4. コネクションを強制解放します。 5. コーラ側：チャンネルを終了します※2（チャンネル使用不可状態※3）。 レスポンド側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます（チャンネル停止状態）。 	<ul style="list-style-type: none"> • 相手システムの状態と自システムのチャンネル定義の -v tim2 を見直してください。 • メッセージを再登録してください。 • 5. のコーラ側の場合は、mqstacha コマンドを入力してください。
継続メッセージ（セグメント）受信監視タイムアウト	<ol style="list-style-type: none"> 1. < 16329-E > 「タイムアウト発生」 2. メッセージの受信を中断します。 3. 受信中のメッセージ※1 を破棄します。 4. コネクションを強制解放します。 5. コーラ側：チャンネルを終了します※2（チャンネル使用不可状態）。 レスポンド側：チャンネルを終了し、再度相手システムからのコネクション確立要求を待ちます（チャンネル停止状態）。 	<ul style="list-style-type: none"> • 相手システムの状態と自システムのチャンネル定義の -v tim3, mtim を見直してください。 • メッセージを再登録してください。 • 5. のコーラ側の場合は、mqstacha コマンドを入力してください。

障害内容	MQT の障害処理	ユーザの処理
終了処理監視タイムアウト	<ol style="list-style-type: none"> 1. < 16312-E > 「タイムアウト発生」 2. メッセージの送受信を中断します。 3. 送信中のメッセージ※1 は破棄し、受信中のメッセージ※1 は破棄します。 4. コネクションを強制解放します。 5. 全チャンネルを終了します。 6. MQT を強制停止します。 	<ul style="list-style-type: none"> • TCP 構成定義の-v etim を見直してください。 • メッセージを再登録してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※1

バッチ転送によって通常メッセージを同時に転送していた場合、送信中のメッセージは転送キューに戻り、受信中のメッセージは破棄されます。

バッチ転送によって送信済みのファーストメッセージは、ロールバックされません。

注※2

チャンネル定義のタイムアウト時のチャンネル確立再試行の要否指定 (TCP 定義の mqtalcccha 定義コマンドの-v vretry オペランド指定) に yes を指定した場合は、チャンネル確立再試行の指定 (-b オプション) に従ってチャンネル確立再試行を実行します。

注※3

クラスタチャンネルの場合は、チャンネル状態は「チャンネル停止」になります。

7.3.9 出口障害 (ファーストメッセージ)

出口障害 (ファーストメッセージ) の障害処理について、次の表に示します。

表 7-28 出口障害 (ファーストメッセージ) の障害処理

障害内容	MQT の障害処理	ユーザの処理
応答パラメタ不正	<ol style="list-style-type: none"> 1. < 16347-E > 「UOC 応答パラメタ不正」 2. メッセージの送受信を中断します。 3. 送信中のメッセージ※は破棄し、受信中のメッセージ※は破棄します。 4. コネクションを強制解放します。 5. チャンネルを終了します (チャンネル使用不可状態)。 	<ul style="list-style-type: none"> • 出口の処理を見直してください。 • メッセージを再登録してください。 • 5.の場合は、mqstacha コマンドを入力してください。

(凡例)

< > : 出力されるメッセージログ ID です。メッセージログ ID には、先頭に KFCA が付きます。

[] : メッセージログの内容です。

注※

バッチ転送によって通常メッセージを同時に転送していた場合、送信中のメッセージは転送キューに戻り、受信中のメッセージは破棄されます。

バッチ転送によって送信済みのファーストメッセージは、ロールバックされません。

7.3.10 OpenTP1 システム, および MQ システム障害 (ファーストメッセージ)

OpenTP1 システム, および MQ システム障害 (ファーストメッセージ) の障害処理について, 次の表に示します。

表 7-29 OpenTP1 システム, および MQ システム障害 (ファーストメッセージ) の障害処理

障害内容	MQT の障害処理	ユーザの処理
OpenTP1 異常終了	<ol style="list-style-type: none">1. OS によるコアダンプを出力します。2. メッセージ送受信を中断します。3. 転送中のファーストメッセージは破棄します※。	障害要因を取り除いたあと OpenTP1 システムを再開してください。

注※

クラスタチャンネルの場合は, チャンネル状態は「チャンネル停止」になります。

7.4 障害時に取得する情報

TP1/Message Queue が障害時に取得する情報について、次の表に示します。

表 7-30 障害時に取得する情報

取得情報	出力情報	参照方法	注意事項
メッセージログファイル (\$DCDIR/spool/dcllog1, dcllog2)	OpenTP1, または UAP が出力するメッセージ	logcat コマンドを実行してください。	古いメッセージログファイルを残しておきたい場合は、メッセージログファイルの交替で内容が失われる前にコピーしてください。
コンソールメッセージ	OpenTP1 が出力するシステム情報	コンソールを見てください。	プリンタ出力, 集中コンソールへの出力は OS の機能です。
コマンド実行時の標準出力, 標準エラー出力	OpenTP1 の運用コマンドの出力情報, およびエラーメッセージ	運用コマンドを入力した画面で見てください。	ファイルに残したい場合は, 必要に応じてリダイレクトしてください。
コアファイル (./core)	OpenTP1 の関連プロセスのデータ, スタック	—	OpenTP1 が起動するプロセスのコアファイルは, \$DCDIR/spool/save ディレクトリ下にサーバ名 n ^{※1} という名称で退避されます。一つのサーバについて 3 個までしか退避されないため, 残しておきたいコアファイルは必要に応じてコピーしてください。
	UAP トレース	uatdump コマンドを実行してください。	
	MQC リスナサーバの内部情報のプロセスなど	—	
共用メモリダンプファイル (\$DCDIR/spool/shmdump)	OpenTP1 が共用メモリに保持するデータ	—	古いダンプファイルを残しておきたい場合は, 必要に応じてコピーしてください。
MQA サービス専用共用メモリダンプファイル (\$DCDIR/spool/shmdump.mXX ^{※2})	MQA サービスが共用メモリに保持するデータ	—	古いダンプファイルを残しておきたい場合は, 必要に応じてコピーしてください。
MQT サービスローカルメモリダンプファイル \$DCDIR/spool/mqtdmpXX ^{※2} _YY...YY ^{※3}	MQT スナップダンプファイル	—	出力したダンプファイルは削除されません。不要なファイルは OS の削除コマンドを使用して削除してください
MQT トレースファイル	MQT トレース	—	古い MQT トレースファイルを残しておきたい場合は, 必要に応じてコピーしてください。

取得情報	出力情報	参照方法	注意事項
チャンネル管理情報格納ファイル	チャンネル管理情報	—	古いチャンネル管理情報格納ファイルを残しておきたい場合は、必要に応じてコピーしてください。
MQR トレースファイル	MQR トレース	—	古い MQR トレースファイルを残しておきたい場合は、必要に応じてコピーしてください。
キューファイル	ユーザデータ	—	—
ジャーナルファイル	ジャーナルデータ	—	—
MQC トレースファイル (<code>\$DCDIR/spool/mqcXXXYYZZZZZ</code> ※ 4)	MQC トレース情報など	—	残しておきたい場合は、必要に応じてコピーしてください。

(凡例)

—：該当しません。

注※1

n：退避コアファイルの通番（1～3）です。

ただし、OpenTP1 が強制停止時（`dcsvstop -f` コマンドを実行した時、または実時間監視タイムアウトになった時）に出力されるコアファイルには、通番は付きません。なお、プロセスサービスのコアファイルが"core"という名称でこのディレクトリに取得されることがあります。また、ユーザ環境設定コマンドが異常終了した場合、そのコアファイルは"_usrCmd"に通番（1～3）が付いた名称で退避されます。

注※2

XX：MQT 通信プロセス識別子

注※3

YY...YY：MQT 通信プロセス開始からの通算秒（10 進数）

注※4

XXXYYZZZZZ の意味を次に示します。

XXX：サーバ種別

lsn：MQC リスナサーバ

gwp：MQC ゲートウェイサーバ

YYY：MQC ゲートウェイサーバ識別子（000～255 の 10 進数）

ZZZZZ：00001～00003

7.4.1 MQT トレースファイルの出力

MQT サーバはプロセスの動作状態を MQT トレース情報として取得しています。

MQT トレース情報を MQT トレースファイルへ出力する方法について説明します。

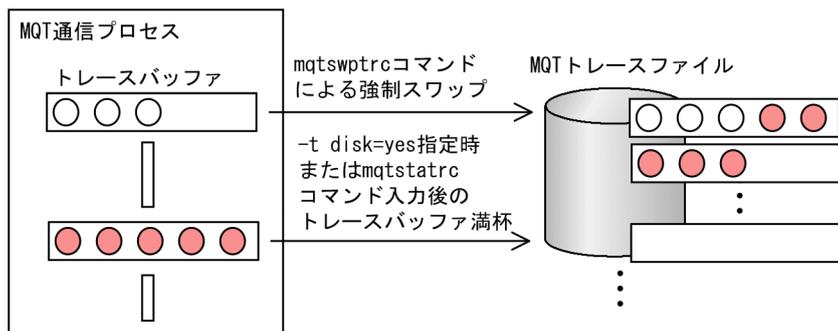
(1) トレース方式の概要

MQT トレース情報はMQT サーバの各通信プロセスのローカルメモリ上でトレースバッファに一定量が取得され、満杯になった時にMQT トレースファイルとしてディスクに出力されます。この情報はMQT サーバに障害が発生した場合に、障害情報として必要になります。出力先ディレクトリのディスク容量に余裕がない場合を除いて、常にMQT トレースファイルに出力しておくことをお勧めします。MQT トレースファイルに出力する場合は、`mqttcp` 定義コマンド、`mqttcps` 定義コマンド、または `mqttcpcr` 定義コマンドの `-t disk` オペランドに `yes` を指定してください。または、オンライン中に `mqttstatrc` コマンドを入力してMQT トレースのディスク出力機能を開始します。

特に、オンライン中にMQT トレースファイルを出力する場合はトレースバッファ上に蓄えているMQT トレース情報を、MQT トレースファイルに強制スワップできます。`mqttswptrc` コマンドを使用して強制スワップしてください。`mqttswptrc` コマンドを使用すると定義コマンドの `-t disk` オペランドに `yes` を指定していない場合でも、メモリ上のMQT トレース情報をMQT トレースファイルに出力できます。ただし、この場合は取得できる情報が少ないため、障害情報として不十分になることがあります。`mqttswptrc` コマンドについては、6章の「`mqttswptrc` (MQT トレースの強制スワップ)」を参照してください。

MQT トレースファイルの出力について、次の図に示します。

図 7-16 MQT トレースファイルの出力



(凡例)

○ : MQT トレース情報

(2) ファイル名とその作成順序

MQT トレースファイルは、MQT サービス定義の `mqtt_trace_file_path` オペランドで指定されたディレクトリに作成されます。トレースファイル名は、"`mqtttrcXXYY`" (XX : MQT 通信プロセス識別子, YY : トレーススワップファイル識別子) です。トレーススワップファイル識別子はその最大値を、TCP 定義の `mqttcp`, `mqttcps`, または `mqttcpcr` 定義コマンドの `-t trcnt` オペランドに指定します。MQT サーバを開始して最大値までファイルを作成すると、ラウンドロビン方式で再度 1 番から上書きしていきます。

(a) オンライン開始後の最初の出力

オンライン開始後、最初に書き込まれるトレースファイルは、`mqtt_trace_file_path` オペランドで指定されたディレクトリにあるトレースファイルのうち、最終更新時刻が最も新しいファイル通番の次の通番の

ファイルになります。ただし、ファイル通番に抜けが存在する場合は、抜けている通番のファイルがオンライン開始後、最初書き込まれるトレースファイルになります。

トレースファイルを書き込む際、`mqt_trace_file_path` オペランドで指定されたディレクトリにあるトレースファイルの更新日付の確認でエラーが発生した場合、オンライン開始後、最初書き込まれるトレースファイルのファイル通番は 1 になります。このエラーが発生した場合は、`KFCA31185-W` がメッセージログファイルに出力され、トレースファイルの出力が継続されます。

(b) オンライン開始後の 2 回目以降の出力

オンライン開始後、2 回目以降に出力されるトレースファイルは、ファイル通番に 1 が加算されたファイルに書き込まれます。最大通番 (TCP 定義の `mqttcp`, `mqttcpcs`, または `mqttcpcr` 定義コマンドの `-t trcnt` オペランドの指定値) まで達した場合は、ラウンドロビン方式で再度 1 番から上書きしていきます。

(3) 取得するトレースの内容

`mqtcd` コマンドを入力すると、MQT トレースファイルの内容が標準出力に出力されます。

なお、MQT サーバで障害が発生し保守員に調査を依頼する場合、MQT トレースファイルが必要になります。障害発生時の MQT トレースファイルを保存することをお勧めします。

7.4.2 MQR トレースファイルの出力

リポジトリ管理サーバは、パーシャルリポジトリとフルリポジトリ間で発生するクラスタについてのメッセージ送受信を MQR トレース情報として取得しています。

MQR トレース情報を MQR トレースファイルへ出力する方法について説明します。

(1) トレース方式の概要

MQR トレース情報は、パーシャルリポジトリとフルリポジトリ間でクラスタについてのメッセージ送受信が発生するたびに、共用メモリ上に一定量が取得され、満杯になった時、または `mqr sup` の正常終了時に MQR トレースファイルに出力されます。この情報はパーシャルリポジトリとフルリポジトリ間の連携に障害が発生した場合に、障害情報として必要になります。

また、`mqr sup` 動作中に MQR トレースファイルを出力する場合はメモリ上に蓄えている MQR トレース情報を MQR トレースファイルに強制スワップできます。`mqr swptrc` コマンドを使用して強制スワップしてください。

MQR トレースファイルを出力しない場合は、リポジトリ管理サーバユーザサービス定義ファイルの `DCMQA_TRACE_STATUS` オペランドに `N` を指定してください。

リポジトリ管理サーバユーザサービス定義ファイルの詳細については、「[2.8.11\(4\) リポジトリ管理サーバの環境変数](#)」を参照してください。

(2) ファイル名とその作成順序

MQR トレースファイルは、DCMQA_TRACE_PATH オペランドで指定されたディレクトリに作成されます。トレースファイル名は、"mqrtrcX" (X: 1~DCMQA_TRACE_FILE_NUM オペランドの指定値) です。リポジトリ管理サーバを開始して最大値までファイルを作成すると、ラウンドロビン方式で再度 1 番から上書きしていきます。

(a) オンライン開始後の最初の出力

オンライン開始後、最初に書き込まれるトレースファイルは、DCMQA_TRACE_PATH オペランドで指定されたディレクトリにあるトレースファイルのうち、最終更新時刻が最も新しいファイル通番の次の通番のファイルになります。ただし、ファイル通番に抜けが存在する場合は、抜けている通番のファイルがオンライン開始後、最初に書き込まれるトレースファイルになります。

トレースファイルに書き込む際、DCMQA_TRACE_PATH オペランドで指定されたディレクトリに存在するトレースファイルの更新日付の確認でエラーが発生した場合、オンライン開始後、最初に書き込まれるトレースファイルのファイル通番は 1 になります。このエラーが発生した場合は、KFCA31185-W がメッセージログファイルに出力され、トレースファイルの出力が継続されます。

(b) オンライン開始後の 2 回目以降の出力

オンライン開始後、2 回目以降に出力されるトレースファイルは、ファイル通番に 1 が加算されたファイルに書き込まれます。最大通番 (DCMQA_TRACE_FILE_NUM オペランドの指定値) まで達した場合は、ラウンドロビン方式で再度 1 番から上書きしていきます。

(3) 取得するトレースの内容

取得するトレースの内容は公開しません。障害発生時に MQR トレースファイルを取得したら、保守員に連絡してください。

(4) 出力失敗時の動作

MQR トレースファイルへ MQR トレース情報を出力する時に、満杯以外のエラーが発生した場合は、機能を停止します。再び使用する場合はリポジトリ管理サーバを再起動してください。

付録

付録 A 共用メモリの見積もり式

ここでは共用メモリの見積もり式について説明します。

付録 A.1 共用メモリを見積もる際の Σ の計算例および注意事項

共用メモリを見積もる際の Σ の計算例， および見積もる際の注意事項について次に示します。

(1) Σ の計算例

MQA サーバ用の静的共用メモリの見積もり式の一部を使用して， Σ の計算例を次に示します。

$$(\Sigma_{i=1}^b c_i) + 1$$

システムに存在する MQT デーモン構成定義に指定した MCA 数を計算することを示します。

$$(\Sigma_{i=1}^b j_i) \times 8$$

システムに存在する MQT デーモン構成定義に設定した総バッチメッセージ数を計算することを示します。

(凡例)

b : mqamqtnam 定義コマンドの定義数 (MQT デーモン数)

c : 該当 MQT プロセスの MQT デーモン構成定義 (MQA サービス定義) に指定した MCA 数

j : mqamqtnam 定義コマンドの -b オプション (総バッチメッセージ数) の指定値

(2) 共用メモリを見積もる際の注意事項

共用メモリを見積もる際は， 次の点に注意してください。

- メモリ所要量とは仮想メモリの値を示します。
- MQA サービス専用共用メモリの固定領域は， mqaquegrp 定義コマンドの -s オプションの定義ごとに複数確保します。領域ごとのサイズは， 次の構成単位で最大 2147483583 バイトになります。
 - mqaquegrp 定義コマンドの -s オプションを指定しないキューファイルグループに対応する領域
 - mqaquegrp 定義コマンドの -s オプションで MQA サービス専用共用メモリ確保要求者識別子を指定した場合は， 異なる識別子ごとの領域
- MQA サービス専用共用メモリ拡張領域で確保できる最大サイズは， 2000000000 バイトです。
- MQA サーバ用の OpenTP1 共用メモリ (静的共用メモリおよび動的共用メモリ) と MQA サービス専用共用メモリの合計値は， OS の共用メモリセグメントサイズの上限を超えないようにしてください。なお， SUP などのアプリケーションで共用メモリセグメントを確保する場合も， この制限が適用されません。OS の共用メモリセグメントサイズの上限については， OS のマニュアルを参照してください。

- HP-UX 版の 32 ビット版の OpenTP1 を使用する場合は、次の点に注意してください。

【MQA サービス専用共用メモリについて】

- MQA サービス専用共用メモリの固定領域：最大サイズの 2147483583 バイトを 1063256064 バイトに読み替えてください。
- MQA サービス専用共用メモリの拡張領域：最大サイズの 20000000000 バイトを 1063256064 バイトに読み替えてください。

32 ビット版の HP-UX では、共用メモリセグメントサイズの上限値は 1 ギガバイトです。

OpenTP1 は、指定したサイズに、内部的に必要とするサイズ（最大 10 メガバイト）を加算して共用メモリの確保要求を行います。このため、1063256064 バイトを超えた値を指定した場合、確保要求を行うサイズが 1 ギガバイトを超えてしまい、OpenTP1 システムの起動に失敗することがあります。

【システム全体で使用できる共用メモリについて】

システム全体で使用できる共用メモリのサイズは、次のとおりです。

HP-UX (IPF) の場合：2 ギガバイト

マルチ OpenTP1 での運用や、共用メモリを確保する他のプログラムを同一マシン上で実行する場合は、共用メモリの合計サイズに注意してください。

付録 A.2 MQA サーバ用の共用メモリの見積もり式

MQA サーバ用の共用メモリの見積もりを次の表に示します。

表 A-1 MQA サーバ用の共用メモリの見積もり

領域	メモリ所要量 (単位：バイト)
静的共用メモリ	32 ビット版の場合： $7516 + 640 \times a + 96 \times b$ $+ (((\sum_{i=1}^b c_i) + 1) \times 568 + 32) \times 2)$ $+ [((\sum_{i=1}^b j_i) \times 8) \times 2] \times 1$ $+ 1232 \times d + 512 \times f + 224 \times g$ 64 ビット版の場合： $8320 + 640 \times a + 112 \times b$ $+ (((\sum_{i=1}^b c_i) + 1) \times 576 + 32) \times 2)$ $+ [((\sum_{i=1}^b j_i) \times 16) \times 2] \times 1$ $+ 1344 \times d + 624 \times f + 288 \times g$
動的共用メモリ	32 ビット版の場合： $512 + 20480 \times q + 512 \times \uparrow q / 5 \uparrow + 1536 \times \uparrow h / 5 \uparrow$ $+ 1536 \times \uparrow v / 5 \uparrow + 512 \times W$ $+ 80384 \times (\sum_{i=1}^p \uparrow w_i / 64 \uparrow$ $+ \sum_{ii=1}^b \uparrow (\sum_{i=1}^{kii} (o_i + 1) + l_{ij} + 2) / 64 \uparrow)$

領域	メモリ所要量 (単位：バイト)
動的共用メモリ	$+ \uparrow (96 + (224 \times \uparrow Z / 5 \uparrow)) / 512 \uparrow \times 512$ 64ビット版の場合： $512 + 20480 \times q + 512 \times \uparrow q / 5 \uparrow + 2560 \times \uparrow h / 5 \uparrow$ $+ 2048 \times \uparrow v / 5 \uparrow + 512 \times W$ $+ 90624 \times (\sum_{i=1}^p \uparrow w_i / 64 \uparrow + \sum_{ii=1}^b \uparrow (\sum_{kii=1}^{kii} (o_i + 1) + l_{ii} + 2) / 64 \uparrow)$ $+ \uparrow (96 + (256 \times \uparrow Z / 5 \uparrow)) / 512 \uparrow \times 512$
MQA サービス専用の共用メモリの固定領域	$\uparrow \{ \uparrow (2048 + \sum_{i=1}^q ((\uparrow (16 + 192 \times n_i) / 32 \uparrow$ $+ [\uparrow (32 + 304 \times \uparrow (n_i \times r_i / 100) \uparrow) / 32 \uparrow]^{*2}$ $+ \uparrow (32 + 4 \times n_i) / 32 \uparrow$ $+ \uparrow (16 \times s_i + 32 \times n_i + 176 + m_i \times s_i) / 32 \uparrow) \times 32) \}$ $/ 1024 \uparrow \times 1024 \uparrow / 1024 \uparrow \times 1024$
MQA サービス専用の共用メモリの拡張領域	$\uparrow \{ \uparrow (2048 + \sum_{i=1}^Z ((\uparrow (16 + 192 \times n_i) / 32 \uparrow$ $+ [\uparrow (32 + 304 \times \uparrow (n_i \times r_i / 100) \uparrow) / 32 \uparrow]^{*2}$ $+ \uparrow (32 + 4 \times n_i) / 32 \uparrow$ $+ \uparrow (16 \times s_i + 32 \times n_i + 176 + m_i \times s_i) / 32 \uparrow) \times 32) \}$ $/ 1024 \uparrow \times 1024 \uparrow / 1024 \uparrow \times 1024$

(説明)

静的共用メモリ

OpenTP1 のシステムサーバとそのシステムサーバが使用する静的共用メモリのサイズに MQA サーバ用の静的共用メモリのサイズを加算し、その合計値をシステム環境定義の static_shmpool_size オペランドに指定してください。

動的共用メモリ

OpenTP1 のシステムサーバとそのシステムサーバが使用する動的共用メモリのサイズに MQA サーバ用の動的共用メモリのサイズを加算し、その合計値をシステム環境定義の dynamic_shmpool_size オペランドに指定してください。

MQA サービス専用の共用メモリの固定領域

MQA サービス専用の共用メモリの固定領域を指定するオペランドはありません。この領域は TP1/Message Queue の起動時に確保されます。

この計算式では、q の値を次に示すように設定してください。

1. mqaquegrp 定義コマンドの -s オプションに指定したキューファイルグループで使用する領域を計算する場合 (異なる MQA サービス専用共用メモリ確保要求者識別子を指定した mqaquegrp 定義コマンドが複数存在する場合は、異なる MQA サービス専用共用メモリ確保要求者識別子ごとに計算する必要があります)

q：同一の MQA サービス専用共用メモリ確保要求者識別子を指定した mqaquegrp 定義コマンドのうち、指定したキューファイルの数（二重化構成では、バックアップキューファイルを除くキューファイル数）

2. mqaquegrp 定義コマンドの -s オプションに指定しないキューファイルグループ全体で使用する領域を計算する場合

q：MQA サービス専用共用メモリ確保要求者識別子を指定しない mqaquegrp 定義コマンドのうち、指定したキューファイルの数（二重化構成では、バックアップキューファイルを除くキューファイル数）

上記の 1. および 2. で計算した値の合計が MQA サービス専用の共用メモリ固定領域全体の所要量です。

MQA サービス専用の共用メモリの拡張領域

計算式で見積もったサイズを MQA サービス定義の mqa_extend_rmshm_size オペランドに指定してください。

(凡例)

↑↑：小数点以下を切り上げます。

a：プロセス定義数（MQA サービス定義の mqaprcdef 定義コマンドの定義数）

b：mqamqtnam 定義コマンドの定義数（MQT デーモン数）

c：該当 MQT プロセスの MQT デーモン構成定義（MQA サービス定義）に指定した MCA 数

d：mqa_max_quenum オペランドの指定値

f：mqa_max_filnum オペランドの指定値

g：mqa_max_grpnum オペランドの指定値

h：同時走行トランザクション数。同時に走行している UAP で発生されるトランザクション数です。具体的には、トランザクション内で TP1/Message Queue を使用している UAP が同時に走行する最大値を指定してください。

j：mqamqtnam 定義コマンドの -b オプション（総バッチメッセージ数）の指定値

k：レシーバ、クラスタレシーバ、またはリクエストのチャンネル数

l：センダ、クラスタセンダ、またはサーバのチャンネル数

m：キューファイルのレコード長

n：mqainit コマンドの -m オプションの指定値（メッセージ数）

o：mqtalccha 定義コマンドの -j オプションの指定値（バッチサイズ）

p：同時走行プロセス数。TP1/Message Queue を使用する UAP の同時走行数です。具体的には、トランザクションの内外に関係なく TP1/Message Queue を使用している UAP が同時に走行する最大値を指定してください。

q：キューファイル数（二重化構成では、バックアップキューファイルを除くキューファイル数）

r：mqaquegrp 定義コマンドの -m オプションの指定値（メッセージエントリ確保率）

s：mqaquegrp 定義コマンドの -n オプションの指定値（入出力バッファ数）

v：同時走行トランザクションブランチ数。同時に走行している UAP で発生されるトランザクション数とトランザクション内で RPC コールされた UAP の数です。大小関係で表すと次のようになります。

「同時実行トランザクション数」 ≤ 「同時実行トランザクションブランチ数」 ≤ 「同時実行プロセス数」

w：一つのプロセスで使用するオブジェクトハンドル数

W：システム全体で同時に使用する UAP プロセスのオブジェクトハンドル数

Z：mqaadd コマンドでオンライン中に組み込むキューファイル数（二重化構成では、バックアップキューファイルを除くキューファイル数）

注※1

mquamqtnam 定義コマンド（MQT デーモン数）を定義しない場合は、[] 内は 0 バイトとなります。

注※2

mqaquegrp 定義コマンドの -m オプションの指定値（メッセージエントリ確保率）が 0 の場合は、[] 内は 0 バイトとして計算します。

付録 A.3 リポジトリ管理機能の共用メモリの見積もり式

リポジトリ管理機能の共用メモリの見積もりを次の表に示します。MQA サービス定義の mqa_mqr_conf オペランドに Y を設定した場合は、表 A-1 で計算した値に、次の表で計算した値を加えてください。

表 A-2 リポジトリ管理機能の共用メモリの見積もり

領域	メモリ所要量 (単位：バイト)
静的共用メモリ	32 ビット版の場合： 832 64 ビット版の場合： 1152
動的共用メモリ	32 ビット版の場合： $\begin{aligned} & ((\uparrow (27264 \times \uparrow (D + E) / 10 \uparrow) / 512 \uparrow) \\ & + (\uparrow (27264 \times \uparrow A / 10 \uparrow) / 512 \uparrow) \\ & + (\uparrow (7424 \times \uparrow F / 10 \uparrow) / 512 \uparrow) \\ & + (\uparrow (7424 \times \uparrow G / 10 \uparrow) / 512 \uparrow) \\ & + (\uparrow (6144 \times \uparrow (H + \sum_{i=1}^S M) / 10 \uparrow) / 512 \uparrow) \\ & + \sum_{i=1}^S (\uparrow (6144 \times \uparrow (K \times 2 + T) / 10 \uparrow) / 512 \uparrow) \\ & + \sum_{i=1}^S (\uparrow (6144 \times \uparrow J / 10 \uparrow) / 512 \uparrow) \\ & + (\uparrow (1664 \times \uparrow K / 10 \uparrow) / 512 \uparrow) \\ & + (\uparrow (1664 \times \uparrow J / 10 \uparrow) / 512 \uparrow) \\ & + \sum_{i=1}^S (\uparrow (1664 \times \uparrow M / 10 \uparrow) / 512 \uparrow) \text{ ※1} \\ & + (\uparrow (384 \times \uparrow L / 5 \uparrow) / 512 \uparrow) \\ & + (\uparrow (96 + N) / 512 \uparrow) \\ & + (\uparrow (64 + 64 \times O) / 512 \uparrow) \\ & + (\uparrow P / 512 \uparrow) + (5 + 4 \times U) \times 512 \end{aligned}$

領域	メモリ所要量 (単位：バイト)
動的共用メモリ	$+ 1024 \times S + 80384 \times (1 + U)$ 64ビット版の場合： $((\uparrow (27776 \times \uparrow (D + E) / 10 \uparrow) / 512 \uparrow)$ $+ (\uparrow (27776 \times \uparrow A / 10 \uparrow) / 512 \uparrow)$ $+ (\uparrow (7776 \times \uparrow F / 10 \uparrow) / 512 \uparrow)$ $+ (\uparrow (7776 \times \uparrow G / 10 \uparrow) / 512 \uparrow)$ $+ (\uparrow (6656 \times \uparrow (H + \sum_{i=1}^S M) / 10 \uparrow) / 512 \uparrow)$ $+ \sum_{i=1}^S (\uparrow (6656 \times \uparrow (K \times 2 + T) / 10 \uparrow) / 512 \uparrow)$ $+ \sum_{i=1}^S (\uparrow (6656 \times \uparrow J / 10 \uparrow) / 512 \uparrow)$ $+ (\uparrow (2336 \times \uparrow K / 10 \uparrow) / 512 \uparrow)$ $+ (\uparrow (2336 \times \uparrow J / 10 \uparrow) / 512 \uparrow)$ $+ \sum_{i=1}^S (\uparrow (2336 \times \uparrow M / 10 \uparrow) / 512 \uparrow) \text{ ※1}$ $+ (\uparrow (576 \times \uparrow L / 5 \uparrow) / 512 \uparrow)$ $+ (\uparrow (112 + N) / 512 \uparrow)$ $+ (\uparrow (64 + 80 \times O) / 512 \uparrow)$ $+ (\uparrow P / 512 \uparrow) + (5 + 4 \times U) \times 512$ $+ 1024 \times S + 90624 \times (1 + U)$

(説明)

静的共用メモリ

OpenTP1 のシステムサーバとそのシステムサーバが使用する静的共用メモリのサイズに加算して、その合計値をシステム環境定義の static_shmpool_size オペランドに指定してください。

動的共用メモリ

OpenTP1 のシステムサーバとそのシステムサーバが使用する動的共用メモリの最大使用時のサイズに加算して、その合計値をシステム環境定義の dynamic_shmpool_size オペランドに指定してください。

(凡例)

- ↑↑：小数点以下を切り上げます。↑↑で計算した結果が0になる場合は、1として計算してください。
- A：クラスタ内にあるリモートのクラスタレシーバチャンネル数（90日以内に削除されたチャンネル数を含みます）※2
- D：事前定義クラスタセンダチャンネル数（90日以内に削除されたチャンネル数を含みます）
- E：事前定義クラスタレシーバチャンネル数（90日以内に削除されたチャンネル数を含みます）
- F：クラスタ内にあるリモートのクラスタキュー数（90日以内に削除されたキュー数を含みます）※2
- G：ローカルのクラスタキュー数（90日以内に削除されたキュー数を含みます）
- H：参加するフルリポジトリマネージャ数（フルリポジトリ数が1の場合は1、フルリポジトリ数が2以上の場合は2）複数のクラスタに参加する場合はクラスタごとに算出した総和。

J: クラスタキュー名の数 (ローカルおよびリモートを合わせてクラスタ内で重複する名前は数えませんが)。アプリケーションで、クラスタ内にないキュー名を、キューマネージャ名を指定しないで MQOPEN 命令で指定した場合のキュー名の数も加えてください。

K: リモートのキューマネージャ名の数

L: 同時走行クラスタアクセスプロセス数

M: クラスタ内にあるフルリポジトリマネージャ数

N: トレースファイルの最大長 (リポジトリ管理サーバのユーザサービス定義ファイルの DCMQA_TRACE_MAX_LEN オペランドの指定値)

O: クラスセンダプロセスの mqamqtnam 定義コマンドの MCA 数

P: $O \geq A$ の場合, $P = 0$

$O < A$ の場合, $P = 384 \times \uparrow (O - A) / 5 \uparrow$

S: 参加するクラスタ数

T: 30 日以内に削除されたりリモートのキューマネージャ数^{※1}

U: mqrsp ervice定義の parallel_count オペランドの最大プロセス数

注※1

複数のクラスタに参加する場合はクラスタごとに算出した総和を示します。

注※2

キューマネージャを再作成 (リポジトリキューを再作成) した場合、再作成前のチャンネル、キューマネージャ、キュー情報は再作成後の情報とは別にカウントしてください。

付録 A.4 MQC サーバ機能の共用メモリの見積もり式

MQC サーバ機能の共用メモリの見積もりを次の表に示します。MQA サービス定義で mqa_mqc_conf オペランドに Y を設定した場合は、表 A-1 で計算した値に、次の表で計算した値を加えてください。

表 A-3 MQC サーバ機能の共用メモリの見積もり

領域	メモリ所要量 (単位: バイト)
静的共用メモリ (MQA サーバ)	32 ビット版の場合: $352 \times g \times 2$ 64 ビット版の場合: $416 \times g \times 2$
静的共用メモリ (MQC サーバ)	32 ビット版の場合: $352 + 480 \times n + 64 \times g$ 64 ビット版の場合: $432 + 576 \times n + 96 \times g$

領域	メモリ所要量 (単位：バイト)
動的共用メモリ	32ビット版の場合： $(\uparrow (352 \times c) / 512 \uparrow + \uparrow (r + 16) / 512 \uparrow) \times 512$ 64ビット版の場合： $(\uparrow (416 \times c) / 512 \uparrow + \uparrow (r + 16) / 512 \uparrow) \times 512$

(説明)

静的共用メモリ

OpenTP1 のシステムサーバとそのシステムサーバが使用する静的共用メモリのサイズに加算して、その合計値をシステム環境定義の `static_shmpool_size` オペランドに指定してください。

動的共用メモリ

OpenTP1 のシステムサーバとそのシステムサーバが使用する動的共用メモリのサイズに加算して、その合計値をシステム環境定義の `dynamic_shmpool_size` オペランドに指定してください。

(凡例)

g：オンライン中に同時に実行できるトランザクションの最大数 (MQA サービス定義の `mqa_mqc_trnnum` オペランドの値)

n：MQC ゲートウェイサーバ数 (MQC サービス定義に指定した `mqcgwpsnam` 定義コマンド数)

c：オンライン中に同時に処理できるクライアントの最大数 (MQA サービス定義の `mqa_mqc_clientnum` オペランドの値)

r：`dcrasget` コマンドで指定する「取得先ディレクトリ」の文字列の長さ (単位：バイト)

付録 A.5 MQT マネジャサーバの共用メモリの見積もり式

MQT マネジャサーバの共用メモリの見積もりを次の表に示します。

表 A-4 MQT マネジャサーバの共用メモリの見積もり

領域	メモリ所要量 (単位：バイト)
静的共用メモリ	32ビット版の場合： $(\uparrow (384 + \uparrow (48 \times b) / 1024 \uparrow \times 1024 + u) / 32 \uparrow + 1) \times 32 + 864^{**}$ 64ビット版の場合： $(\uparrow (384 + \uparrow (48 \times b) / 1024 \uparrow \times 1024 + u) / 32 \uparrow + 1) \times 32 + 896^{**}$

(凡例)

b：`mquamqtnam` 定義コマンドの定義数 (MQT デーモン数)

u：

- 使用するチャンネル管理情報ファイルが通常ファイルの場合

↑ (32×前回オンライン時のmqamqtnam定義コマンドの定義数) /1024↑×1024

- 使用するチャンネル管理情報ファイルが OpenTP1 ファイルの場合

↑ (↑ (32×前回オンライン時のmqamqtnam定義コマンドの定義数) /1024↑×1024)
/OpenTP1ファイルシステムのセクタ長↑×OpenTP1ファイルシステムのセクタ長

注※

MQA サービス定義の set mqa_channel_inf_file_name_a オペランドと set mqa_channel_inf_file_name_b オペランドの指定を同時に省略した場合、32 ビット版は 864 バイト、64 ビット版は 896 バイトになります。

付録 B ファイルの見積もり式

OpenTP1 ファイルの見積もり式について説明します。

付録 B.1 ステータスファイルのサイズの見積もり式

TP1/Message Queue では、TP1/Server Base のステータスファイルを使用します。

ステータスファイルのサイズの見積もり式を次に示します。

表 B-1 ステータスファイルの使用容量とキー数 (MQA サーバ)

サービス	1 キー当たりのステータスファイルの使用容量 (単位：バイト)	キー数
TP1/Message Queue MQA サーバ	288	1
	$48 \times a$	1
	256	1
	$640 \times b$	1
	$64 \times c$	1
	$160 \times d$	1
	$416 \times e$	1
	$240 \times f$	1
	$336 \times g$	1
	$16 \times h$	1

(凡例)

- a : MQA サービス定義の MQT デーモン構成定義の指定数 (mqamqtnam 定義コマンドで指定)
- b : MQA サービス定義のプロセス定義数 (mqaprcdef 定義コマンドで指定)
- c : MQA サービス定義のオンライン中に使用するキューファイルグループの最大数 (mqa_max_grpnum オペランドで指定)
- d : MQA サービス定義のオンライン中に使用するキューファイルの最大数 (mqa_max_filnum オペランドで指定)
- e : MQA サービス定義のモデルキューの属性定義数 (mqaqueatl 定義コマンドで指定)
- f : MQA サービス定義の別名キューの属性定義数 (mqaalsque 定義コマンドで指定)
- g : MQA サービス定義のリモートキューのローカル定義数 (mqaremque 定義コマンドで指定)
- h : MQA サービス定義のオンライン中に使用するキューファイルの数 (ただし、二重化構成時のバックアップキューファイルは除く)

MQC サーバ機能を使用する場合（MQA サービス定義の mqa_mqc_conf オペランドに Y を指定する場合）、上記の MQA サーバの見積もり式に TP1/Message Queue Access が使用するステータスファイルのサイズを加算してください。

TP1/Message Queue Access のサーバ機能は TP1/Server Base のステータスファイルを使用します。ステータスファイルのサイズの見積もり式を次に示します。

表 B-2 ステータスファイルの使用容量とキー数（MQC リスナサーバ）

サービス	1 キー当たりのステータスファイルの使用容量 (単位：バイト)	キー数
TP1/Message Queue Access	96	1
MQC リスナサーバ	48×a	1

(凡例)

a：MQC サービス定義の MQC ゲートウェイサーバ名の定義数（mqcgwpsnam オペランドで指定）

付録 B.2 ジャーナルファイルのサイズの見積もり式

TP1/Message Queue では、TP1/Server Base のジャーナルを取得します。

ジャーナルファイルのサイズの見積もり方法については、マニュアル「OpenTP1 運用と操作」を参照してください。なお、次の表の項番 6～項番 10 は、マニュアル「OpenTP1 運用と操作」に記載している内容を次のように読み替えて見積もりをしてください。

- 「1 トランザクション」を「同一バッチ内で送受信するメッセージ数」に読み替えてください。
 - 「オンライン開始～終了までの総トランザクション数」を「オンライン開始～終了までの同一バッチによるメッセージ送受信回数」に読み替えてください。
- 1 バッチ単位で送受信されるメッセージの送受信数は、使用されるシステムによって異なるため安全値として「オンライン開始～終了までの送受信されるメッセージ数の総和」にすることをお勧めします。

(1) ジャーナルレコード長

TP1/Message Queue が使用するジャーナルレコード長を次に示します。

表 B-3 ジャーナルレコード長 (TP1/Message Queue)

項番	条件	ジャーナルレコード長 (単位：バイト)		条件に一致したときに 取得されるジャーナル レコード種別
		メッセージジャーナル の取得回復機能を使用 する場合	メッセージジャーナル の取得回復機能を使用 しない場合	
1	トランザクション内で永続メッセージを登録または削除する場合、そのトランザクションごと	$160 + \Sigma P + \Sigma G$	$160 + \Sigma P + \Sigma G$	cj
2	トランザクション内で永続メッセージを登録する場合、MQPUT 命令または MQPUT1 命令の発行ごと	$160 + Q$	—	cj
3	トランザクション外で永続メッセージを登録する場合、MQPUT 命令または MQPUT1 命令の発行ごと	$320 + P + Q$	$160 + P$	cj
4	トランザクション外で永続メッセージを削除する場合、MQGET 命令ごと	$160 + G$	$160 + G$	cj
5	永続メッセージの受信時、メッセージの受信ごと	$160 + Q$	—	cj
6	永続メッセージの送信時、同一バッチ内の最終メッセージの送信ごと※1	$160 + \Sigma G$	$160 + \Sigma G$	cj
7	永続メッセージの受信時、同一バッチ内の最終メッセージの受信ごと※2	$160 + \Sigma P$	$160 + \Sigma P$	cj
8	メッセージの送信時、同一バッチ内の最終メッセージの送信ごと※1, ※3	$780 + 16 \times m$	$780 + 16 \times m$	cj
9	メッセージの送信時、同一バッチ内の最終メッセージの送信ごと※1, ※3	780	780	cj
10	メッセージの受信時、同一バッチ内の最終メッセージの受信ごと※2, ※4	780	780	cj
11	自動クラスタセンダチャンネルの作成および削除ごと	780	780	cj

(凡例)

—：該当しません。

P：メッセージ登録ジャーナルデータ

32 ビット版の場合：

$$P=644 + p$$

64 ビット版の場合：

$$P=660 + p$$

p: 1 メッセージの最大レコード数

$$p=4 \times \uparrow (L + 512) / (((\uparrow (512 + M) / (S-8) \uparrow) \times S) - 8) \uparrow$$

Q: メッセージ登録メッセージジャーナルデータ

32 ビット版の場合:

$$Q=516 + \uparrow (2 + f) / 4 \uparrow \times 4 + \uparrow (2 + g) / 4 \uparrow \times 4 + \uparrow (L + 512) / (((\uparrow (512 + M) / (S-8) \uparrow) \times S) - 8) \uparrow \times 4 + \uparrow L / 4 \uparrow \times 4$$

64 ビット版の場合:

$$Q=532 + \uparrow (2 + f) / 4 \uparrow \times 4 + \uparrow (2 + g) / 4 \uparrow \times 4 + \uparrow (L + 512) / (((\uparrow (512 + M) / (S-8) \uparrow) \times S) - 8) \uparrow \times 4 + \uparrow L / 4 \uparrow \times 4$$

G: メッセージ削除ジャーナルデータ

32 ビット版の場合:

$$G=96 + \uparrow (2 + f) / 8 \uparrow \times 8 + \uparrow (2 + g) / 8 \uparrow \times 8$$

64 ビット版の場合:

$$G=112 + \uparrow (2 + f) / 8 \uparrow \times 8 + \uparrow (2 + g) / 8 \uparrow \times 8$$

Σ: 項番 1 については、トランザクション内での総和を示します。項番 6 および項番 7 については、1 回のバッチで送受信するメッセージ数の総和を示します。なお、「オンライン開始～終了までの送受信されるメッセージ数の総和」で計算する場合、項番 6 および項番 7 は 1 メッセージ単位の計算になります。

上記の計算式の変数の意味は次のとおりです。

↑↑: 小数点以下を切り上げます。

L: メッセージ長

M: mqainit コマンドの-s オプション (メッセージ長) の指定値

S: OpenTP1 ファイルシステムのセクタ長

f: 該当キューが所属するキューファイルグループで定義されている中で最も長いキューファイル名長

g: 該当キューを作成した際に使用したモデルキューの属性定義のキューファイル名長

m: 同一バッチ内のメッセージ数

「オンライン開始～終了までの送受信されるメッセージ数の総和」で計算する場合、m の値は 1 になります。

注※1

項番 6, 項番 8 および項番 9 は別のジャーナルレコードとして取得されます。

注※2

項番 7 および項番 10 は別のジャーナルレコードとして取得されます。

注※3

ファーストメッセージだけ送信する場合は取得されません。

注※4

ファーストメッセージだけ受信する場合は取得されません。

MQC サーバ機能を使用する場合（MQA サービス定義の mqa_mqc_conf オペランドに Y を指定する場合）、上記の MQA サーバの見積もり式に TP1/Message Queue Access が使用するジャーナルレコード長のサイズを加算してください。

TP1/Message Queue Access のサーバ機能は TP1/Server Base のジャーナルを取得します。ジャーナルレコード長の見積もり式を次に示します。

表 B-4 ジャーナルレコード長 (TP1/Message Queue Access)

条件	ジャーナルレコード長 (単位：バイト)	条件に一致したときに取得されるジャーナルレコード種別
XA 連携で次の指示が通知されること <ul style="list-style-type: none"> • コミット • ロールバック • トランザクション終了 • プリペア 	288	cj

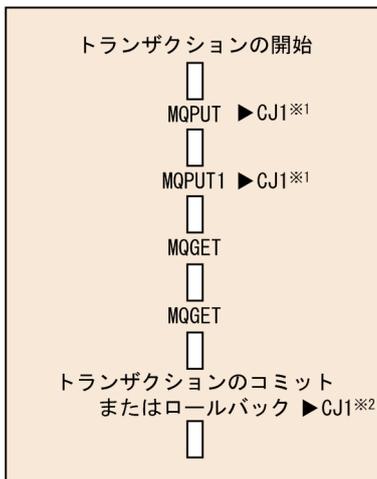
(2) ジャーナルの取得契機

TP1/Message Queue のジャーナルの取得契機を次に示します。

(a) 同期点でメッセージを登録、または削除する場合

同期点でメッセージを登録、または削除する場合のジャーナルの取得契機を次に示します。

アプリケーション



(凡例)

CJ1：回復用ジャーナルの取得（バッファリング）を示します。

注※1

メッセージジャーナルの取得回復機能を使用する場合に取得されます。表 B-3 の項番 2 に該当します。

注※2

表 B-3 の項番 1 に該当します。

図中の各命令での構造体の情報を次に示します。

MQPUT 命令, または MQPUT1 命令

- MQPMO 構造体：Options フィールドに MQPMO_SYNCPOINT を設定
- 登録するメッセージの MQMD 構造体：Persistence フィールドに MQPER_PERSISTENT を設定

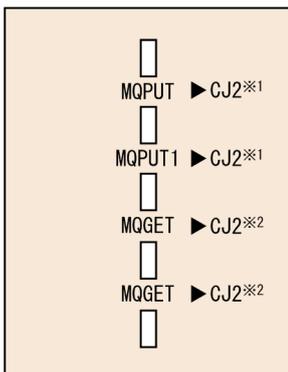
MQGET 命令

- MQGMO 構造体：Options フィールドに MQGMO_SYNCPOINT を設定, MQGMO_BROWSE_XXX は未設定
- 削除対象となるメッセージの MQMD 構造体：Persistence フィールドに MQPER_PERSISTENT を設定

(b) 同期点なしでメッセージを登録, または削除する場合

同期点なしでメッセージを登録, または削除する場合のジャーナルの取得契機を次に示します。

アプリケーション



(凡例)

CJ2：回復用ジャーナルの取得（即書き）を示します。

注※1

表 B-3 の項番 3 に該当します。

注※2

表 B-3 の項番 4 に該当します。

図中の各命令での構造体の情報を次に示します。

MQPUT 命令, または MQPUT1 命令

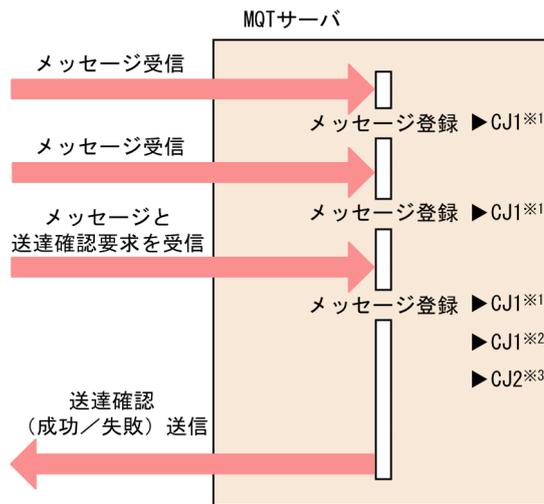
- MQPMO 構造体：Options フィールドに MQPMO_NO_SYNCPOINT を設定
- 登録するメッセージの MQMD 構造体：Persistence フィールドに MQPER_PERSISTENT を設定

MQGET 命令

- MQGMO 構造体：Options フィールドに MQGMO_NO_SYNCPOINT を設定, MQGMO_BROWSE_XXX は未設定
- 削除対象となるメッセージの MQMD 構造体：Persistence フィールドに MQPER_PERSISTENT を設定

(c) チャネル経由でメッセージを受信する場合

チャネル経由でメッセージを受信する場合のジャーナルの取得契機を次に示します。



(凡例)

CJ1：回復用ジャーナルの取得（バッファリング）を示します。

CJ2：回復用ジャーナルの取得（即書き）を示します。

注※1

メッセージジャーナルの取得回復機能を使用する場合に取得されます。表 B-3 の項番 5 に該当します。

注※2

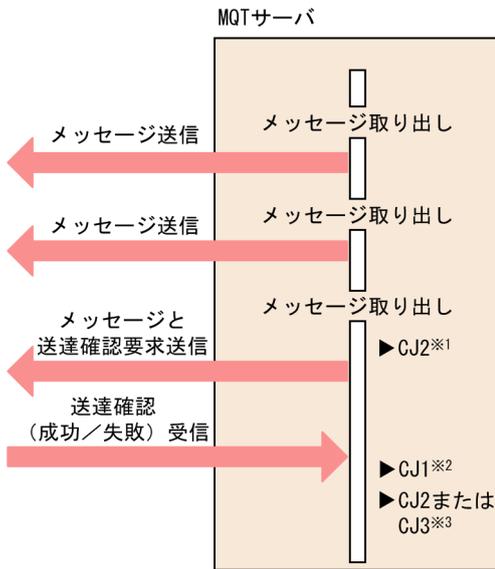
ファーストメッセージだけを受信する場合、または非永続メッセージだけを受信する場合は取得されません。表 B-3 の項番 7 に該当します。

注※3

ファーストメッセージだけを受信する場合は取得されません。表 B-3 の項番 10 に該当します。

(d) チャネル経由でメッセージを送信する場合

チャネル経由でメッセージを送信する場合のジャーナルの取得契機を次に示します。



(凡例)

CJ1：回復用ジャーナルの取得（バッファリング）を示します。

CJ2：回復用ジャーナルの取得（即書き）を示します。

CJ3：mqtalcccha 定義コマンドの-jn オプションの sndjnl オペランドに noflush を指定した場合、回復用ジャーナルの取得（バッファリング）を示します。

注※1

ファーストメッセージだけを送信する場合は取得されません。表 B-3 の項番 8 に該当します。

注※2

ファーストメッセージだけを送信する場合、または非永続メッセージだけを送信する場合は取得されません。表 B-3 の項番 6 に該当します。

注※3

ファーストメッセージだけを送信する場合は取得されません。表 B-3 の項番 9 に該当します。

付録 B.3 チェックポイントダンプファイルのサイズの見積もり式

TP1/Message Queue は TP1/Server Base のチェックポイントダンプを取得します。チェックポイントダンプファイルの使用容量を次の表に示します。

表 B-5 チェックポイントダンプファイルの使用容量（TP1/Message Queue）

取得元	チェックポイントダンプファイルの使用容量 (単位：バイト)
TP1/Message Queue	32 ビット版の場合： (総 MCA 数 (各 mqamqtnam 定義コマンドで指定した合計値) + 1) × 568 + 総 バッチメッセージ数 (各 mqamqtnam 定義コマンドの-b オプションの合計値) × 8 + 32

取得元	チェックポイントダンプファイルの使用容量 (単位：バイト)
TP1/Message Queue	64 ビット版の場合： (総 MCA 数 (各 mqamqtnam 定義コマンドで指定した合計値) + 1) × 576 + 総 バッチメッセージ数 (各 mqamqtnam 定義コマンドの -b オプションの合計値) × 16 + 32

MQA サービス定義の mqamqtnam 定義コマンドが指定されていない場合、チェックポイントダンプファイルの容量は 0 になります。

MQC サーバ機能を使用する場合 (MQA サービス定義の mqa_mqc_conf オペランドに Y を指定する場合)、上記の見積もり式に TP1/Message Queue Access が使用するチェックポイントダンプファイルのサイズを加算してください。

TP1/Message Queue Access のサーバ機能は TP1/Server Base のチェックポイントダンプを取得します。MQC チェックポイントダンプファイルのサイズの見積もり式を次に示します。

表 B-6 MQC チェックポイントダンプファイルの使用容量 (TP1/Message Queue Access)

取得元	MQC チェックポイントダンプファイルの使用容量 (単位：バイト)
TP1/Message Queue Access	32 ビット版の場合： 総トランザクション数 (mqa_mqc_trnnum オペランド) × 352 64 ビット版の場合： 総トランザクション数 (mqa_mqc_trnnum オペランド) × 416

MQA サービス定義の mqa_mqc_conf オペランドに Y が指定されていない場合、MQC チェックポイントダンプファイルの容量は 0 になります。

注意事項

これらの計算式は、チェックポイントダンプファイル内で TP1/Message Queue、または TP1/Message Queue Access が使用する容量サイズの見積もりを示しています。jnlinit コマンドを実行して実際に作成されるチェックポイントダンプファイルのサイズについては、次の計算式で見積もってください。

$$(\uparrow (A + B) / 4096 \uparrow + 3) \times 4096$$

(凡例)

↑↑：小数点以下を切り上げます。

A：TP1/Message Queue のチェックポイントダンプファイル使用容量

B：TP1/Message Queue Access の MQC チェックポイントダンプファイル使用容量

付録 B.4 キューファイルのサイズの見積もり式

キューファイルのサイズの見積もり式を次に示します。

表 B-7 キューファイルの使用容量

項目	ディスク占有量 (単位：バイト)
メッセージ情報管理ファイル	32 ビットの場合： $(2 + \uparrow 516 \times Q / S \uparrow + \uparrow 256 \times N / S \uparrow + \uparrow 4 \times N / S \uparrow) \times S$ 64 ビットの場合： $(2 + \uparrow 512 \times Q / S \uparrow + \uparrow 256 \times N / S \uparrow + \uparrow 4 \times N / S \uparrow) \times S$
メッセージファイル	$((\uparrow (520 + M) / S \uparrow) \times S) \times N$

(凡例)

$\uparrow \uparrow$ ：小数点以下を切り上げます。

Q：mqainit コマンドのキュー数

M：mqainit コマンドのメッセージ長

N：mqainit コマンドのメッセージ数

S：OpenTPI ファイルシステムのセクタ長（1024 未満の場合は 1024）

付録 B.5 チャネル管理情報格納ファイルのサイズの見積もり式

TP1/Message Queue の正常開始時のチャネル状態引き継ぎ機能を使用した場合、チャネル管理情報格納ファイルが必要です。チャネル管理情報格納ファイルのサイズの見積もり式を次に示します。

表 B-8 チャネル管理情報格納ファイルの使用容量

項目	ファイル使用容量 (単位：バイト)
チャネル管理情報格納ファイル（MQA サービス定義の mqa_channel_inf_file オペランドに normal を指定した場合）	32 ビット版の場合： $((\sum_{x=1}^A Bx) + 1) \times 568 + (\sum_{x=1}^A Cx) \times 8 + \uparrow (A \times 32 / 1024) \uparrow \times 1024 + 96$ 64 ビット版の場合： $((\sum_{x=1}^A Bx) + 1) \times 576 + (\sum_{x=1}^A Cx) \times 16 + \uparrow (A \times 32 / 1024) \uparrow \times 1024 + 136$
チャネル管理情報格納ファイル（MQA サービス定義の mqa_channel_inf_file オペランドに tp1file を指定した場合）	32 ビット版の場合： $(\uparrow 64 / D \uparrow + \uparrow (((\sum_{x=1}^A Bx) + 1) \times 568 + (\sum_{x=1}^A Cx) \times 8) / D \uparrow + \uparrow (\uparrow A \times 32 / 1024 \uparrow \times 1024) / D \uparrow + \uparrow 32 / D \uparrow) \times D$

項目	ファイル使用容量 (単位：バイト)
チャンネル管理情報格納ファイル (MQA サービス定義の mqa_channel_inf_file オペランドに tp1file を指定した場合)	64 ビット版の場合： $(\uparrow 88 / D \uparrow + \uparrow ((\sum_{x=1}^A Bx) + 1) \times 576 + (\sum_{x=1}^A Cx) \times 16) / D \uparrow + \uparrow (\uparrow A \times 32 / 1024 \uparrow \times 1024) / D \uparrow + \uparrow 48 / D \uparrow) \times D$

(凡例)

↑↑：小数点以下を切り上げます。

A：mqamqtnam 定義コマンドの定義数 (MQT デーモン数)

B：MCA 数 (mqamqtnam 定義コマンドの指定値)

C：総バッチメッセージ数 (mqamqtnam 定義コマンドの -b オプションの指定値)

D：OpenTP1 ファイルシステムのセクタ長

付録 B.6 スナップdumpファイルのサイズの見積もり式 (チャンネルの状態表示コマンド)

TP1/Message Queue のチャンネルの状態表示コマンドのスナップdump機能 (mqtlscha コマンドの -D オプション) で出力される、スナップdumpファイルのサイズの見積もり式を次の表に示します。ただし、チャンネルの状態や属性によってこの計算式より少なくなることがあります。

表 B-9 スナップdumpファイルの使用容量 (mqtlscha コマンドの -D オプション)

項目	1 プロセス当たりのファイル使用容量 (単位：バイト)
MQT スナップdumpファイル (通常プロセス)	32 ビット版の場合：
MQT スナップdumpファイル (クラスタセンダプロセス)	$4300 + A \times 7368 + \sum_{x=1}^A (Bx \times 8) + \sum_{x=1}^C (Dx \times 568 + Ex \times 8)$
	64 ビット版の場合：
	$4800 + A \times 8040 + \sum_{x=1}^A (Bx \times 16) + \sum_{x=1}^C (Dx \times 576 + Ex \times 16)$
MQT スナップdumpファイル (クラスタレシーバプロセス)	32 ビット版の場合：
	$7700 + A \times 7240 + \sum_{x=1}^C (Dx \times 568 + Ex \times 8)$
	64 ビット版の場合：
	$8600 + A \times 7904 + \sum_{x=1}^C (Dx \times 576 + Ex \times 16)$

(凡例)

A：事前定義または自動定義によって定義されたチャンネル数

B：送信側チャンネルのバッチサイズ

C：mqamqtnam 定義コマンドの定義数 (MQT デーモン数)

D：MCA 数 (mqamqtnam 定義コマンドの指定値)

E：総バッチメッセージ数 (mqamqtnam 定義コマンドの-b オプションの指定値)

付録 B.7 MQC トレースファイルのサイズの見積もり式

MQC サーバ機能を使用する場合、\$DCDIR/spool ディレクトリ下に MQC トレースファイルが作成されます。MQC トレースファイルのサイズの見積もり式を次に示します。

491712 + 起動済みのMQCゲートウェイサーバ数 × 491712 (単位：バイト)

付録 B.8 MQR トレースファイルのサイズの見積もり式

キューマネージャクラスタ機能を使用する場合、\$DCDIR/spool ディレクトリ下に MQR トレースファイルが作成されます。

MQR トレースファイルのサイズの見積もり式を次に示します。

DCMQA_TRACE_MAX_LENオペランドの指定値 × DCMQA_TRACE_FILE_NUMオペランドの指定値 (単位：バイト)

付録 B.9 MQT トレースファイルのサイズの見積もり式

MQT トレースファイルのサイズの見積もり式を次の表に示します。

表 B-10 MQT トレースファイルの使用容量

分類	条件	MQT トレースファイルの使用容量 (単位：バイト)	変数の説明
サービス開始	MQT サービスを開始するごと	32 ビット版の場合： 2200 + Σ (40) 64 ビット版の場合： 2500 + Σ (70)	Σ：mqtalcccha 定義コマンドの定義数の総数を意味します。 (例) mqtalcccha 定義コマンドの定義数=5 の場合、MQT サービス開始時のトレース容量は、次のようになります。 32 ビット版の場合： 2200 + 40 × 5 = 2400 バイト 64 ビット版の場合： 2500 + 70 × 5 = 2850 バイト
	MQ クラスタのMQT サービスを開始するごと	32 ビット版の場合： 4200	—

分類	条件	MQT トレースファイルの使用容量 (単位：バイト)	変数の説明
サービス開始	MQ クラスタのMQT サービスを開始すること	64 ビット版の場合： 6000	—
運用コマンド	MQT 運用コマンドを実行すること	Σ (1000)	Σ：チャンネル名または相手キューマネージャ名を指定するMQT 運用コマンドはその条件に一致するチャンネル数の総数を意味します。チャンネル名または相手キューマネージャ名を指定しないMQT 運用コマンドは、チャンネル名=1 とします。 (例) mqtalcccha 定義コマンドの定義数=16 で mqstacha -n "*" の場合、チャンネル数の総数=16 となります。
MQ クラスタ (クラスタセンダ)	チャンネル要求監視間隔 (mqttcps -v stim) ごと	450	—
MQ クラスタ (クラスタレシーバ)	チャンネル要求監視間隔 (mqttcpcr -v stim) ごと	330	—
イニシエーション キュー監視	タイマ方式によるイニシエーション キュー監視 (mqttcp -z time) ごと	32 ビット版の場合： 300 64 ビット版の場合： 350	(例) 1 秒間隔 (mqttcp -v itim=1) の監視を行う場合、1 分間のMQT トレース容量は、次のようになります。 32 ビット版の場合：300×60 = 18000 バイト 64 ビット版の場合：350×60 = 21000 バイト
	イベント方式によるイニシエーション キュー監視 (mqttcp -z event) を行う場合	32 ビット版の場合： Σ (60) 64 ビット版の場合： Σ (70)	Σ：イニシエーションキュー監視を行っている時間 (単位：秒) の総数を意味します。 (例) 1 分間のMQT トレース容量は、次のようになります。 32 ビット版の場合：60×60 = 3600 バイト 64 ビット版の場合：70×60 = 4200 バイト
チャンネル開始 (コーラ側)	コーラ側チャンネルを開始すること (チャンネルの確立を再試行する場合は、その再試行処理ごと)	32 ビット版の場合： Σ (500× tretrycnt) ※1 64 ビット版の場合： Σ (520× tretrycnt) ※1	tretrycnt：mqtalcccha -v tretrycnt の値 Σ：開始するコーラ側チャンネル数の総数を意味します。
チャンネル開始 (レスポンド側)	レスポンド側チャンネルを開始すること	32 ビット版の場合： Σ (3000)	Σ：開始するレスポンド側チャンネル数の総数を意味します。

分類	条件	MQT トレースファイルの使用容量 (単位：バイト)	変数の説明
チャンネル開始 (レスポンド側)	レスポンド側チャンネルを開始すること	64 ビット版の場合： Σ (3500)	Σ ：開始するレスポンド側チャンネル数の総数を意味します。
チャンネル終了 (コーラ側)	コーラ側チャンネルを終了すること	32 ビット版の場合： Σ (1500) 64 ビット版の場合： Σ (1800)	Σ ：終了するコーラ側チャンネル数の総数を意味します。
チャンネル終了 (レスポンド側)	レスポンド側チャンネルを終了すること	32 ビット版の場合： Σ (1000) 64 ビット版の場合： Σ (1400)	Σ ：終了するレスポンド側チャンネル数の総数を意味します。
メッセージ送受信	メッセージ送受信ごと	Σ (2000 + XQH)	XQH：第 1 セグメントの場合は 500，第 2 セグメント以降の場合は 0 となります。 (転送するメッセージ長 + 476) / 最大セグメントサイズ ≤ 1 の場合は，第 1 セグメントだけとなります。 (転送するメッセージ長 + 476) / 最大セグメントサイズ > 1 の場合は，その値がセグメント数となります。 Σ ：メッセージ送受信を行うチャンネル数の総数を意味します。
ハートビート	チャンネルが「チャンネル動作中」にハートビート電文を送受信すること	32 ビット版の場合： Σ (1200) 64 ビット版の場合： Σ (1500)	Σ ：ハートビート電文を送受信するチャンネル数の総数を意味します。「チャンネル動作中」以外の状態のチャンネル数はカウントしません。
転送キュー監視	タイマ方式による転送キュー監視 (mqtalccha -z time) を行う場合，転送キュー監視間隔 (mqtalccha -v tim4) ごと	32 ビット版の場合： Σ (400) 64 ビット版の場合： Σ (750)	Σ ：タイマ方式による転送キュー監視を行うチャンネル数の総数を意味します。「チャンネル動作中」以外の状態のチャンネル数はカウントしません。
	イベント方式による転送キュー監視 (mqtalccha -z event) を行う場合	32 ビット版の場合： $\Sigma 1 \Sigma 2$ (80)	$\Sigma 1$ ：イベント方式による転送キュー監視を行うチャンネル数の総数を意味します。「チャンネル動作中」以外の状態のチャンネル数はカウントしません。 $\Sigma 2$ ：イベント方式による転送キュー監視を行っている時間 (単位：秒) の総数を意味します。

分類	条件	MQT トレースファイルの使用容量 (単位：バイト)	変数の説明
転送キュー監視	イベント方式による転送キュー監視 (mqtalcccha -z event) を行う場合	64 ビット版の場合： $\Sigma 1 \Sigma 2$ (150)	(例) 二つのチャンネルが1分間イベント方式による転送キュー監視を行う場合、次のようになります。 32 ビット版の場合： $2 \times 60 \times 80 = 9600$ バイト 64 ビット版の場合： $2 \times 60 \times 150 = 18000$ バイト
サービス終了	MQT サービスを終了すること	32 ビット版の場合： 1300 64 ビット版の場合： 2000	—
UOC※2	チャンネルを開始すること	200	—
	クラスタレシーバチャンネルが確立すること	260	—
	メッセージ送受信ごと	260	—
	チャンネルを終了すること	260	—

(凡例)

—：該当しません。

注※1

TCP/IP コネクションの状況によっては、計算式よりも少なくなる場合があります。

注※2

64 ビット版の場合にだけ該当します。

付録 B.10 ソケット用ファイル記述子のサイズの見積もり式 (ユーザサーバで使用)

ユーザサーバで使用するソケット用ファイル記述子 (max_socket_descriptors オペランド) の最大数の見積もり式を次に示します。

$$\uparrow \text{メッセージを登録するキュー数} \times 2 / 0.8 \uparrow$$

(凡例)

↑↑：小数点以下を切り上げます。

付録 C 系切り替え機能使用時の注意

系切り替え機能を使用する場合の注意を次に示します。

- 系切り替え構成を構築する場合、各マシンのシステム時刻を合わせてください。
タイマ値が異なると、系切り替え発生時に時間が戻ることになるため、メッセージの登録順序が入れ代わることがあります。
- 系切り替え構成を構築し、TP1/Message Queue 以外のシステムとチャンネルを接続する場合、系切り替えによって引き継ぐ IP アドレスを、MQT 通信構成定義のチャンネル定義 (mqtalcccha 定義コマンド) に自システムの IP アドレスとして指定 (-r オプションの ipaddr オペランドまたは hostname オペランドに指定) してください。IP アドレスを指定しないと、シーケンス番号の不一致が発生することがあります。

付録 D TP1/Message Queue が出力するファイル一覧

TP1/Message Queue (UNIX 版・Windows 版共通) が出力するファイルの一覧を次の表に示します。各ファイルの詳細な説明については、表 D-2 を参照してください。

表 D-1 TP1/Message Queue (UNIX 版・Windows 版共通) が出力するファイル一覧

項番	名称	ファイル名またはディレクトリ名	バージョン	タイプ	ファイル形式	取得タイミング	削除可否
1	MQT トレースファイル	UNIX 版： \$DCDIR/spool/mqttrcXXYY ※1 Windows 版： %DCDIR% %spool%mqttrcXXYY※1 (XX：MQT 通信プロセス識別子，YY：トレーススワップファイル識別子)	初期	A	バイナリ	MQT トレースのディスク出力機能を使用している場合は、次のタイミングで取得します。 <ul style="list-style-type: none"> • mqtswptrc コマンド実行時 • メモリ上に蓄積された MQT トレース情報のバッファが満杯になったとき • OpenTP1 停止時 (MQT 通信プロセス終了時) • mqtstptrc コマンド実行時 MQT トレースのディスク出力機能を使用していない場合は、次のタイミングで取得します。 <ul style="list-style-type: none"> • mqtswptrc コマンド実行時 	△
2	MQT サービスローカルメモリダンプファイル	< mqtlscha -D コマンド実行時 > UNIX 版： \$DCDIR/spool/mqtdmpXX Windows 版： %DCDIR%*spool*mqtdmpXX < MQT 通信プロセス異常終了時 > UNIX 版： \$DCDIR/spool/ mqtdmpXX_YY....YY Windows 版： %DCDIR% %spool*mqtdmpXX_YY....YY	05-10	B, D	バイナリ	<ul style="list-style-type: none"> • mqtlscha -D コマンド実行時 • 障害や論理矛盾を検知して MQT 通信プロセスが異常終了したとき 	△

項番	名称	ファイル名またはディレクトリ名	バージョン	タイプ	ファイル形式	取得タイミング	削除可否
2	MQT サービスローカルメモリダンプファイル	(XX : MQT 通信プロセス識別子, YY...YY : MQT 通信プロセス開始からの通算秒 (10 進数))	05-10	B, D	バイナリ	<ul style="list-style-type: none"> mqtlscha -D コマンド実行時 障害や論理矛盾を検知して MQT 通信プロセスが異常終了したとき 	△
3	MQR トレースファイル	UNIX 版 : \$DCDIR/spool/mqrtrcXX*2 Windows 版 : %DCDIR%*spool*mqrtrcXX*2 (XX : 1~99)	05-10	A	バイナリ	<ul style="list-style-type: none"> メモリ上に蓄積された MQR トレース情報のバッファが満杯になったとき リポジトリ管理サーバ (mqrsup) 停止時 mqrswptrc コマンド実行時 	△
4	MQC トレースファイル	UNIX 版 : \$DCDIR/spool/mqcXXXYYYZZZZZ Windows 版 : %DCDIR%*spool*mqcXXXYYYZZZZZ (XXX : サーバ種別 (lsn : MQC リスナサーバ, gwp : MQC ゲートウェイサーバ), YYY : MQC ゲートウェイサーバ識別子 (000~255 の 10 進数), ZZZZ : 00001~00003)	05-00	A	バイナリ	<ul style="list-style-type: none"> メモリ上に蓄積された MQC トレース情報のバッファが満杯になったとき MQC ゲートウェイサーバ停止時 OpenTP1 停止時 mqcswptrc コマンド実行時 	△
5	チャンネル管理情報格納ファイル	MQA サービス定義の mqa_channel_inf_file_name_a オペランド, および mqa_channel_inf_file_name_b オペランドの指定値	01-03	B	バイナリ	<ul style="list-style-type: none"> MQT サーバサービス開始時 MQT サーバサービス終了時 	▲
6	mqafrc コマンド用引き継ぎファイル	UNIX 版 : mqafrc コマンド実行のレントディレクトリ/mqafrc*** Windows 版 : mqafrc コマンド実行のレントディレクトリ*mqafrc*** (*** : mqafrc -k で指定した値)	05-12	B	バイナリ	mqafrc -k コマンド実行時	□ (mqafrc -e)
7	mqafrc コマンド用引き継ぎファイルのバックアップファイル	UNIX 版 : mqafrc コマンド実行のレントディレクトリ/mqafrc***.bak Windows 版 :	05-12	C	バイナリ	mqafrc -k コマンド実行時に, 同名の引き継ぎファイルがあるとき	□ (mqafrc -e)

項番	名称	ファイル名またはディレクトリ名	バージョン	タイプ	ファイル形式	取得タイミング	削除可否
7	mqafrc コマンド用引き継ぎファイルのバックアップファイル	mqafrc コマンド実行のカルレントディレクトリ¥mqafrc***.bak (*** : mqafrc -k で指定した値)	05-12	C	バイナリ	mqafrc -k コマンド実行時に、同名の引き継ぎファイルがあるとき	<input type="checkbox"/> (mqafrc -e)

(凡例)

A : ラウンドロビン (一定量に達した直後の出力で、新しいファイルに切り替わるタイプ)

B : 制御ファイル, 一時ファイル

C : バックアップファイル

D : コマンド実行などで 1 回ごとに出力し、最大容量が決まっているファイル

△ : 削除してはいけません。ただし、障害調査が不要であれば、ユーザ判断で削除できます。

▲ : 削除してはいけません。ただし、チャンネル管理情報格納ファイルを使ったチャンネル情報引き継ぎ機能を使用しない場合は、ユーザ判断で削除できます。

□ : コマンド終了時に削除します。() 内は該当するコマンド名です。

注※1

MQT サービス定義の mqt_trace_file_path オペランドで、ファイルの出力先を変更できます。

注※2

リポジトリ管理サーバユーザサービス定義ファイル (\$DCCONFPATH/mqrsup) の DCMQA_TRACE_PATH オペランドで、ファイルの出力先を変更できます。

TP1/Message Queue (UNIX 版・Windows 版共通) が出力するファイルの説明を次の表に示します。

表 D-2 TP1/Message Queue (UNIX 版・Windows 版共通) が出力するファイルの説明

項番	名称	関連する定義	サイズ	最大ファイル数	説明
1	MQT トレースファイル	<p>< TCP 構成定義 ></p> <p>< クラスタセンダ TCP 構成定義 ></p> <p>< クラスタレシーバ TCP 構成定義 ></p> <ul style="list-style-type: none"> • -t disk オペランド MQT トレースのディスク出力機能を使用するかどうかを指定 • -t bufcnt オペランド MQT トレースファイルのサイズ • -t trccnt オペランド MQT トレースファイルの個数 • -t bufsize オペランド MQT トレースを格納するバッファの大きさ 	「付録 B.9 MQT トレースファイルのサイズの見積もり式」を参照	1 プロセス当たり n 世代 (n : mqttcp, mqttcpes または mqttcpcr 定義コマンドの -t trccnt オペランドの値)	<p>MQT サーバはプロセスの動作状態を MQT トレース情報として取得します。</p> <p>< 初期化方法 ></p> <p>OpenTP1 開始時 (MQT 通信プロセス開始時) は MQT トレースファイル名の YY に、前回の値の次の値を設定し、ラウンドロビン方式で出力します。同名のファイルがある場合は上書きします。</p>

項番	名称	関連する定義	サイズ	最大ファイル数	説明
2	MQT サービスローカルメモリダンプファイル	なし	「付録 B.6 スナップダンプファイルのサイズの見積もり式 (チャンネルの状態表示コマンド)」を参照	1 プロセス当たり 1 ファイル	MQT 通信プロセスの保守情報をスナップダンプファイルとして取得します。 mqtdmped コマンドで編集・表示ができます。 <初期化方法> 同名のファイルがあれば上書きします。
3	MQR トレースファイル	<\$DCCONFPATH/mqrsup のオペランド> <ul style="list-style-type: none"> DCMQA_TRACE_STATUSES MQR トレース情報を出力するかどうかを指定 DCMQA_TRACE_PATH MQR トレースファイルのパス名 DCMQA_TRACE_FILE_NUM MQR トレースファイルの最大数 DCMQA_TRACE_MAX_LEN MQR トレースファイルの容量 	「付録 B.8 MQR トレースファイルのサイズの見積もり式」を参照	n 世代 (n : \$DCCONFPATH/mqrsup に定義した環境変数 DCMQA_TRACE_FILE_NUM の指定値)	クラスタ機能についてのシステムメッセージ送受信の情報を MQR トレース情報として取得します。 <初期化方法> リポジトリ管理サーバ起動時の通番 (MQR トレースファイル名の XX) は、前回の値の次の値を設定し、ラウンドロビン方式で出力します。同名のファイルがある場合は上書きします。
4	MQC トレースファイル	なし	「付録 B.7 MQC トレースファイルのサイズの見積もり式」を参照	1 プロセス当たり 3 世代	MQC サーバの回線トレース情報です。 <初期化方法> MQC サーバ起動時の通番 (MQC トレースファイル名の ZZZZ) は、前回の値の次の値を設定し、ラウンドロビン方式で出力します。同名のファイルがある場合は上書きします。
5	チャンネル管理情報格納ファイル	<MQA サービス定義> <ul style="list-style-type: none"> mqa_channel_inf_file_name_a オペランド チャンネル管理情報格納 A 系ファイル mqa_channel_inf_file_name_b オペランド 	「付録 B.5 チャンネル管理情報格納ファイルのサイズの見積もり式」を参照	2 ファイル	MQT サーバのチャンネルについての情報を格納するファイルです。 <初期化方法> 同名のファイルがあれば上書きします。

項番	名称	関連する定義	サイズ	最大ファイル数	説明
5	チャンネル管理情報格納ファイル	チャンネル管理情報格納 B 系ファイル • mqa_channel_inf_file オプション チャンネル管理情報格納ファイルの入出力形式	「付録 B.5 チャンネル管理情報格納ファイルのサイズの見積もり式」を参照	2 ファイル	MQT サーバのチャンネルについての情報を格納するファイルです。 <初期化方法> 同名のファイルがあれば上書きします。
6	mqafrfc コマンド用引き継ぎファイル	なし	最大で 4096 + X バイト X : mqafrfc コマンド実行時に指定したアンロードジャーナルファイルの総ディスク容量※	1 ファイル	mqafrfc コマンドでキューファイルを回復する際にコマンド内部で使用する引き継ぎファイルです。 <初期化方法> 同名の引き継ぎファイルがあるときは、既存の引き継ぎファイルの名称を「mqafrfc***.bak」に変更して退避します。
7	mqafrfc コマンド用引き継ぎファイルのバックアップファイル	なし	最大で 4096 + X バイト X : mqafrfc コマンド実行時に指定したアンロードジャーナルファイルの総ディスク容量※	1 ファイル	mqafrfc コマンド用引き継ぎファイルのバックアップファイルです。

注※

アンロードジャーナルファイルの総ディスク容量は、UNIX の ls コマンドなどで参照できます。コマンド引数に複数のアンロードジャーナルファイルを指定した場合は、その合計になります。

付録 E 用語解説

TP1/Message Queue で使用する用語について説明します。

(英字)

FAP

米国 International Business Machines Corporation によって策定され、MCP に関するフォーマットとプロトコル (FAP : Format and Protocols) が定義されたものです。

IBM MQ

米国 IBM 社が開発した、メッセージ蓄積型の通信を実現する製品群のことです。OpenTP1 では、TP1/Message Queue を使ってメッセージ蓄積型の通信を実現できます。

MCP

TCP/IP プロトコル上で動作するメッセージチャンネルプロトコル (MCP) のことです。メッセージを送受信するためのチャンネルの機能を提供します。

MQ システム

メッセージキューイング機能のキューマネージャがあるシステムのことです。

(ア行)

アプリケーション

目的の業務に応じてユーザが作成するプログラムのことです。TP1/Message Queue の機能を利用するアプリケーションは、OpenTP1 の UAP として作成します。

オブジェクト

アプリケーションがメッセージキューイング機能の命令で利用する対象のことです。キュー、キューマネージャなどがあります。

(カ行)

キューファイル

メッセージキューを格納するファイルのことです。一つのキューファイルに、複数のキューが格納されます。

キューマネージャ

メッセージキューイング機能を管理するソフトウェア製品のことで、メッセージキューイング機能を使って通信するシステムには、キューマネージャが必要になります。OpenTP1 の場合は、TP1/Message Queue がキューマネージャの役割をします。

クラスタ

クラスタは、何らかの方法によって論理的に関連づけられたキューマネージャのネットワークです。クラスタに属しているキューマネージャは、それぞれが物理的にリモートであることもあります。世界的なチェーンストアをクラスタに例えて、キューマネージャをチェーンストアの支店に例えると、それぞれの支店同士は、物理的に別々の国にあることもあります。一つの企業内の各クラスタは、ユニークな名前を持っていないければなりません。

クラスタキュー

クラスタキューは、クラスタキューマネージャが保持し、クラスタのほかのキューマネージャが利用できるキューです。クラスタキューマネージャでは、ローカルなキューを作成し、キューを使用できるクラスタの名前を指定します。この指定は、クラスタのほかのキューマネージャにキューの存在を通知する効果を持ちます。

クラスタのほかのキューマネージャは、対応するリモートキューのローカル定義なしで、クラスタキューにメッセージを登録できます。クラスタキューは、複数のクラスタに通知されることもあります。

クラスタキューマネージャ

クラスタキューマネージャは、クラスタのメンバであるキューマネージャです。キューマネージャは、複数のクラスタのマネージャになることもできます。各クラスタキューマネージャは、自分がメンバであるすべてのクラスタでユニークな名前を持っていないければなりません。

クラスタキューマネージャは、クラスタのほかのキューマネージャに通知するキューを保持できます。クラスタキューマネージャは、すべてのキューを保持したり、通知したりする必要はありません。メッセージをクラスタに提供し、通知されるキューにではなく、明示的にそのクラスタに向けられた応答だけを受信します。

クラスタキューマネージャは自律的です。自分が定義しているキューやチャンネルを全面的にコントロールします。その定義をほかのキューマネージャは変更できません。キューマネージャでの定義を作成または変更する場合には、フルリポジトリキューマネージャに情報を送信し、それに従ってクラスタのリポジトリが更新されます。

クラスタセンダチャンネル

クラスタセンダ (CLUSDR) チャンネル定義は、クラスタキューマネージャがフルリポジトリの一つにクラスタ情報を送信できるチャンネルの送信端を定義しています。クラスタセンダチャンネルを使用して、キューマネージャの状態の変更 (例えば、キューの追加や削除) をリポジトリに通知します。また、メッセージを転送するのにも使用されます。

フルリポジトリキューマネージャ自身は、相互にポイントしているクラスタセンダチャンネルを持っています。これを使用して、クラスタ状態の変更を相互に伝達します。

キューマネージャのクラスタセンダチャンネル定義がどのフルリポジトリをポイントしているかは、ほとんど問題になりません。最初のコンタクトが実行されれば、クラスタセンダチャンネルが自動的に定義されるので、キューマネージャはどのフルリポジトリへもクラスタ情報を送信でき、どのキューマネージャへもメッセージを送信できます。

クラスタ転送キュー

各クラスタキューマネージャは、SYSTEM.CLUSTER.TRANSMIT.QUEUE というクラスタ転送キューを持っています。クラスタ転送キューは、キューマネージャからのすべてのメッセージを、同じクラスタにあるほかのキューマネージャに送信します。

クラスタレシーバチャンネル

クラスタレシーバ (CLUSRCVR) チャンネル定義は、クラスタキューマネージャがクラスタのほかのキューマネージャからメッセージを受信できるチャンネルの受信端を定義しています。また、クラスタレシーバチャンネルは、クラスタについての情報 (リポジトリに向けられた情報) を通信することもできます。クラスタレシーバチャンネルの定義は、キューマネージャがメッセージを受信できることを、クラスタのほかのキューマネージャに通知する効果を持ちます。各クラスタキューマネージャに少なくとも一つのクラスタレシーバチャンネルが必要です。

(サ行)

サービス名ファイル

サービス名称とポート番号とを対応づけたファイルです。標準的なパス名を次に示します。

- UNIX の場合
/etc/services
- Windows の場合
C:¥WINDOWS¥system32¥drivers¥etc¥services

属性

オブジェクトに対応づけられた名称、タイプなど、オブジェクトの特性を示すものです。オブジェクトの種類によって、対応づけられる属性は異なります。

(タ行)

チャンネル

自システムと特定の MQ システムとの間に対応づけた論理的な通信路のことです。メッセージチャンネルエージェントによって管理されます。

トリガ

メッセージキューにメッセージが到着したことを自システムのアプリケーションに知らせる機能のことです。トリガイベントを監視するアプリケーションをトリガモニタアプリケーションといいます。

(八行)

ハートビートメッセージ

送信側 MCA の転送キューにメッセージがない場合、送信側 MCA が受信側 MCA に対して一定間隔で送受信するメッセージです。

バインディング

複数のキューマネージャが、同じクラスタキューの 1 インスタンスを保持するクラスタを作成できます。その場合には、連続するメッセージがすべてキューの同じインスタンスに送信されるようにしてください。MQOPEN 命令で MQOO_BIND_ON_OPEN オプションを使用して、特定のキューに連続するメッセージをバインドできます。

フルリポジトリキューマネージャ

フルリポジトリキューマネージャとは、フルリポジトリを保持しているクラスタキューマネージャです。可用性を確保するために、各クラスタで二つ以上のキューマネージャを選択してフルリポジトリを確保するようお勧めします。フルリポジトリキューマネージャは、クラスタのほかのキューマネージャが送信した情報を受信し、それに従ってフルリポジトリを更新します。フルリポジトリキューマネージャは、クラスタについての情報を最新状態に保つために、相互にメッセージを送信します。

フルリポジトリとパーシャルリポジトリ

フルリポジトリキューマネージャは、クラスタのすべてのキューマネージャについての完全な情報を保持します。この情報をクラスタのフルリポジトリといいます。

クラスタのほかのキューマネージャは、フルリポジトリに情報を照会し、パーシャルリポジトリにその情報のサブセットを構築します。キューマネージャのパーシャルリポジトリは、相互にメッセージを交換する必要があるキューマネージャについての情報だけを含んでいます。キューマネージャは自分が必要とする情報についての更新を要求し、その情報が変更されている場合、フルリポジトリキューマネージャは、新しい情報を送信します。ほとんどの場合、あるキューマネージャのパーシャルリポジトリは、クラスタ内で実行するために必要なすべての情報を持っています。追加情報が必要になると、キューマネージャはフルリポジトリに問い合わせ、自分のパーシャルリポジトリを更新します。キューマネージャは、リポジトリに対して更新を要求したり、更新を受信したりするために、SYSTEM.CLUSTER.COMMAND.QUEUE というキューを使用します。

ホスト名ファイル

ホスト名称と IP アドレスとを対応づけたファイルです。標準的なパス名を次に示します。

- UNIX の場合
/etc/hosts
- Windows の場合
C:¥WINDOWS¥system32¥drivers¥etc¥hosts

(マ行)

命令

アプリケーションからメッセージキューイング機能を利用するためのインタフェースのことです。C 言語では関数、COBOL 言語では CALL 文で呼び出すプログラムに相当します。

メッセージキュー

メッセージキューイング機能を使った通信をする場合に、メッセージを格納するために使用するキューのことです。メッセージキューに登録したメッセージは、キューマネージャによって通信相手に送信されます。

メッセージキューイング機能

米国 IBM 社が開発した、メッセージ蓄積型の通信手順のことです。メッセージキューイング機能を使った通信では、アプリケーションが登録したメッセージをキューマネージャが送受信するため、システム間の通信手順や通信障害時の処理をアプリケーションで意識しなくて済みます。また、通信相手のアプリケーションが稼働している必要がないため、アプリケーションの構造を簡単にできます。アプリケーションからキューへのアクセスには、IBM MQ 標準の API を使用します。

メッセージチャネルエージェント

キューからキューへメッセージを送信するプログラムのことです。送信側と受信側にそれぞれあり、対で動作します。メッセージチャネルエージェントには、送信側か受信側か、呼び出し側かそれを受ける側かなどによって、四つのチャネルタイプがあります。

(ラ行)

リポジトリ

リポジトリは、クラスタのメンバであるキューマネージャについての情報の集まりです。この情報には、キューマネージャの名前、その位置、チャネル、および保持するキューなどが含まれます。情報は SYSTEM.CLUSTER.REPOSITORY.QUEUE (TP1/Message Queue の場合、ユーザが作成するキュー) というキューにメッセージの形で保存されます。通常、一つクラス

タ内で二つのキューマネージャがフルリポジトリを保持し、ほかのキューマネージャは、パーシャルリポジトリを保持します。

TP1/Message Queue がサポートするのは、パーシャルリポジトリだけです。IBM MQ のキューマネージャがフルリポジトリを保持するクラスタに参加することになります。

索引

A

- adoptchk [mqtalcccha (MQT 通信構成定義)] 378
- atomic_update [MQC ゲートウェイサービス定義] 408
- auto_restart [MQC ゲートウェイサービス定義] 409

B

- bmtim [mqtalcccha (MQT 通信構成定義)] 372
- bmtim [mqtttim (MQT 通信構成定義)] 342
- bretrycnt [mqtalcccha (MQT 通信構成定義)] 365
- bretryint [mqtalcccha (MQT 通信構成定義)] 365
- bretrylgcnt [mqtalcccha (MQT 通信構成定義)] 365
- bretrylgint [mqtalcccha (MQT 通信構成定義)] 365
- bretrylg [mqtalcccha (MQT 通信構成定義)] 365
- bretrymcp [mqtalcccha (MQT 通信構成定義)] 366
- bretry [mqtalcccha (MQT 通信構成定義)] 365
- btim [mqtalcccha (MQT 通信構成定義)] 371
- btim [mqtttim (MQT 通信構成定義)] 342
- bufcnt [mqttcp (MQT 通信構成定義)] 349
- bufcnt [mqttcpcr (MQT 通信構成定義)] 357
- bufcnt [mqttcpcs (MQT 通信構成定義)] 353
- bufsize [mqttcp (MQT 通信構成定義)] 349
- bufsize [mqttcpcr (MQT 通信構成定義)] 357
- bufsize [mqttcpcs (MQT 通信構成定義)] 353
- buftype [mqtalcccha (MQT 通信構成定義)] 363

C

- clusrcvr 73
- clusdr 73
- cluster [mqtalcccha (MQT 通信構成定義)] 377
- cnvccsid [mqtalcccha (MQT 通信構成定義)] 375
- CONNNAME パラメタ 165
- count [mqttbuf (MQT 通信構成定義)] 344

critical [MQC ゲートウェイサービス定義] 409

D

- dcmtcq_uoc_bufflist 243
- dcmtcq_uoc_mqcd 229
- dcmtcq_uoc_mqcxp 224
- dcmtcq_uoc_parmlist 223
- DEFINE CHANNEL 165
- disk [mqttcp (MQT 通信構成定義)] 349
- disk [mqttcpcr (MQT 通信構成定義)] 356
- disk [mqttcpcs (MQT 通信構成定義)] 352
- dtimefct [mqtalcccha (MQT 通信構成定義)] 371
- dtim [mqtalcccha (MQT 通信構成定義)] 371

E

- etim [mqttcp (MQT 通信構成定義)] 348
- etim [mqttcpcr (MQT 通信構成定義)] 356
- etim [mqttcpcs (MQT 通信構成定義)] 352
- extend [mqttbuf (MQT 通信構成定義)] 345

F

- FAP 71
- FAP [用語解説] 697

G

- groupno [mqttbuf (MQT 通信構成定義)] 343

H

- hostname [mqtalcccha (MQT 通信構成定義)] 367
- htim [mqtalcccha (MQT 通信構成定義)] 371

I

- IBM MQ [用語解説] 697
- ipaddr [mqtalcccha (MQT 通信構成定義)] 366
- itim [mqttcp (MQT 通信構成定義)] 348

L

length [mqttbuf (MQT 通信構成定義)] 343
libmqcgwp.a 214
libmqcgwp.lib 214
libmqcgwp.so 214
libmqcs.a 214
libmqcs.lib 214
libmqcs.so 214

M

max_socket_descriptors [MQA サービス定義] 291
max_socket_descriptors [MQT サービス定義] 329
maxmsg [mqtalcccha (MQT 通信構成定義)] 362
maxseg [mqtalcccha (MQT 通信構成定義)] 362
MCA 71
mcatype [mqtalcccha (MQT 通信構成定義)] 379
mcauser [mqtalcccha (MQT 通信構成定義)] 379
MCA 起動中 75
MCA 数 [mqamqtnam (MQA サービス定義)] 326
MCA 数オーバ 650
MCA タイプ [dcmtcq_uoc_mqcd] 242
MCA のチャンネルタイプ 72
MCA ユーザID [dcmtcq_uoc_mqcd] 242
MCP 障害 (チャンネル確立後, 相手システムで検出したMQ プロトコル障害) 639
MCP 障害 (チャンネル確立後, 相手システムで検出したMQ プロトコル障害・ファーストメッセージ) 654
MCP 障害 (チャンネル確立後, 自システムで検出したMQ プロトコル障害) 639
MCP 障害 (チャンネル確立後, 自システムで検出したMQ プロトコル障害・ファーストメッセージ) 653
MCP 障害 (チャンネル確立時の障害) 638
MCP 障害 (デッドレターキュー障害) 644
MCP 障害 (デッドレターキュー障害・ファーストメッセージ) 656
MCP 障害 (メッセージ受信時の転送障害) 642

MCP 障害 (メッセージ受信時の転送障害・ファーストメッセージ) 655
MCP 障害 (メッセージ送信時の転送障害) 641
MCP 障害 (メッセージ送信時の転送障害・ファーストメッセージ) 654
MCP [用語解説] 697
medbuf [mqtalcccha (MQT 通信構成定義)] 363
medcnt [mqtalcccha (MQT 通信構成定義)] 364
metexdata [mqtalcccha (MQT 通信構成定義)] 373
MHP 46
module [MQC ゲートウェイサーバユーザサービス定義] 408
mqa_channel_inf_file_name_a [MQA サービス定義] 283
mqa_channel_inf_file_name_b [MQA サービス定義] 283
mqa_channel_inf_file [MQA サービス定義] 284
mqa_expiry [MQA サービス定義] 286
mqa_extend_rmshm_size [MQA サービス定義] 291
mqa_filerr_continue [MQA サービス定義] 292
mqa_getwait_timeout [MQA サービス定義] 292
mqa_ioproc_num [MQA サービス定義] 286
mqa_jnl_conf [MQA サービス定義] 288
mqa_local_ccsid [MQA サービス定義] 287
mqa_max_filnum [MQA サービス定義] 286
mqa_max_grpnum [MQA サービス定義] 285
mqa_max_msg_recnum [MQA サービス定義] 285
mqa_max_quenum [MQA サービス定義] 284
mqa_maxmsg_expiry [MQA サービス定義] 287
mqa_maxqueue_expiry [MQA サービス定義] 286
mqa_mqc_clientnum [MQA サービス定義] 289
mqa_mqc_conf [MQA サービス定義] 288
mqa_mqc_trnnum [MQA サービス定義] 289
mqa_mqo_conf [MQA サービス定義] 289
mqa_mqo_queue_alarm_level [MQA サービス定義] 290
mqa_mqo_queue_recover_level [MQA サービス定義] 290

mqa_mqo_queuefile_alarm_level [MQA サービス定義] 290
 mqa_mqo_queuefile_recover_level [MQA サービス定義] 290
 mqa_mqr_conf [MQA サービス定義] 287
 mqa_msg_stay_time [MQA サービス定義] 290
 mqa_pool_attri [MQA サービス定義] 291
 mqa_prf_trace_level [MQA サービス定義] 293
 mqa_quefil_inf [MQA サービス定義] 287
 mqaadd 453
 mqaalsque 315
 mqaabkup 458
 mqachgque 459
 mqaconvert 462
 mqadel 464
 mqadelmsg 465
 mqadelque 467
 mqadf 469
 mqafilinf 472
 mqafills 478
 mqafrc 481
 MQA FRC 58
 mqagrpobs 485
 mqainit 487
 mqainq 490
 mqalsmsg 501
 mqamkque 507
 mqamqtnam 325
 mqaprcdef 318
 mqaqueatl 303
 mqaquegrp 297
 mqaquemgr 321
 mqarcvr 509
 mqaremque 312
 mqarles 511
 mqarm 513
 mqarstr 516
 mqaset 518
 MQA サーバ 31
 MQA サービス専用共用メモリ 35
 MQA サービス専用共用メモリ確保要求者識別子 [mqaquegrp (MQA サービス定義)] 301
 MQA サービス定義 276
 MQA サービス定義 [set 形式] 282
 mqc.def 214
 mqccontrn 520
 mqcdtcp 214
 mqcdtcp.exe 214
 mqcenv 403
 mqcgwp.o 214
 mqcgwpsnam 406
 mqcls 523
 mqclstrn 526
 mqcmkgwp 214
 mqcmkgwp.bat 214
 mqcmkgwp コマンド 215
 mqcswptrc 529
 mqctcp 214
 MQC クライアント機能 39
 MQC ゲートウェイサーバ 40
 MQC ゲートウェイサーバ監視タイマ値 [mqcenv (MQC サービス定義)] 404
 MQC ゲートウェイサーバの開始と終了 217
 MQC ゲートウェイサーバの環境設定での注意事項 215
 MQC ゲートウェイサーバの作成 215
 MQC ゲートウェイサーバの状態表示 [コマンド] 523
 MQC ゲートウェイサーバ名 [mqcgwpsnam (MQC サービス定義)] 406
 MQC ゲートウェイサーバ名定義 406
 MQC ゲートウェイサーバユーザサービス定義 408
 MQC コネクション確立完了監視タイマ値 [mqcenv (MQC サービス定義)] 404
 MQC サーバ機能 214
 MQC サーバ機能のインストール時に作成されるファイル 214
 MQC サーバ機能のセットアップ 214
 MQC サーバ機能の定義 215

MQC サービス定義 401
 MQC トランザクション状態の表示 [コマンド] 526
 MQC トランザクション状態の変更 [コマンド] 520
 MQC トレースの強制スワップ [コマンド] 529
 MQC リスナサーバ 39
 MQC リスナサーバ通信環境定義 403
 MQC リスナサーバの開始と終了 216
 MQC リスナサーバの終了モード 217
 MQI とクラスタ 160
 MQI 命令使用時の OpenTP1 アプリケーションとの関係 46
 MQR 189
 mqrls 530
 MQRMH 構造体 212
 mqrrefresh 546
 mqrremove 550
 mqrsp 190
 mqrsp のサービス定義に関する注意事項 192
 mqrsp 190
 mqrswptrc 552
 MQR トレースの強制スワップ [コマンド] 552
 MQR トレースファイルの出力 663
 mqtt_tcpnodelay [MQT サービス定義] 330
 mqtt_trace_file_path [MQT サービス定義] 330
 mqtalccha 359
 mqtalced 390
 mqaltcha 553
 mqtdmped 557
 mqted 567
 mqtlscha 576
 mqtpngcha 587
 mqtrlvcha 590
 mqtrstcha 593
 mqstacha 597
 mqstatrc 601
 mqstpcha 603
 mqstpptrc 606
 mqtswptrc 608
 mqttbuf 343
 mqtttcp 346
 mqttcpcr 354
 mqttcpcs 351
 mqttenv 341
 mqttim 342
 MQT 環境定義 341
 MQT サーバ 31
 MQT サーバのトリガ起動によるチャンネル開始 81
 MQT サーバ名または MQC リスナサーバ名 [mqamqtnam (MQA サービス定義)] 325
 MQT サービス定義 328
 MQT サービス定義 [set 形式] 329
 MQT 実行形式プログラムの作成 (UNIX) 247
 MQT 実行形式プログラムの作成 (Windows) 249
 MQT 通信構成定義 331
 MQT 通信プロセス識別子 [mqttenv (MQT 通信構成定義)] 341
 MQT 通信プロセス情報の表示 [コマンド] 557
 MQT 定義オブジェクト生成ユーティリティ 396
 MQT 定義オブジェクトファイルの作成手順 391
 MQT 定義結合ユーティリティ 399
 MQT デーモン構成定義 325
 MQT トレースの強制スワップ [コマンド] 608
 MQT トレースのディスク出力機能の開始 [コマンド] 601
 MQT トレースのディスク出力機能の終了 [コマンド] 606
 MQT トレースファイルの出力 661
 MQT トレースファイルの編集出力 [コマンド] 567
 MQT マネージャ 31
 MQT メイン関数の作成 246
 MQ システム [用語解説] 697
 MQ プロトコルエラー時のチャンネル確立再試行の有無 [dcmtcq_uoc_mqcd] 240
 msgexit [mqtalccha (MQT 通信構成定義)] 373
 mtim [mqtalccha (MQT 通信構成定義)] 372

N

netprty [mqtalccha (MQT 通信構成定義)] 377

nice [MQC ゲートウェイサーバユーザサービス定義] 409

npmsspeed [mqtalcccha (MQT 通信構成定義)] 376

O

ohostname [mqtalcccha (MQT 通信構成定義)] 369

oipaddr [mqtalcccha (MQT 通信構成定義)] 369

OpenTP1 システム, および MQ システム障害 647

OpenTP1 システム, および MQ システム障害 (ファーストメッセージ) 659

oportno [mqtalcccha (MQT 通信構成定義)] 369

oservname [mqtalcccha (MQT 通信構成定義)] 370

P

portno [mqtalcccha (MQT 通信構成定義)] 367

portno [mqtttcp (MQT 通信構成定義)] 347

portno [mqttcpcr (MQT 通信構成定義)] 354

portnum [mqtalcccha (MQT 通信構成定義)] 368

R

rcvbuf [mqtalcccha (MQT 通信構成定義)] 363

rcvbuf [mqtttcp (MQT 通信構成定義)] 346

rcvbuf [mqttcpcr (MQT 通信構成定義)] 354

rcvexdata [mqtalcccha (MQT 通信構成定義)] 372

rcvexit [mqtalcccha (MQT 通信構成定義)] 373

receive_from [MQC ゲートウェイサーバユーザサービス定義] 409

receiver 73

requester 73

RESET CLUSTER コマンドを実行できない場合の再参加 (オフライン) 203

RESET CLUSTER コマンドを実行できない場合の再参加 (オンライン) 202

RESET CLUSTER コマンドを実行できない場合の削除 201

RESET CLUSTER コマンドを実行できる場合の再参加 200

RESET CLUSTER コマンドを実行できる場合の削除 199

RESUME QMGR コマンド 189

RPC 49

rtim [mqtttcp (MQT 通信構成定義)] 348

rtim [mqttcpcr (MQT 通信構成定義)] 356

S

scyexdata [mqtalcccha (MQT 通信構成定義)] 373

scyexit [mqtalcccha (MQT 通信構成定義)] 373

sender 73

sendexit [mqtalcccha (MQT 通信構成定義)] 373

server 73

servname [mqtalcccha (MQT 通信構成定義)] 367

servname [mqtttcp (MQT 通信構成定義)] 347

servname [mqttcpcr (MQT 通信構成定義)] 355

sndbuf [mqtalcccha (MQT 通信構成定義)] 363

sndexdata [mqtalcccha (MQT 通信構成定義)] 372

sndjnl [mqtalcccha (MQT 通信構成定義)] 380

SPP 46

stim [mqttcpcr (MQT 通信構成定義)] 355

stim [mqttcpcs (MQT 通信構成定義)] 351

SUP 46

SUSPEND QMGR コマンド 189

SYSTEM.CLUSTER.COMMAND.QUEUE 193

SYSTEM.CLUSTER.REMOTE.QUEUE 193

SYSTEM.CLUSTER.REPOSITORY.QUEUE 193

SYSTEM.CLUSTER.TRANSMIT.QUEUE 193

T

TCP/IP 確立確認の回数 [dcmtcq_uoc_mqcd] 242

TCP/IP 確立確認の間隔 (秒数) [dcmtcq_uoc_mqcd] 242

TCP/IP 出力用バッファサイズ [dcmtcq_uoc_mqcd] 240

TCP/IP 入力用バッファサイズ [dcmtcq_uoc_mqcd] 240

tcprcvbuf [mqtalcccha (MQT 通信構成定義)] 374
tcpsndbuf [mqtalcccha (MQT 通信構成定義)] 374
TCP 構成定義 346
tim1 [mqtalcccha (MQT 通信構成定義)] 370
tim2 [mqtalcccha (MQT 通信構成定義)] 370
tim3 [mqtalcccha (MQT 通信構成定義)] 370
tim4cnt [mqtalcccha (MQT 通信構成定義)] 370
tim4 [mqtalcccha (MQT 通信構成定義)] 370
TP1/Message Queue Access 39
TP1/Message Queue で使用できる命令 47
TP1/Message Queue の運用コマンド 450
TP1/Message Queue の環境作成手順 266
TP1/Message Queue のジャーナルの取得契機 679
trccnt [mqttcp (MQT 通信構成定義)] 349
trccnt [mqttcpcr (MQT 通信構成定義)] 357
trccnt [mqttcps (MQT 通信構成定義)] 353
tretrycnt [mqtalcccha (MQT 通信構成定義)] 375
tretryint [mqtalcccha (MQT 通信構成定義)] 375
trn_crm_use [トランザクションサービス定義] 410
trn_rm_open_close_scope [MQC ゲートウェイ
サーバユーザサービス定義] 409
type [mqtalcccha (MQT 通信構成定義)] 360

U

UOC が動作するプロセス 252
UOC で使用できる関数 253
UOC とのインタフェース 222
UOC に渡されるデータ形式 220
UOC の異常処理 252
UOC の開始条件 220
UOC の組み込み 246
UOC の作成についての注意事項 252
UOC のスタック領域 252
UOC の呼び出し形式 222
UOC パラメタリスト 223
UOC 呼び出しの流れ 245

V

vretry [mqtalcccha (MQT 通信構成定義)] 371

W

watch_time [MQA サービス定義] 282

あ

相手システムのキューマネージャ名
[dcmtcq_uoc_mqcxp] 228
相手ホストの IP アドレス [dcmtcq_uoc_mqcd]
238
相手ホストのサービス名 [dcmtcq_uoc_mqcd]
238
相手ホストのポート番号 [dcmtcq_uoc_mqcd]
238
相手ホストのホスト名 [dcmtcq_uoc_mqcd] 238
空き 55
アプリケーション 31
アプリケーションからのメッセージ操作とキュー操作
46
アプリケーションからメッセージキューにアクセスす
る場合の条件 49
アプリケーション記述子 [mqaprcdef (MQA サー
ビス定義)] 318
アプリケーションタイプ [mqaprcdef (MQA サー
ビス定義)] 318
アプリケーションとキューマネージャの接続 47
アプリケーションの見直し 133
アプリケーション [用語解説] 697

い

一時的障害閉塞 55
一時的動的キュー 45
イニシエーションキュー監視 117
イニシエーションキュー障害 645
イニシエーションキューの障害 616
イニシエーションキュー名 [mqaqueatl (MQA サー
ビス定義)] 307
イニシエーションキュー名 [mqttcp (MQT 通信構
成定義)] 346

う

運用コマンドの概要 448

え

永続的動的キュー 45

お

応答キューの別名 147

オブジェクト属性の設定〔コマンド〕 518

オブジェクトの属性の設定（キューに登録の許可／禁止および取り出しの許可／禁止を設定） 57

オブジェクトの属性表示 57

オブジェクトの属性表示〔コマンド〕 490

オブジェクトハンドル 47

オブジェクト〔用語解説〕 697

か

開始要求応答受信監視タイム値
〔dcmtcq_uoc_mqcd〕 238

回線監視〔mqcenv (MQC サービス定義)〕 405

各チャネルタイプに指定できるオプション 383

確認メッセージ受信監視タイム値
〔dcmtcq_uoc_mqcd〕 238

環境データ〔mqaprcdef (MQA サービス定義)〕
319

環境不正によるリポジトリ情報の有効期限切れの通知
184

関連製品 39

き

機能フラグ〔dcmtcq_uoc_mqcxp〕 228

キュー、メッセージの削除〔コマンド〕 467

キューおよびキューファイルの障害とユーザの処理
611

キュー記述子〔mqaalsque (MQA サービス定義)〕
315

キュー記述子〔mqaqueatl (MQA サービス定義)〕
303

キュー記述子〔mqaremque (MQA サービス定義)〕
312

キューとキューファイルの関係 52

キューの削除 54

キューの作成〔コマンド〕 507

キューの作成と削除 53

キューの使用方法による分類 44

キューの説明 43

キューの属性 45

キューの属性の変更 54

キューの属性の変更〔コマンド〕 459

キューファイル 51

キューファイルグループ 51

キューファイルグループ監視の要否〔mqaquegrp
(MQA サービス定義)〕 300

キューファイルグループ登録メッセージサイズ回復値
〔mqaquegrp (MQA サービス定義)〕 301

キューファイルグループ登録メッセージサイズしきい
値〔mqaquegrp (MQA サービス定義)〕 301

キューファイルグループ内のキューファイル状態変更
〔コマンド〕 485

キューファイルグループの構成定義 297

キューファイルグループの使用状況表示 69

キューファイルグループの使用状況表示〔コマンド〕
469

キューファイルグループ名〔mqaquegrp (MQA
サービス定義)〕 297

キューファイル内の情報表示 58

キューファイル内の情報表示〔コマンド〕 472

キューファイル二重化構成 67

キューファイルの回復 58

キューファイルの回復〔コマンド〕 481

キューファイルの切り離し〔コマンド〕 513

キューファイルの組み込み〔コマンド〕 453

キューファイルの組み込みと切り離し 58

キューファイルのコンバート〔コマンド〕 462

キューファイルの削除 67

キューファイルの削除〔コマンド〕 464

キューファイルの作成 53

キューファイルの状態 54

キューファイルの情報表示 69

キューファイルの情報表示〔コマンド〕 478

キューファイルの入出力エラー 616

- キューファイルのバックアップ [コマンド] 458
- キューファイルのバックアップとリストア 58
- キューファイルの閉塞解除 [コマンド] 511
- キューファイルのリストア [コマンド] 516
- キューファイルの論理閉塞と閉塞解除 58
- キューファイルの割り当て [コマンド] 487
- キューファイル名 [mqaqueatl (MQA サービス定義)] 310
- キューファイル名 [mqaquegrp (MQA サービス定義)] 302
- キューファイル [用語解説] 697
- キュー保持時間 [mqaqueatl (MQA サービス定義)] 306
- キューマネージャ 31
- キューマネージャからクラスタキューを削除 198
- キューマネージャ記述子 [mqaquemgr (MQA サービス定義)] 321
- キューマネージャ定義 321
- キューマネージャの脱退 178
- キューマネージャの別名 146
- キューマネージャのリフレッシュ 177
- キューマネージャ名 [mqaquemgr (MQA サービス定義)] 321
- キューマネージャ [用語解説] 698
- 強制停止 441
- 共用性 [mqaqueatl (MQA サービス定義)] 305
- 共用メモリの見積もり式 666
- クラスタ環境のシステムキュー 193
- クラスタ環境の通信構成 166
- クラスタ管理の検討事項 177
- クラスタキュー 142
- クラスタキュー登録不可時の処理 179
- クラスタキューの属性を変更 203
- クラスタキューマネージャ [用語解説] 698
- クラスタキュー [用語解説] 698
- クラスタ使用時の障害対策 208
- クラスタ情報の表示 [コマンド] 530
- クラスタ設計の検討事項 172
- クラスタセンダ TCP 構成定義 351
- クラスタセンダチャンネル 167
- クラスタセンダチャンネルの自動定義 169
- クラスタセンダチャンネル [用語解説] 698
- クラスタセンダプロセス 167
- クラスタチャンネル情報待ち時間 191
- クラスタチャンネルの削除 171
- クラスタチャンネルの属性変更 171
- クラスタチャンネルの属性を変更 204
- クラスタで使用する定義コマンド 164
- クラスタでネットワークを設定する定義 132
- クラスタ転送キュー 142
- クラスタ転送キューの保守 178
- クラスタ転送キュー [用語解説] 699
- クラスタ内共用クラスタ名称 [mqaalsque (MQA サービス定義)] 317
- クラスタ内共用クラスタ名称 [mqaremque (MQA サービス定義)] 314
- クラスタ内での別名の使用例 148
- クラスタに関連する別名の使用とリモートキューのローカル定義 146
- クラスタの概要 126
- クラスタの管理 172
- クラスタの機能 140
- クラスタの構成要素 140
- クラスタの使用によるシステム管理の軽減 130
- クラスタの使用方法 130
- クラスタの設定方法 133

<

- クラスタ運用時の注意事項 196
- クラスタ外のキューマネージャからの応答 152
- クラスタ外のキューマネージャによるキューへのメッセージ登録 149
- クラスタ外のキューマネージャによるキューへのメッセージ登録 (別名使用) 150
- クラスタ外のキューマネージャへの応答 149
- クラスタ外のキューマネージャへのメッセージ登録 150
- クラスタからキューマネージャを削除および再参加 199
- クラスタからクラスタへのメッセージ登録 152
- クラスタからの脱退 [コマンド] 550

クラスタのセットアップ 134
クラスタの長所 128
クラスタへの新しいキューマネージャの追加 138
クラスタへのゲートウェイとしてのキューマネージャの使用 147
クラスタ名 [dcmtcq_uoc_mqcd] 236
クラスタ [用語解説] 698
クラスタリングの動作 144
クラスタレシーバ TCP 構成定義 354
クラスタレシーバチャンネル 168
クラスタレシーバチャンネルのマルチインスタンス 170
クラスタレシーバチャンネル [用語解説] 699
クラスタレシーバプロセス 168

け

計画停止 A 441
計画停止 B 441
継続セグメント受信監視タイマ値 [dcmtcq_uoc_mqcd] 238
継続セグメント受信監視タイマ値 [mqcenv (MQC サービス定義)] 404
継続メッセージ受信監視タイマ値 [dcmtcq_uoc_mqcd] 239
経路の選択 163
ゲートウェイキューマネージャ 147

こ

構造体識別子 [dcmtcq_uoc_mqcxp] 225
構造体識別子 [dcmtcq_uoc_parmlist] 223
構造体バージョン番号 [dcmtcq_uoc_mqcd] 235
構造体バージョン番号 [dcmtcq_uoc_mqcxp] 225
構造体バージョン番号 [dcmtcq_uoc_parmlist] 223
コーラ 72
コーラとレスポンド 72
コネクション障害 (コーラ側) 636
コネクション障害 (コーラ側・ファーストメッセージ) 652
コネクション障害 (レスポンド側) 637

コネクション障害 (レスポンド側・ファーストメッセージ) 652
コネクションハンドル 47

さ

サーバ 73
サービス名ファイル [用語解説] 699
最大セグメントサイズ [dcmtcq_uoc_mqcd] 236
最大ハンドル数 [mqaquemgr (MQA サービス定義)] 322
最大メッセージ長 [dcmtcq_uoc_mqcd] 236
最大メッセージ長 [mqaqueatl (MQA サービス定義)] 305
最大メッセージ長 [mqaquemgr (MQA サービス定義)] 323
最大メッセージ登録数 [mqaqueatl (MQA サービス定義)] 305
最大メッセージ優先度 [mqaquemgr (MQA サービス定義)] 322

し

時間監視機能 114
時間監視障害 645
時間監視障害 (ファーストメッセージ) 657
自システムのサービス名 [mqcenv (MQC サービス定義)] 403
自システムのサービス名 [mqcgwpsnam (MQC サービス定義)] 406
自システムのポート番号 [mqcenv (MQC サービス定義)] 403
自システムのポート番号 [mqcgwpsnam (MQC サービス定義)] 406
自システムのポート番号使用数 [dcmtcq_uoc_mqcd] 242
システム起動 83
システム構成例 412
システムサービス情報定義の作成 250
自ホストの IP アドレス [dcmtcq_uoc_mqcd] 237
自ホストのサービス名 [dcmtcq_uoc_mqcd] 238
自ホストのポート番号 [dcmtcq_uoc_mqcd] 238
自ホストのホスト名 [dcmtcq_uoc_mqcd] 237

終了処理監視 116
終了処理監視タイマ値 [mqcenv (MQC サービス定義)] 404
受信出口ユーザデータ [dcmtcq_uoc_mqcd] 239
受信バッファグループ番号 [dcmtcq_uoc_mqcd] 237
受信バッファ方式 103
障害時に取得する情報 660
障害閉塞 55
使用種別 [mqaqueatl (MQA サービス定義)] 307
使用ライブラリの制限 252
省略時の転送キュー名 [mqaquemgr (MQA サービス定義)] 322
省略時の取り出しオプション [mqaqueatl (MQA サービス定義)] 306
省略時のメッセージ永続性 [mqaalsque (MQA サービス定義)] 316
省略時のメッセージ永続性 [mqaqueatl (MQA サービス定義)] 304
省略時のメッセージ永続性 [mqaremque (MQA サービス定義)] 313
省略時のメッセージ優先度 [mqaalsque (MQA サービス定義)] 316
省略時のメッセージ優先度 [mqaqueatl (MQA サービス定義)] 304
省略時のメッセージ優先度 [mqaremque (MQA サービス定義)] 313

せ

正常 54
正常開始する場合の準備 112
正常終了 441
セキュリティ出口名 [dcmtcq_uoc_mqcd] 241
セキュリティ出口ユーザデータ [dcmtcq_uoc_mqcd] 241
セグメント 102
セグメント最大長 [dcmtcq_uoc_mqcxp] 227
切断時間間隔 117
切断時間間隔 [dcmtcq_uoc_mqcd] 239
切断時間間隔使用の有無 [dcmtcq_uoc_mqcd] 239

センダ 73

そ

送受信監視 116
送受信バッファリストのアドレス [dcmtcq_uoc_parmlist] 224
送信チャンネルジャーナル出力条件 [dcmtcq_uoc_mqcd] 242
送信出口ユーザデータ [dcmtcq_uoc_mqcd] 239
送信バッファグループ番号 [dcmtcq_uoc_mqcd] 237
送信バッファ方式 102
総バッチメッセージ数 [mqamqtnam (MQA サービス定義)] 325
属性 [用語解説] 699

た

タイマ定義 342
タイムアウト時チャンネル確立再試行の要否 [dcmtcq_uoc_mqcd] 239
短期確立再試行の回数 [dcmtcq_uoc_mqcd] 241
短期確立再試行の間隔 [dcmtcq_uoc_mqcd] 241
短期確立再試行の要否 [dcmtcq_uoc_mqcd] 241

ち

チャンネル 71
チャンネル解放中 75
チャンネル確立拒否応答 86
チャンネル確立再試行 86
チャンネル確立中 75
チャンネル確立方式 [dcmtcq_uoc_mqcd] 237
チャンネル確立リクエスト中 75
チャンネル確立リトライ中 75
チャンネル管理情報格納ファイル障害 648
チャンネル終了定義 390
チャンネル受信出口名 [dcmtcq_uoc_mqcd] 241
チャンネル状態 74
チャンネル状態の不一致時の強制再接続条件 [dcmtcq_uoc_mqcd] 240
チャンネル使用不可 75

チャンネル送信出口名 [dcmtcq_uoc_mqcd] 241
チャンネル属性の変更 [コマンド] 553
チャンネルタイプ 72
チャンネルタイプ [dcmtcq_uoc_mqcd] 236
チャンネル定義 359
チャンネル定義の更新時刻 [dcmtcq_uoc_mqcd] 240
チャンネル停止 74
チャンネルデータ定義ブロック 229
チャンネルデータ定義ブロック (dcmtcq_uoc_mqcd) の長さ [dcmtcq_uoc_mqcd] 242
チャンネルデータ定義ブロックのアドレス [dcmtcq_uoc_parmlist] 224
チャンネル出口パラメタブロック 224
チャンネル出口パラメタブロックのアドレス [dcmtcq_uoc_parmlist] 224
チャンネル動作中 75
チャンネルの開始 83
チャンネルの開始 [コマンド] 597
チャンネルの解放 99
チャンネルの確立 86
チャンネルの機能 71
チャンネルの終了 [コマンド] 603
チャンネルの状態表示 [コマンド] 576
チャンネルのテスト接続 [コマンド] 587
チャンネルのネゴシエーション 388
チャンネル名 [dcmtcq_uoc_mqcd] 235
チャンネル名 [mqtalcccha (MQT 通信構成定義)] 360
チャンネル要求監視 117
チャンネル [用語解説] 699
注釈 [dcmtcq_uoc_mqcd] 242
長期確立再試行の回数 [dcmtcq_uoc_mqcd] 237
長期確立再試行の間隔 [dcmtcq_uoc_mqcd] 237
長期確立再試行の要否 [dcmtcq_uoc_mqcd] 237

つ

通常メッセージ 124
通信障害 636

て

定義情報の作成手順 268
定義済みキュー 45
定義済みキューの作成 53
定義タイプ [mqaqueatl (MQA サービス定義)] 307
定義の記述形式 274
出口からの応答 [dcmtcq_uoc_mqcxp] 226
出口からの二次応答 [dcmtcq_uoc_mqcxp] 227
出口障害 647
出口障害 (ファーストメッセージ) 658
出口データ [dcmtcq_uoc_mqcxp] 227
出口のタイプ [dcmtcq_uoc_mqcxp] 225
出口ユーザ領域 [dcmtcq_uoc_mqcxp] 227
出口を呼び出すための二次理由 [dcmtcq_uoc_mqcxp] 226
出口を呼び出すための理由 [dcmtcq_uoc_mqcxp] 225
デッドレターキュー名 [mqaquemgr (MQA サービス定義)] 322
デフォルトチャンネル定義 381
デフォルトのバインディング [mqaalsque (MQA サービス定義)] 316
デフォルトのバインディング [mqaqueatl (MQA サービス定義)] 309
デフォルトのバインディング [mqaremque (MQA サービス定義)] 313
転送キュー監視 117
転送キュー監視回数 [dcmtcq_uoc_mqcd] 239
転送キュー監視間隔 [dcmtcq_uoc_mqcd] 238
転送キュー監視方式 [dcmtcq_uoc_mqcd] 240
転送キュー名 [dcmtcq_uoc_mqcd] 236
転送キュー名 [mqaremque (MQA サービス定義)] 314
転送キュー名 [mqtalcccha (MQT 通信構成定義)] 361

と

動的キュー 45
動的キューの作成 54

同名クラスタキュー 154
登録許可 [mqaalsque (MQA サービス定義)] 316
登録許可 [mqaqueatl (MQA サービス定義)] 304
登録許可 [mqaremque (MQA サービス定義)] 312
トランザクションサービス定義 410
トランザクションとメッセージキューアクセスとの関係 49
トランザクションブランチ終了監視タイマ値 [mqcenv (MQC サービス定義)] 405
トランザクション有効性 [mqaquemgr (MQA サービス定義)] 323
トリガイベント 49
トリガ間隔 [mqaquemgr (MQA サービス定義)] 321
トリガ起動処理障害 649
トリガ機能 49
トリガ制御 [mqaqueatl (MQA サービス定義)] 307
トリガタイプ [mqaqueatl (MQA サービス定義)] 308
トリガデータ [mqaqueatl (MQA サービス定義)] 309
トリガのためのメッセージ登録数の下限値 [mqaqueatl (MQA サービス定義)] 309
トリガのためのメッセージ優先度の下限値 [mqaqueatl (MQA サービス定義)] 308
トリガメッセージ 49
トリガモニタアプリケーション 50
トリガ [用語解説] 700
取り出し許可 [mqaalsque (MQA サービス定義)] 315
取り出し許可 [mqaqueatl (MQA サービス定義)] 304

に

二重受信メッセージの破棄 112
入出力バッファ数 [mqaquegrp (MQA サービス定義)] 298

ね

ネゴシエーション 86

ネットワーク接続優先順位 [dcmtcq_uoc_mqcd] 240
ネットワークリクエスト 83

は

バージョンアップ時の注意 252
ハートビート間隔 121
ハートビート間隔 [dcmtcq_uoc_mqcd] 239
ハートビート機能 119
ハートビートメッセージ 119
ハートビートメッセージ [用語解説] 700
配布リスト 122
配布リストのサポート [mqaqueatl (MQA サービス定義)] 309
配布リストのサポート [mqaquemgr (MQA サービス定義)] 323
バインディング [用語解説] 700
バックアップキューファイル名 [mqaquegrp (MQA サービス定義)] 297
バッチ 113
バッチサイズ 113
バッチサイズ [dcmtcq_uoc_mqcd] 236
バッチサイズ [mqtalcccha (MQT 通信構成定義)] 361
バッチ終了待ちタイマ (バッチインターバル) 119
バッチ終了待ちタイマ値 (ミリ秒) [dcmtcq_uoc_mqcd] 242
バッチ終了待ちタイマ値 [dcmtcq_uoc_mqcd] 239
バッチ転送 113
バッファアドレス [dcmtcq_uoc_bufflist] 244
バッファグループ定義 343
バッファ情報領域 [dcmtcq_uoc_bufflist] 243
バッファ数 [dcmtcq_uoc_bufflist] 243
バッファ方式 [dcmtcq_uoc_mqcd] 236
バッファリスト 243

ひ

非永続メッセージ転送速度 [dcmtcq_uoc_mqcd] 240

ふ

- ファーストチャネル 124
- ファーストメッセージ 124
- ファーストメッセージ機能 124
- ファーストメッセージ障害 652
- ファイル障害 611
- ファイル二重化の回復 [コマンド] 509
- フィードバックコード [dcmtcq_uoc_mqcxp] 227
- 負荷分散 155
- フルリポジトリキューマネージャ [用語解説] 700
- フルリポジトリとパーシャルリポジトリ [用語解説] 700
- フルリポジトリを保持するキューマネージャの選択 172
- プロセス記述子 [mqaprcdef (MQA サービス定義)] 318
- プロセス定義 318
- プロセス名 [mqaprcdef (MQA サービス定義)] 319
- プロセス名 [mqaqueatl (MQA サービス定義)] 305
- プロトコル名称 [mqamqtnam (MQA サービス定義)] 326
- プロトコルレベル [dcmtcq_uoc_mqcxp] 228
- 分散キューイング 126
- 分散キューイングでネットワークを設定する定義 131

へ

- ベースキュー名 [mqaalsque (MQA サービス定義)] 317
- 別名キュー 148
- 別名キューの属性定義 315
- 別名キュー名 [mqaalsque (MQA サービス定義)] 317
- 変換後 CCSID [dcmtcq_uoc_mqcd] 240
- 編集用バッファエントリ番号 [dcmtcq_uoc_bufflist] 243
- 編集用バッファリストのアドレス [dcmtcq_uoc_parmlist] 224

ほ

- ポート番号 [mqttenv (MQT 通信構成定義)] 341
- ホスト名ファイル [用語解説] 701

ま

- マルチキャスト機能 122
- マルチクラスタ 152

め

- 命令 46
- 命令 [用語解説] 701
- メッセージエントリ確保率 [mqaquegrp (MQA サービス定義)] 300
- メッセージキューイング 30
- メッセージキューイング機能 42
- メッセージキューイング機能の特長 42
- メッセージキューイング機能 [用語解説] 701
- メッセージキューイングの基本構成要素 42
- メッセージキュー [用語解説] 701
- メッセージシーケンス番号 105
- メッセージシーケンス番号でのメッセージの管理 105
- メッセージシーケンス番号の最大値 [dcmtcq_uoc_mqcd] 236
- メッセージシーケンス番号の最大値 [mqtalccha (MQT 通信構成定義)] 362
- メッセージシーケンス番号のリセット 112
- メッセージシーケンス番号のリセット [コマンド] 593
- メッセージ受信時のあて先の決定 147
- メッセージ受信中の障害 616
- メッセージ受信の流れ 103
- メッセージ情報管理ファイル 51
- メッセージ送受信機能 101
- メッセージ送受信中の障害 615
- メッセージ送信時のキューマネージャの再マッピング 146
- メッセージ送信時の転送キューの変更または指定 146
- メッセージ送信中の障害 615
- メッセージ送信の流れ 101
- メッセージ送達未確認の解決 [コマンド] 590

メッセージ滞留時間監視の要否 [mqaqueatl (MQA サービス定義)] 310
メッセージチャネルエージェント 31
メッセージチャネルエージェント [用語解説] 701
メッセージチャネルプロトコル 37
メッセージ出口名 [dcmtcq_uoc_mqcd] 241
メッセージとセグメントの関係 104
メッセージの削除 70
メッセージの削除 [コマンド] 465
メッセージの受信 103
メッセージの送信 101
メッセージの送達未確認状態 111
メッセージの送達未確認状態の解決 112
メッセージの表示 70
メッセージの表示 [コマンド] 501
メッセージ配布順序 [mqaqueatl (MQA サービス定義)] 306
メッセージファイル 51
メッセージ編集出口 UOC 220
メッセージ編集出口 UOC を使用したファイル転送 211
メッセージ編集出口バッファグループ番号 [dcmtcq_uoc_mqcd] 237
メッセージ編集出口バッファ数 [dcmtcq_uoc_mqcd] 237
メッセージ編集出口ユーザデータ [dcmtcq_uoc_mqcd] 239
メッセージ類似性 129
メッセージ類似性の処置 158
メモリ不足 640

も

モデルキューの属性定義 303
モデルキューの定義名 [mqaqueatl (MQA サービス定義)] 310

ゆ

ユーザOWNコーディング 219
ユーザデータ [mqaprcdef (MQA サービス定義)] 319

よ

要求受信監視タイム値 [mqcenv (MQC サービス定義)] 404

り

リクエスト 73
リポジトリ管理サーバ 189
リポジトリ管理サーバ起動待ち合わせ時間 191
リポジトリ管理サーバの開始方法 193
リポジトリ管理サーバの環境変数 190
リポジトリ管理サーバの作成 189
リポジトリ管理サーバの動作 189
リポジトリ管理サーバユーザサービス定義ファイル 190
リポジトリ管理の構成 189
リポジトリ情報送信契機の通知 186
リポジトリ情報の再作成 [コマンド] 546
リポジトリ情報の保持期間 180
リポジトリへの登録失敗時の処理 179
リポジトリ [用語解説] 701
リモートキューのローカル定義 312
リモートキューのローカル定義名 [mqaremque (MQA サービス定義)] 314
リモートキューマネージャ名 [mqaremque (MQA サービス定義)] 314
リモートキュー名 [mqaremque (MQA サービス定義)] 313

れ

レシーバ 73
レスポンド 72

ろ

ローカルキューの作成方法による分類 45
論理閉塞 55

わ

ワークロード管理アルゴリズム 162
ワークロード管理へのクラスタの使用 154