

OpenTP1 Version 7

OpenTP1 メッセージキューイング機能

TP1/Message Queue プログラム作成リファレンス

文法書

3000-3-D93-50

---

## 前書き

### ■ 対象製品

マニュアル「TP1/Message Queue 使用の手引」を参照してください。

### ■ 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-D93-50

### ■ 著作権

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

(C) Copyright International Business Machines Corporation 1994, 2003. All rights reserved.

## 変更内容

### 変更内容 (3000-3-D93-50) TP1/Message Queue 07-60

追加・変更内容	変更箇所
記載内容は変更なし。	—

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-D93-40) TP1/Message Queue 07-53

追加・変更内容
記載内容は変更なし。

### 変更内容 (3000-3-D93-31) TP1/Message Queue 07-52

追加・変更内容
対象製品の記載を、マニュアル「TP1/Message Queue 使用の手引」を参照する内容に変更した。
MQGMO_CONVERT の説明を変更した。
次の理由コードの説明を変更した。 <ul style="list-style-type: none"><li>• MQRC_SOURCE_CCSID_ERROR</li><li>• MQRC_TARGET_CCSID_ERROR</li><li>• MQRC_NOT_CONVERTED</li></ul>
コード変換対象の文字コードセットの次の説明を変更した。 <ul style="list-style-type: none"><li>• 注※4 の変換対象外の製品</li></ul>
日立コード変換を使用した場合の文字コード変換の対応 (HP-UX) の次の説明を変更した。 <ul style="list-style-type: none"><li>• 凡例</li></ul>

### 変更内容 (3000-3-D93-30) TP1/Message Queue 07-52

追加・変更内容
コード変換対象の文字コードセットの次の説明を変更した。 <ul style="list-style-type: none"><li>• 注※2 の変換対象の製品</li><li>• 注※3 の変換対象外の製品</li></ul>

### 変更内容 (3000-3-D93-20) TP1/Message Queue 07-50

追加・変更内容
注意事項に説明を追加した。

## 変更内容 (3000-3-D93-10) TP1/Message Queue 07-01

追加・変更内容
MQPTR 型のデータタイプの説明を変更した。
COBOL 言語で追加が必要なフィールドについて説明を追加した。
MQMD 構造体の Feedback フィールドの、次の項目の説明を変更した。 <ul style="list-style-type: none"><li>• MQFB_DATA_LENGTH_ZERO</li><li>• MQFB_DATA_LENGTH_NEGATIVE</li><li>• MQFB_DATA_LENGTH_TOO_BIG</li><li>• MQFB_BUFFER_OVERFLOW</li><li>• MQFB_LENGTH_OFF_BY_ONE</li><li>• MQFB_IH_ERROR</li><li>• MQFB_NOT_AUTHORIZED_FOR_IMS</li><li>• MQFB_IMS_ERROR</li><li>• MQFB_IMS_FIRST</li><li>• MQFB_IMS_LAST</li><li>• MQFB_CICS_APPL_ABENDED</li><li>• MQFB_CICS_APPL_NOT_STARTED</li><li>• MQFB_CICS_BRIDGE_FAILURE</li><li>• MQFB_CICS_CCSID_ERROR</li><li>• MQFB_CICS_CIH_ERROR</li><li>• MQFB_CICS_COMMAREA_ERROR</li><li>• MQFB_CICS_CORREL_ID_ERROR</li><li>• MQFB_CICS_DLQ_ERROR</li><li>• MQFB_CICS_ENCODING_ERROR</li><li>• MQFB_CICS_INTERNAL_ERROR</li><li>• MQFB_CICS_NOT_AUTHORIZED</li><li>• MQFB_CICS_UOW_BACKED_OUT</li><li>• MQFB_CICS_UOW_ERROR</li></ul>
MQMD 構造体の次の値の説明を変更した。 <ul style="list-style-type: none"><li>• Persistence フィールドの MQPER_PERSISTENCE_AS_Q_DEF</li><li>• Priority フィールドの MQPRI_PRIORITY_AS_Q_DEF</li></ul>
理由コードの数値 (10 進数) から「L」を削除した。
次の理由コードの説明を変更した (括弧内は 10 進数の値)。 <ul style="list-style-type: none"><li>• MQRC_CONNECTION_BROKEN (2009)</li><li>• MQRC_Q_MGR_NAME_ERROR (2058)</li><li>• MQRC_Q_MGR_NOT_AVAILABLE (2059)</li><li>• MQRC_SECOND_MARK_NOT_ALLOWED (2062)</li><li>• MQRC_SYNCPOINT_NOT_AVAILABLE (2072)</li><li>• MQRC_STORAGE_CLASS_ERROR (2105)</li></ul>

## 追加・変更内容

- MQRC\_COD\_NOT\_VALID\_FOR\_XCF\_Q (2106)
- MQRC\_XWAIT\_CANCELED (2107)
- MQRC\_XWAIT\_ERROR (2108)
- MQRC\_NOT\_CONVERTED (2119)
- MQRC\_ADAPTER\_STORAGE\_SHORTAGE (2127)
- MQRC\_ADAPTER\_CONN\_LOAD\_ERROR (2129)
- MQRC\_ADAPTER\_SERV\_LOAD\_ERROR (2130)
- MQRC\_ADAPTER\_DEFS\_ERROR (2131)
- MQRC\_ADAPTER\_DEFS\_LOAD\_ERROR (2132)
- MQRC\_ADAPTER\_CONV\_LOAD\_ERROR (2133)
- MQRC\_ADAPTER\_DISC\_LOAD\_ERROR (2138)
- MQRC\_CICS\_WAIT\_FAILED (2140)
- MQRC\_ASID\_MISMATCH (2157)
- MQRC\_CONN\_ID\_IN\_USE (2160)
- MQRC\_Q\_MGR\_STOPPING (2162)
- MQRC\_DUPLICATE\_RECOV\_COORD (2163)
- MQRC\_API\_EXIT\_LOAD\_ERROR (2183)
- MQRC\_CICS\_BRIDGE\_RESTRICTION (2187)
- MQRC\_CLUSTER\_RESOLUTION\_ERROR (2189)
- MQRC\_PAGESET\_FULL (2192)
- MQRC\_STORAGE\_MEDIUM\_FULL (2192)
- MQRC\_PAGESET\_ERROR (2193)
- MQRC\_NAME\_IN\_USE (2201)
- MQRC\_CONNECTION QUIESCING (2202)
- MQRC\_CONNECTION\_STOPPING (2203)
- MQRC\_ADAPTER\_NOT\_AVAILABLE (2204)
- MQRC\_CONNECTION\_NOT\_AUTHORIZED (2217)
- MQRC\_Q\_MGR\_ACTIVE (2222)
- MQRC\_CONN\_TAG\_IN\_USE (2271)
- MQRC\_CONNECTION\_ERROR (2273)
- MQRC\_OPTION\_ENVIRONMENT\_ERROR (2274)
- MQRC\_CD\_ERROR (2277)
- MQRC\_HCONFIG\_ERROR (2280)
- MQRC\_FUNCTION\_ERROR (2281)
- MQRC\_SERVICE\_NOT\_AVAILABLE (2285)
- MQRC\_INITIALIZATION\_FAILED (2286)
- MQRC\_TERMINATION\_FAILED (2287)
- MQRC\_UNKNOWN\_Q\_NAME (2288)
- MQRC\_SERVICE\_ERROR (2289)
- MQRC\_Q\_ALREADY\_EXISTS (2290)

## 追加・変更内容

- MQRC\_USER\_ID\_NOT\_AVAILABLE (2291)
- MQRC\_UNKNOWN\_ENTITY (2292)
- MQRC\_UNKNOWN\_AUTH\_ENTITY (2293)
- MQRC\_UNKNOWN\_REF\_OBJECT (2294)
- MQRC\_CHANNEL\_NOT\_ACTIVATED (2296)
- MQRC\_UOW\_CANCELED (2297)
- MQRC\_MSG\_TOKEN\_ERROR (2331)
- MQRC\_MISSING\_WIH (2332)
- MQRC\_WIH\_ERROR (2333)
- MQRC\_CHAR\_CONVERSION\_ERROR (2340)
- MQRC\_UCS2\_CONVERSION\_ERROR (2341)
- MQRC\_DB2\_NOT\_AVAILABLE (2342)
- MQRC\_OBJECT\_NOT\_UNIQUE (2343)
- MQRC\_CONN\_TAG\_NOT\_RELEASED (2344)
- MQRC\_CF\_NOT\_AVAILABLE (2345)
- MQRC\_CF\_STRUC\_IN\_USE (2346)
- MQRC\_CF\_STRUC\_LIST\_HDR\_IN\_USE (2347)
- MQRC\_CF\_STRUC\_AUTH\_FAILED (2348)
- MQRC\_CF\_STRUC\_ERROR (2349)
- MQRC\_CONN\_TAG\_NOT\_USABLE (2350)
- MQRC\_UOW\_MIX\_NOT\_SUPPORTED (2355)
- MQRC\_OBJECT\_LEVEL\_INCOMPATIBLE (2360)
- MQRC\_BACKOUT\_THRESHOLD\_REACHED (2362)
- MQRC\_SEGMENTS\_NOT\_SUPPORTED (2365)
- MQRC\_WRONG\_CF\_LEVEL (2366)
- MQRC\_CF\_STRUC\_FAILED (2373)
- MQRC\_NO\_DATA\_AVAILABLE (2379)
- MQRC\_Q\_INDEX\_TYPE\_ERROR (2394)
- MQRC\_REOPEN\_EXCL\_INPUT\_ERROR (6100)
- MQRC\_REOPEN\_INQUIRE\_ERROR (6101)
- MQRC\_REOPEN\_SAVED\_CONTEXT\_ERR (6102)
- MQRC\_REOPEN\_TEMPORARY\_Q\_ERROR (6103)
- MQRC\_ATTRIBUTE\_LOCKED (6104)
- MQRC\_CURSOR\_NOT\_VALID (6105)
- MQRC\_ENCODING\_ERROR (6106)
- MQRC\_STRUC\_ID\_ERROR (6107)
- MQRC\_NULL\_POINTER (6108)
- MQRC\_NO\_CONNECTION\_REFERENCE (6109)
- MQRC\_NO\_BUFFER (6110)
- MQRC\_BINARY\_DATA\_LENGTH\_ERROR (6111)

## 追加・変更内容

- MQRC\_BUFFER\_NOT\_AUTOMATIC (6112)
- MQRC\_INSUFFICIENT\_BUFFER (6113)
- MQRC\_INSUFFICIENT\_DATA (6114)
- MQRC\_DATA\_TRUNCATED (6115)
- MQRC\_ZERO\_LENGTH (6116)
- MQRC\_NEGATIVE\_LENGTH (6117)
- MQRC\_NEGATIVE\_OFFSET (6118)
- MQRC\_INCONSISTENT\_FORMAT (6119)
- MQRC\_INCONSISTENT\_OBJECT\_STATE (6120)
- MQRC\_CONTEXT\_OBJECT\_NOT\_VALID (6121)
- MQRC\_CONTEXT\_OPEN\_ERROR (6122)
- MQRC\_STRUC\_LENGTH\_ERROR (6123)
- MQRC\_NOT\_CONNECTED (6124)
- MQRC\_NOT\_OPEN (6125)
- MQRC\_DISTRIBUTION\_LIST\_EMPTY (6126)
- MQRC\_INCONSISTENT\_OPEN\_OPTIONS (6127)
- MQRC\_WRONG\_VERSION (6128)
- MQRC\_REFERENCE\_ERROR (6129)

次の理由コードの説明を追加した（括弧内は 10 進数の値）。

- MQRC\_API\_EXIT\_NOT\_FOUND (2182)
- MQRC\_FILE\_NOT\_AUDITED (2216)
- MQRC\_PARTIALLY\_CONVERTED (2272)
- MQRC\_NEXT\_RECORD\_ERROR (2361)

リンケージ方法の説明を、OS 別に記載した。

適用 OS から Solaris を削除した。

適用 OS が 64 ビット版の場合の説明を追加した。

日立コード変換を使用した場合の文字コードの対応について説明を追加した。

## はじめに

このマニュアルは、TP1/Message Queue を使用する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：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 を使用するアプリケーションプログラマ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。

- TP1/Message Queue 使用の手引 (3000-3-D90)
- TP1/Message Queue プログラム作成の手引 (3000-3-D92)

### ■ 謝辞

COBOL 言語仕様は、CODASYL (the Conference on Data Systems Languages : データシステムズ言語協議会) によって、開発された。OpenTP1 のアプリケーションプログラムのインタフェース仕様のうち、データ操作言語 (DML Data Manipulation Language) の仕様は、CODASYL COBOL (1981) の通信節、RECEIVE 文、SEND 文、COMMIT 文、及び ROLLBACK 文を参考にし、それに日立製作所独自の解釈と仕様を追加して開発した。原開発者に対し謝意を表すとともに、CODASYL の要求に従って以下の謝辞を掲げる。なお、この文章は、COBOL の原仕様書「CODASYL COBOL JOURNAL OF DEVELOPMENT 1984」の謝辞の一部を再掲するものである。

いかなる組織であっても、COBOL の原仕様書とその仕様の全体又は一部分を複製すること、マニュアルその他の資料のための土台として原仕様書のアイデアを利用することは自由である。ただし、その場合には、その刊行物のまえがきの一部として、次の謝辞を掲載しなければならない。書評などに短い文章を引用するときは、"COBOL" という名称を示せば謝辞全体を掲載する必要はない。

COBOL は産業界の言語であり、特定の団体や組織の所有物ではない。

CODASYL COBOL 委員会又は仕様変更の提案者は、このプログラミングシステムと言語の正確さや機能について、いかなる保証も与えない。さらに、それに関連する責任も負わない。

次に示す著作権表示付資料の著作者及び著作権者

FLOW-MATIC (Sperry Rand Corporation の商標),

Programming for the Univac (R) I and II,

Data Automation Systems, Sperry Rand Corporation 著作権表示 1958 年, 1959 年 ;

IBM Commercial Translator Form No.F 28-8013, IBM 著作権表示 1959 年;

FACT, DSI 27A5260-2760, Minneapolis-Honeywell, 著作権表示 1960 年

は、これら全体又は一部分を COBOL の原仕様書中に利用することを許可した。この許可は、COBOL 原仕様書をプログラミングマニュアルや類似の刊行物に複製したり、利用したりする場合にまで拡張される。

## ■ 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「TP1/Message Queue 使用の手引」を参照してください。

# 目次

前書き	2
変更内容	3
はじめに	8

## 1 データタイプ 12

データタイプの概要	13
MQCFH 構造体 - PCF ヘッダ	17
MQCFIL 構造体 - PCF 整数リスト引数	20
MQCFIN 構造体 - PCF 整数引数	22
MQCFSL 構造体 - PCF 文字列リスト引数	24
MQCFST 構造体 - PCF スtring引数	27
MQDH 構造体 - 配布ヘッダ	30
MQDLH 構造体 - デッドレターヘッダ	36
MQGMO 構造体 - メッセージ取り出しオプション	44
MQMD 構造体 - メッセージ記述子	72
MQMDE 構造体 - メッセージ記述子の拡張	124
MQOD 構造体 - オブジェクト記述子	130
MQOR 構造体 - オブジェクトレコード	142
MQPMO 構造体 - メッセージ登録オプション	144
MQPMR 構造体 - 登録メッセージレコード	164
MQRMH 構造体 - メッセージ参照ヘッダ	167
MQRR 構造体 - 応答レコード	175
MQTM 構造体 - トリガメッセージ	177
MQXQH 構造体 - 転送キューヘッダ	182

## 2 メッセージキューイング機能の命令 188

命令の説明形式	189
MQCLOSE 命令 - オブジェクトのクローズ	191
MQCONN 命令 - キューマネージャへの接続	196
MQDISC 命令 - キューマネージャからの切り離し	199
MQGET 命令 - メッセージの取り出し	201
MQINQ 命令 - オブジェクトの属性の照会	210
MQOPEN 命令 - オブジェクトのオープン	219
MQPUT 命令 - メッセージの登録	233
MQPUT1 命令 - 1メッセージの登録	243
MQSET 命令 - オブジェクトの属性の設定	250

## 3 オブジェクトの属性 255

キューの属性	256
--------	-----

プロセス定義の属性	279
キューマネージャの属性	282

## 付録 287

付録 A	命令および引数の一覧	288
付録 A.1	MQCLOSE 命令の引数一覧	288
付録 A.2	MQCONN 命令の引数一覧	288
付録 A.3	MQDISC 命令の引数一覧	289
付録 A.4	MQGET 命令の引数一覧	289
付録 A.5	MQINQ 命令の引数一覧	293
付録 A.6	MQOPEN 命令の引数一覧	294
付録 A.7	MQPUT 命令および MQPUT1 命令の引数一覧	296
付録 A.8	MQSET 命令の引数一覧	301
付録 B	エラーコード	302
付録 B.1	完了コード	302
付録 B.2	理由コード	302
付録 C	マシンコード形式	380
付録 C.1	2 進整数のマシンコード形式	380
付録 C.2	10 進パック形式整数のマシンコード形式	381
付録 C.3	浮動小数点のマシンコード形式	382
付録 C.4	マシンコード形式の作成と検証	382
付録 C.5	マシンアーキテクチャのマシンコード形式の要約	383
付録 D	報告オプションとメッセージフラグ	384
付録 D.1	報告オプション	384
付録 D.2	メッセージフラグ	385
付録 E	各プログラミング言語での実現例	388
付録 E.1	C 言語での実現例	388
付録 E.2	COBOL 言語での実現例	402
付録 F	TP1/Message Queue の定数値	420
付録 G	コード変換対象の文字セット識別子一覧	447
付録 G.1	文字セット識別子と各 OS でサポートされている文字コードセットとの対応	447
付録 G.2	日立コード変換を使用した場合の文字コードの対応	449

## 索引 455

# 1

## データタイプ

この章では、メッセージキューイング機能の命令で使用する、データタイプについて説明します。

# データタイプの概要

---

メッセージキューイング機能の命令で使用するデータタイプには次の 2 種類があります。

- 基本データタイプ
- 構造体データタイプ

この章では、プログラミング言語に依存しない形式でこれらのデータタイプを説明します。なお、これらのデータタイプは、ユーザ定義のデータタイプをサポートする言語（例：C 言語）で宣言されるデータタイプに相当します。ただし、ユーザ定義のデータタイプは、基本データタイプとその組み合わせ（配列または構造体）で構成されます。

## 基本データタイプ

メッセージキューイング機能の命令では、次の基本データタイプを使用します。

**MQBYTE** : 1 バイト (8 ビット列)

**MQCHAR** : 1 文字 (定義された文字セットの中の、シングルキャラクタ文字)

**MLONG** : 整数 (符号付き 4 バイト 2 進数)

ここで示すほかのデータタイプは、上記の基本データタイプと同じものか、それらを組み合わせたもの（配列または構造体）です。

基本データタイプの実現例については、「付録 E 各プログラミング言語での実現例」を参照してください。

### ● MQBYTE 型 - 1 バイト

MQBYTE 型データは、1 バイトのデータを表します。バイトに対する特別な解釈はなく、2 進数や文字としてではなく、ビット文字列として扱われます。バウンダリ調整は不要です。

文字セットとマシンコードが異なるキューマネージャ間で MQBYTE データが送信される場合、MQBYTE データは変換されません。MQMD 構造体の MsgId および CorrelId フィールドについても同様です。

MQBYTE 型の配列で、キューマネージャ管理外の記憶領域を表す場合があります。例えば、アプリケーションメッセージデータや構造体を格納する領域です。この領域のバウンダリ調整では、領域内のデータの性質と矛盾が起きないように注意してください。

C 言語では、通常、MQBYTE 型の配列で指定する引数はアドレスで渡されます。そのため、関数の引数を MQBYTE 型の配列で指定すると、すべてのデータタイプを使用できます。C 言語では、関数の該当する引数は、void 型のポインタとして宣言されます。

### ● MQBYTEn 型 - n バイト

MQBYTEn 型データは、n バイトのバイト列を表します。n として、次の値を取ることができます。

n : 8, 16, 24, 32, 40, または 128

それぞれのバイトは、MQBYTE型で記述されます。バウンダリ調整は必要ありません。

なお、バイト型属性の各フィールドの長さが、定数 MQ\_\*として定義されています。

## ● MQCHAR型 - 1バイト文字

MQCHAR型データは、1バイトの文字、または2バイト文字やマルチバイト文字の1バイトを表します。

文字セットとマシンコードが異なるキューマネージャ間で、MQCHARデータが送信される場合、データが正しく解釈されるように通常はMQCHARデータの変換が必要です。MQMD構造体中のMQCHARデータについては、キューマネージャが自動的に変換を実行します。アプリケーションメッセージデータ内のMQCHARデータの変換については、MQGET命令発行時に指定するMQGMO\_CONVERTオプションによって制御できます。

この文字の文字セット識別子には、該当するキューマネージャの文字セット識別子が使用されます（「3. オブジェクトの属性」の「キューマネージャの属性」で、CodedCharSetId属性の説明を参照してください）。バウンダリ調整は必要ありません。

### 注意事項

MQGET命令、MQPUT命令、およびMQPUT1命令で指定するアプリケーションメッセージデータは、MQBYTE型で記述します。MQCHAR型では記述しません。

## ● MQCHARn型 - nバイト文字列

MQCHARn型データは、nバイトの文字列を表します。nとして、次の値を取ることができます。

n: 4, 8, 12, 20, 28, 32, 48, 64, 128, または 256

それぞれの文字は、MQCHAR型で記述されます。バウンダリ調整は必要ありません。

定義された文字列の長さよりもデータが短いとき、余りを空白で埋める必要があります。場合によっては、空白で埋める代わりに、ヌル文字で最後尾を示してもかまいません。データ領域内のヌル文字以降の文字は、空白として扱われます。ヌル文字を使用できるかどうかについては、この章で説明する各構造体と、「2. メッセージキューイング機能の命令」で説明する各命令を参照してください。

キューマネージャからアプリケーションに文字列が返される時（例えば、MQGET命令）、文字列の余りは空白で埋められます。キューマネージャはヌル文字を使用しません。

なお、文字型属性の各フィールドの長さが、定数 MQ\_\*として定義されています。

## ● MQHCONN型 - コネクションハンドル

MQHCONN型データは、コネクションハンドルを表します。このデータは、特定のキューマネージャへの接続を示します。コネクションハンドルは、4バイトのバウンダリ調整が必要です。

## 注意事項

アプリケーションで必要なコネクションハンドルの操作は、値が等しいかどうかを調べることです。

### ● MQHOBJ 型 – オブジェクトハンドル

MQHOBJ 型データは、オブジェクトを利用するためのハンドルを表します。オブジェクトハンドルは、4 バイトのバウンダリ調整が必要です。

## 注意事項

アプリケーションで必要なオブジェクトハンドルの操作は、値が等しいかどうかを調べることです。

### ● MQLONG 型 – ロング整数

MQLONG 型データは、32 ビットの符号付き 2 進数字を表します。このデータは、環境（使用言語など）で制限されないかぎり、-2147483648 から+2147483647 の範囲で値を取ることができます。COBOL 言語の場合は、-999999999 から+999999999 の範囲に制限されます。ロング整数は、4 バイトのバウンダリ調整が必要です。

### ● MQPTR 型 – ポインタ

MQPTR 型データは、任意の型のアドレスです。ポインタは 4 バイト（32 ビット版）、または 8 バイト（64 ビット版）のバウンダリ調整が必要です。

プログラミング言語で型付きポインタをサポートしていることがあり、MQI でも使用します。例えば、C 言語の PMQCHAR および PMQLONG 型です。

### ● PMQCHAR 型 – MQCHAR 型データへのポインタ

PMQCHAR 型データは、MQCHAR 型データへのポインタです。

### ● PMQLONG 型 – MQLONG 型データへのポインタ

PMQLONG 型データは、MQLONG 型データへのポインタです。

### ● PMQMD 型 – MQMD 構造体へのポインタ

PMQMD 型データは、MQMD 構造体へのポインタです。

## 構造体データタイプ

構造体の扱いは、プログラミング言語によって異なります。メッセージキューイング機能では、構造体について規則を定めています。この規則によって、それぞれのプログラミング言語で矛盾なく、構造体を使用できるようにしています。

構造体の規則を次に示します。

1. 構造体はバウンダリ調整します。メッセージキューイング機能で使用する構造体の多くは、4バイトの倍数を先頭アドレスとします。
2. 構造体中の各フィールドも、それぞれのデータタイプに応じてバウンダリ調整します。MQLONG型に相当するフィールドでは、4バイトの倍数を先頭アドレスとします。そのほかのフィールドでは、1バイト単位で先頭アドレスを設定できます。
3. 構造体の長さは、バウンダリ調整の単位の倍数とします。メッセージキューイング機能で使用する構造体は、4バイトの倍数の長さになります。
4. 上記の規則に適合するようにバイトやフィールドを追加してパディングします。

構造体データタイプの実現例については、「[付録 E 各プログラミング言語での実現例](#)」を参照してください。

# MQCFH 構造体 – PCF ヘッダ

MQCFH 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
Type (MQLONG 型)	構造体タイプ	MQCFT_COMMAND
StrucLength (MQLONG 型)	構造体長	MQCFH_STRUC_LENGTH
Version (MQLONG 型)	構造体バージョン番号	MQCFH_VERSION_1
Command (MQLONG 型)	命令識別子	0
MsgSeqNumber (MQLONG 型)	メッセージシーケンス番号	1
Control (MQLONG 型)	制御オプション	MQCFC_LAST
CompCode (MQLONG 型)	完了コード	MQCC_OK
Reason (MQLONG 型)	理由コード	MQRC_NONE
ParameterCount (MQLONG 型)	引数構造体の数	0

## 概要

MQCFH 構造体には、命令メッセージまたは応答メッセージのメッセージデータの開始位置にある情報を記述します。どちらの場合も、メッセージ記述子の Format フィールドは MQFMT\_ADMIN です。

MQCFH 構造体は、ユーザ定義のメッセージデータに使用することもできます。この場合、メッセージ記述子の Format フィールドは MQFMT\_PCF となります。

なお、この場合、構造体の中にある一部のフィールドは意味を持たなくなります。大部分のフィールドに提供されている初期値を使用できますが、StrucLength フィールド、および ParameterCount フィールドには、アプリケーション側でデータに適した値を設定する必要があります。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● Command (MQLONG 型) 命令識別子

TP1/Message Queue では使用されません。

命令メッセージは、実行される機能を識別します。

このフィールドの初期値は 0 です。

### ● CompCode (MQLONG 型) 完了コード

TP1/Message Queue では使用されません。

このフィールドは応答のときだけ意味があります。

## 1. データタイプ

このフィールドの初期値は MQCC\_OK です。

### ● Control (MQLONG 型) 制御オプション

TP1/Message Queue では使用されません。

次の値を取ります。

#### MQCFC\_LAST

グループの最終メッセージ

命令では、この値が常に設定されなければなりません。

#### MQCFC\_NOT\_LAST

グループの最終ではないメッセージ

このフィールドの初期値は MQCFC\_LAST です。

### ● MsgSeqNumber (MQLONG 型) メッセージシーケンス番号

TP1/Message Queue では使用されません。

これは、関連するメッセージのグループ内のメッセージシーケンス番号です。命令では、この値は 1 でなければなりません。それは、命令は常に単一のメッセージで構成されるためです。応答では、命令の応答の最初、またはこのシーケンス番号に対してだけ 1 であり、その命令に後続する応答のたびに 1 ずつ増加します。グループの最後の、または一つだけのメッセージは、Control フィールドに MQFSC\_LAST フラグが設定されます。

このフィールドの初期値は 1 です。

### ● ParameterCount (MQLONG 型) 引数構造体の数

MQCFH 構造体に続く引数構造体 (MQCFIL 構造体, MQCFIN 構造体, MQCFST 構造体, および MQCFSL 構造体) の数です。このフィールドの値は 0 以上です。

このフィールドの初期値は 0 です。

### ● Reason (MQLONG 型) 理由コード

TP1/Message Queue では使用されません。

理由コードです。

このフィールドの初期値は MQRC\_NONE です。

### ● StrucLength (MQLONG 型) 構造体長

MQCFH 構造体のバイト長です。次の値でなければなりません。

#### MQCFH\_STRUC\_LENGTH

命令フォーマットヘッダ構造体長

このフィールドの初期値は MQCFH\_STRUC\_LENGTH です。

### ● Type (MQLONG 型) 構造体タイプ

TP1/Message Queue では使用されません。

このフィールドは、メッセージの内容を示します。

このフィールドの初期値は MQCFT\_COMMAND です。

### ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

`MQCFH_VERSION_1`

命令フォーマットヘッダのバージョン番号

次の定数は現在のバージョンのバージョン番号を指定します。

`MQCFH_CURRENT_VERSION`

現在のバージョンの命令形式

このフィールドの初期値は `MQCFH_VERSION_1` です。

# MQCFIL 構造体 — PCF 整数リスト引数

MQCFIL 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
Type (MQLONG 型)	構造体タイプ	MQCFT_INTEGER_LIST
StrucLength (MQLONG 型)	構造体長	MQCFIL_STRUC_LENGTH_FIXED
Parameter (MQLONG 型)	引数識別子	0
Count (MQLONG 型)	引数の値の数	0
Values (MQLONG×Count)	引数の値	なし

## 概要

MQCFIL 構造体には、命令または命令の応答であるメッセージの整数リスト引数を記述します。どちらの場合も、メッセージ記述子のフォーマット名称は MQFMT\_ADMIN です。

MQCFIL 構造体は、イベントメッセージでも使用します。この場合、メッセージ記述子の Format フィールドは MQFMT\_EVENT です。

MQCFIL 構造体は、ユーザ定義のメッセージデータにも使用できます。この場合、メッセージ記述子の Format フィールドは、MQFMT\_PCF です。なお、この場合は、構造体の中にある一部のフィールドは意味を持たなくなります。大部分のフィールドには提供されている初期値を使用できますが、StrucLength フィールド、Count フィールド、および Values フィールドには、アプリケーション側でデータに適した値を設定する必要があります。

構造体は、可変長の整数 (Values フィールド) で終了します。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● Count (MQLONG 型) 引数の値の数

Values 配列の要素数です。0 以上でなければなりません。

このフィールドの初期値は 0 です。

### ● Parameter (MQLONG 型) 引数識別子

TP1/Message Queue では使用されません。

構造体に含まれる値の引数を示します。このフィールドに現れる値は、MQCFH 構造体の Command フィールドの値に依存します。

このフィールドの初期値は 0 です。

## 1. データタイプ

## ● StrucLength (MQLONG 型) 構造体長

構造体の最後 (Values フィールド) の可変長整数配列を含む, MQCFIL 構造体のバイト長です。長さは 4 の倍数でなければなりません。また, 配列を含むのに十分な長さでなければなりません。配列と StrucLength フィールドで定義される長さ間のバイトは意味を持ちません。

次の定数は構造体の固定部分の長さを表し, Values フィールドを除いた長さです。

### MQCFIL\_STRUC\_LENGTH\_FIXED

命令フォーマットの整数リスト引数構造体の固定部分の長さ

このフィールドの初期値は MQCFIL\_STRUC\_LENGTH\_FIXED です。

## ● Type (MQLONG 型) 構造体タイプ

構造体は整数リスト引数を表す MQCFIL 構造体であることを示します。

次の値を取ります。

### MQCFT\_INTEGER\_LIST

整数リストを定義する構造体

このフィールドの初期値は MQCFT\_INTEGER\_LIST です。

## ● Values (MQLONG × Count) 引数の値

これは, Parameter フィールドで識別される引数の値の配列です。例えば, MQIACF\_Q\_ATTRS では, 属性セクタ (MQCA\_\*および MQIA\_\*の値) のリストです。

C 言語の場合, このフィールドは 1 要素の配列として定義されます。構造体の格納域は動的に割り付けられる必要があります。また, ポインタはその中のフィールドを指すために使用される必要があります。

COBOL 言語の場合, 構造体の宣言にこのフィールドが入っていません。MQCFIL 構造体をより大きい構造体を含め, MQCFIL 構造体のあとに必要な長さの Values フィールドを宣言する必要があります。

## MQCFIN 構造体 — PCF 整数引数

MQCFIN 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
Type (MQLONG 型)	構造体タイプ	MQCFIN_INTEGER
StrucLength (MQLONG 型)	構造体長	MQCFIN_STRUC_LENGTH
Parameter (MQLONG 型)	引数識別子	0
Value (MQLONG 型)	引数の値	0

### 概要

MQCFIN 構造体は、命令または命令の応答であるメッセージに整数引数を記述します。どちらの場合もメッセージ記述子のフォーマット名称は MQFMT\_ADMIN です。

MQCFIN 構造体は、イベントメッセージでも使用します。この場合、メッセージ記述子の Format フィールドは MQFMT\_EVENT です。

MQCFIN 構造体は、ユーザ定義のメッセージデータにも使用できます。この場合、メッセージ記述子の Format フィールドは、MQFMT\_PCF です。なお、この場合は、構造体の中にある一部のフィールドは意味を持たなくなります。大部分のフィールドには提供されている初期値を使用できますが、Value フィールドには、アプリケーション側でデータに適した値を設定する必要があります。

### フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

#### ● Parameter (MQLONG 型) 引数識別子

TP1/Message Queue では使用されません。

構造体に含まれる値の引数を示します。このフィールドに現れる値は、MQCFH 構造体の Command フィールドの値に依存します。

このフィールドの初期値は 0 です。

#### ● StrucLength (MQLONG 型) 構造体長

MQCFIN 構造体のバイト長さです。

次の値を取ります。

#### MQCFIN\_STRUC\_LENGTH

命令フォーマットの整数引数構造体の長さ

このフィールドの初期値は MQCFIN\_STRUC\_LENGTH です。

## ● Type (MQLONG 型) 構造体タイプ

構造体は整数引数を表す MQCFIN 構造体であることを示します。

次の値を取ります。

`MQCFT_INTEGER`

整数を定義する構造体

このフィールドの初期値は `MQCFT_INTEGER` です。

## ● Value (MQLONG 型) 引数の値

このフィールドは、Parameter フィールドで識別される引数の値です。

このフィールドの初期値は 0 です。

# MQCFSL 構造体 — PCF 文字列リスト引数

MQCFSL 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
Type (MQLONG 型)	構造体タイプ	MQCFT_STRING_LIST
StrucLength (MQLONG 型)	構造体長	MQCFSL_STRUC_LENGTH_FIXED
Parameter (MQLONG 型)	引数識別子	0
CodedCharSetId (MQLONG 型)	文字セット識別子	MQCCSI_DEFAULT
Count (MQLONG 型)	引数の値の数	0
StringLength (MQLONG 型)	文字列長	なし
Strings (MQCHAR×StringLength×Count)	文字列の値	ヌル文字列

## 概要

MQCFSL 構造体には、命令または命令の応答であるメッセージの文字列リスト引数を記述します。どちらの場合も、メッセージ記述子のフォーマット名は MQFMT\_ADMIN となります。

構造体は可変長の文字列 (Strings フィールド) で終了します。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● CodedCharSetId (MQLONG 型) 文字セット識別子

Strings フィールドのデータの文字セット識別子を指定します。

次の値を取ります。

#### MQCCSI\_DEFAULT

デフォルトの文字セット識別子

文字データはメッセージ記述子 MQMD 構造体の CodedCharSetId フィールドによって定義された文字セットです。

このフィールドの初期値は MQCCSI\_DEFAULT です。

### ● Count (MQLONG 型) 引数の値の数

String フィールドにある文字列の数を表します。値は 0 以上でなければなりません。

このフィールドの初期値は 0 です。

## ● Parameter (MQLONG 型) 引数識別子

構造体に含まれる値の引数を示します。このフィールドの値は、MQCFH 構造体の Command フィールドの値と同じです。

このフィールドの初期値は 0 です。

## ● StringLength (MQLONG 型) 文字列長

引数の値の長さであり、Strings フィールドの 1 文字列の長さを表します。全文字列がこの長さとなります。値は 0 以上でなければなりません。なお、この長さは 4 の倍数である必要はありません。

## ● Strings (MQCHAR×StringLength×Count) 文字列の値

Parameter フィールドで識別される引数の 1 組の文字列の値を表します。文字列の数は Count フィールドで、各文字列の長さは StringLength フィールドで与えられます。文字列同士は結合され、文字列間に間隔はありません。文字列の合計長は、1 文字列の長さ×文字列の数を掛けたもの (String フィールドの値×Count フィールドの値) です。

指定された文字列が引数の標準の長さより短い場合、省略された文字は空白に仮定されます。指定された文字列が引数の標準の長さより長い場合、標準の長さを超える文字は空白でなければなりません。

文字列は CodedCharSetId フィールドで定義された文字セットにある文字列を含み、Parameter フィールドで指定された引数に適した値となります。

### ■ 注意事項

文字列中のヌル文字は、通常データとして扱われます。文字列の分離を示す記号として扱われません。

このフィールドの初期値は、C 言語ではヌル文字列です。

C 言語の場合、このフィールドは 1 要素の配列として定義されます。構造体の格納域は動的に割り付けられる必要があります。また、ポインタはその中のフィールドを指すために使用される必要があります。

COBOL 言語の場合、構造体の宣言にこのフィールドが入っていません。MQCFSL 構造体をより大きい構造体を含め、MQCFSL 構造体のあとに必要な長さの Strings フィールドを宣言する必要があります。

## ● StrucLength (MQLONG 型) 構造体長

構造体の最後である String フィールドの可変長文字列を含む、MQCFSL 構造体のバイト長を表します。長さは 4 の倍数でなければなりません。また、文字列を含むのに十分な長さでなければなりません。

文字列の最終と StrucLength フィールドで定義される長さ間のバイトは意味を持ちません。

次の定数は構造体の固定部分の長さを表し、String フィールドを除いた長さです。

## MQCFSL\_STRUC\_LENGTH\_FIXED

文字列リスト引数構造体の固定部分の長さ

このフィールドの初期値は MQCFSL\_STRUC\_LENGTH\_FIXED です。

### ● Type (MQLONG 型) 構造体タイプ

構造体が、文字列リスト引数である MQCFSL 構造体であることを示します。

次の値を取ります。

## MQCFT\_STRING\_LIST

文字列リストを定義する構造体

このフィールドの初期値は MQCFT\_STRING\_LIST です。

# MQCFST 構造体 – PCF スtring引数

MQCFST 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
Type (MQLONG 型)	構造体タイプ	MQCFT_STRING
StrucLength (MQLONG 型)	構造体長	MQCFST_STRUC_LENGTH_FIXED
Parameter (MQLONG 型)	引数識別子	0
CodedCharSetId (MQLONG 型)	文字セット識別子	MQCCSI_DEFAULT
StringLength (MQLONG 型)	文字列長	0
String (MQCHAR×StringLength)	文字列の値	ヌル文字列

## 概要

MQCFST 構造体には、命令または命令の応答であるメッセージに文字列引数を記述します。どちらの場合もメッセージ記述子のフォーマット名称は、MQFMT\_ADMIN です。

MQCFST 構造体は、イベントメッセージでも使用します。この場合、メッセージ記述子の Format フィールドは MQFMT\_EVENT です。

MQCFST 構造体は、ユーザ定義のメッセージデータにも使用できます。この場合、メッセージ記述子の Format フィールドは MQFMT\_PCF です。なお、この場合は、構造体の中にある一部のフィールドは意味を持たなくなります。大部分のフィールドには提供されている初期値を使用できますが、StrucLength フィールド、StringLength フィールド、および String フィールドにはアプリケーション側でデータに適した値を設定する必要があります。

構造体は可変長の文字列 (String フィールド) で終了します。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● CodedCharSetId (MQLONG 型) 文字セット識別子

String フィールドのデータの文字セット識別子を指定します。

次の値を取ります。

#### MQCCSI\_DEFAULT

デフォルトの文字セット識別子

文字データはメッセージ記述子 MQMD 構造体の CodedCharSetId フィールドによって定義された文字セットです。

このフィールドの初期値は MQCCSI\_DEFAULT です。

### ● Parameter (MQLONG 型) 引数識別子

TP1/Message Queue では使用されません。

構造体に含まれる値の引数を示します。このフィールドに現れる値は、MQCFH 構造体の Command フィールドの値に依存します。

このフィールドの初期値は 0 です。

### ● String (MQCHAR × StringLength) 文字列の値

指定された文字列が引数の標準の長さより短い場合、省略された文字は空白に仮定します。指定された文字列が引数の標準の長さより長い場合、標準の長さを超える文字は空白でなければなりません。

#### ■ 注意事項

文字列のヌル文字は通常データとして扱われ、文字列の分離記号としては扱われません。

このフィールドの初期値は、C 言語ではヌル文字列です。

C 言語の場合、このフィールドは 1 要素の配列として定義されます。構造体の格納域は動的に割り付けられる必要があります。また、ポインタはその中のフィールドを指すために使用される必要があります。

COBOL 言語の場合、構造体の宣言にこのフィールドが入っていません。MQCFST 構造体をより大きい構造体を含め、MQCFST 構造体のあとに必要な長さの String フィールドを宣言する必要があります。

### ● StringLength (MQLONG 型) 文字列長

String フィールドにあるデータのバイトの長さを示します。0 以上でなければなりません。この長さは 4 の倍数である必要はありません。

このフィールドの初期値は 0 です。

### ● StrucLength (MQLONG 型) 構造体長

構造体の最後 (String フィールド) の可変長文字列を含む、MQCFST 構造体のバイト長です。長さは 4 の倍数でなければなりません。また、文字列を含むのに十分でなければなりません。

文字列の最終と StrucLength フィールドで定義される長さ間のバイトは意味を持ちません。

次の定数は構造体の固定部分の長さを表し、String フィールドを除いた長さです。

#### MQCFST\_STRUC\_LENGTH\_FIXED

命令フォーマットの文字列引数構造体の固定部分の長さ

このフィールドの初期値は MQCFST\_STRUC\_LENGTH\_FIXED です。

## ● Type (MQLONG 型) 構造体タイプ

構造体は文字列引数を表す MQCFST 構造体であることを示しています。

次の値を取ります。

### MQCFT\_STRING

文字列を定義する構造体

このフィールドの初期値は MQCFT\_STRING です。

# MQDH 構造体 – 配布ヘッダ

MQDH 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQDH_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQDH_VERSION_1
StrucLength (MQLONG 型)	構造体長	0
Encoding (MQLONG 型)	MQPMR 構造体の配列に続くデータのマシンコード形式	0
CodedCharSetId (MQLONG 型)	MQPMR 構造体の配列に続くデータの文字セット識別子	MQCCSI_UNDEFINED
Format (MQCHAR8 型)	MQPMR 構造体の配列に続くデータのフォーマット名	MQFMT_NONE
Flags (MQLONG 型)	フラグ	MQDHF_NONE
PutMsgRecFields (MQLONG 型)	MQPMR 構造体フィールドの存在	MQPMRF_NONE
RecsPresent (MQLONG 型)	オブジェクトレコード数	0
ObjectRecOffset (MQLONG 型)	最初のオブジェクトレコードまでのオフセット	0
PutMsgRecOffset (MQLONG 型)	最初の登録メッセージレコードまでのオフセット	0

## 概要

### 目的

MQDH 構造体には、転送キューに存在する配布リストメッセージ中の、追加データを記述します。配布リストメッセージは、複数のあて先キューに送信されるメッセージです。追加データは、MQDH 構造体、MQOR 構造体の配列および MQPMR 構造体の配列の順序で構成されます。

この構造体は、転送キューに直接メッセージを登録したり、転送キューからメッセージを取り出したりする特別なアプリケーション（例：MCA）で使用されます。

MQDH 構造体は、配布リストにメッセージを登録するだけの通常の実アプリケーションで使用しないでください。通常の実アプリケーションを使用する場合、配布リストのあて先を指定するには MQOD 構造体を使用してください。また、メッセージの属性や、それぞれのあて先に送信されたメッセージに関する情報を指定するには、MQPMO 構造体を使用してください。

### フォーマット名

MQFMT\_DIST\_HEADER

#### 1. データタイプ

## 文字セットおよびマシンコード

MQDH 構造体のデータは、ローカルキューマネージャの文字セットとマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE オプションで指定します。

MQDH 構造体の文字セットとマシンコードは、次に示す構造体の CodedCharSetId および Encoding フィールドに指定します。

- MQMD 構造体 (MQDH 構造体がメッセージデータの先頭にある場合)
- MQDH 構造体に先行するヘッダ構造体 (その他の場合)

## 使用方法

アプリケーションがメッセージを配布リストに登録する場合、その幾つかまたはすべてのあて先がリモートであるとき、キューマネージャはアプリケーションデータの前に MQXQH 構造体と MQDH 構造体を付けます。その後、対応する転送キューにそのメッセージに登録します。そのため、転送キューにあるメッセージのデータは次の順序となります。

1. MQXQH 構造体
2. MQDH 構造体、および MQOR 構造体と MQPMR 構造体の配列
3. アプリケーションメッセージ

あて先によっては、そのようなメッセージが一つ以上、キューマネージャによって生成されて、異なる転送キューに登録されます。この場合、それらのメッセージの MQDH 構造体は、アプリケーションがオープンした配布リストによって指定したあて先の中で異なる組み合わせを示します。

配布リストメッセージを直接転送キューに登録するアプリケーションは、上記の順序や、その MQDH 構造体が正しいことを保証しなければなりません。MQDH 構造体が誤っている場合、キューマネージャは、MQPUT 命令または MQPUT1 命令を MQRC\_DH\_ERROR の理由コードで失敗させることがあります。

メッセージは、キューが配布リストメッセージをサポートすると定義してある場合だけ、配布リスト形式でキューに登録されます。配布リストをサポートしていないキューに、アプリケーションが直接配布リストメッセージに登録した場合、キューマネージャは配布リストメッセージを個々のメッセージに分解し、キューに登録します。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● CodedCharSetId (MQLONG 型) 文字セット識別子

MQOR および MQPMR 構造体の配列に続くデータの文字セット識別子です。

MQOR および MQPMR 構造体の配列に続くデータの文字セット識別子を指定します。MQDH 構造体自身の文字データには適用されません。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。

## MQCCSI\_INHERIT

この構造体の文字セット識別子を引き継ぎます。

この構造体に続くデータの文字データは、この構造体と同じ文字セットです。

メッセージで送信される構造体のこの値を、キューマネージャは構造体の実際の文字セット識別子に変更します。エラーが発生しない場合、MQGET 命令では MQCCSI\_INHERIT は返されません。

MQMD 構造体の PutApplType フィールドの値が MQAT\_BROKER の場合、MQCCSI\_INHERIT は使用できません。

このフィールドの初期値は MQCCSI\_UNDEFINED です。

## ● Encoding (MQLONG 型) マシンコード形式

MQOR および MQPMR 構造体の配列に続くデータのマシンコード形式です。

MQOR および MQPMR 構造体の配列に続くデータのマシンコード形式を指定します。MQDH 構造体自身の数値データには適用されません。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。

このフィールドの初期値は 0 です。

## ● Flags (MQLONG 型) フラグ

次に示すフラグが指定できます。

### MQDHF\_NEW\_MSG\_IDS

新しいメッセージ識別子を生成します。

このフラグは、新しいメッセージ識別子が配布リストのあて先ごとに生成されることを示しています。登録メッセージレコードが存在しない場合、またはレコードは存在するがレコードに MQMD 構造体の MsgId フィールドが含まれない場合にだけ、フラグを指定できます。

このフラグを使用した場合、メッセージ記述子の生成は配布リストメッセージがおのこのメッセージに分割されるときまで延期されます。これによって、配布リストメッセージと伝達されなければならない制御情報を最小にできます。

アプリケーションが配布リストにメッセージを登録した場合、次に示す両方が成り立つときに、キューマネージャは生成した MQDH 構造体に MQDHF\_NEW\_MSG\_IDS を設定します。

- アプリケーションによって生成された登録メッセージレコードが存在しない場合、または生成されたレコードに MsgId フィールドが含まれない場合
- MQMD 構造体の MsgId フィールドが MQMI\_NONE である場合、または MQPMO 構造体の Options フィールドに MQPMO\_NEW\_MSG\_ID が指定されている場合

フラグが必要ない場合は、次の指定ができます。

## MQDHF\_NONE

フラグなし。

この定数は、フラグが指定されなかったことを示しています。

MQDHF\_NONE はプログラムの文書化のために定義されています。値としては 0 が定義されていますが、プログラムを文書化する以外の目的はありません。

このフィールドの初期値は MQDHF\_NONE です。

## ● Format (MQCHAR8 型) フォーマット名

MQOR および MQPMR 構造体の配列に続くデータのフォーマット名です。

MQOR および MQPMR 構造体の配列に続くデータのフォーマット名を指定します。どちらが最後になるかは関係しません。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。このフィールドの設定規則は MQMD 構造体の Format フィールドと同じです。

このフィールドの初期値は MQFMT\_NONE です。

## ● ObjectRecOffset (MQLONG 型) 最初のオブジェクトレコードまでのオフセット

あて先キューの名称を含む、MQOR 構造体オブジェクトレコード配列にある最初のレコードのバイト位置を示します。この配列に RecsPresent フィールドのレコードがあります。これらのレコード（最初のオブジェクトレコードとその前のフィールドの間とのスキップされたバイトを加えて）は、StrucLength フィールドの長さに含まれます。

配布リストは少なくとも一つのあて先を持つ必要があり、ObjectRecOffset フィールドは常に 0 より大きくなければなりません。

このフィールドの初期値は 0 です。

## ● PutMsgRecFields (MQLONG 型) MQPMR 構造体フィールドの存在

0 または次に示すフラグが指定できます。

### MQPMRF\_MSG\_ID

メッセージ記述子フィールドが存在します。

### MQPMRF\_CORREL\_ID

相関識別子フィールドが存在します。

### MQPMRF\_GROUP\_ID

グループ識別子フィールドが存在します。

### MQPMRF\_FEEDBACK

フィードバックフィールドが存在します。

## MQPMRF\_ACCOUNTING\_TOKEN

課金トークンフィールドが存在します。

MQPMR 構造体フィールドが存在しない場合、次のフラグが指定できます。

## MQPMRF\_NONE

登録メッセージレコードが存在しません。

MQPMRF\_NONE はプログラムの文書化のために定義されています。値としては 0 が定義されていますが、プログラムを文書化する以外の目的はありません。

このフィールドの初期値は MQPMRF\_NONE です。

### ● PutMsgRecOffset (MQLONG 型) 最初の MQPMR 構造体までのオフセット

メッセージ属性を含む、MQPMR 登録メッセージレコード配列にある最初のレコードのバイト位置を示します。存在する場合は、この配列に RecsPresent フィールドのレコードがあります。

これらのレコード（最初のオブジェクトレコードとその前のフィールドの間とのスキップされたバイトを加えたレコード）は、StrucLength フィールドの長さに含まれます。

登録メッセージレコードはオプションです。レコードがない場合、PutMsgRecOffset フィールドは 0 となります。なお、PutMsgRecFields フィールドは MQPMRF\_NONE の値になります。

このフィールドの初期値は 0 です。

### ● RecsPresent (MQLONG 型) MQOR 構造体の数

あて先の数を定義します。配布リストは必ず最低一つのあて先を含まなければなりません。したがって、RecsPresent フィールドは常に 0 より大きくなければなりません。

このフィールドの初期値は 0 です。

### ● StrucId (MQCHAR4 型) 構造体識別子

次の値を取ります。

## MQDH\_STRUC\_ID

配布ヘッダ構造体の識別子

C 言語では、MQDH\_STRUC\_ID\_ARRAY 定数も定義されています。これは、MQDH\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの初期値は MQDH\_STRUC\_ID です。

### ● StrucLength (MQLONG 型) MQDH 構造体とそれに続く MQOR 構造体および MQPMR 構造体を足した長さ

MQDH 構造体長とそれに続くレコードを足した長さです。

これは、MQDH 構造体の先頭から、MQOR 構造体、または MQPMR 構造体レコードの配列に続くメッセージデータの先頭までのバイト数です。データは次の順序で現れます。

1. MQDH 構造体
2. MQOR レコードの配列
3. MQPMR レコードの配列
4. メッセージデータ

MQOR 構造体、MQPMR 構造体レコードの配列は、MQDH 構造体に含まれるオフセット値によって位置づけられます。それらのオフセットが、MQDH 構造体およびレコードの配列とメッセージデータの間で使用されないバイトを生じさせた場合、使用されないバイト数も StrucLength フィールドの値に含まれなければなりません。ただし、使用されないバイトの内容を、キューマネージャは保存しません。なお、MQPMR 構造体のレコードの配列は、MQOR 構造体にあるレコードの配列の前にあるとき有効です。

このフィールドの初期値は 0 です。

### ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

#### MQDH\_VERSION\_1

配布ヘッダの構造体バージョン番号

次の値は、現在のバージョンのバージョン番号を示します。

#### MQDH\_CURRENT\_VERSION

配布ヘッダ構造体の現在のバージョン

このフィールドの初期値は MQDH\_VERSION\_1 です。

# MQDLH 構造体 – デッドレターヘッダ

MQDLH 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQDLH_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQDLH_VERSION_1
Reason (MQLONG 型)	メッセージ通知理由	MQRC_NONE
DestQName (MQCHAR48 型)	あて先キュー名	ヌル文字列/空白
DestQMgrName (MQCHAR48 型)	あて先キューマネージャ名	ヌル文字列/空白
Encoding (MQLONG 型)	MQDLH 構造体に続くデータのマシンコード形式	0
CodedCharSetId (MQLONG 型)	MQDLH 構造体に続くデータの文字セット識別子	MQCCSI_UNDEFINED
Format (MQCHAR8 型)	MQDLH 構造体に続くデータのフォーマット名	MQFMT_NONE
PutApplType (MQLONG 型)	登録アプリケーションタイプ	0
PutApplName (MQCHAR28 型)	登録アプリケーション名	ヌル文字列/空白
PutDate (MQCHAR8 型)	登録日付	ヌル文字列/空白
PutTime (MQCHAR8 型)	登録時刻	ヌル文字列/空白

## 概要

### 目的

MQDLH 構造体には、デッドレターキューに格納されるメッセージのアプリケーションデータの先頭に付けられる情報を記述します。デッドレターキューにメッセージが届くのは、次のどちらかの場合です。

- キューマネージャまたは MCA が、デッドレターキューにメッセージを転送した場合
- アプリケーションが直接、デッドレターキューにメッセージを登録した場合

セグメントであるメッセージが、MQDLH 構造体先頭に付けて登録される場合、キューマネージャは特別な処理をします。詳細については、この章の「[MQMDE 構造体 – メッセージ記述子の拡張](#)」を参照してください。

### フォーマット名

MQFMT\_DEAD\_LETTER\_HEADER

## 文字セットおよびマシンコード

MQDLH 構造体のフィールドは、MQDLH 構造体の前にあるヘッダ構造体の CodedCharSetId および Encoding フィールドの文字セットおよびマシンコードに従います。また、MQDLH 構造体がアプリケーションメッセージデータの先頭にある場合、MQMD 構造体の CodedCharSetId および Encoding フィールドに従います。

文字セットは、キュー名に有効な 1 バイト文字セットにする必要があります。

## 使用方法

アプリケーションが直接デッドレターキューにメッセージを登録する場合、メッセージデータの先頭に MQDLH 構造体を付ける必要があります。また、構造体の各フィールドを適切に初期化してください。MQDLH 構造体を付けたかどうか、および各フィールドの値が正しいかどうかは、キューマネージャによってチェックされません。

デッドレターキューに登録するメッセージが長過ぎる場合、アプリケーションは次のどれかの処置をしてください。

- デッドレターキューに登録できるように、メッセージの後部を切り捨てます。
- 補助記憶装置にメッセージを格納し、その旨を示す例外報告メッセージをデッドレターキューに登録します。
- メッセージを破棄し、登録元アプリケーションにエラーを通知します。重要なメッセージの場合は、メッセージを破棄するときに注意が必要です。この場合、登録元アプリケーションがメッセージの複製を保持していると判明しているときだけ、破棄してください。例えば、MCA によって通信回線から受け取ったメッセージの場合などです。

上記のどの方法を採用するかは、アプリケーションの設計方法によって異なります。

セグメントであるメッセージが MQDLH 構造体を先頭にして登録されたとき、キューマネージャは特別な処理をします。詳細については、この章の「MQMDE 構造体 - メッセージ記述子の拡張」を参照してください。

## デッドレターキューへのメッセージの登録

メッセージをデッドレターキューへ登録するとき、MQPUT または MQPUT1 命令に使用する MQMD 構造体は、そのメッセージに関連する MQMD 構造体と同一にする必要があります（通常は、MQGET 命令によって返される MQMD 構造体です）。ただし、次の例外があります。

- CodedCharSetId フィールドと Encoding フィールドには、MQDLH 構造体の各フィールドで使用する値を設定します。
- Format フィールドには、MQFMT\_DEAD\_LETTER\_HEADER を設定します。この値は、MQDLH 構造体でデータが始まることを示します。
- 次に示すコンテキストフィールドには、状況に合ったコンテキストオプションを使用して設定します。
  - UserIdentifier

- AccountingToken
  - ApplIdentityData
  - PutApplType
  - PutApplName
  - PutDate
  - PutTime
  - ApplOriginData
- デッドレターキューに登録するメッセージがそれ以前のメッセージと無関係の場合、MQPMO\_DEFAULT\_CONTEXT オプションを使用します。このオプションを使用すると、すべてのコンテキストフィールドに省略時の値が設定されます。
  - サーバアプリケーションが受信したメッセージをデッドレターキューに登録する場合、元のコンテキスト情報を保存するために、MQPMO\_PASS\_ALL\_CONTEXT オプションを使用します。
  - サーバアプリケーションが受信したメッセージに対する応答をデッドレターキューに登録する場合、MQPMO\_PASS\_IDENTITY\_CONTEXT オプションを使用します。このオプションを使用すると、識別コンテキストの情報は保存されます。しかし、登録元コンテキストの情報は、サーバアプリケーションのコンテキスト情報に変更されます。
  - MCA が通信回線から受け取ったメッセージをデッドレターキューに登録する場合、元のコンテキスト情報を保存するために MQPMO\_SET\_ALL\_CONTEXT オプションを使用します。

MQDLH 構造体中の各フィールドは、次のように設定してください。

- CodedCharsetId フィールド、Encoding フィールドおよび Format フィールドには、MQDLH 構造体に続くデータに応じた値を設定します。通常は、登録元のメッセージ記述子の値を設定します。
- 次に示すコンテキストフィールドには、デッドレターキューにメッセージを登録するアプリケーションに応じた値を設定します。これらの値は、登録元メッセージとは関係ありません。
  - PutApplType
  - PutApplName
  - PutDate
  - PutTime
- そのほかのフィールドには、適切な値を設定します。

アプリケーションは、各フィールドの値が適切かどうか、および文字フィールドの余りが空白で埋められているかどうかを確認する必要があります。MQDLH 構造体中の文字データは、ヌル文字で終了できません。キューマネージャは、MQDLH 構造体のヌル文字以降の文字を、空白に置き換えません。

## デッドレターキューからのメッセージの取り出し

アプリケーションは、デッドレターキューから取り出したメッセージが、MQDLH 構造体で始まっているかどうかを確認する必要があります。メッセージ記述子 MQMD 構造体の Format フィールドを調べるこ

とで、それを確認できます。Format フィールドが MQFMT\_DEAD\_LETTER\_HEADER の場合、メッセージデータが MQDLH 構造体で始まっていると判断できます。なお、メッセージを取り出すアプリケーションは、メッセージが長過ぎて後部が切り捨てられた場合があることに注意してください。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● CodedCharSetId (MQLONG 型) 文字セット識別子

MQDLH 構造体に続くデータの文字セット識別子です。

MQDLH 構造体に続くデータ（通常は、元のメッセージから取られたデータ）の文字セット識別子を指定します。MQDLH 構造体自身の文字データには適用されません。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。

### MQCCSI\_INHERIT

この構造体の文字セット識別子を引き継ぎます。

この構造体に続くデータの文字データは、この構造体と同じ文字セットです。

メッセージで送信される構造体のこの値を、キューマネージャは構造体の実際の文字セット識別子に変更します。エラーが発生しない場合、MQGET 命令では MQCCSI\_INHERIT は返されません。

MQMD 構造体の PutApplType フィールドの値が MQAT\_BROKER の場合、MQCCSI\_INHERIT は使用できません。

このフィールドの初期値は MQCCSI\_UNDEFINED です。

### ● DestQMgrName (MQCHAR48 型) あて先キューマネージャ名

登録元メッセージのあて先であるキューマネージャの名称です。

このフィールドの長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

### ● DestQName (MQCHAR48 型) あて先キュー名

登録元メッセージのあて先キューの名称です。

このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

### ● Encoding (MQLONG 型) マシンコード形式

MQDLH 構造体に続くデータのマシンコード形式です。

MQDLH 構造体に続くデータ（通常は、元のメッセージから取られたデータ）のマシncode形式を指定します。MQDLH 構造体自身の数値データには適用されません。

MQPUT および MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。

このフィールドの初期値は 0 です。

### ● Format (MQCHAR8 型) フォーマット名

MQDLH 構造体に続くデータのフォーマット名です。

MQDLH 構造体に続くデータ（通常は元のメッセージから取られたデータ）のフォーマット名を指定します。

MQPUT および MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。このフィールドの設定方法は MQMD 構造体の Format フィールドと同じです。

このフィールドの長さは、MQ\_FORMAT\_LENGTH に定義されています。

このフィールドの初期値は MQFMT\_NONE です。

### ● PutApplName (MQCHAR28 型) 登録アプリケーション名

メッセージをデッドレターキューに登録するアプリケーションの名称です。

名称の形式は、PutApplType フィールドの値に合わせます。この章の「MQMD 構造体 – メッセージ記述子」で PutApplName フィールドの説明を参照してください。

キューマネージャがデッドレターキューにメッセージを転送する場合、PutApplName フィールドの値はキューマネージャ名の先頭から 28 文字までです。キューマネージャは、必要に応じて余りを空白で埋めます。

このフィールドの長さは、MQ\_PUT\_APPL\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、28 個の空白です。

### ● PutApplType (MQLONG 型) 登録アプリケーションタイプ

メッセージをデッドレターキューに登録するアプリケーションのタイプです。

このフィールドの意味は、メッセージ記述子 MQMD 構造体の PutApplType フィールドと同じです。

キューマネージャがデッドレターキューにメッセージを転送する場合、PutApplType フィールドの値は MQAT\_QMGR です。

このフィールドの初期値は 0 です。

## ● PutDate (MQCHAR8 型) 登録日付

メッセージがデッドレターキューに転送された日付です。

キューマネージャは次の形式を使用します。

YYYYMMDD

YYYY : 年 (4 けたの 10 進数)

MM : 月 (01~12 の 10 進数)

DD : 日 (01~31 の 10 進数)

PutDate フィールドと PutTime フィールドには、グリニッジ標準時 (GMT) を使用します。システムの時計を正しく設定してください。

このフィールドの長さは、MQ\_PUT\_DATE\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、8 個の空白です。

## ● PutTime (MQCHAR8 型) 登録時刻

メッセージがデッドレターキューに転送された時刻です。

キューマネージャは次の形式を使用します。

HHMMSSSTH

HH : 時 (00~23 の 10 進数)

MM : 分 (00~59 の 10 進数)

SS : 秒 (00~59 の 10 進数)

T : 1/10 秒 (0~9 の 10 進数)

H : 1/100 秒 (0~9 の 10 進数)

PutDate フィールドと PutTime フィールドには、グリニッジ標準時 (GMT) を使用します。システムの時計を正しく設定してください。

このフィールドの長さは、MQ\_PUT\_TIME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、8 個の空白です。

## ● Reason (MQLONG 型) メッセージ通知理由

メッセージがデッドレターキューに届いた理由を示します。MQFB\_\*, または MQRC\_\* のどちらか (例: MQRC\_Q\_FULL) で示されます。

MQFB\_\* の中には、このフィールドでだけ使用されるものがあります。それらの値は、デッドレターキューへ転送されるトリガメッセージに関するものです。

## MQFB\_APPL\_CANNOT\_BE\_STARTED

アプリケーションを開始できませんでした。

トリガモニタアプリケーションが、トリガメッセージの ApplId フィールドに指定されたアプリケーションを起動できませんでした。詳細は、この章の「MQTM 構造体 – トリガメッセージ」を参照してください。

## MQFB\_TM\_ERROR

トリガメッセージが不正です。

トリガモニタアプリケーションが、MQTM 構造体で次のどれかの不正を発見しました。

- StrucId フィールドが不正です。
- Version フィールドを認識できません。
- トリガメッセージ長が MQTM 構造体に対して不十分です。

## MQFB\_APPL\_TYPE\_ERROR

アプリケーションタイプが不正です。

トリガメッセージの ApplType フィールドが不正のため、アプリケーションを起動できませんでした。詳細は、この章の「MQTM 構造体 – トリガメッセージ」を参照してください。

## MQFB\_BIND\_OPEN\_CLUSRCVR\_DEL

クラスタレシーバチャンネルが削除されました。

MQOO\_BIND\_ON\_OPEN オプションでオープンされたクラスタキューあてのメッセージが SYSTEM.CLUSTER.TRANSMIT.QUEUE にありました。しかし、メッセージが送信される前に、あて先キューにメッセージを転送するために使用するリモートのクラスタレシーバチャンネルが削除されました。MQOO\_BIND\_ON\_OPEN が指定されたので、キューがオープンされたときに選択されたチャンネルだけがメッセージを転送できます。このチャンネルが使用不能なので、メッセージはデッドレターキューに登録されました。

## MQFB\_NOT\_A\_REPOSITORY\_MSG

メッセージはリポジトリメッセージではありません。

## MQFB\_STOPPED\_BY\_CHAD\_EXIT

メッセージがチャンネル自動定義出口によって停止されました。

## MQFB\_XMIT\_Q\_MSG\_ERROR

転送キュー上のメッセージのフォーマットが不正です。

転送キュー上のメッセージのフォーマット不正を、MCA が発見しました。MCA は、このフィールドバックコードを使用して、メッセージをデッドレターキューに登録します。

このフィールドの初期値は MQRC\_NONE です。

### ● StrucId (MQCHAR4 型) 構造体識別子

次の値を取ります。

## MQDLH\_STRUC\_ID

デッドレターヘッダの構造体識別子

C 言語では、MQDLH\_STRUC\_ID\_ARRAY も定義されています。これは、MQDLH\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの初期値は MQDLH\_STRUC\_ID です。

## ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

### MQDLH\_VERSION\_1

デッドレターヘッダの構造体バージョン番号

次の値には現在のバージョン番号を指定します。

### MQDLH\_CURRENT\_VERSION

デッドレターヘッダの現在の構造体バージョン番号

このフィールドの初期値は MQDLH\_VERSION\_1 です。

# MQGMO 構造体 – メッセージ取り出しオプション

MQGMO 構造体は、MQGET 命令で使用する入出力用の引数です。次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQGMO_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQGMO_VERSION_1
Options (MQLONG 型)	取り出しオプション	MQGMO_NO_WAIT
WaitInterval (MQLONG 型)	待ち合わせ最大時間	0
Signal1 (MQLONG 型)	シグナル	0
Signal2 (MQLONG 型)	(予備)	0
ResolvedQName (MQCHAR48 型)	受信キュー名	ヌル文字列/空白
MatchOptions (MQLONG 型) ※	一致オプション	MQMO_MATCH_CORREL_ID, MQMO_MATCH_MSG_ID
GroupStatus (MQCHAR 型) ※	メッセージグループフラグ	MQGS_NOT_IN_GROUP
SegmentStatus (MQCHAR 型) ※	論理メッセージフラグ	MQSS_NOT_A_SEGMENT
Segmentation (MQCHAR 型) ※	セグメント分割フラグ	MQSEG_INHIBITED
Reserved1 (MQCHAR 型) ※	(予備)	空白

注※

Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ存在するフィールドです。

## 概要

### 目的

MQGMO 構造体を使用すれば、キューからのメッセージの取り出し方法を制御するオプションをアプリケーションで指定できます。この構造体は、MQGET 命令の入出力用の引数です。

### バージョン

MQGMO 構造体のバージョンには 1 および 2 があります。しかし、すべての環境で MQGMO\_VERSION\_2 がサポートされているわけではありません。複数の環境にアプリケーションを移植する場合は、使用するバージョンの MQGMO 構造体が、すべての環境でサポートされている必要があります。MQGMO\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことをフィールドの説明に記載しています。

サポートするプログラミング言語の COPY ファイルと INCLUDE ファイルでは、最新バージョンの MQGMO 構造体が提供されます。しかし、Version フィールドの初期値は MQGMO\_VERSION\_1 です。バージョン 1 の構造体がないフィールドを使用する場合は、アプリケーションで使用したいバージョンの番号を Version フィールドに設定してください。

#### 1. データタイプ

## 文字セットおよびマシンコード

MQGMO 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。TP1/Message Queue Access のクライアントアプリケーションの場合、MQGMO 構造体はクライアントの文字セットとマシンコードに従います。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● GroupStatus (MQCHAR 型) メッセージグループフラグ

取り出されたメッセージがグループに属するかどうかを示すフラグです。

次の値を取ります。

#### MQGS\_NOT\_IN\_GROUP

メッセージはグループに属しません。

#### MQGS\_MSG\_IN\_GROUP

メッセージはグループに属しますが、グループの最終メッセージではありません。

#### MQGS\_LAST\_MSG\_IN\_GROUP

メッセージはグループの最終メッセージです。

これは、グループがただ一つのメッセージで構成されている場合にも返される値です。

これは出力用のフィールドです。

このフィールドの初期値は MQGS\_NOT\_IN\_GROUP です。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

### ● MatchOptions (MQLONG 型) 一致オプション

アプリケーションが MQGET 命令で返されるメッセージを選択するのに使用する MsgDesc 引数のどのフィールドを選ぶかを指定できます。アプリケーションはこのフィールドに必要なオプションを設定し、MsgDesc 引数の対応するフィールドに要求する値を設定します。MQMD 構造体の値を持つメッセージだけ、MQGET 命令で MsgDesc 引数を使用して取り出される可能性があります。対応する一致オプションが指定されないフィールドは、返すメッセージを選ぶときに無視されます。MQGET 命令でメッセージを選択しないで、どんなメッセージも受け入れられる場合、MatchOptions フィールドには MQMO\_NONE が設定されなければなりません。

MQGMO\_LOGICAL\_ORDER が指定された場合、特定のメッセージだけが次の MQGET 命令で返す対象となります。

- 現在のグループまたは現在の論理メッセージがない場合、MsgSeqNumber フィールドが 1 で、かつ Offset フィールドが 0 のメッセージだけが返す対象となります。この場合、実際に返されるメッセージを選択するには、次に示す一致オプションのうち、一つ以上のオプションを使用できます。

- MQMO\_MATCH\_MSG\_ID
- MQMO\_MATCH\_CORREL\_ID
- MQMO\_MATCH\_GROUP\_ID
- 現在のグループまたは現在の論理メッセージがある場合、そのグループの次のメッセージ、またはその論理メッセージの次のセグメントだけが返されます。これは、MQMO\_\*オプションを指定しても変更できません。

上記の条件が両方とも該当する場合も、ほかの一致オプションを指定できます。ただし、MsgDesc 引数の対応するフィールドの値は、返す必要があるメッセージのフィールドの値と一致しなければなりません。この条件が満たされない場合は、MQRC\_MATCH\_OPTIONS\_ERROR となり失敗します。

MQGMO\_MSG\_UNDER\_CURSOR, または MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR が指定された場合、MatchOptions フィールドは無視されます。

次に示す一致オプションを、一つ以上指定できます。

### MQMO\_MATCH\_MSG\_ID

指定されたメッセージ識別子のメッセージを取り出します。

取り出されるメッセージは、MQGET 命令の MsgDesc 引数の MsgId フィールドの値と一致するメッセージ識別子を持つ必要があることを指定します。

このオプションは、適用できるほかの一致オプション（例：相関識別子）と同時に使用できます。

このオプションが指定されない場合、MsgDesc 引数の MsgId フィールドは無視され、どんなメッセージ識別子とも一致します。

#### 注意事項

メッセージ識別子 MQMI\_NONE は、MQMD 構造体のどんなメッセージ識別子とも一致する特別な値です。そのため、MQMI\_NONE と一緒に MQMO\_MATCH\_MSG\_ID を指定しても意味がありません。

### MQMO\_MATCH\_CORREL\_ID

指定された相関識別子のメッセージを取り出します。

取り出されるメッセージは、MQGET 命令の MsgDesc 引数の CorrelId フィールドの値と一致する相関識別子を持つ必要があることを指定します。このオプションは、適用できるほかの一致オプション（例：メッセージ識別子）と同時に使用できます。

このオプションが指定されない場合は、MsgDesc 引数の CorrelId フィールドは無視され、どんな相関識別子とも一致します。

#### 注意事項

相関識別子 MQCI\_NONE は、MQMD 構造体のどんな相関識別子とも一致する特別な値です。そのため、MQCI\_NONE と一緒に MQMO\_MATCH\_CORREL\_ID を指定しても意味がありません。

## MQMO\_MATCH\_GROUP\_ID

指定されたグループ識別子のメッセージを取り出します。

取り出されるメッセージは、MQGET 命令の MsgDesc 引数の GroupId フィールドの値と一致するグループ識別子を持つ必要があることを指定します。このオプションは、適用できるほかの一致オプション（例：相関識別子）と同時に使用できます。

このオプションが指定されない場合、MsgDesc 引数の GroupId フィールドは無視され、どんなグループ識別子とも一致します。

### 注意事項

グループ識別子 MQGI\_NONE は、MQMD 構造体のどんなグループ識別子とも一致する特別な値です。そのため、MQGI\_NONE と一緒に MQMO\_MATCH\_GROUP\_ID を指定しても意味がありません。

## MQMO\_MATCH\_MSG\_SEQ\_NUMBER

指定されたメッセージシーケンス番号のメッセージを取り出します。

取り出されるメッセージが、MQGET 命令の MsgDesc 引数の MsgSeqNumber フィールドの値と一致するメッセージシーケンス番号を持つ必要があることを指定します。このオプションは、適用できるほかの一致オプション（例：グループ識別子）と同時に指定できます。

このオプションが指定されない場合、MsgDesc 引数の MsgSeqNumber フィールドは無視され、どんなメッセージシーケンス番号とも一致します。

## MQMO\_MATCH\_OFFSET

指定されたオフセットのメッセージを取り出します。

取り出されるメッセージは、MQGET 命令の MsgDesc 引数の Offset フィールドの値と一致するオフセットを持つ必要があることを指定します。このオプションは、適用できるほかの一致オプション（例：メッセージシーケンス番号）と同時に使用できます。

このオプションが指定されない場合、MsgDesc 引数の MsgSeqNumber フィールドは無視され、どんなオフセットとも一致します。

上記のオプションがどれも指定されなかった場合、次のオプションが指定できます。

## MQMO\_NONE

一致指定なし。

返されるメッセージを選択するときに、一致指定を使用しないことを指定します。したがって、キューに存在するすべてのメッセージが、取り出し対象となります。ただし、MQGMO\_ALL\_MSGS\_AVAILABLE オプション、MQGMO\_ALL\_SEGMENTS\_AVAILABLE オプション、および MQGMO\_COMPLETE\_MSG オプションによって制御できます。MQMO\_NONE はプログラムの文書化のために定義されています。ほかの MQMO\_\* オプションと組み合わせて使用するオプションではありません。値は 0 なので、これ以外の用途では使用できません。

これは入力用のフィールドです。

このフィールドの初期値は MQMO\_MATCH\_CORREL\_ID と MQMO\_MATCH\_MSG\_ID です。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

## 注意事項

MatchOptions フィールドの初期値は、以前のキューマネージャと互換性を保つために定義されています。ただし、選択基準なしでキューから一連のメッセージを取り出すときは、アプリケーションは、MQGET 命令が実行される前に MsgId フィールドと CorrelId フィールドを、MQMI\_NONE と MQCI\_NONE にリセットする必要があります。ただし、Version フィールドに MQGMO\_VERSION\_2 を、MatchOptions フィールドに MQMO\_NONE を設定すれば、MsgId フィールドと CorrelId フィールドをリセットする必要はありません。

## ● Options (MQLONG 型) 取り出しオプション

MQGET 命令の動作を制御します。

指定できるオプションを説明します。なお、指定しなくてもかまいません。二つ以上を指定する場合は、次のどちらかに従ってください。

- それぞれを加算します。ただし、同じ値は 2 回以上加算しないでください。
- プログラミング言語がビット演算をサポートしている場合に、それぞれのビット論理和を取ります。

同時に指定できない組み合わせは、そのつど示します。それ以外の組み合わせは自由です。オプションおよびその指定方法の一覧については、「付録 A 命令および引数の一覧」を参照してください。

### 待ち合わせオプション：

メッセージがキューに到着するまでの待ち合わせに関連するオプションを次に示します。

### MQGMO\_WAIT

該当するメッセージの到着を待ちます。

アプリケーションは、該当するメッセージが到着するまで待ちます。アプリケーションの待ち合わせ最大時間は、WaitInterval フィールドに指定します。

取り出しが禁止されている場合、または待ち合わせの間に禁止された場合は、メッセージの待ち合わせは取り消されます。このとき、完了コード MQCC\_FAILED、理由コード MQRC\_GET\_INHIBITED で命令は終了します。該当するメッセージがキューにあるかどうかは、関係ありません。

このオプションは、MQGMO\_BROWSE\_FIRST オプションまたは MQGMO\_BROWSE\_NEXT オプションと同時に使用できます。

複数のアプリケーションが同じキューで待ち合わせをしている場合、該当するメッセージが到着したときの動作は、次のようになります。なお、ここでは、検索オプションを指定して MQGMO\_LOCK オプションを指定しないものを、検索 MQGET 命令と呼びます。MQGMO\_LOCK を指定したものは、非検索 MQGET 命令として扱われます。

- 一つ以上の非検索 MQGET 命令が待ち合わせをしているが、検索 MQGET 命令は待ち合わせをしていない場合、一つの命令がメッセージを取り出せます。
- 一つ以上の検索 MQGET 命令が待ち合わせをしているが、非検索 MQGET 命令は待ち合わせをしていない場合、すべての命令がメッセージを検索できます。
- 一つ以上の非検索 MQGET 命令と一つ以上の検索 MQGET 命令が待ち合わせをしている場合、一つの非検索 MQGET 命令だけがメッセージを取り出せます。同時に、幾つか（すべての場合もあるし、まったくない場合もある）の検索 MQGET 命令がメッセージを検索できます。なお、検索 MQGET 命令が実行できる数は、オペレーティングシステムのスケジュール管理やほかの要素に左右されます。

一つ以上の非検索 MQGET 命令が同じキューについて待ち合わせをしている場合、一つの命令だけがメッセージを取り出せます。このときキューマネージャは、次のように優先順位を付けます。

#### 1. 指定付きメッセージの待ち合わせ

この場合は、指定に該当するメッセージだけが対象になります（例：指定した MsgId または CorrelId のどちらかの場合、またはその両方を満たすメッセージの場合）。

#### 2. 一般のメッセージの待ち合わせ

この場合は、すべてのメッセージが対象になります。

また、次のことに注意してください。

- 1 の指定付きメッセージの待ち合わせでは、それよりほかの優先度を付け加えることはできません。
- 1, 2 の各優先順位内では、どのアプリケーションが選択されるか確定していません。特に、最も長く待ち合わせをしているアプリケーションが選択されるとは限りません。
- パスの長さ、およびオペレーティングシステムのスケジュール管理が影響するため、オペレーティングシステムのプロセス実行優先度の低いアプリケーションが、先にメッセージを取り出せることがあります。
- 上記の理由のため、待ち合わせをしていないアプリケーションがメッセージを取り出せることがあります。

MQGMO\_WAIT オプションと次に示すオプションを同時に指定した場合、MQGMO\_WAIT は無視されますが、エラーにはなりません。

- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_MSG\_UNDER\_CURSOR

### MQGMO\_NO\_WAIT

該当するメッセージがない場合、即時に制御を返します。

該当するメッセージがない場合、アプリケーションは待ち合わせをしません。これは MQGMO\_WAIT オプションの反対で、プログラムのコーディングをわかりやすくするために定義されています。どちらも指定しなかった場合、この値が仮定されます。

MQGMO\_NO\_WAIT オプションと MQGMO\_WAIT オプションを同時に指定した場合、MQGMO\_NO\_WAIT は無視されますが、エラーにはなりません。

## MQGMO\_FAIL\_IF\_QUIESCING

TP1/Message Queue では意味がありません。

このオプションを指定しても、TP1/Message Queue の動作には影響しません。これは、アプリケーションの移植性を向上させるために定義されています。

## 同期点オプション：

MQGET 命令のトランザクションへの参加に関連するオプションを次に示します。

## MQGMO\_SYNCPOINT

同期点でメッセージを取り出します。

通常のトランザクションの範囲内で取り出します。取り出したメッセージは、ほかのアプリケーションから使用できなくなります。ただし、キューから削除されるのは、トランザクションがコミットしたときだけです。トランザクションがロールバックすると、ほかのアプリケーションから使用できるようになります。

このオプションは、ローカルキューマネージャが同期点処理をサポートしている場合だけ指定できます。同期点処理をサポートしていないキューマネージャで MQGMO\_SYNCPOINT を指定した場合、完了コード MQCC\_FAILED、理由コード MQRC\_SYNCPOINT\_NOT\_AVAILABLE で命令は終了します。同期点の取得とトランザクションについては、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。

このオプションと MQGMO\_NO\_SYNCPOINT オプションのどちらも指定しなかった場合、取り出し命令がトランザクション内に含まれるかどうかは、環境に依存します。TP1/Message Queue では、この命令をトランザクション内から呼び出したかどうかで、省略時にトランザクション内に含まれるかが決まります。省略時の動作が環境によって異なるため、移植を考慮するアプリケーションでは、オプションを省略しないでください。つまり、MQGMO\_SYNCPOINT オプションか MQGMO\_NO\_SYNCPOINT オプションのどちらかを、明確に指定してください。

次に示すオプションは、MQGMO\_SYNCPOINT オプションと同時に指定できません。

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_LOCK
- MQGMO\_NO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

## MQGMO\_NO\_SYNCPOINT

同期点の処理なしでメッセージを取り出します。

通常のトランザクションの範囲外で取り出します。検索要求の場合を除いて、メッセージは取り出し直後にキューから削除されます。トランザクションがロールバックしても、メッセージは再び有効になりません。

このオプションは、MQGMO\_BROWSE\_FIRST、またはMQGMO\_BROWSE\_NEXT オプションを指定した場合を想定しています。

このオプションとMQGMO\_SYNCPOINT オプションのどちらも指定しなかった場合、取り出し命令がトランザクション内に含まれるかどうかは、環境に依存します。TP1/Message Queue では、この命令をトランザクション内から呼び出したかどうかで、省略時にトランザクション内に含まれるかどうかが決まります。省略時の動作が環境によって異なるため、移植を考慮するアプリケーションでは、オプションを省略しないでください。つまり、MQGMO\_SYNCPOINT オプションかMQGMO\_NO\_SYNCPOINT オプションのどちらかを、明確に指定してください。

次に示すオプションは、MQGMO\_NO\_SYNCPOINT オプションと同時に指定できません。

- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT

同期点のオプションとキューマネージャの SyncPoint 属性の関係を、次の表に示します。

キューマネージャの SyncPoint 属性	MQGET 命令の呼び出し位置	MQGET 命令の同期点のオプション		
		MQGMO_SYNCPOINT	MQGMO_NO_SYNCPOINT	省略
MQSP_AVAILABLE	トランザクション内	1	2	1
	トランザクション外	×	2	2
MQSP_NOT_AVAILABLE	トランザクション内	×	2	×
	トランザクション外	×	2	2

(凡例)

- 1：命令はトランザクション内に含まれます。
- 2：命令はトランザクション内に含まれません。
- ×：命令は理由コード MQRC\_SYNCPOINT\_NOT\_AVAILABLE で失敗します。

## MQGMO\_SYNCPOINT\_IF\_PERSISTENT

取り出すメッセージが永続メッセージの場合、キューマネージャはアプリケーションでMQGMO\_SYNCPOINT が指定されている場合と同じように取り出し処理を実行します。

取り出すメッセージが非永続メッセージの場合、キューマネージャはアプリケーションでMQGMO\_NO\_SYNCPOINT が指定されている場合と同じように取り出し処理を実行します。

次に示すオプションは、MQGMO\_SYNCPOINT\_IF\_PERSISTENT オプションと同時に指定できません。

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_BROWSE\_NEXT
- MQGMO\_COMPLETE\_MSG
- MQGMO\_NO\_SYNCPOINT
- MQGMO\_SYNCPOINT

- MQGMO\_UNLOCK
- MQGMO\_LOCK

## 検索オプション：

キューからのメッセージ検索に関連するオプションを次に示します。

### MQGMO\_BROWSE\_FIRST

キューの先頭から検索します。

MQOO\_BROWSE オプションでキューをオープンした場合、検索カーソルが生成されます。検索カーソルは、キューの最初のメッセージを論理的に指します。以降の MQGET 命令では、MQGMO\_BROWSE\_FIRST、MQGMO\_BROWSE\_NEXT、または MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR オプションを指定することで、メッセージを削除しないで検索できます。検索カーソルは、キュー内のメッセージの位置を指します。このため、MQGMO\_BROWSE\_NEXT を指定した次の MQGET 命令で、適切なメッセージを検索できます。

MQGET 命令で MQGMO\_BROWSE\_FIRST を指定すると、それまでの検索カーソルの位置は無視されます。そして、メッセージ記述子の指定に該当する最初のメッセージが検索されます。検索されたメッセージはキューに残ります。また、検索カーソルはこのメッセージを指したままです。

この命令が終了すると、検索カーソルは返されたメッセージを指します。MQGMO\_BROWSE\_NEXT を指定した次の MQGET 命令を実行するまでにそのメッセージが削除された場合、その位置にメッセージがなくても、検索カーソルはその位置を指したままです。

検索したメッセージをキューから削除するために、このあとの非検索 MQGET 命令で、MQGMO\_MSG\_UNDER\_CURSOR オプションを指定できます。

同じ Hobj ハンドルを使用して非検索 MQGET 命令を実行しても、検索カーソルは移動しないことに注意してください。また、完了コード MQCC\_FAILED、または理由コード MQRC\_TRUNCATED\_MSG\_FAILED で終了した検索 MQGET 命令でも、検索カーソルは移動しません。

検索したメッセージを排他状態にするために、このオプションと同時に MQGMO\_LOCK オプションを指定できます。

MQGMO\_BROWSE\_FIRST は、グループ内のメッセージおよび論理メッセージのセグメントの処理を制御する MQGMO 構造体、および MQMO 構造体のオプションの正しい組み合わせと一緒に指定できます。

MQGMO\_LOGICAL\_ORDER が指定された場合、メッセージは論理的な順序で検索されます。省略された場合、メッセージは物理的な順序で検索されます。MQGMO\_BROWSE\_FIRST が指定されると、論理順序と物理順序をスイッチできますが、MQGMO\_BROWSE\_NEXT を使用した MQGET 命令は、キューハンドルに対して MQGMO\_BROWSE\_FIRST を最後に指定した命令と同じ順序でキューを検索しなければなりません。

キューにあるメッセージを検索する MQGET 命令で、キューマネージャに保持されるグループおよびセグメントの情報は、キューからメッセージを取り出す MQGET 命令でキューマネージャに保持されるグループおよびセグメントの情報とは分離されています。MQGMO\_BROWSE\_FIRST が指定されると、キューマネージャは、グループおよびセグメントの情報を無視し、現在のグループおよび論理メッセージ

がないかのようにキューを探します。MQGET 命令が、完了コード MQCC\_OK または MQCC\_WARNING で成功すると、検索のためのグループおよびセグメントの情報は、返却されたメッセージの値に設定されます。命令が失敗すると、グループおよびセグメントの情報は命令前と同じになります。

次に示すオプションは、MQGMO\_BROWSE\_FIRST オプションと同時に指定できません。

- MQGMO\_BROWSE\_NEXT
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_MSG\_UNDER\_CURSOR
- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

また、キューを検索用にオープンしていない場合もエラーになります。

## MQGMO\_BROWSE\_NEXT

キューの現在位置から検索します。

検索カーソルは、メッセージ記述子の指定に該当する次のメッセージに移ります。そして、そのメッセージがアプリケーションに返されます。検索されたメッセージは、キューに残ります。

メッセージが検索される順序は、キューにメッセージが格納されているとおりです。メッセージが検索される順序の種類を次に示します。

- 優先度内での FIFO 順序 (MQMDS\_PRIORITY)
- 優先度に関係ない FIFO 順序 (MQMDS\_FIFO)

詳細は、「3. オブジェクトの属性」の「[キューの属性](#)」で MsgDeliverySequence 属性の説明を参照してください。

キューを検索用にオープンしたあと、そのハンドルを使用した最初の検索 MQGET 命令は、MQGMO\_BROWSE\_FIRST を指定した場合も MQGMO\_BROWSE\_NEXT を指定した場合も同じになります。

MQGMO\_BROWSE\_NEXT を指定した次の MQGET 命令を実行するまでにそのメッセージが削除された場合、その位置にメッセージがなくても、検索カーソルはその位置を指したままです。

### 注意事項

メッセージをキューに登録するとき、そのキューでのメッセージの位置が決定される時点と、メッセージの位置が有効になってほかのアプリケーションが参照できるようになる時点とは異なります。そのため、複数のアプリケーションまたはチャンネルが同一のキューにメッセージに登録する場合には注意が必要です。つまり、メッセージを参照するアプリケーションが検索 MQGET 命令を実行して、位置が有効になる前のメッセージより後ろに検索カーソルを進めた場合、以降の MQGMO\_BROWSE\_NEXT を使用した検索ではそのメッセージは見つかりません。この場合は、MQGMO\_BROWSE\_FIRST を使用して、キューの先頭から検索する必要があります。

キューの `MsgDeliverySequence` 属性が `MQMDS_PRIORITY` の場合で、現在の検索カーソルが指すメッセージよりも優先度の高いメッセージが到着したとき、そのメッセージは `MQGMO_BROWSE_NEXT` を使用した検索では見つかりません。`MQGMO_BROWSE_FIRST` オプションで検索カーソルをリセットするか、再びキューをオープンしないと、そのメッセージは検索できません。

検索したメッセージをキューから削除するには、このあとの非検索 `MQGET` 命令で、`MQGMO_MSG_UNDER_CURSOR` オプションを指定します。同じ `Hobj` ハンドルを使用して非検索 `MQGET` 命令を実行しても、検索カーソルは移動しないことに注意してください。

検索したメッセージを排他状態にするには、このオプションと同時に `MQGMO_LOCK` オプションを指定します。

なお、`MQGMO_BROWSE_NEXT` は、グループ内のメッセージおよび論理メッセージのセグメントの処理を制御する `MQGMO` 構造体、および `MQMO` 構造体のオプションの正しい組み合わせと一緒に指定できます。

`MQGMO_LOGICAL_ORDER` が指定された場合、メッセージは論理的な順序で検索されます。省略された場合、メッセージは物理的な順序で検索されます。`MQGMO_BROWSE_FIRST` が指定されると、論理順序と物理順序でスイッチできますが、`MQGMO_BROWSE_NEXT` を使用した `MQGET` 命令はキューハンドルに対して `MQGMO_BROWSE_FIRST` を指定した最後の命令と同じ順序でキューを検索しなければなりません。この条件が満たされない場合、`MQRC_INCONSISTENT_BROWSE` で失敗となります。

## 注意事項

`MQGMO_LOGICAL_ORDER` を指定しないで、`MQGET` 命令でメッセージグループ、またはグループに属さない論理メッセージの終わりを超えて検索した場合は、特に注意が必要です。例えば、グループの最終メッセージが、偶然そのグループの最初のメッセージよりも先にキューにある場合に、グループの終わりを超えて検索するとします。この場合、次のグループにある最初のメッセージを見つけるために、`MQGMO_BROWSE_NEXT` で `MQMO_MATCH_MSG_SEQ_NUMBER` の `MsgSeqNumber` を 1 に設定すると、すでに検索したグループの最初のメッセージが再度返却されます。また、間にグループがある場合に、複数の `MQGET` 命令のあとにも、すでに検索したグループの最初のメッセージが再度返却されることがあります。

このような不確定のループは、検索用にキューを二度オープンすることで避けられます。また、次のことに留意すると、不確定のループを避けられます。

- 最初のハンドルは各グループの最初のメッセージだけを検索するのに使用します。
- 2 番目のハンドルは指定したグループ内のメッセージだけを検索するのに使用します。
- 2 番目の検索カーソルを最初の検索カーソルの位置に移動するときは、`MQMO` 構造体のオプションをグループのメッセージ検索の前に使用します。
- グループを超えて検索する場合、`MQGMO_BROWSE_NEXT` は使用しないようにします。

キューにあるメッセージを検索する MQGET 命令でキューマネージャに保持されるグループおよびセグメントの情報は、キューからメッセージを取り出す MQGET 命令によってキューマネージャに保持されるグループおよびセグメントの情報とは分離されています。

次に示すオプションは、MQGMO\_BROWSE\_NEXT オプションと同時に指定できません。

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_MSG\_UNDER\_CURSOR
- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

また、キューを検索用にオープンしていない場合もエラーになります。

## MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR

検索カーソルの位置を検索します。

このオプションは、MQGMO 構造体の MatchOptions フィールドに指定された MQMO 構造体のオプションに関係なく、検索のために検索カーソルがメッセージにポイントされます。

また、このオプションでは、検索カーソルが指すメッセージを削除しないで検索します。MsgDesc 引数の MsgId フィールドおよび CorrelId フィールドに指定した値は無視されます。

検索カーソルが指すメッセージは、MQGMO\_BROWSE\_FIRST または MQGMO\_BROWSE\_NEXT オプションで、一つ前に検索されたものです。オープンしてから一度も検索しなかった場合、または該当するメッセージが削除されていた場合は、命令は失敗します。

このオプションの命令で、検索カーソルの位置が移動することはありません。検索したメッセージをキューから削除するために、このあとの非検索 MQGET 命令で、MQGMO\_MSG\_UNDER\_CURSOR オプションを指定できます。

同じ Hobj ハンドルを使用して非検索 MQGET 命令を実行しても、検索カーソルは移動しないことに注意してください。また、完了コード MQCC\_FAILED、または理由コード

MQRC\_TRUNCATED\_MSG\_FAILED で終了した検索 MQGET 命令でも、検索カーソルは移動しません。

MQGMO\_LOCK オプションと同時に MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR オプションを指定した場合、次のようになります。

- すでに排他状態のメッセージがある場合、検索カーソルはそのメッセージを指しています。この場合、そのメッセージを排他状態にしたままで検索できます。
- 排他状態のメッセージがない場合、検索カーソルが指すメッセージが排他状態になり、アプリケーションに返されます。検索カーソルが指す位置にメッセージがない場合、命令は失敗します。

MQGMO\_LOCK オプションを指定しないで MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR オプションを指定した場合、次のようになります。

- すでに排他状態のメッセージがある場合、検索カーソルはそのメッセージを指しています。そのメッセージはアプリケーションに返され、排他状態は解除されます。メッセージの排他状態が解除され

るため、再び検索したり取り出したりできる保証はありません。したがって、そのメッセージは、ほかのアプリケーションによって削除されることがあります。

- 排他状態のメッセージがない場合、検索カーソルが指すメッセージがアプリケーションに返されません。検索カーソルが指す位置にメッセージがない場合、命令は失敗します。

MQGMO\_COMPLETE\_MSG が MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR と一緒に指定される場合、検索カーソルは、MQMD 構造体の Offset フィールドが 0 であるメッセージを特定しなければなりません。この条件が満たされない場合、理由コード MQRC\_INVALID\_MSG\_UNDER\_CURSOR で失敗となります。

キューにあるメッセージを検索する MQGET 命令でキューマネージャに保持されるグループおよびセグメントの情報は、キューからメッセージを取り出す MQGET 命令によってキューマネージャに保持されるグループ、およびセグメントの情報とは分離されています。

次に示すオプションは、MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR オプションと同時に指定できません。

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_NEXT
- MQGMO\_MSG\_UNDER\_CURSOR
- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

また、キューを検索用にオープンしていない場合もエラーになります。

## MQGMO\_MSG\_UNDER\_CURSOR

検索カーソルの位置で取り出します。

このオプションを指定すると、MQGMO 構造体の MatchOptions フィールドに指定された MQMO 構造体のオプションには関係なく、検索カーソルが指すメッセージを取り出します。MsgDesc 引数の MsgId フィールドまたは CorrelId フィールドで指定した値は無視されます。このメッセージは、キューから削除されます。

検索カーソルが指すメッセージは、MQGMO\_BROWSE\_FIRST または MQGMO\_BROWSE\_NEXT オプションで、一つ前に検索されたものです。

MQGMO\_COMPLETE\_MSG が MQGMO\_MSG\_UNDER\_CURSOR と一緒に指定された場合、検索カーソルは MQMD 構造体の Offset フィールドが 0 であるメッセージを特定しなければなりません。この条件が満たされない場合、理由コード MQRC\_INVALID\_MSG\_UNDER\_CURSOR で、MQGET 命令は失敗します。

次に示すオプションは、MQGMO\_MSG\_UNDER\_CURSOR オプションと同時に指定できません。

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_NEXT
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR
- MQGMO\_UNLOCK

また、キューを検索用と取り出し用の両方でオープンしていない場合もエラーになります。検索カーソルが取り出せるメッセージを指していなかった場合、MQGET 命令は失敗します。

## ロックオプション：

キュー上のメッセージの排他に関連するオプションを次に示します。

### MQGMO\_LOCK

メッセージを排他状態にします。

検索したメッセージを排他状態にします。そのキューのほかのオープンハンドルからは、そのメッセージを参照できなくなります。このオプションは、MQGMO\_BROWSE\_FIRST、MQGMO\_BROWSE\_NEXT、または MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR オプションのどれかと同時に指定してください。

それぞれのハンドルでは、一つのメッセージだけを排他状態にできます。排他状態にできるメッセージは、論理メッセージまたは物理メッセージとなります。

- MQGMO\_COMPLETE\_MSG が指定されると、論理メッセージを構成するすべてのメッセージセグメントが、キューハンドルにロックされます。ただし、すべてのメッセージがキューに存在し、取り出し可能でなければなりません。
- MQGMO\_COMPLETE\_MSG が指定されない場合、単一の物理メッセージだけがキューハンドルにロックされます。このメッセージが論理メッセージのセグメントであるとき、ロックされたセグメントは MQGMO\_COMPLETE\_MSG を使用するほかのアプリケーションが論理メッセージを取り出したり、検索したりするのを防止します。

完了コード MQCC\_FAILED が返された場合、または完了コード MQCC\_WARNING で理由コード MQRC\_TRUNCATED\_MSG\_FAILED が返された場合、メッセージは排他状態になりません。

あるハンドルで排他状態にしているメッセージがある場合は、そのメッセージは常に検索カーソルで指しているメッセージになります。

排他状態のメッセージは、次の MQGET 命令で取り出せます。その MQGET 命令では、同じ Hobj ハンドルを使用して、MQGMO\_MSG\_UNDER\_CURSOR オプション、またはそのほかの取り出し用オプションを指定します。

アプリケーションがメッセージを取り出さない場合は、次のようにして排他状態を解除します。

- 該当するハンドルを使用して、MQGMO\_BROWSE\_FIRST または MQGMO\_BROWSE\_NEXT オプションで、MQGET 命令を実行します。このとき、MQGMO\_LOCK オプションは、指定しなくても指定しなくてもかまいません。完了コード MQCC\_OK または MQCC\_WARNING で命令が終了した場合、メッセージの排他状態は解除されます。MQCC\_FAILED で完了した場合は、排他状態は解除されません。ただし、次の例外があります。
  - 完了コード MQCC\_WARNING、理由コード MQRC\_TRUNCATED\_MSG\_FAILED で終了した場合、メッセージの排他状態は解除されません。
  - 完了コード MQCC\_FAILED、理由コード MQRC\_NO\_MSG\_AVAILABLE で終了した場合、メッセージの排他状態は解除されます。

同時に MQGMO\_LOCK オプションを指定すると、ほかのメッセージが排他状態になります。MQGMO\_LOCK オプションを指定しない場合は、命令終了後には、どんなメッセージも排他状態になりません。

MQGMO\_WAIT オプションを指定した場合で、該当するメッセージが直後にはないときは、待ち合わせが始まる前に、元のメッセージの排他状態が解除されます。このため、エラーは発生しません。

- 該当するハンドルを使用して、MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR オプションで MQGET 命令を実行します。このとき、MQGMO\_LOCK は指定しません。完了コード MQCC\_OK または MQCC\_WARNING で命令が終了した場合、メッセージの排他状態は解除されます。MQCC\_FAILED で完了した場合は、メッセージの排他状態は解除されません。ただし、次の例外があります。
  - 完了コード MQCC\_WARNING, 理由コード MQRC\_TRUNCATED\_MSG\_FAILED で終了した場合、メッセージの排他状態は解除されません。
- 該当するハンドルを使用して、MQGMO\_UNLOCK オプションで MQGET 命令を実行します。
- 該当するハンドルを使用して、MQCLOSE 命令を実行します。これは、アプリケーションの終了によってキューがクローズされる場合も含まれます。

このオプションを指定するための、特別なオープンオプションは必要ありません。ただし、検索オプションを使用するためには、MQOO\_BROWSE オプションが必要です。

次に示すオプションは、MQGMO\_LOCK オプションと同時に指定できません。

- MQGMO\_SYNCPOINT
- MQGMO\_SYNCPOINT\_IF\_PERSISTENT
- MQGMO\_UNLOCK

## MQGMO\_UNLOCK

メッセージの排他状態を解除します。

MQGMO\_LOCK オプションを指定した以前の MQGET 命令で排他状態にしたメッセージがある場合だけ、このオプションを指定できます。このハンドルで排他状態にしたメッセージがない場合、完了コード MQCC\_WARNING, 理由コード MQRC\_NO\_MSG\_LOCKED で命令は終了します。

MQGMO\_UNLOCK オプションを指定した場合、MsgDesc 引数、BufferLength 引数、Buffer 引数、および DataLength 引数は参照・変更されません。また、Buffer 引数にメッセージは返されません。

オプションを省略して仮定された場合も含めて、MQGMO\_UNLOCK オプションと同時に指定できるのは、MQGMO\_NO\_WAIT オプションと MQGMO\_NO\_SYNCPOINT オプションだけです。そのほかのオプションと同時に指定すると不正になります。

このオプションを指定するための、特別なオープンオプションは必要ありません。ただし、メッセージを排他状態にする時点では、MQOO\_BROWSE オプションが必要です。

### メッセージデータオプション：

メッセージがキューから取り出されるとき処理に関連するオプションを次に示します。

## MQGMO\_ACCEPT\_TRUNCATED\_MSG

後部が切り捨てられたメッセージでも、取り出しを許容します。

メッセージを格納する十分なバッファ領域がなかった場合にこのオプションを指定していると、バッファ領域に格納できる部分のメッセージだけでも取り出しを許容します。このとき MQGET 命令は警告の完了コードを返して、次のように処理を完了します。

- メッセージの検索の場合は、検索カーソルは返されたメッセージの位置に進みます。
- メッセージの取り出しの場合は、返されたメッセージはキューから削除されます。
- そのほかにエラーが発生しなかった場合、理由コード MQRC\_TRUNCATED\_MSG\_ACCEPTED が返されます。

このオプションを指定しなかった場合も、バッファ領域分のメッセージが返されて警告の完了コードが返されますが、処理は完了しません。この場合は、次のように処理されます。

- メッセージの検索の場合は、検索カーソルは進みません。
- メッセージの取り出しの場合は、メッセージはキューから削除されません。
- そのほかにエラーが発生しなかった場合、理由コード MQRC\_TRUNCATED\_MSG\_FAILED が返されます。

## MQGMO\_CONVERT

メッセージデータを変換します。

このオプションは、メッセージ内のアプリケーションデータが、Buffer 引数にコピーされる前に、MQGET 命令の MsgDesc 引数で指定される CodedCharSetId フィールドおよび Encoding フィールドの値に従って変換されることを要求します。

メッセージが登録される時に指定される Format フィールドは、変換プロセスによってメッセージ内のデータ構造を記述しているとみなされます。メッセージデータは、組み込みフォーマットに関しては、キューマネージャで変換されます。

変換が成功した場合、MsgDesc 引数に指定された CodedCharSetId フィールドおよび Encoding フィールドは変更されません。すべての関連するデータタイプに関して変換できない場合は、MQGET 命令は失敗しないで完了しますが、メッセージデータのフィールドは変換されません。この場合は、MsgDesc 引数に指定された CodedCharSetId フィールドおよび Encoding フィールドには、未変換のメッセージの値が設定されます。

したがって、どちらの場合でも、これらのフィールドは Buffer 引数に設定されたメッセージデータの文字セット識別子およびマシンコード形式を表しています。

キューマネージャが変換するフォーマット名のリストの詳細は、この章の「[MQMD 構造体 - メッセージ記述子](#)」で Format フィールドの説明を参照してください。

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、「[付録 G コード変換対象の文字セット識別子一覧](#)」を参照してください。

### グループとセグメントに関するオプション：

グループ内のメッセージおよび論理メッセージのセグメントの処理に関連するオプションを次に示します。オプションを理解するために、次の用語について説明します。

#### 1. データタイプ

## 物理メッセージ

キューに登録したり取り出したりすることのできる最も小さい情報単位です。通常、単一の MQPUT 命令、MQPUT1 命令、または MQGET 命令で指定したり取り出したりする情報に該当します。

それぞれの物理メッセージは、メッセージ記述子 (MQMD 構造体) を持っています。

キューマネージャによって定義されているわけではありませんが、一般的に、物理メッセージはメッセージ記述子 (MQMD 構造体の MsgId フィールド) の値で区別されます。

## 論理メッセージ

アプリケーション情報の一単位です。システムの強制がない場合、論理メッセージは物理メッセージと同じである場合があります。ただし、論理メッセージが極端に大きい場合、論理メッセージを二つまたはそれ以上のセグメントと呼ばれる物理メッセージに分割することをお勧めします。また、分割が必要となる場合もあります。

セグメント分割された論理メッセージは、同一のヌルではないグループ記述子 (MQMD 構造体の GroupId フィールド) と同一のメッセージシーケンス番号 (MQMD 構造体の MsgSeqNumber フィールド) を持つ二つ以上の物理メッセージによって構成されます。セグメントはセグメントオフセット (MQMD 構造体の Offset フィールド) の値で区別されます。セグメントオフセットは、論理メッセージのデータ先頭からの、物理メッセージのデータの位置を表します。それぞれのセグメントは物理メッセージであるため、論理メッセージのセグメントは、通常、異なるメッセージ記述子を持ちます。

セグメント分割はされていないメッセージのうち、送信アプリケーションによってセグメント分割が許可されている論理メッセージも、ヌルではないグループ記述子を持ちます。この場合、論理メッセージがメッセージグループに属していないときは、そのグループ記述子を持つ物理メッセージはただ一つしか存在しないことになります。送信アプリケーションによってセグメント分割が禁止されている論理メッセージがメッセージグループに属していない場合は、ヌルのグループ記述子 (MQGL\_NONE) を持っています。

## メッセージグループ

ヌルではない同一のグループ記述子を持つ一つ以上の物理メッセージの集まりです。

グループ内の論理メッセージは、メッセージシーケンス番号の値で区別されます。メッセージシーケンス番号は、1 から N の整数範囲となります。なお、N はグループ内の論理メッセージの数を表します。一つ以上の論理メッセージがセグメント分割されている場合、グループ内には N 個以上の物理メッセージが存在します。

次に、メッセージが、MQGET 命令で返却される方法を制御する個々のオプションについて説明します。

## MQGMO\_LOGICAL\_ORDER

グループ内のメッセージや論理メッセージのセグメントを論理的な順序で返します。

キューハンドルに対し、成功した MQGET 命令によって返されるメッセージの順序を制御します。このオプションは、おのおのの命令に指定した場合に有効となります。

キューハンドルに対する MQGET 命令が成功した場合、MQGMO\_LOGICAL\_ORDER が指定されていると、グループ内のメッセージはメッセージシーケンス番号の順序で返されます。また、論理メッセージのセグメントは、セグメントオフセットの順序で返されます。この順序は、メッセージとセグメントのキュー内での位置には関係ありません。

キューマネージャは要求された順序でメッセージを返すために、成功した MQGET 命令間でのグループとセグメントの情報を保持しています。この情報で、キューハンドルに対する現在のメッセージグループと現在の論理メッセージとその位置、およびメッセージが同期点で取り出されているかどうかを識別します。キューマネージャがこの情報を保持しているため、アプリケーションはそれぞれの MQGET 命令に先立って、グループとセグメントの情報を指定する必要はありません。つまり、アプリケーションは MQMD 構造体の GroupId フィールドや MsgSeqNumber フィールド、および Offset フィールドを指定する必要はありません。ただし、アプリケーションはそれぞれの命令で MQGMO\_SYNCPOINT または MQGMO\_NO\_SYNCPOINT オプションを正しく指定しなければなりません。

キューのオープン時、現在のメッセージグループおよび現在の論理メッセージは存在しません。

MQMF\_MSG\_IN\_GROUP フラグが設定されたメッセージが MQGET 命令で返された場合は、そのメッセージグループが現在のメッセージグループとなります。成功した MQGET 命令で指定された MQGMO\_LOGICAL\_ORDER によって次に示すどちらかのメッセージが返されるまで、そのグループが現在のグループとなります。

- MQMF\_SEGMENT なしの MQMF\_LAST\_MSG\_IN\_GROUP  
(グループ内のセグメント分割されていない最終論理メッセージ)
- MQMF\_LAST\_SEGMENT ありの MQMF\_LAST\_MSG\_IN\_GROUP  
(グループ内の最終論理メッセージの最終セグメント)

上記のメッセージが返されると、メッセージグループは終了し、MQGET 命令の成功で現在のグループは存在しなくなります。同様に、論理メッセージは、MQGET 命令で MQMF\_SEGMENT フラグが設定されたメッセージが返されたときに現在の論理メッセージとなります。MQMF\_LAST\_SEGMENT フラグを持つメッセージが返されると、現在の論理メッセージは終了します。

何も選択されない場合、成功した MQGET 命令は、キューの最初にあるメッセージグループのメッセージ、2 番目にあるメッセージグループのメッセージの順に、有効なメッセージがなくなるまで正しい順序で返します。このとき、MatchOptions フィールドに、次に示すオプションを一つ以上指定することで、特定のメッセージグループを選択できます。

- MQMO\_MATCH\_MSG\_ID
- MQMO\_MATCH\_CORREL\_ID
- MQMO\_MATCH\_GROUP\_ID

ただし、これらのオプションは、現在のメッセージグループまたは現在の論理メッセージが存在しないときだけ有効となります。詳細については、「MatchOptions フィールド」を参照してください。

MQGET 命令が返すメッセージの検索時に、キューマネージャが参照する MsgId フィールド、CorrelId フィールド、GroupId フィールド、MsgSeqNumber フィールド、および Offset フィールドの値を次の表に示します。この値は、キューからメッセージを取り出すときと、キューのメッセージを参照するときの両方に適用されます。

指定した オプション	命令に先行するグループ と論理メッセージの状態		キューマネージャが検索する値				
	命令に先行する現在のメッセージグループがある	命令に先行する現在の論理メッセージがある	Msg Id フィールド	Correl Id フィールド	Group Id フィールド	MsgSeq Number フィールド	Offset フィールド
○	×	×	Match Options で 制御される	Match Options で 制御される	Match Options で 制御される	1	0
○	×	○	メッセージ 識別子	相関識別子	グループ識 別子*	1	先行するオフセット値とセグメント長の和
○	○	×	メッセージ 識別子	相関識別子	グループ識 別子*	シーケンス 番号 + 1*	0
○	○	○	メッセージ 識別子	相関識別子	グループ識 別子*	シーケンス 番号*	先行するオフセット値とセグメント長の和
×	—	—	Match Options で 制御される				

(凡例)

- ：該当します。
- ×
- ：有無に関係ありません。

注※

キューハンドルに対する前の命令で返された値です。

複数のメッセージグループがキューに存在する場合、各グループの最初の論理メッセージの先頭セグメントがキュー内のどこに位置するかによって、メッセージの返される順序が決まります。つまり、メッセージシーケンス番号が1でオフセットが0である物理メッセージによって、メッセージが返される時の順序が決まります。

MQGMO\_LOGICAL\_ORDER オプションは次に示すように、同期点に影響を及ぼします。

- グループの最初の論理メッセージまたはセグメントがトランザクション内で取り出された場合で、同じキューハンドルを使うときは、そのグループにあるほかのすべての論理メッセージとセグメントが、トランザクション内で取り出されなければなりません。しかし、それらのメッセージが同じトランザクション内で取り出される必要はありません。これによって、多くの物理メッセージで構成されるメッセージグループは、そのキューハンドルで二つ以上の連続したトランザクションに分割できます。

- グループの最初の論理メッセージまたはセグメントがトランザクション内で取り出されなかった場合で、同じキューハンドルを使うときは、そのグループのほかの論理メッセージとセグメントは、トランザクション内で取り出せません。

これらの条件が満たされなかった場合、MQGET 命令は理由コード MQRC\_INCONSISTENT\_UOW で失敗となります。MQGMO\_LOGICAL\_ORDER が指定されたとき、MQGET 命令で指定された MQGMO 構造体は、MQGMO\_VERSION\_2 より前のバージョンではいけません。また、MQMD 構造体は MQMD\_VERSION\_2 より前のバージョンではいけません。この条件が満たされない場合、理由コード MQRC\_WRONG\_GMO\_VERSION または MQRC\_WRONG\_MD\_VERSION で失敗となります。

キューハンドルで MQGET 命令に MQGMO\_LOGICAL\_ORDER が指定されない場合、メッセージはメッセージグループまたは論理メッセージのセグメントに属しているかどうかに関係なく返されます。これは、特定のグループまたは論理メッセージから、メッセージまたはセグメントが順序に従わないで返されることを示します。または、メッセージやセグメントが、ほかのグループや論理メッセージからのメッセージ、セグメント、およびグループに属していないセグメントのメッセージと混在していることを示しています。この場合、MQGET 命令で返される特定のメッセージは、それぞれの命令で指定される MQMO\_\* オプションで制御されます。詳細については、「MatchOptions フィールド」を参照してください。

システムに障害が起きた場合は、次に示す方法でメッセージグループや論理メッセージを途中から再開できます。システム再開時、アプリケーションは GroupId フィールド、MsgSeqNumber フィールド、Offset フィールド、および MatchOptions フィールドに適切な値を設定します。その後、MQGMO\_SYNCPOINT または MQGMO\_NO\_SYNCPOINT に適切な値を設定します。これによって、MQGMO\_LOGICAL\_ORDER を指定しないで MQGET 命令を発行できるようになります。この MQGET 命令が成功したとき、キューマネージャはグループとセグメントの情報を保持できます。したがって、そのキューハンドルを使用した次の MQGET 命令では、通常の MQGMO\_LOGICAL\_ORDER を指定できます。

キューマネージャが MQGET 命令用に保持しているグループとセグメントの情報は、MQPUT 命令用に保持しているグループとセグメントの情報とは別です。さらに、キューマネージャは、次に示す命令について別の情報を保持しています。

- キューからメッセージを取り出す MQGET 命令
- キューのメッセージを参照する MQGET 命令

どのキューハンドルについても、アプリケーションは MQGMO\_LOGICAL\_ORDER を指定した MQGET 命令と、指定しない MQGET 命令を混在させてもかまいません。ただし、次の点に注意してください。

- MQGMO\_LOGICAL\_ORDER を指定していない場合、MQGET 命令が成功するたびに、キューマネージャは返されたメッセージに対応する値を、保持しているグループおよびセグメント情報に設定します。その情報で、キューハンドルに対してキューマネージャが保持していたグループとセグメントの情報を置き換えます。そのとき、参照または取り出し命令の動作に必要な情報だけが変更されます。

- MQGMO\_LOGICAL\_ORDER が指定されない場合、現在のメッセージグループまたは論理メッセージが存在するとき、その命令は失敗しません。命令は完了コード MQCC\_WARNING で成功します。

グループおよびセグメントの情報と一致しない MQGET 命令および MQCLOSE 命令の結果を次の表に示します。

実行時の命令	直前の命令が MQGMO_LOGICAL_ORDER 指定ありの MQGET 命令	直前の命令が MQGMO_LOGICAL_ORDER 指定なしの MQGET 命令
MQGMO_LOGICAL_ORDER 指定ありの MQGET 命令	MQCC_FAILED	MQCC_FAILED
MQGMO_LOGICAL_ORDER 指定なしの MQGET 命令	MQCC_WARNING	MQCC_OK
非永続グループ、または論理メッセージのある MQCLOSE 命令	MQCC_WARNING	MQCC_OK

上記の表の場合で、完了コードが MQCC\_OK 以外するとき、理由コードは次のどれかになります。

MQRC\_INCOMPLETE\_GROUP

MQRC\_INCOMPLETE\_MSG

MQRC\_INCONSISTENT\_UOW

## 注意事項

キューマネージャは、参照するためにオープンされていますが、入力用にオープンされていないキューを参照したり、クローズしたりした場合には、グループとセグメントの情報をチェックしません。この場合、完了コードは常に MQCC\_OK（ほかのエラーがない場合）となります。

論理的な順序でメッセージとセグメントを取り出したいアプリケーションでは、MQGMO\_LOGICAL\_ORDER を指定することをお勧めします。このオプションを指定すると、キューマネージャがグループとセグメントの情報を管理するため、アプリケーションではキューマネージャとセグメントの情報を管理する必要がありません。しかし、アプリケーションによっては、MQGMO\_LOGICAL\_ORDER オプションが提供する機能より高い制御機能を必要とする場合があります。その場合、MQGMO\_LOGICAL\_ORDER オプションを指定しないでください。MQGMO\_LOGICAL\_ORDER を指定しなかったとき、アプリケーションは、MQGET 命令に先立って、MQMD 構造体の MsgId フィールド、CorrelId フィールド、GroupId フィールド、MsgSeqNumber フィールド、Offset フィールド、および MQGMO 構造体の MatchOptions フィールドの MQMO\_\* オプションを正しく指定しなければなりません。

例えば、受信した物理メッセージがグループまたは論理メッセージのセグメントに属する場合でも、転送するアプリケーションに MQGMO\_LOGICAL\_ORDER を指定しないでください。送信側と受信側のキューマネージャ間に複数の通信路がある複雑なネットワークの場合、MQGMO\_LOGICAL\_ORDER を指定すると、物理メッセージが順序どおりに到着しません。

MQGMO\_LOGICAL\_ORDER および MQPUT 命令で対応する MQPMO\_LOGICAL\_ORDER を指定しない場合は、各メッセージが到着すると、転送するアプリケーションは次の論理的な順序のメッセージを待たないで、そのメッセージを取り出したり転送したりできます。

MQGMO\_LOGICAL\_ORDER は、ほかの MQGMO\_\* オプションのどれとも同時に指定できます。また、上記に示したような適当な環境では、MQMO\_\* オプションの組み合わせとともに指定できます。

## MQGMO\_COMPLETE\_MSG

完全な論理メッセージだけ取り出します。

このオプションは、MQGET 命令で完全な論理メッセージだけ返すことを指定します。論理メッセージがセグメント分割されている場合、キューマネージャはセグメントを組み立てて、アプリケーションに完全な論理メッセージを返します。なお、論理メッセージがセグメント分割されていても、取り出すアプリケーションからは見えません。

### 注意事項

このオプションは、キューマネージャにメッセージセグメントを組み立てさせるためだけのオプションです。指定されなかった場合、セグメントがキューに存在していて、MQGET 命令で指定されたほかの条件を満たすときには、セグメントごとにアプリケーションに返されます。したがって、個々にセグメントを受信しないアプリケーションでは、常に MQGMO\_COMPLETE\_MSG を指定してください。

MQGMO\_COMPLETE\_MSG を使用する場合、アプリケーションは完全なメッセージを格納するのに十分なバッファを用意するか、MQGMO\_ACCEPT\_TRUNCATED\_MSG オプションを指定する必要があります。

ネットワークの遅延などによってキューにセグメントはなく、かつセグメント分割されたメッセージがある場合、MQGMO\_COMPLETE\_MSG を指定すると、不完全な論理メッセージに属するセグメントは取り出されません。ただし、そのメッセージセグメントは、CurrentQDepth キュー属性の値と関係があります。つまり、CurrentQDepth 属性の値が 0 より大きくても、取り出せる論理メッセージが存在しない場合があります。

キューマネージャはトランザクション内でだけセグメントを組み立てることができます。例えば、MQGET 命令がユーザの定義したトランザクション内で発行された場合、そのトランザクションが使用されます。命令が組み立てる途中で部分的に失敗した場合、キューマネージャは組み立て中に取り出されたセグメントをキューに戻します。その失敗によって、トランザクションのコミットの成功が妨げられることはありません。

セグメントであるそれぞれの物理メッセージは、それ自身のメッセージ記述子を持っています。一つの論理メッセージを構成するセグメントについては、メッセージ記述子のほとんどのフィールドが、論理メッセージ内のすべてのセグメントで同じになります。通常、論理メッセージ内のセグメント間で異なるのは、MsgId フィールド、Offset フィールド、および MsgFlags フィールドだけです。しかし、セグメントがネットワークの異なる通信路を通り、そのうちの幾つかの通信路で MCA 送信変換ができる場合、セグメントが最終的にあて先キューに到着したとき、セグメント間で CodedCharSetId フィールドと Encoding フィールドが異なる場合があります。CodedCharSetId フィールドまたは Encoding フィールドが異なるセグメントで構成されている論理メッセージは、キューマネージャで一つの論理メッセージとして組み立てることはできません。その代わりに、キューマネージャは、論理メッセージの最初か

ら同じ文字セット識別子とマシンコード形式を持つ連続した一部のセグメントを返し、MQGET 命令は MQCC\_WARNING の完了コードと MQRC\_INCONSISTENT\_CCSDS, または MQRC\_INCONSISTENT\_ENCODINGS の理由コードで完了します。これは、MQGMO\_CONVERT の指定には関係ありません。残りのセグメントを取り出すために、アプリケーションは MQGMO\_COMPLETE\_MSG オプションを指定しないで MQGET 命令を再発行し、一つ一つセグメントを取り出さなければなりません。このとき、MQGMO\_LOGICAL\_ORDER は残ったセグメントを順番に取り出すために使用できます。

セグメントを登録するアプリケーションは、メッセージ記述子のほかのフィールドに、セグメント間で異なる値を設定することもできます。しかし、受信するアプリケーションが MQGMO\_COMPLETE\_MSG を使用して論理メッセージを取り出す場合は意味がありません。キューマネージャが論理メッセージを組み立てるとき、メッセージ記述子の値に最初のセグメントの値を返します。唯一の例外は MsgFlags フィールドです。キューマネージャは、組み立てられたメッセージがただ一つのセグメントであることを示す値を設定します。

報告メッセージに MQGMO\_COMPLETE\_MSG が指定されると、キューマネージャは次に示す特別な処理をします。

キューマネージャは、論理メッセージ内の異なるセグメントに関する報告形式の報告メッセージが、キューにすべて存在するかチェックします。存在する場合、MQGMO\_COMPLETE\_MSG を指定すると、それらは 1 メッセージとして取り出されます。この処理を可能にするために、報告メッセージはセグメント分割をサポートするキューマネージャまたは MCA によって生成されなければなりません。または、報告を要求したアプリケーションが少なくとも 100 バイトのメッセージデータを要求しなければなりません。つまり、MQRO\*\_WITH\_DATA オプションまたは MQRO\*\_WITH\_FULL\_DATA オプションが指定されなければなりません。セグメントのデータがアプリケーションデータのすべてより小さい場合、失われたバイトは報告メッセージでヌルに置き換えられて返されます。

MQGMO\_COMPLETE\_MSG が MQGMO\_MSG\_UNDER\_CURSOR または MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR とともに指定された場合、参照カーソルは MQMD 構造体の Offset フィールドの値が 0 のメッセージに位置していなければなりません。この条件が満たされない場合、理由コード MQRC\_INVALID\_MSG\_UNDER\_CURSOR で失敗します。

MQGMO\_COMPLETE\_MSG は MQGMO\_ALL\_SEGMENTS\_AVAILABLE と同じ機能です。したがって、MQGMO\_ALL\_SEGMENTS\_AVAILABLE を指定する必要はありません。

なお、MQGMO\_COMPLETE\_MSG は、MQGMO\_SYNCPOINT\_IF\_PERSISTENT 以外の MQGMO\_\* オプションおよび MQMO\_MATCH\_OFFSET 以外の MQMO\_\* オプションと組み合わせて指定できます。

## MQGMO\_ALL\_MSGS\_AVAILABLE

グループのすべてのメッセージが有効である場合に取り出すことを指定します。

グループ内のメッセージは、すべてのメッセージが有効な場合だけ取り出し可能になります。ネットワークの遅延などによってキューに幾つかのメッセージが抜けているグループがある場合、MQGMO\_ALL\_MSGS\_AVAILABLE を指定すると、不完全なグループに属するメッセージを取り出すことを抑止できます。ただし、そのメッセージは CurrentQDepth キュー属性の値と関係があります。つまり、CurrentQDepth 属性の値が 0 より大きくても、取り出せるメッセージグループが存在しない場合があります。取り出せるメッセージがない場合、待ち時間が指定されていると、待ち時間が経過したあとで MQRC\_NO\_MSG\_AVAILABLE の理由コードが返されます。

MQGMO\_ALL\_MSGS\_AVAILABLE が有効となるかどうかは、MQGMO\_LOGICAL\_ORDER が指定されているかどうかによって、次に示すように決まります。

- MQGMO\_ALL\_MSGS\_AVAILABLE と MQGMO\_LOGICAL\_ORDER が両方とも指定された場合、MQGMO\_ALL\_MSGS\_AVAILABLE は現在のグループまたは現在の論理メッセージがないときだけ有効となります。現在のグループまたは現在の論理メッセージがある場合、MQGMO\_ALL\_MSGS\_AVAILABLE は無視されます。したがって、MQGMO\_ALL\_MSGS\_AVAILABLE は、論理メッセージのメッセージを処理するときに無効となります。
- MQGMO\_LOGICAL\_ORDER なしで MQGMO\_ALL\_MSGS\_AVAILABLE が指定された場合、MQGMO\_ALL\_MSGS\_AVAILABLE は常に有効となります。したがって、グループの最初のメッセージがキューから取り出されたあと、そのグループの残りのメッセージを削除するためには、オプションの指定を外さなければなりません。

MQGMO\_ALL\_MSGS\_AVAILABLE を指定する MQGET 命令が正常に完了した場合、MQGET 命令が発行された時点でグループ内のすべてのメッセージがキュー上にあることになります。しかし、ほかのアプリケーションがグループからメッセージを除去できることに注意してください。グループは、グループ内の最初のメッセージを検索するアプリケーションにロックされていません。

MQGMO\_ALL\_MSGS\_AVAILABLE が指定されない場合、グループに属するメッセージはグループが不完全なときも取り出すことができます。

MQGMO\_ALL\_MSGS\_AVAILABLE は MQGMO\_ALL\_SEGMENTS\_AVAILABLE と同じ機能です。したがって、MQGMO\_ALL\_SEGMENTS\_AVAILABLE を指定する必要はありません。

MQGMO\_ALL\_MSGS\_AVAILABLE は MQGMO 構造体のほかのオプション、または MQMO 構造体のオプションのどのオプションとも同時に指定できます。

## MQGMO\_ALL\_SEGMENTS\_AVAILABLE

論理メッセージのすべてのセグメントが有効である場合に取り出すことを指定します。

論理メッセージ内のセグメントは、論理メッセージ内のすべてのセグメントが有効にならない限り取り出せません。ネットワークの遅延などによってキューに幾つかのセグメントが抜けている論理メッセージがある場合、MQGMO\_ALL\_SEGMENTS\_AVAILABLE を指定すると、不完全な論理メッセージに属するセグメントを取り出すことを抑止できます。ただし、そのセグメントは CurrentQDepth キュー属性の値と関係があります。つまり、CurrentQDepth 属性の値が 0 より大きくても、取り出せるメッセージが存在しない場合があります。取り出せるメッセージがない場合、待ち時間が指定されていると、その待ち時間が経過したあとで MQRC\_NO\_MSG\_AVAILABLE の理由コードが返されます。

MQGMO\_ALL\_SEGMENTS\_AVAILABLE が有効であるかどうかは、MQGMO\_LOGICAL\_ORDER が指定されているかどうかによって次に示すように決まります。

- MQGMO\_ALL\_SEGMENTS\_AVAILABLE と MQGMO\_LOGICAL\_ORDER が両方とも指定された場合、MQGMO\_ALL\_SEGMENTS\_AVAILABLE は現在の論理メッセージがないときだけ有効となります。現在の論理メッセージがある場合、MQGMO\_ALL\_SEGMENTS\_AVAILABLE は無視されます。したがって、MQGMO\_ALL\_SEGMENTS\_AVAILABLE は、論理メッセージのメッセージを処理するときに無効となります。

- MQGMO\_LOGICAL\_ORDER なしで MQGMO\_ALL\_SEGMENTS\_AVAILABLE が指定された場合、MQGMO\_ALL\_SEGMENTS\_AVAILABLE は常に有効となります。したがって、論理メッセージの最初のセグメントがキューから取り出されたあと、その論理メッセージの残りのセグメントを削除するためには、オプションの指定を外さなければなりません。

MQGMO\_ALL\_SEGMENTS\_AVAILABLE が指定されない場合、論理メッセージに属するセグメントは、論理メッセージが不完全なときも取り出すことができます。

MQGMO\_COMPLETE\_MSG と MQGMO\_ALL\_SEGMENTS\_AVAILABLE はすべてのセグメントが有効になるまで、セグメントの取り出しを禁止します。MQGMO\_COMPLETE\_MSG は完全なメッセージを返します。MQGMO\_ALL\_SEGMENTS\_AVAILABLE は、セグメントを一つ一つ取り出せるようにします。

MQGMO\_ALL\_SEGMENTS\_AVAILABLE が報告メッセージに指定された場合、キューマネージャは次の特別な処理をします。

キューマネージャは、完全な論理メッセージを構成する各セグメントに対して、報告メッセージが少なくとも一つ存在するかどうか、キューをチェックします。報告メッセージが存在する場合は、MQGMO\_ALL\_SEGMENTS\_AVAILABLE の条件が満たされています。

ただし、キューマネージャは報告メッセージの形式まではチェックしないので、論理メッセージのセグメントに関連するさまざまな報告メッセージの形式が混在している可能性があります。つまり、MQGMO\_ALL\_SEGMENTS\_AVAILABLE が成功しても、MQGMO\_COMPLETE\_MSG が成功するとは限りません。特別な論理メッセージのセグメントに対する報告形式が混在している場合、その報告メッセージは一つ一つ取り出されなければなりません。

MQGMO\_ALL\_SEGMENTS\_AVAILABLE は、ほかの MQGMO\_\* オプション、または MQMO\_\* オプションと同時に指定できます。

## デフォルトオプション：

ここまでで説明されたオプションを指定しない場合、次に示すオプションを使用できます。

### MQGMO\_NONE

オプションを指定しません。

このオプションは、ほかにオプションを指定しない場合に指定します。このほかのオプションには、どれも省略時の仮定値があります。MQGMO\_NONE オプションは、プログラムのコーディングをわかりやすくするために定義されています。値としては 0 が定義されていますが、コーディングをわかりやすくする以外の目的はありません。

このフィールドの初期値は MQGMO\_NO\_WAIT です。

### ● Reserved1 (MQCHAR 型) 予備

これは予備のフィールドです。このフィールドの初期値は空白です。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

## ● ResolvedQName (MQCHAR48 型) 受信キュー名

キューマネージャによって設定される出力用のフィールドです。メッセージを取り出したキューの名称が返されます。返される名称はローカルキューマネージャに定義された名称です。

次に示す場合はオープン時に指定したキューの名称とは異なる名称が返されます。

- 別名キューをオープンした場合は、別名キューのベースキューの名称が返されます。
- モデルキューをオープンした場合は、動的ローカルキューの名称が返されます。

このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● Segmentation (MQCHAR 型) セグメント分割フラグ

取り出されたメッセージがセグメント分割できるかどうかを示すフラグです。

次の値を取ります。

### MQSEG\_INHIBITED

セグメント分割は許可されていません。

### MQSEG\_ALLOWED

セグメント分割が許可されています。

これは出力用のフィールドです。

このフィールドの初期値は MQSEG\_INHIBITED です。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

## ● SegmentStatus (MQCHAR 型) 論理メッセージフラグ

取り出されたメッセージが、論理メッセージグループのセグメントかどうかを示すフラグです。

次の値を取ります。

### MQSS\_NOT\_A\_SEGMENT

メッセージはセグメントではありません。

### MQSS\_SEGMENT

メッセージはセグメントですが、論理メッセージの最終セグメントではありません。

### MQSS\_LAST\_SEGMENT

メッセージは論理メッセージの最終セグメントです。

これは、論理メッセージがただ一つのセグメントで構成されている場合にも返される値です。

これは出力用のフィールドです。

このフィールドの初期値は MQSS\_NOT\_A\_SEGMENT です。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

### ● Signal1 (MQLONG 型) シグナル

TP1/Message Queue では、このフィールドは使用されません。

このフィールドの初期値は 0 です。

### ● Signal2 (MQLONG 型) 予備

予備のフィールドです。キューマネージャはこの内容を特に使用しません。

このフィールドの初期値は 0 です。

### ● StrucId (MQCHAR4 型) 構造体識別子

次の値を取ります。

#### MQGMO\_STRUC\_ID

メッセージ取り出しオプションの構造体識別子

C 言語では、MQGMO\_STRUC\_ID\_ARRAY も定義されています。これは、MQGMO\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

これは入力用のフィールドです。

このフィールドの初期値は MQGMO\_STRUC\_ID です。

### ● Version (MQLONG 型) 構造体バージョン番号

次に示す値を取ります。

#### MQGMO\_VERSION\_1

バージョン 1 のメッセージ取り出しオプションの構造体バージョン番号

#### MQGMO\_VERSION\_2

バージョン 2 のメッセージ取り出しオプションの構造体バージョン番号

MQGMO\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことがフィールドの説明に記載されています。次に示す定数は、現在のバージョン番号を指定します。

#### MQGMO\_CURRENT\_VERSION

メッセージ取り出しオプションの現在の構造体バージョン番号

これは入力用のフィールドです。

このフィールドの初期値は MQGMO\_VERSION\_1 です。

## ● WaitInterval (MQLONG 型) 待ち合わせ最大時間

MQGET 命令が該当するメッセージの到着を待つ最大時間を、ミリ秒単位で指定します。該当するメッセージとは、MQGET 命令の MsgDesc 引数に指定した選択規準を満たすものです。詳細は、この章の「MQMD 構造体 – メッセージ記述子」で MsgId フィールドの説明を参照してください。この時間が経過しても該当するメッセージが到着しなかった場合、完了コード MQCC\_FAILED, 理由コード MQRC\_NO\_MSG\_AVAILABLE で命令は終了します。

このフィールドは、MQGMO\_WAIT オプションと組み合わせて使用します。MQGMO\_WAIT オプションを指定しなかった場合、このフィールドは無視されます。指定した場合は、WaitInterval フィールドには 0 以上を指定してください。また、次の定数値も指定できます。

### MQWI\_UNLIMITED

無制限に待ち合わせをします。無制限に待ち合わせ中に、アプリケーションまたは OpenTP1 が終了した場合、MQGET 命令は MQRC\_Q\_MGR\_STOPPING の理由コードで完了します。

このフィールドの初期値は 0 です。TP1/Message Queue では待ち合わせ最大時間の精度を MQA サービス定義の mqa\_getwait\_timeout オペランドで指定できます。指定を省略した場合、秒単位の精度で待ち合わせます。

## MQMD 構造体 – メッセージ記述子

MQMD 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQMD_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQMD_VERSION_1
Report (MQLONG 型)	報告メッセージオプション	MQRO_NONE
MsgType (MQLONG 型)	メッセージタイプ	MQMT_DATAGRAM
Expiry (MQLONG 型)	メッセージ保持時間	MQEI_UNLIMITED
Feedback (MQLONG 型)	報告メッセージ返答コード	MQFB_NONE
Encoding (MQLONG 型)	メッセージデータのマシンコード形式	MQENC_NATIVE
CodedCharSetId (MQLONG 型)	メッセージデータの文字セット識別子	MQCCSI_Q_MGR
Format (MQCHAR8 型)	メッセージデータのフォーマット名	MQFMT_NONE
Priority (MQLONG 型)	メッセージ優先度	MQPRI_PRIORITY_AS_Q_DEF
Persistence (MQLONG 型)	メッセージ永続性	MQPER_PERSISTENCE_AS_Q_DEF
MsgId (MQBYTE24 型)	メッセージ識別子	MQMI_NONE
CorrelId (MQBYTE24 型)	相関識別子	MQCI_NONE
BackoutCount (MQLONG 型)	ロールバック回数	0
ReplyToQ (MQCHAR48 型)	応答キュー名	ヌル文字列/空白
ReplyToQMgr (MQCHAR48 型)	応答キューマネージャ名	ヌル文字列/空白
UserIdentifier (MQCHAR12 型)	ユーザ識別子	ヌル文字列/空白
AccountingToken (MQBYTE32 型)	課金トークン	MQACT_NONE
ApplIdentityData (MQCHAR32 型)	アプリケーション識別データ	ヌル文字列/空白
PutApplType (MQLONG 型)	登録アプリケーションタイプ	MQAT_NO_CONTEXT
PutApplName (MQCHAR28 型)	登録アプリケーション名	ヌル文字列/空白
PutDate (MQCHAR8 型)	登録日付	ヌル文字列/空白
PutTime (MQCHAR8 型)	登録時刻	ヌル文字列/空白
ApplOriginData (MQCHAR4 型)	登録元データ	ヌル文字列/空白
GroupId (MQBYTE24 型) *	グループ識別子	MQGI_NONE
MsgSeqNumber (MQLONG 型) *	メッセージシーケンス番号	1
Offset (MQLONG 型) *	オフセット	0

### 1. データタイプ

フィールド (データタイプ)	内容	初期値
MsgFlags (MQLONG 型) ※	メッセージフラグ	MQMF_NONE
OriginalLength (MQLONG 型) ※	登録元メッセージ長	MQOL_UNDEFINED

注※

Version フィールドの値が MQMD\_VERSION\_2 以降の場合だけ存在するフィールドです。

## 概要

### 目的

MQMD 構造体には、アプリケーション間でデータをやり取りするために必要な制御情報を記述します。この構造体は MQGET, MQPUT および MQPUT1 命令の入出力用の引数です。

### バージョン

MQMD 構造体のバージョンには 1 および 2 があります。しかし、すべての環境で MQMD\_VERSION\_2 がサポートされているわけではありません。複数の環境にアプリケーションを移植する場合は、使用するバージョンの MQMD 構造体が、すべての環境でサポートされている必要があります。

MQMD\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことがフィールドの説明に記載されています。

サポートするプログラミング言語の COPY ファイルと INCLUDE ファイルでは、最新バージョンの MQMD 構造体が提供されます。ただし、Version フィールドの初期値は MQMD\_VERSION\_1 です。バージョン 1 の構造体がないフィールドを使用する場合は、アプリケーションで使用したいバージョンの番号を Version フィールドに設定してください。

バージョン 1 の構造体宣言には、MQMD1 型を使用できます。

### 文字セットおよびマシンコード

MQMD 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。TP1/Message Queue Access のクライアントアプリケーションの場合、MQMD 構造体はクライアントの文字セットおよびマシンコードにします。

送信側および受信側キューマネージャが使用する文字セットまたはマシンコードが異なる場合、MQMD 構造体の中のデータは自動的に変換されます。アプリケーションが MQMD 構造体を変換する必要はありません。

### 異なるバージョンの MQMD 構造体の使用

バージョン 1 の MQMD 構造体のアプリケーションデータの前に MQMDE 構造体を追加すると、バージョン 2 の MQMD 構造体と同じ機能になります。ただし、MQMDE 構造体の全フィールドが初期値の場合、MQMDE 構造体は省略できます。バージョン 1 の MQMD 構造体と MQMDE 構造体は次のように使用されます。

- MQPUT 命令および MQPUT1 命令で、アプリケーションがバージョン 1 の MQMD 構造体を使用する場合、アプリケーションは、MQMDE 構造体があることを示すために、MQMD 構造体の Format フィールドに MQFMT\_MD\_EXTENSION を設定します。これによって、メッセージデータの前に、MQMDE 構造体をオプションで追加できます。

## 注意事項

バージョン 2 の MQMD 構造体に存在し、バージョン 1 の MQMD 構造体に存在しない幾つかのフィールドは、MQPUT 命令および MQPUT1 命令の入出力フィールドです。ただし、キューマネージャは、MQPUT 命令および MQPUT1 命令からの出力では、MQMDE 構造体の対応するフィールドへ値を返しません。アプリケーションが出力された値を必要とする場合は、バージョン 2 の MQMD 構造体を使用しなければなりません。

- MQGET 命令でアプリケーションがバージョン 1 の MQMD 構造体を指定すると、キューマネージャは返すメッセージの前に MQMDE 構造体を追加します。ただし、MQMDE 構造体の一つ以上のフィールドが初期値以外の値を持つ場合だけです。MQMD 構造体の Format フィールドは MQMDE 構造体が存在することを示すために、MQFMT\_MD\_EXTENSION となります。

この構造体は、MQGET 命令、MQPUT 命令、および MQPUT1 命令の入出力用の引数です。

サポートされるすべてのプログラム言語について、MQMD 構造体のデフォルト値はバージョン 1 構造体のデフォルト値と同じです。また、明示的な MQMD1 構造体を利用できます (COBOL 言語では COPY ファイルの CMQMD1 と CMQMD1V)。この構造体は、バージョン 1 の MQMD 構造体とバージョン 2 の MQMD 構造体を扱う必要があるアプリケーションのために提供されています。

メッセージが転送キュー上にある場合、MQMD 構造体内のフィールドの一部が特定の値に設定されます。詳細については、この章の「MQXQH 構造体 - 転送キューヘッダ」を参照してください。

## メッセージコンテキスト

MQMD 構造体の特定のフィールドには、メッセージコンテキストが含まれます。メッセージテキストには、識別コンテキストと登録元コンテキストがあります。

### 識別コンテキスト

該当するメッセージを最初に登録したアプリケーションに関係します。

### 登録元コンテキスト

該当するメッセージを最後に登録したアプリケーションに関係します。

これらの二つのアプリケーションは、同一のアプリケーションである場合と、同一のアプリケーションでない場合 (例えば、メッセージがアプリケーションからアプリケーションへ転送される場合) があります。

識別コンテキストと登録元コンテキストには通常、上記のような意味があります。しかし実際には、MQMD 構造体のコンテキストフィールドの内容は、メッセージ登録時に指定された MQPMO\_\*\_CONTEXT オプションによって決定されます。そのため、必ずしも上記のとおりではありません。アプリケーションの設計に依存します。

メッセージのコンテキストをまったく変更しないアプリケーションのクラスには、メッセージチャネルエージェント (MCA) があります。リモートキューマネージャからメッセージを受信する MCA は、MQPUT または MQPUT1 命令にコンテキストオプション MQPMO\_SET\_ALL\_CONTEXT を使用します。これによって受信側の MCA は、送信側の MCA からメッセージとともに送信されたメッセージコンテキストを、そのままの状態でも保存できます。ただし、その結果として、登録元コンテキストはメッセージを最後に登録したアプリケーション (受信側 MCA) に関係づけられないで、それ以前にメッセージを登録したアプリケーション (送信側アプリケーション自身を含む) に関係づけられます。

以降の説明では、上記に示した通常の方法でコンテキストフィールドが使用されている場合について説明します。メッセージのコンテキストの詳細については、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。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● AccountingToken (MQBYTE32 型) 課金トークン

メッセージの識別コンテキストの一部です。メッセージコンテキストについては、この章の「MQMD 構造体 - メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

アプリケーションが課金情報を必要とするときに使用できます。

この情報は、キューマネージャではビット列として扱われます。内容はチェックされません。キューマネージャによってこの情報が生成される時、次の値が設定されます。

- 1 バイト目に、2 バイト目以降に格納する課金情報の長さが設定されます。長さの範囲は 0~31 で、2 進数として格納されます。
- 2 バイト目以降に、上記に示された長さで環境に応じた課金情報が設定されます。
- 上記以降のバイトには、2 進数の 0 が設定されます。

MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_IDENTITY\_CONTEXT または MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。

MQPMO\_SET\_IDENTITY\_CONTEXT も MQPMO\_SET\_ALL\_CONTEXT も指定しなかった場合、このフィールドは入力時に無視されて出力専用のフィールドとなり、このフィールドは「¥0」で埋められます。メッセージコンテキストおよびコンテキスト情報の制御については、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。

MQPUT 命令または MQPUT1 命令が成功すると、このフィールドには、転送されたメッセージの AccountingToken が設定されます。メッセージにコンテキストがなかった場合、このフィールドは 0 となります。

MQGET 命令では、出力用フィールドとなります。

このフィールドは、キューマネージャの文字セットの変換対象にはなりません。このフィールドは、文字列ではなく、ビット列として扱われます。

キューマネージャはこの情報を操作しません。この情報を課金目的で使用するときは、アプリケーションがそれを解釈してください。

次の定数値も使用できます。

## MQACT\_NONE

課金トークンを指定しません。

この定数値は、2進数の0が設定されます。

C言語では、MQACT\_NONE\_ARRAYも定義されています。これは、MQACT\_NONEと同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの長さは、MQ\_ACCOUNTING\_TOKEN\_LENGTHに定義されています。

このフィールドの初期値はMQACT\_NONEです。

## ● ApplIdentityData (MQCHAR32型) アプリケーション識別データ

メッセージの識別コンテキストの一部です。メッセージコンテキストについては、この章の「MQMD 構造体 - メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

このフィールドでは、アプリケーションの組を定義します。また、メッセージまたは登録元に関する付加情報に使用できます。

この情報は、キューマネージャでは文字データとして扱われます。ただし、データの形式は規定されません。キューマネージャによってこの情報が生成される時、空白が設定されます。

MQPUT 命令およびMQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_IDENTITY\_CONTEXT または MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。このフィールドにヌル文字があった場合、ヌル文字以降の文字は、キューマネージャによって空白に置き換えられます。MQPMO\_SET\_IDENTITY\_CONTEXT も MQPMO\_SET\_ALL\_CONTEXT も指定しなかった場合、このフィールドは入力時に無視されて出力専用フィールドとなり、このフィールドは空白で埋められます。メッセージコンテキストおよびコンテキスト情報の制御については、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。

MQPUT 命令またはMQPUT1 命令が成功すると、このフィールドには、転送されたメッセージの ApplIdentityData が設定されます。メッセージにコンテキストがなかった場合、このフィールドは空白で埋められます。

MQGET 命令では、出力用フィールドとなります。

このフィールドの長さは、MQ\_APPL\_IDENTITY\_DATA\_LENGTHに定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、32 個の空白です。

### ● ApplOriginData (MQCHAR4 型) 登録元データ

メッセージの登録元コンテキストの一部です。メッセージコンテキストについては、この章の「MQMD 構造体 – メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

ApplOriginData は、アプリケーションの組で定義する情報で、メッセージの登録元に関する付加情報として使用できます。例えば、適切な権限を持つアプリケーションが、識別データが信用できるかどうかを示すために設定できます。

この情報は、キューマネージャでは文字データとして扱われます。ただし、データの形式は規定されません。キューマネージャによってこの情報が生成される時、空白が設定されます。

MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。このフィールドのヌル文字以降の情報は破棄されます。ヌル文字以降の文字は、キューマネージャによって空白に置き換えられます。MQPMO\_SET\_ALL\_CONTEXT を指定しなかった場合、このフィールドは入力時に無視されて出力専用フィールドとなり、このフィールドは空白で埋められます。

MQPUT 命令または MQPUT1 命令が成功すると、このフィールドには転送されたメッセージの ApplOriginData が設定されます。メッセージにコンテキストがなかった場合、このフィールドは空白で埋められます。

MQGET 命令では、出力用フィールドとなります。

このフィールドの長さは、MQ\_APPL\_ORIGIN\_DATA\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、4 個の空白です。

### ● BackoutCount (MQLONG 型) ロールバック回数

トランザクション内の MQGET 命令でメッセージが返されて、そのあとにロールバックした回数です。アプリケーションはこの回数によって、メッセージの内容が原因のエラーを検出できます。この回数には、MQGMO\_BROWSE\_\* オプションを指定した MQGET 命令は含まれません。

これは、MQGET 命令の出力用フィールドです。MQPUT 命令および MQPUT1 命令のときは無視されます。

このフィールドの初期値は 0 です。

### ● CodedCharSetId (MQLONG 型) 文字セット識別子

メッセージデータの文字セット識別子です。

## 注意事項

命令引数である MQMD 構造体およびほかの MQ データ構造体の文字データは、キューマネージャの文字セットである必要があります。これは、キューマネージャの CodedCharSetId 属性によって定義されている文字セットです。属性の詳細については、「3. オブジェクトの属性」の「[キューマネージャの属性](#)」を参照してください。

次の値が使用できます。

### MQCCSI\_Q\_MGR

キューマネージャの文字セット識別子

アプリケーションメッセージデータの文字データは、キューマネージャの文字セット（MQA サービス定義の mqa\_local\_ccsid オペランド指定値）を使用します。

MQPUT と MQPUT1 命令では、キューマネージャはアプリケーションメッセージとともに設定する MQMD 構造体のフィールドの値をキューマネージャの文字セット識別子（MQA サービス定義の mqa\_local\_ccsid オペランド指定値）に変更します。そのため、MQGET 命令の際に、MQCCSI\_Q\_MGR という値が返されることはありません。

### MQCCSI\_DEFAULT

デフォルトの文字セット識別子

String フィールドのデータの CodedCharSetId は、MQCFH 構造体に先行するヘッダ構造体の CodedCharSetId フィールドによって定義されます。MQCFH 構造体がメッセージの先頭にある場合は MQMD 構造体の CodedCharSetId フィールドによって定義されます。

### MQCCSI\_INHERIT

この構造体の文字セット識別子を引き継ぎます。

メッセージの文字データは、この構造体と同じ文字セットです。つまり、キューマネージャの文字セットです（MQMD 構造体の場合だけ、MQCCSI\_INHERIT は MQCCSI\_Q\_MGR と同じ意味になります）。

メッセージで送信される MQMD 構造体のこの値を、キューマネージャは MQMD 構造体の実際の文字セット識別子に変更します。エラーが発生しない場合、MQGET 命令では MQCCSI\_INHERIT は返されません。

MQMD 構造体の PutApplType フィールドの値が MQAT\_BROKER の場合、MQCCSI\_INHERIT は使用できません。

### MQCCSI\_EMBEDDED

組み込みの文字セット識別子

メッセージ内の文字データの文字セットは、メッセージデータ自身に含まれている識別子の文字セットです。メッセージデータに組み込まれていて、データの異なる部分に適用される文字セット識別子は幾つあってもかまいません。この値は、幾つかの文字セットによるデータが混在した PCF メッセージでは必ず使用されます。PCF メッセージのフォーマット名は MQFMT\_PCF です。

この値は MQPUT 命令および MQPUT1 命令にだけ指定します。MQGET 命令にこの値を指定すると、メッセージは変換されません。

MQPUT および MQPUT1 命令では、メッセージとともに送信される MQMD 構造体中の MQCCSI\_Q\_MGR および MQCCSI\_INHERIT という値を、キューマネージャはキューマネージャの文字セット識別子と同じ値に変更します。しかし、MQPUT または MQPUT1 命令に指定された MQMD 構造体は変更しません。指定された値について、ほかのチェックはありません。

メッセージを取り出すアプリケーションは、このフィールドが期待した値かどうかを比較します。期待した値以外の場合、アプリケーションはメッセージ中の文字データを変換する必要があります。

MQGMO\_CONVERT オプションが MQGET 命令で指定される場合、このフィールドは入出力用フィールドです。アプリケーションで指定された値は、メッセージデータの必要に応じて変換される文字セット識別子です。変換が成功した場合、または変換が不要の場合、値は変更されません。ただし、MQCCSI\_Q\_MGR および MQCCSI\_INHERIT が実際の値に変換される場合を除きます。変換が成功しなかった場合、MQGET 命令後の値はアプリケーションに設定される未変換メッセージの文字セット識別子を表します。

MQGMO\_CONVERT オプションが MQGET 命令で指定されない場合、このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQCCSI\_Q\_MGR です。

## ● CorrelId (MQBYTE24 型) 相関識別子

MQGET 命令では、取り出すメッセージの相関識別子を指定します。このフィールドの指定方法は MsgId フィールドと同様です。

MQGET 命令の GetMsgOpts 引数で MQGMO\_MSG\_UNDER\_CURSOR オプションを指定した場合、このフィールドは無視されます。

MQGET 命令が返されると、取り出したメッセージの相関識別子が CorrelId フィールドに設定されます。

MQPUT 命令および MQPUT1 命令では、アプリケーションは任意の相関識別子を指定できます。その値は、キューマネージャによってメッセージを取り出すアプリケーションに返されます。

MQPUT および MQPUT1 命令に MQPMO\_NEW\_CORREL\_ID オプションを指定する場合、登録したメッセージの相関識別子に、ユニークな相関識別子<sup>※</sup>がキューマネージャによって設定されます。

### 注※

キューマネージャによって生成される相関識別子 (CorrelId) は、4 バイトのプロダクト識別子と、それに続くプロダクト固有の文字列で構成されます。この CorrelId は、EBCDIC コードまたは ASCII コードの A~I の範囲の文字で始まっています。アプリケーションで CorrelId を指定する場合は、キューマネージャによって生成される値との重複を避けるために、A~I 以外の文字で始まる文字列を指定してください。

報告が生成されるとき、登録元メッセージの Report フィールド (MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID または MQRO\_PASS\_CORREL\_ID) の方法で, CorrelId が設定されます。アプリケーションが報告を生成するときも、同様の処理をする必要があります。

このフィールドは、キューマネージャの文字セットの変換対象にはなりません。このフィールドは、文字列ではなく、ビット列として扱われます。

次の定数値を使用できます。

## MQCI\_NONE

関連識別子を指定しません。

この定数の値は、2 進数の 0 です。

C 言語では、MQCI\_NONE\_ARRAY も定義されています。これは、MQCI\_NONE と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

これは、MQGET 命令の入出力用フィールドです。また、MQPUT 命令と MQPUT1 命令では、MQPMO\_NEW\_CORREL\_ID の指定がない場合は入力用フィールドとなり、指定された場合は出力用フィールドとなります。

このフィールドの長さは、MQ\_CORREL\_ID\_LENGTH に定義されています。

このフィールドの初期値は MQCI\_NONE です。

## ● Encoding (MQLONG 型) マシンコード形式

メッセージ内の数値データのマシンコード形式を指定します。ただし、MQMD 構造体自身の数値データには適用されません。マシンコード形式は、2 進整数、10 進パック形式整数および浮動小数点の表記を定義します。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。キューマネージャは、フィールドが有効かどうかをチェックしません。次に示す特別な値が定義されます。

## MQENC\_NATIVE

計算機固有のマシンコード形式

このマシンコード形式は、アプリケーションを実行するプログラミング言語と計算機に固有の値です。

### ■ 注意事項

この定数の値は、プログラミング言語と環境に依存します。このため、アプリケーションは、実行する環境に合った COPY ファイルと INCLUDE ファイルを使用してコンパイルする必要があります。

メッセージを登録するアプリケーションは、通常、MQENC\_NATIVE を指定します。メッセージを取り出すアプリケーションは、このフィールドが MQENC\_NATIVE と等しいかどうかを比較します。等しくない場合は、アプリケーションはメッセージ中の数値データを変換する必要があります。MQGET 命令の

処理でキューマネージャがメッセージを変換するように指定するには、MQGMO\_CONVERT オプションを使用できます。Encoding フィールドの指定方法については、「付録 C マシンコード形式」を参照してください。

MQGMO\_CONVERT オプションが MQGET 命令で指定される場合、このフィールドは入出力用フィールドです。アプリケーションで指定された値は、メッセージデータの必要に応じて変換されるマシンコード形式です。変換が成功した場合、または変換が不要な場合、値は変更されません。変換が成功しなかった場合、MQGET 命令後の値はアプリケーションに設定される未変換メッセージのマシンコード形式を表します。

MQGMO\_CONVERT オプションが MQGET 命令で指定されない場合、このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQENC\_NATIVE です。

## ● Expiry (MQLONG 型) メッセージ保持時間

メッセージを登録するアプリケーションが指定する、1/10 秒単位の時間です。この時間が経過する前にメッセージがあて先キューから取り出されないと、このメッセージは破棄できる状態になります。

あて先キューで時間が経過するごとに、この時間が減少します。リモートキューに登録した場合は、仲介するすべての転送キューで時間が減少します。また、転送時間が大きく影響する場合、MCA がこの時間を減少させます。必要に応じて、メッセージを次のキューへ進めるアプリケーションで同様に時間を減少させてください。ただし、メッセージ保持時間は概算で扱われるため、少しの時間では減少しません。

アプリケーションが MQGET 命令でメッセージを取り出したとき、Expiry フィールドにはメッセージ保持時間の残りが示されます。

メッセージ保持時間が経過すると、キューマネージャによって破棄できる状態になります。破棄できる状態のメッセージは、検索 MQGET 命令や非検索 MQGET 命令が実行されると破棄されます。また、保持時間を経過したメッセージは、実行された命令が検索 MQGET 命令と非検索 MQGET 命令のどちらの場合も、返されません。例えば、MQGMO 構造体の MatchOptions フィールドの設定値が MQMO\_NONE の非検索 MQGET 命令で、FIFO 方式のキューから読み取りを実行するとします。この場合、終了していないメッセージが最初に出現するまでの、終了しているメッセージはすべて破棄されます。一方、優先順位方式のキューから同様の読み取りを実行した場合、終了していない最初のメッセージより先にキューに到着したメッセージのうち、優先順位が高いメッセージと優先順位が等しいメッセージが破棄されます。したがって、MQGET 命令が成功した場合の Expiry フィールドの値は、0 以上の数値または MQEI\_UNLIMITED です。

メッセージをリモートキューへ登録した場合、メッセージがあて先キューに到着する前に転送キューに格納されている状態で保持時間が終了することがあります。その後、そのメッセージは破棄されます。

メッセージの報告オプションで MQRO\_EXPIRATION を指定した場合、保持時間終了でメッセージが破棄されたときに報告が生成されます。これらのオプションを指定しなかった場合、報告は生成されません。通常、代替りのメッセージが送られるため、保持時間が経過したあとは、そのメッセージは無関連のものとみなされます。

保持時間終了でメッセージを破棄するプログラムは、報告が要求されている場合、報告メッセージを通知する必要があります。

## 注意事項

1. TP1/Message Queue では時間の減少を 1/10 秒単位の精度で実行しています。このフィールドに 1~9 の値を指定した場合、0 秒として扱います。このため、10 未満の値を指定した場合、または相手システムから受信したメッセージ保持時間が 10 未満の値である場合、保持時間が経過したメッセージとして扱われます。
2. 保持時間に 0 を指定して登録した場合、MQPUT 命令または MQPUT1 命令は理由コード MQRC\_EXPIRY\_ERROR で失敗します。この場合、報告メッセージは生成されません。
3. 保持時間が終了したメッセージはあとになるまで破棄されないため、それらのメッセージが取り出せない状態でキューに残ります。これらも、キューのメッセージ登録数として数えられます。この数え方は、トリガを起動する際のメッセージ登録数などのすべての場合に適用されます。
4. 例外報告を要求している場合、保持時間が終了したときではなく、メッセージが実際に破棄されたときに例外報告が生成されます。
5. 保持時間が終了したメッセージの破棄および例外報告の生成は、トランザクションの処理には含まれません。トランザクション内の MQGET 命令の結果でメッセージが破棄された場合でも含まれません。
6. トランザクション内の MQGET 命令で保持時間終了の際のメッセージを取り出したあとにトランザクションがロールバックした場合、再びメッセージを取り出す前に保持時間が終了することがあります。
7. 同様に、MQGMO\_LOCK オプションを指定した MQGET 命令で保持時間終了間際のメッセージを排他状態にした場合、MQGMO\_MSG\_UNDER\_CURSOR オプションを指定した MQGET 命令でメッセージを取り出す前に保持時間が終了することがあります。このとき、理由コード MQRC\_NO\_MSG\_UNDER\_CURSOR が MQGET 命令で返されます。
8. メッセージ保持時間がゼロよりも大きい問い合わせメッセージを取り出す場合、アプリケーションは応答メッセージを送信するときに次に示すどれかの処理を実行できます。
  - ・残りの保持時間を問い合わせメッセージから応答メッセージにコピーする。
  - ・応答メッセージ中の保持時間をゼロより大きい明示的な値に設定する。
  - ・応答メッセージ中の保持時間を MQEI\_UNLIMITED に設定する。実行される動作は、アプリケーションの設計に依存します。ただし、デッドレターキューにメッセージを登録するときのデフォルトでは、メッセージの残りの保持時間を保存し、減算を継続します。
9. トリガメッセージは、常に MQEI\_UNLIMITED オプションで生成されます。
10. Format フィールドの名称が MQFMT\_XMIT\_Q\_HEADER であるメッセージには、MQXQH 構造体に二つ目のメッセージ記述子があります。通常、これらのメッセージは転送キューに格

納されています。つまり、二つの Expiry フィールドがあります。この場合、次の点に注意してください。

#### 注 1

アプリケーションがリモートキューへメッセージを登録すると、キューマネージャによって転送キューへ格納されます。このとき、メッセージに MQXQH ヘッダが付けられ、アプリケーションが指定した同じ値が Expiry フィールドへ設定されます。Format の名称が MQFMT\_XMIT\_Q\_HEADER のメッセージをアプリケーションが直接登録する場合、アプリケーションは二つの Expiry フィールドに同じ値を指定する必要はありません。キューマネージャは、MQXQH 構造体の Expiry フィールドの値が正しいかどうかチェックしません。また、メッセージデータの長さが、MQXQH 構造体を含むのに十分かどうかもチェックしません。

#### 注 2

Format の名称が MQFMT\_XMIT\_Q\_HEADER のメッセージが、キュー（通常キューまたは転送キュー）から取り出されたとき、キューマネージャは、キューで経過した時間を両方の Expiry フィールドの値から減算します。メッセージデータの長さが MQXQH 構造体を含むのに十分ではない場合も、エラーは発生しません。

#### 注 3

メッセージを破棄できるかどうか判断するとき、キューマネージャによって「通常」のメッセージ記述子の Expiry フィールドが使用されます。MQXQH 構造体のメッセージ記述子ではないため、注意が必要です。

#### 注 4

二つの Expiry フィールドの初期値が異なった場合、MQXQH 構造体の Expiry フィールドの時間が経過しても、通常メッセージ記述子の Expiry フィールドの値が 0 以上のときはメッセージを取り出せます。この場合、MQXQH 構造体の Expiry フィールドには 0 が設定されます。

特殊な定数値として、次を指定できます。

### MQEI\_UNLIMITED

無制限

メッセージは、保持時間で制限されません。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQEI\_UNLIMITED です。

### ● Feedback (MQLONG 型) 報告メッセージ返答コード

このフィールドは、MQMT\_REPORT タイプのメッセージで使用するもので、報告の性質を示します。メッセージタイプが MQMT\_REPORT のときだけ有効です。

報告メッセージ返答コードは、次のように分けられます。

## MQFB\_NONE

返答コードなし

## MQFB\_SYSTEM\_FIRST

システムが生成する返答コードの最小値

## MQFB\_SYSTEM\_LAST

システムが生成する返答コードの最大値

MQFB\_SYSTEM\_FIRST から MQFB\_SYSTEM\_LAST までのシステムが生成する返答コードの範囲には、後述する特殊な返答コード (MQFB\_\*) が含まれます。また、メッセージをあて先キューへ登録できなかったときに発生する理由コード (MQRC\_\*) も含まれます。

## MQFB\_APPL\_FIRST

アプリケーションが生成できる返答コードの最小値

## MQFB\_APPL\_LAST

アプリケーションが生成できる返答コードの最大値

報告メッセージを生成するアプリケーションは、システムの範囲の返答コード (MQFB\_QUIT 以外) を使用できません。ただし、キューマネージャや MCA が生成する報告メッセージをシミュレートする場合は例外です。

MQPUT または MQPUT1 命令では、MQFB\_NONE、システムの範囲またはアプリケーションの範囲の値を使用してください。この値は、MsgType フィールドの値に関係なくチェックされます。

一般的な返答コード：

## MQFB\_COA

メッセージ到着確認

「Report フィールド」の MQRO\_COA オプションの説明を参照してください。

## MQFB\_COD

メッセージ配布確認

「Report フィールド」の MQRO\_COD オプションの説明を参照してください。

## MQFB\_EXPIRATION

メッセージ保持時間終了

保持時間が終了してもキューから削除されなかったため、メッセージは破棄されました。

## MQFB\_PAN

肯定動作通知

## MQFB\_NAN

否定動作通知

## MQFB\_QUIT

アプリケーション終了指示

このコードは、実行中のアプリケーションプログラムの数を制御するために、負荷スケジューリングプログラムで使用できます。この返答コードの MQMT\_REPORT タイプのメッセージが通知されると、そのアプリケーションに終了が指示されたことを意味します。ただし、この指示に従うかどうかはアプリケーションに任せられます。キューマネージャはそれを強要しません。

#### IMS ブリッジの返答コード：

IMS ブリッジが非ゼロの IMS-OTMA センスコードを受け取るとき、センスコードを 16 進から 10 進に変換し、値 MQFB\_IMS\_ERROR (300) を追加し、応答メッセージの Feedback フィールドに格納します。そのため、IMS-OTMA エラーが発生した場合は、返答コードの値は MQFB\_IMS\_FIRST (301) から MQFB\_IMS\_LAST (399) の範囲になります。

IMS ブリッジが生成する返答コードを次に示します。これらの値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。

- MQFB\_DATA\_LENGTH\_ZERO
- MQFB\_DATA\_LENGTH\_NEGATIVE
- MQFB\_DATA\_LENGTH\_TOO\_BIG
- MQFB\_BUFFER\_OVERFLOW
- MQFB\_LENGTH\_OFF\_BY\_ONE
- MQFB\_IIH\_ERROR
- MQFB\_NOT\_AUTHORIZED\_FOR\_IMS
- MQFB\_IMS\_ERROR
- MQFB\_IMS\_FIRST
- MQFB\_IMS\_LAST

#### CICS ブリッジの返答コード：

CICS ブリッジが生成する返答コードを次に示します。これらの値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。

- MQFB\_CICS\_APPL\_ABENDED
- MQFB\_CICS\_APPL\_NOT\_STARTED
- MQFB\_CICS\_BRIDGE\_FAILURE
- MQFB\_CICS\_CCSID\_ERROR
- MQFB\_CICS\_CIH\_ERROR
- MQFB\_CICS\_COMMAREA\_ERROR
- MQFB\_CICS\_CORREL\_ID\_ERROR
- MQFB\_CICS\_DLQ\_ERROR

- MQFB\_CICS\_ENCODING\_ERROR
- MQFB\_CICS\_INTERNAL\_ERROR
- MQFB\_CICS\_NOT\_AUTHORIZED
- MQFB\_CICS\_UOW\_BACKED\_OUT
- MQFB\_CICS\_UOW\_ERROR

理由コード：

例外報告の場合は、Feedback フィールドに次の理由コードが設定されます。

#### MQRC\_PUT\_INHIBITED

指定されたキューへの登録は禁止されています。

#### MQRC\_Q\_FULL

キューが満杯です。

#### MQRC\_NOT\_AUTHORIZED

アクセスの権限がありません。

#### MQRC\_Q\_SPACE\_NOT\_AVAILABLE

キューに対応するディスクに空き領域がありません。

#### MQRC\_PERSISTENT\_NOT\_ALLOWED

キューは永続メッセージをサポートしていません。

#### MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR

メッセージ長がリモートキューマネージャに許される値を超えています。

#### MQRC\_MSG\_TOO\_BIG\_FOR\_Q

メッセージ長がキューに定義された最大値を超えています。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQFB\_NONE です。

### ● Format (MQCHAR8 型) フォーマット名

メッセージデータの性質を示す名称です。メッセージの送信側が受信側に示します。この名称には、キューマネージャの文字セットの文字を使用します。ただし、次に示す文字だけが使用できます。

- 英大文字の A~Z
- アラビア数字の 0~9

このほかの文字を使用すると、送信側と受信側のキューマネージャの間で、名称を変換できないことがあります。

この名称の余りは、空白で埋めます。または、ヌル文字で終了します。ヌル文字以降の文字は、空白として扱われます。先頭または中間に空白を含んだ名称は指定しないでください。MQGET 命令では、キューマネージャによって、フィールドの余りが空白で埋められた名称が返されます。

使用文字の条件を満たしているかどうか、キューマネージャはチェックしません。

大文字、小文字および大文字小文字混合の「MQ」で始まる名称は、キューマネージャで定義されたことを意味します。ユーザ独自のフォーマットには、この文字で始まる名称を使用できません。キューマネージャのフォーマットは、次のとおりです。

## MQFMT\_NONE

フォーマット名なし。

データの性質は定義されていません。これは、メッセージがキューから取り出されるときに、MQGMO\_CONVERT オプションを使用してデータを変換できないことを意味します。

MQGET 命令に MQGMO\_CONVERT が指定され、メッセージ中のデータの文字セットまたはマシンコードが MsgDesc 引数に指定されたものと異なる場合、メッセージが次に示す完了コードと理由コードとともに返されます（その他のエラーはないとします）。

- メッセージの先頭に MQFMT\_NONE データがある場合は、完了コード MQCC\_WARNING と理由コード MQRC\_FORMAT\_ERROR が返されます。
- MQFMT\_NONE データがメッセージの末尾にある場合（つまり、一つ以上の MQ ヘッダ構造体が先行している場合）、完了コード MQCC\_OK と理由コード MQRC\_NONE が返されます。この場合、MQ ヘッダ構造体は、要求された文字セットとマシンコードに変換されます。

C 言語では、MQFMT\_NONE\_ARRAY も定義されています。これは、MQFMT\_NONE と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_ADMIN

コマンドサーバの問い合わせ／応答メッセージ

プログラマブルコマンドフォーマット (PCF) の問い合わせまたは応答メッセージです。

C 言語では、MQFMT\_ADMIN\_ARRAY も定義されています。これは、MQFMT\_ADMIN と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_DEAD\_LETTER\_HEADER

デッドレターヘッダ

デッドレターヘッダ (MQDLH 構造体) で始まるメッセージデータです。MQDLH 構造体の後ろに、登録元メッセージのデータが続きます。登録元メッセージデータのフォーマット名は、MQDLH 構造体の Format フィールドに指定されています。MQDLH 構造体の詳細については、この章の「[MQDLH 構造体 - デッドレターヘッダ](#)」を参照してください。

Format フィールドが MQFMT\_DEAD\_LETTER\_HEADER のメッセージには、到着確認および配布確認報告は生成されません。

C 言語では、MQFMT\_DEAD\_LETTER\_HEADER\_ARRAY も定義されています。これは、MQFMT\_DEAD\_LETTER\_HEADER と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_DIST\_HEADER

### 配布リストヘッダ

メッセージデータは配布リストヘッダ (MQDH 構造体) で始まり、MQOR 構造体および MQPMR 構造体レコードの配列を含んでいます。そのあとに、配布リストヘッダに追加する形式でデータが続きます。追加されたデータがある場合、そのフォーマットは MQDH 構造体の Format フィールドによって与えられます。MQDH 構造体の詳細については、この章の「[MQDH 構造体 – 配布ヘッダ](#)」を参照してください。MQGMO\_CONVERT オプションが MQGET 命令に指定されると、フォーマットのメッセージを変換できます。

C 言語では、MQFMT\_DIST\_HEADER\_ARRAY も定義されています。これは、MQFMT\_DIST\_HEADER と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_EVENT

### イベントメッセージ

TP1/Message Queue では意味がありません。

このオプションを指定しても、TP1/Message Queue の動作には影響しません。これは、アプリケーションの移植性を向上させるために定義されています。

発生したイベントを報告するメッセージキューイング機能のイベントメッセージです。イベントメッセージの構造は、プログラマブルコマンドの場合と同じです。

C 言語では、MQFMT\_EVENT\_ARRAY も定義されています。これは、MQFMT\_EVENT と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_MD\_EXTENSION

### メッセージ記述子の拡張

メッセージデータはメッセージ記述拡張子で始まり、ほかのデータ (通常はアプリケーションメッセージデータ) が続きます。MQMDE 構造体に続くデータのフォーマット名、文字セットおよびマシンコード形式は、MQMDE 構造体の Format フィールド、CodedCharSetId フィールド、および Encoding フィールドによって与えられます。MQMDE 構造体の詳細については、この章の「[MQMDE 構造体 – メッセージ記述子の拡張](#)」を参照してください。このフォーマットのメッセージは、MQGET 命令に MQGMO\_CONVERT オプションを指定すると変換できます。

## MQFMT\_PCF

### プログラマブルコマンドフォーマット (PCF) のユーザ定義メッセージ

プログラマブルコマンドフォーマット (PCF) メッセージの構造に従ったユーザ定義のメッセージです。MQGET 命令で MQGMO\_CONVERT オプションが指定されると、このフォーマットのメッセージは変換されます。

C 言語では、MQFMT\_PCF\_ARRAY も定義されています。これは、MQFMT\_PCF と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_REF\_MSG\_HEADER

### 参照メッセージヘッダ

メッセージデータは参照メッセージヘッダ (MQRMH 構造体) で始まり、ほかのデータが続く場合もあります。MQRMH 構造体に続くデータのフォーマット名、文字セットおよびマシンコード形式は、

MQMDE 構造体の Format フィールド、CodedCharSetId フィールド、および Encoding フィールドによって与えられます。MQMDE 構造体の詳細については、この章の「[MQMDE 構造体 - メッセージ記述子の拡張](#)」を参照してください。このフォーマットのメッセージは、MQGET 命令で MQGMO\_CONVERT オプションを指定すると変換できます。

C 言語では、定数 MQFMT\_REF\_MSG\_HEADER\_ARRAY も定義されています。これは、MQFMT\_REF\_MSG\_HEADER と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_STRING

文字だけのメッセージ

MQGET 命令で MQGMO\_CONVERT オプションが指定されると、このフォーマットのメッセージは変換されます。アプリケーションメッセージデータは、SBCS 文字列（1 バイト文字セット）でも DBCS 文字列（2 バイト文字セット）でもかまいません。

C 言語では、MQFMT\_STRING\_ARRAY も定義されています。これは、MQFMT\_STRING と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_TRIGGER

トリガメッセージ

MQTM 構造体で記述されるトリガメッセージです。MQTM 構造体の詳細については、この章の「[MQTM 構造体 - トリガメッセージ](#)」を参照してください。

C 言語では、MQFMT\_TRIGGER\_ARRAY も定義されています。これは、MQFMT\_TRIGGER と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

## MQFMT\_XMIT\_Q\_HEADER

転送キューヘッダ

転送キューヘッダ（MQXQH 構造体）で始まるメッセージデータです。MQXQH 構造体の後ろに、登録元メッセージのデータが続きます。登録元データのフォーマット名は、転送キューヘッダ MQXQH 構造体の中の MQMD 構造体の Format フィールドに指定されています。MQXQH 構造体の詳細については、この章の「[MQXQH 構造体 - 転送キューヘッダ](#)」を参照してください。

COA 報告と COD 報告は、MQFMT\_XMIT\_Q\_HEADER の Format フィールドを持つメッセージに対しては生成されません。

C 言語では、MQFMT\_XMIT\_Q\_HEADER\_ARRAY も定義されています。これは、MQFMT\_XMIT\_Q\_HEADER と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの長さは、MQ\_FORMAT\_LENGTH に定義されています。

このフィールドの初期値は MQFMT\_NONE です。

## ● GroupId (MQBYTE24 型) グループ識別子

物理メッセージが属する特定のメッセージグループまたは論理メッセージを識別するのに使用するバイト列です。GroupId フィールドは、セグメント分割が許可されているときにも使用されます。ただし、すべてのケースで GroupId フィールドはヌル以外の値であり、次に示すフラグの一つ以上が MsgFlags フィールドに指定されていなければなりません。

- MQMF\_MSG\_IN\_GROUP
- MQMF\_LAST\_MSG\_IN\_GROUP
- MQMF\_SEGMENT
- MQMF\_LAST\_SEGMENT
- MQMF\_SEGMENTATION\_ALLOWED

これらのフラグが一つも指定されない場合、GroupId フィールドには特別なヌルの値である MQGI\_NONE が設定されます。

次の場合、アプリケーションによって MQPUT 命令または MQGET 命令に GroupId フィールドが設定される必要はありません。

- MQPUT 命令で MQPMO\_LOGICAL\_ORDER が指定された場合
- MQGET 命令で MQMO\_MATCH\_GROUP\_ID が指定されない場合

報告メッセージではないメッセージに対して、MQPUT 命令または MQGET 命令を使用するときには、上記のように命令を使用することをお勧めします。ただし、アプリケーションがさらに制御を必要としたり、MQPUT1 命令を使用したりする場合、アプリケーションは GroupId フィールドに適切な値が設定されていることを保証しなければなりません。

メッセージグループとセグメントはグループ識別子がユニークである場合だけ、正しく処理できます。このため、次に示す場合を除いては、アプリケーションによって自身のグループ識別子を生成しないでください。

- MQPMO\_LOGICAL\_ORDER を指定すると、キューマネージャが自動的にグループの最初のメッセージまたは論理メッセージのセグメントにユニークなグループ識別子を生成します。その後、グループの残りのメッセージや論理メッセージのセグメントに対してそのグループ識別子を使用するので、アプリケーションが特別な処理をする必要がなくなります。MQPMO\_LOGICAL\_ORDER を指定できる場合は、指定することをお勧めします。
- MQPMO\_LOGICAL\_ORDER を指定しない場合は、アプリケーションはグループメッセージまたは論理メッセージのセグメントに対する最初の MQPUT 命令または MQPUT1 命令で、GroupId フィールドに MQGI\_NONE を指定して、キューマネージャにグループ識別子の生成を要求しなければなりません。出力値としてキューマネージャから返却されるグループ識別子は、グループの残りのメッセージや論理メッセージのセグメントに使用できます。メッセージグループにセグメント分割されたメッセージが含まれていれば、同一のグループ識別子がグループ内のすべてのセグメントとメッセージに使用されなければなりません。

MQPMO\_LOGICAL\_ORDER が指定されない場合、グループ内のメッセージや論理メッセージのセグメントはどんな順序（例：逆順）でも登録できますが、グループ識別子はそれらのメッセージに対して発行される最初の MQPUT 命令または MQPUT1 命令で割り当てられなければなりません。

MQPUT 命令または MQPUT1 命令で、キューマネージャは MQPMO 構造体の MQPMO\_LOGICAL\_ORDER の表中に示される値を使用します。MQPUT 命令または MQPUT1 命令で値を返す場合、オープンされているオブジェクトが単一のキューであり、配布リストではないときに、キューマネージャは、このフィールドにメッセージとともに送信される値を設定します。一方、オープンされているオブジェクトが配布リストのときは、フィールドの値は変更されません。配布リストがオープンされている場合、アプリケーションが生成されたグループ識別子を知る必要があるときは、GroupId フィールドを含む MQPMR 構造体を指定しなければなりません。

MQGET 命令での入力について、キューマネージャは MQGMO 構造体の MQGMO\_LOGICAL\_ORDER の表中に示される値を使用します。MQGET 命令の出力について、キューマネージャはこのフィールドに取り出されたメッセージの値を設定します。

次の特別な値が定義されています。

## MQGI\_NONE

グループ識別子を指定しません。

この定数の値は 2 進数の 0 です。

グループに属さない、かつ論理メッセージのセグメントではなく、かつセグメント分割もできないメッセージに使用されます。

C 言語では、MQGI\_NONE\_ARRAY も定義されています。これは MQGI\_NONE と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの長さは、MQ\_GROUP\_ID\_LENGTH に定義されています。

このフィールドの初期値は MQGI\_NONE です。

このフィールドは、Version フィールドの値が MQMD\_VERSION\_2 以降の場合だけ有効です。

## ● MsgFlags (MQLONG 型) メッセージフラグ

メッセージの属性を指定したり、処理を制御したりするフラグです。次のフラグがあります。

- セグメンテーションフラグ
- ステータスフラグ
- デフォルトフラグ

### セグメンテーションフラグ：

セグメンテーションフラグは、セグメント分割の際に使用されます。通常、メッセージがキューに対して大き過ぎる場合、キューへのメッセージ登録は失敗します。セグメント分割は、キューマネージャやアプリケーションがメッセージをセグメントと呼ばれる小さな部分に分割し、それぞれのセグメントを分離され

た物理メッセージとしてキューに配置する方法です。メッセージを取り出すアプリケーションは一つ一つセグメントを取り出すか、キューマネージャにセグメントを MQGET 命令で返される単一のメッセージに組み立てさせることができます。メッセージを組み立てる場合は、MQGET 命令で MQGMO\_COMPLETE\_MSG オプションを指定し、完全なメッセージを格納するのに十分なバッファを用意します。MQGMO\_COMPLETE\_MSG オプションの詳細は、この章の「MQGMO 構造体 – メッセージ取り出しオプション」を参照してください。メッセージのセグメント分割は、送信キューマネージャ、中間のキューマネージャまたはあて先キューマネージャで発生する可能性があります。

メッセージのセグメント分割を制御するために、次の値の一つを指定できます。

## MQMF\_SEGMENTATION\_INHIBITED

### セグメント分割の禁止

キューマネージャがメッセージをセグメント分割しないようにします。すでにセグメント分割されているメッセージが指定された場合、さらに小さなセグメントに分割することを防止できます。このフラグの初期値は 2 進数の 0 です。

## MQMF\_SEGMENTATION\_ALLOWED

### セグメント分割の許可

キューマネージャがメッセージをセグメント分割することを指定します。すでにセグメント分割されているメッセージが指定された場合、さらに小さなセグメントに分割することを許可します。

MQMF\_SEGMENTATION\_ALLOWED は MQMF\_SEGMENT または MQMF\_LAST\_SEGMENT の指定なしで指定できます。

キューマネージャは、必要に応じてセグメント、つまり、必要となるヘッダデータを加算したものがキューに当てはまるように、メッセージをセグメント分割します。キューマネージャは次のような処理をします。

- ユーザ定義フォーマットを 16 バイトの倍数の境界で分割します。
- MQFMT\_STRING 以外の組み込みフォーマットは、データの属性に適応した個所で分割します。ただし、キューマネージャは、メッセージを MQ ヘッダ構造体の中間で分割しません。なお、2 番目以降のキューマネージャによって生成されたセグメントは、次のうちの一つでだけ始まります。

MQ ヘッダ構造体

アプリケーションメッセージデータの開始

アプリケーションメッセージデータの一部

- MQFMT\_STRING では、データの属性 (SBCS, DBCS, SBCS/DBCS) が混在していても、それに関係なく分割されます。文字列に DBCS または SBCS/DBCS が混在している場合、セグメントがある文字セットから別の文字セットに変換できません。
- キューマネージャによって、各セグメントの MQMD 構造体の Format フィールド、CodedCharSetId フィールド、および Encoding フィールドは、セグメントの先頭からそのデータを正確に表すように設定されます。フォーマット名は組み込みフォーマットまたはユーザ定義フォーマットです。
- 0 より大きい Offset フィールドを持つセグメントの MQMD 構造体の Report フィールドは次のように調整されます。

それぞれの報告タイプについて、報告オプションが MQRO\*\_WITH\_DATA であるが、セグメントがユーザデータ、つまり、MQ ヘッダ構造体続くデータの最初の 100 バイトをまったく含むことができない場合、報告オプションは MQRO\_\*に変更されます。

キューマネージャは上記の規則に従いますが、それ以外は適当な単位でメッセージを分割します。キューマネージャが特定のメッセージの分割を選択する方法については、何も仮定されません。

TP1/Message Queue では、キューマネージャは、次に示すようにトランザクションの中でだけセグメント分割できます。

- MQPUT 命令または MQPUT1 命令がユーザ定義のトランザクションで実行される場合、そのトランザクションが使用されます。命令がセグメント分割の処理中に失敗した場合、キューマネージャはその呼び出しによって、キューに入れられたセグメントを削除します。命令がセグメント分割の処理中に失敗しても、トランザクションは正常にコミットされます。
- 命令がトランザクション外で実行された場合、キューマネージャはセグメント分割の処理をしません。ただし、メッセージのセグメント分割が必要な場合、MQRC\_UOW\_NOT\_AVAILABLE の理由コードで失敗します。

セグメント分割されたメッセージのデータ変換に関しては、次に示す考慮が必要です。

- 送信側での変換が、MCA に指定されたとき、MCA は論理メッセージのセグメントではないメッセージだけを変換します。MCA はセグメント分割されたメッセージの変換をすることはありません。

セグメンテーションフラグは MQPUT 命令および MQPUT1 命令の入力フラグです。また、MQGET 命令の出力フラグです。MQGET 命令では、キューマネージャがフラグの値を MQGMO 構造体の Segmentation フィールドにも反映します。

このフラグの初期値は MQMF\_SEGMENTATION\_INHIBITED です。

## ステータスフラグ：

ステータスフラグは、物理メッセージがメッセージグループに属するか、論理メッセージのセグメントか、その両方か、またはどちらでもないのかを示すフラグです。次に示す一つ以上の値を、MQPUT 命令または MQPUT1 命令で指定し、MQGET 命令で返すことができます。

### MQMF\_MSG\_IN\_GROUP

メッセージはグループのメンバです。

### MQMF\_LAST\_MSG\_IN\_GROUP

メッセージはグループの最終論理メッセージです。

このフラグが設定されると、キューマネージャはメッセージとともに送信される MQMD 構造体のコピーに、MQMF\_MSG\_IN\_GROUP を設定します。ただし、アプリケーションが MQPUT 命令または MQPUT1 命令で作成した MQMD 構造体にあるこれらのフラグの設定は変更しません。

グループがただ一つの論理メッセージで構成されることがあります。この場合、MQMF\_LAST\_MSG\_IN\_GROUP が設定されますが、MsgSeqNumber フィールドの値は 1 となります。

### MQMF\_SEGMENT

メッセージは論理メッセージのセグメントです。

MQMF\_SEGMENT が MQMF\_LAST\_SEGMENT なしで指定された場合、MQ ヘッダ構造体が存在するときはその長さを除いたセグメントのアプリケーションメッセージデータ長は、小さくても 1 でなければなりません。長さが 0 の場合、MQPUT 命令または MQPUT1 命令は MQRC\_SEGMENT\_LENGTH\_ZERO となり失敗します。

## MQMF\_LAST\_SEGMENT

メッセージは論理メッセージの最終セグメントです。

このフラグが設定されると、キューマネージャは、メッセージとともに送信される MQMD 構造体の複写に、MQMF\_SEGMENT を設定します。ただし、アプリケーションが MQPUT 命令または MQPUT1 命令で作成した MQMD 構造体のフラグの設定は変更しません。

論理メッセージがただ一つのセグメントで構成されることがあります。この場合、MQMF\_SEGMENT が設定されますが、Offset フィールドの値は 0 となります。MQMF\_LAST\_SEGMENT が指定された場合、MQ ヘッダ構造体が存在するときはその長さを除いたセグメントのアプリケーションメッセージのデータ長が 0 であることもあります。

アプリケーションは、メッセージ登録時に、これらのフラグが正しく設定されることを保証しなければなりません。キューハンドルに対して、MQPMO\_LOGICAL\_ORDER を指定する、または以前の MQPUT 命令で指定していた場合、フラグの設定はそのキューハンドルに対してキューマネージャの保持しているグループとセグメントの情報と一致していなければなりません。

MQPMO\_LOGICAL\_ORDER が指定されたとき、キューハンドルに対して成功する MQPUT 命令には、次の条件が適用されます。

- 現在のグループ、および現在の論理メッセージがない場合、フラグのすべてとその組み合わせが正しくなければなりません。
- いったん MQMF\_MSG\_IN\_GROUP を指定すると、MQMF\_LAST\_MSG\_IN\_GROUP を指定するまで MQMF\_MSG\_IN\_GROUP を指定しなければなりません。この条件が満たされない場合、MQRC\_INCOMPLETE\_GROUP の理由コードで失敗します。
- いったん MQMF\_SEGMENT を指定すると、MQMF\_LAST\_SEGMENT を指定するまで MQMF\_SEGMENT を指定しなければなりません。この条件が満たされない場合、MQRC\_INCOMPLETE\_MSG の理由コードで失敗します。
- いったん MQMF\_SEGMENT を MQMF\_MSG\_IN\_GROUP なしに指定すると、MQMF\_LAST\_SEGMENT を指定するまで MQMF\_MSG\_IN\_GROUP を指定しないでください。この条件が満たされない場合、MQRC\_INCOMPLETE\_MSG の理由コードで失敗します。

フラグの正しい組み合わせとさまざまなフィールドに使用される値については、この章の「[MQPMO 構造体 - メッセージ登録オプション](#)」で、Options フィールドの MQPMO\_LOGICAL\_ORDER の表で示しています。

このフラグは MQPUT 命令および MQPUT1 命令の入力フラグです。また、MQGET 命令の出力フラグです。MQGET 命令では、キューマネージャがフラグの値を MQGMO 構造体の GroupStatus フィールドおよび SegmentStatus フィールドにも反映します。

## デフォルトフラグ：

デフォルトフラグはメッセージのデフォルトの属性を示すために使用します。次の値は、メッセージがデフォルトの属性を持つことを示すために指定します。

## MQMF\_NONE

メッセージフラグなし（メッセージ属性の初期値）。

セグメント分割が禁止されている、かつメッセージがグループにも属さない、かつ論理メッセージのセグメントでもないことを示しています。MQMF\_NONE はプログラムの文書化の目的で定義されています。値としては 0 が定義されていますが、プログラムを文書化する以外の目的はありません。

MsgFlags フィールドはサブフィールドに分割されています。詳細は「付録 D 報告オプションとメッセージフラグ」を参照してください。

このフィールドの初期値は MQMF\_NONE です。

このフィールドは、Version フィールドの値が MQMD\_VERSION\_2 以降の場合だけ有効です。

## ● MsgId (MQBYTE24 型) メッセージ識別子

MQGET 命令では、取り出すメッセージのメッセージ識別子を指定します。通常、MQGET 命令に指定したメッセージ記述子の、MsgId フィールドと CorrelId フィールドに該当する最初のメッセージが、キューマネージャによって返されます。ただし、特殊な定数値として、MQMI\_NONE および MQCI\_NONE を使用できます。

- MQGET 命令のメッセージ識別子として MQMI\_NONE を指定すると、どんなメッセージ識別子でも該当します。
- MQGET 命令の相関識別子として MQCI\_NONE を指定すると、どんな相関識別子でも該当します。

なお、MQGET 命令では、キューから特定のメッセージを選択して取り出すために、次に示すフィールドを使用できます。

- MsgId
- CorrelId
- GroupId
- MsgSeqNumber
- Offset

MsgId フィールドは、これらのフィールドの一つとして使用されます。

通常、MQGET 命令は、キューにある次のメッセージを取り出します。特定のメッセージが必要な場合には、選択基準として一つまたは複数のフィールドを任意の組み合わせで指定すれば、目的のメッセージを取得します。

したがって、アプリケーションでは、一つ以上のフィールドを必要な値に設定した上で、MQGMO 構造体の MatchOptions フィールドに対応する MQMO\_\* 一致オプションを設定します。これによって、フィールドを選択基準として使用できるようになり、フィールドに指定した値を持つメッセージだけが取り出し

の対象になります。アプリケーションで MQMO\_\*一致オプションを設定しない場合は、MatchOptions フィールドのデフォルトでは、メッセージ識別子および相関識別子の二つが選択規準として使用されます。

## 注意事項

選択規準に該当するメッセージは、キュー内で順次に検索されます。このため、選択規準がないときよりも、取り出しに時間が掛かります。特に、該当するメッセージを発見するまでに多くのメッセージを検索するほど、時間が掛かります。

MQGET 命令の GetMsgOpts 引数で MQGMO\_MSG\_UNDER\_CURSOR オプションを指定した場合、このフィールドは無視されます。

MQGET 命令が返される時、取り出したメッセージのメッセージ識別子が、MsgId フィールドに設定されます。

MQPUT 命令および MQPUT1 命令で MQMI\_NONE オプションを指定した場合、登録したメッセージのメッセージ記述子に、ユニークなメッセージ識別子\*がキューマネージャによって設定されます。設定されたメッセージ識別子は、それを送信したアプリケーションにも、メッセージ記述子の一部として返されます。アプリケーションは、特定のメッセージに関する情報を記録するために、この値を使用できます。また、アプリケーションのほかの部分からの問い合わせに対する応答にも使用できます。

### 注※

キューマネージャによって生成されるメッセージ識別子 (MsgId) は、4 バイトのプロダクト識別子と、それに続くプロダクト固有の文字列で構成されます。この MsgId は、EBCDIC コードまたは ASCII コードの A~I の範囲の文字で始まっています。アプリケーションで MsgId を指定する場合は、キューマネージャによって生成される値との重複を避けるために、A~I 以外の文字で始まる文字列を指定してください。

メッセージが配布リストに登録された場合、キューマネージャは要求されたとおりにユニークなメッセージ識別子を生成します。ただし、MQMI\_NONE が指定されている場合でも、MQMD 構造体の MsgId フィールドの値は返される時にも変更されません。アプリケーションがキューマネージャの生成したメッセージ識別子を知る必要がある場合、アプリケーションは MsgId フィールドを含む MQPMR レコードを提供しなければなりません。

送信するアプリケーションは、MQMI\_NONE 以外の定数値も、メッセージ識別子として指定できます。ただし、MQMI\_NONE 以外の値では、キューマネージャによって、ユニークなメッセージ識別子は生成されません。メッセージを転送するアプリケーションが、登録元メッセージのメッセージ識別子を引き渡すために、この機能を使用できます。

キューマネージャは、次の場合を除いて、このフィールドを使用しません。

- 上記に示したように、要求に応じてユニークな値を生成する場合
- メッセージを取り出したアプリケーションに値を返す場合

- 報告オプションに依存して、このメッセージに関連した報告を生成し、かつ報告メッセージの CorrelId フィールドに値を複写する場合

キューマネージャまたは MCA によって報告が生成されるとき、登録元メッセージの Report フィールド (MQRO\_NEW\_MSG\_ID または MQRO\_PASS\_MSG\_ID) に従って MsgId が設定されます。アプリケーションが報告を生成するときも、同様の処理をする必要があります。

MQPER\_PERSISTENT のメッセージでは、キューマネージャが再開しても、メッセージ記述子は失われません。

このフィールドは、キューマネージャの文字セットの変換対象にはなりません。このフィールドは、文字列ではなく、ビット列として扱われます。

次の定数値を使用できます。

## MQMI\_NONE

メッセージ識別子を指定しません。

この定数の値は、2 進数の 0 です。

C 言語では、MQMI\_NONE\_ARRAY も定義されています。これは、MQMI\_NONE と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドは、MQGET 命令、MQPUT 命令、および MQPUT1 命令の入出力用のフィールドです。

このフィールドの長さは、MQ\_MSG\_ID\_LENGTH に定義されています。

このフィールドの初期値は MQMI\_NONE です。

## ● MsgSeqNumber (MQLONG 型) メッセージシーケンス番号

グループ内の論理メッセージのシーケンス番号を表すフィールドです。シーケンス番号は 1 から始まり、グループ内の新しい論理メッセージごとに 1 ずつ加算されます。最大値は 999999999 です。グループに属さない物理メッセージのシーケンス番号は 1 です。次の場合、MsgSeqNumber フィールドは、MQPUT 命令または MQPUT1 命令でアプリケーションによって設定される必要はありません。

- MQPMO\_LOGICAL\_ORDER が MQPUT 命令で指定される場合
- MQMO\_MATCH\_MSG\_SEQ\_NUMBER が MQGET 命令で指定されない場合

報告メッセージではないメッセージに対して、MQPUT 命令または MQGET 命令を使用するときには、上記のように命令を使用することをお勧めします。ただし、アプリケーションがさらに制御を必要としたり、MQPUT1 命令を使用したりする場合、アプリケーションはシーケンス番号が適切な値に設定してあることを保証しなければなりません。

MQPUT 命令または MQPUT1 命令で、キューマネージャは MQPMO 構造体の MQPMO\_LOGICAL\_ORDER の表中に示される値を使用します。MQPUT 命令または MQPUT1 命令で値を返す場合、キューマネージャは、このフィールドにメッセージとともに送信される値を設定します。

MQGET 命令での入力について、キューマネージャは MQGMO 構造体の MQGMO\_LOGICAL\_ORDER の表中に示される値を使用します。MQGET 命令の出力について、キューマネージャはこのフィールドに取り出されたメッセージの値を設定します。

このフィールドの初期値は 1 です。

このフィールドは、Version フィールドの値が MQMD\_VERSION\_2 以降の場合だけ有効です。

## ● MsgType (MQLONG 型) メッセージタイプ

メッセージの種類を示します。メッセージタイプは次のように分けられます。

### MQMT\_SYSTEM\_FIRST

システムが定義したメッセージタイプの最小値

### MQMT\_SYSTEM\_LAST

システムが定義したメッセージタイプの最大値

次の値は、システムの範囲に定義されています。

### MQMT\_DATAGRAM

非問い合わせメッセージ

応答を要求しないメッセージです。

### MQMT\_REQUEST

問い合わせメッセージ

応答を要求するメッセージです。

応答を受け取りたいキューの名称を、ReplyToQ フィールドに指定する必要があります。応答メッセージの Report フィールドには、応答メッセージの MsgId と CorrelId の設定方法が示されます。

### MQMT\_REPLY

直前の問い合わせメッセージに対する応答メッセージ

直前の問い合わせメッセージ (MQMT\_REQUEST) へ応答するメッセージです。このメッセージは、問い合わせメッセージの ReplyToQ フィールドのキューへ登録します。問い合わせメッセージの Report フィールドには、応答メッセージの MsgId と CorrelId の設定方法が示されます。

#### ■ 注意事項

キューマネージャは、問い合わせに対する応答の有無について関知しません。アプリケーションの責任で行ってください。

### MQMT\_REPORT

報告メッセージ

各種の事象を報告するメッセージです。通常は、ほかのメッセージに関連して生成します。例えば、不正なデータを含んだ問い合わせメッセージを受信した場合などです。このメッセージは、登録元メッセージのメッセージ記述子の ReplyToQ フィールドのキューへ登録します。Feedback フィールドに

は、報告の性質を示します。登録元メッセージの Report フィールドには、報告メッセージの MsgId と CorrelId の設定方法が示されます。

キューマネージャまたは MCA によって生成される報告メッセージは、Feedback フィールドと CorrelId フィールドに設定された値が使用されて、常に ReplyToQ のキューへ登録されます。

システムの範囲で上記以外の値は、MQI の次バージョン以降に定義されることがあります。それらの値は、MQPUT 命令または MQPUT1 命令で、エラーが発生しないで受け付けられます。

また、アプリケーションで定義した値も使用できます。それらの値は、次の範囲で定義してください。

#### MQMT\_APPL\_FIRST

アプリケーションが定義できるメッセージタイプの最小値

#### MQMT\_APPL\_LAST

アプリケーションが定義できるメッセージタイプの最大値

MQPUT 命令および MQPUT1 命令の MsgType フィールドには、システムの範囲かアプリケーションの範囲の値を使用してください。使用しない場合、理由コード MQRC\_MSG\_TYPE\_ERROR で命令は失敗します。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQMT\_DATAGRAM です。

### ● Offset (MQLONG 型) オフセット

論理メッセージのデータ部の先頭からの物理メッセージデータのオフセットです。このデータをセグメントと呼びます。オフセットは 0~999999999 の範囲です。論理メッセージのセグメントではない物理メッセージは、0 のオフセットを持ちます。

次の場合、Offset フィールドは、MQPUT 命令または MQGET 命令でアプリケーションによって設定される必要はありません。

- MQPMO\_LOGICAL\_ORDER が MQPUT 命令で指定される場合
- MQMO\_MATCH\_OFFSET が MQGET 命令で指定されない場合

報告メッセージではないメッセージに対して、MQPUT 命令または MQGET 命令を使用するときには、上記のように命令を使用することをお勧めします。ただし、アプリケーションが上記の条件に合わなかったり、MQPUT1 命令を使用したりする場合、アプリケーションはオフセットが適切な値に設定してあることを保証しなければなりません。

MQPUT 命令または MQPUT1 命令で、キューマネージャは MQPMO 構造体の MQPMO\_LOGICAL\_ORDER の表中で示される値を使用します。MQPUT 命令または MQPUT1 命令で値を返す場合、キューマネージャは、このフィールドにメッセージとともに送信される値を設定します。

論理メッセージのセグメントに対する報告メッセージでは、MQOL\_UNDEFINED ではない場合に生成される OriginalLength フィールドは、キューマネージャによって保持されているセグメント情報に関するオフセット値の更新に使用されます。

MQGET 命令での入力について、キューマネージャは MQGMO 構造体の MQGMO\_LOGICAL\_ORDER の表中で示される値を使用します。MQGET 命令の出力で、キューマネージャはこのフィールドに取り出されたメッセージの値を設定します。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQMD\_VERSION\_2 以降の場合だけ有効です。

## ● OriginalLength (MQLONG 型) 登録元メッセージ長

このフィールドはセグメントである報告メッセージにだけ関連します。報告メッセージが関連するメッセージセグメントの長さを指定します。セグメントによって構成されている論理メッセージの長さや、報告メッセージ中のデータの長さは指定しません。

### 注意事項

セグメントであるメッセージの報告メッセージを生成するときは、キューマネージャおよびメッセージチャンネルエージェントが、元のメッセージから GroupId, MsgSeqNumber, Offset および MsgFlags フィールドを報告メッセージの MQMD 構造体にコピーします。そのため、その報告メッセージもセグメントです。報告メッセージを生成するアプリケーションでは、同じことを実行し、OriginalLength フィールドを正しく設定することをお勧めします。

次に示す特別な値があります。

### MQOL\_UNDEFINED

メッセージの元の長さが定義されていません。

このフィールドは、MQPUT 命令や MQPUT1 命令の入力用フィールドです。ただし、アプリケーションが指定する値は次に示す特定の場合にだけ受け入れられます。

- 登録されたメッセージがセグメントであり、報告メッセージでもある場合、キューマネージャはこのフィールドに指定された値を受け取ります。ただし、その値は次の条件を満たさなければなりません。
  - セグメントが最終セグメントではない場合、0 より大きい
  - セグメントが最終セグメントの場合、0 以上
  - メッセージのデータ長以上である

上記の条件が満たされない場合、MQRC\_ORIGINAL\_LENGTH\_ERROR の理由コードで失敗します。

- 登録されるメッセージがセグメントであり、報告メッセージではない場合、キューマネージャはこのフィールドを無視して、アプリケーションメッセージ長を使用します。
- そのほかの場合、キューマネージャはこのフィールドを無視して MQOL\_UNDEFINED を使用します。

OriginalLength フィールドは、MQGET 命令の出力用フィールドです。

このフィールドの初期値は MQOL\_UNDEFINED です。

このフィールドは、Version フィールドの値が MQMD\_VERSION\_2 以降の場合だけ有効です。

## ● Persistence (MQLONG 型) メッセージ永続性

このフィールドは、システム障害後とキューマネージャを再度開始したあとにメッセージが残るかどうかを示します。MQPUT 命令および MQPUT1 命令では、次のどれかの値を指定します。

### MQPER\_PERSISTENT

永続メッセージにします。

メッセージはシステム障害後とキューマネージャを再度開始したあとも残ります。いったんメッセージが登録され、登録側のトランザクションがコミットされると（メッセージがトランザクションの一部として登録される場合）、そのメッセージは補助記憶装置に保存されます。メッセージがキューから取り出され、取り出し側のトランザクションがコミットされる（メッセージがトランザクションの一部として取り出される場合）まで残ります。

永続メッセージをリモートキューへ送信するときは、格納・転送機構が使用されて、次のキューマネージャへ到着するまでメッセージが保持されます。

永続メッセージは、一時動的キューに登録できません。永続的動的キューおよび定義済みキューに登録できます。

### MQPER\_NOT\_PERSISTENT

非永続メッセージにします。

通常、メッセージはシステム障害後やキューマネージャを再度開始したあとに残りません。キューマネージャを再度開始するときにメッセージの完全なコピーが補助記憶装置で見つかった場合も適用されます。

### MQPER\_PERSISTENCE\_AS\_Q\_DEF

省略時のメッセージ永続性を使用します。

- クラスタキューの場合、MQOPEN 命令で指定する MQOD 構造体の ObjectQMgrName フィールドの値によって、使用される永続性が異なります。

メッセージがシステム転送キューに登録されるときに、DefPersistence 属性の値が Persistence フィールドにコピーされます。その後、DefPersistence 属性が変更されても、キューに登録済みのメッセージは影響を受けません。

#### MQOD 構造体の ObjectQMgrName フィールドに空白を指定した場合

あて先キュー（クラスタキュー）に定義された DefPersistence 属性の値が、メッセージ永続性として使用されます。

ただし、MQOD 構造体の ObjectName フィールドにリモートキューのローカル定義を指定して、あて先キュー（クラスタキュー）へ名称解決される場合は、システム転送キューに定義された DefPersistence 属性の値が、メッセージ永続性として使用されます。

#### MQOD 構造体の ObjectQMgrName フィールドにあて先キューマネージャを指定した場合

システム転送キューに定義された DefPersistence 属性の値が、メッセージ永続性として使用されます。

- クラスタキュー以外の場合、あて先キューマネージャがリモートキューマネージャであっても、ローカルキューマネージャに定義されたあて先キューの DefPersistence 属性の値がメッセージ永続性として使用されます。

メッセージが登録されたとき、DefPersistence 属性の値が Persistence フィールドに設定されます。そのあとに DefPersistence 属性の値を変更しても、すでに登録したメッセージには影響しません。キューの実名をたどるまでに複数の定義がある場合（クラスタキューに名称解決される場合を除きます）、最初に登録したキューの定義のこの属性が、省略時のメッセージ永続性になります。この属性は、次に示すオブジェクトに適用されます。

- 別名キュー
- ローカルキュー
- リモートキューのローカル定義
- キューマネージャの別名
- 転送キュー（例えば、DefXmitQName キュー）

永続メッセージと非永続メッセージを、同じキューに格納できます。

応答を返す場合、アプリケーションは通常、問い合わせメッセージの永続性を応答メッセージに使用します。

MQGET 命令で返される値は、MQPER\_PERSISTENT または MQPER\_NOT\_PERSISTENT のどちらかです。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQPER\_PERSISTENCE\_AS\_Q\_DEF です。

## ● Priority (MQLONG 型) メッセージ優先度

MQPUT 命令および MQPUT1 命令では、0 以上の値か、次に示す値を指定してください。

### MQPRI\_PRIORITY\_AS\_Q\_DEF

省略時のメッセージ優先度を使用します。

- クラスタキューの場合、MQOPEN 命令で指定する MQOD 構造体の ObjectQMgrName フィールドの値によって、使用される優先度が異なります。

メッセージがシステム転送キューに登録されるときに、DefPriority 属性の値が Priority フィールドにコピーされます。その後、DefPriority 属性が変更されても、キューに登録済みのメッセージは影響を受けません。

使用される優先度を次に示します。

MQOD 構造体の ObjectQMgrName フィールドに空白を指定した場合

あて先キュー（クラスタキュー）に定義された DefPriority 属性の値が、メッセージ優先度として使用されます。

ただし、MQOD 構造体の ObjectName フィールドにリモートキューのローカル定義を指定して、あて先キュー（クラスタキュー）へ名称解決される場合は、システム転送キューに定義された DefPriority 属性の値が、メッセージ優先度として使用されます。

#### MQOD 構造体の ObjectQMgrName フィールドにあて先キューマネージャを指定した場合

システム転送キューに定義された DefPriority 属性の値が、メッセージ優先度として使用されます。

- クラスタキュー以外の場合、あて先キューマネージャがリモートキューマネージャであっても、ローカルキューマネージャに定義されたあて先キューの DefPriority 属性の値がメッセージ優先度として使用されます。

キューの実名をたどるまでに複数の定義がある場合（クラスタキューに名称解決される場合を除きます）、最初に登録したキューの定義のこの属性が省略時のメッセージ優先度になります。この属性は、次に示すオブジェクトに適用されます。

- 別名キュー
- ローカルキュー
- リモートキューのローカル定義
- キューマネージャの別名
- 転送キュー（例えば、DefXmitQName キュー）

MQGET 命令で返される値は、常に 0 以上です。MQPRI\_PRIORITY\_AS\_Q\_DEF は返されません。

ローカルキューマネージャに定義された最大値、つまり、キューマネージャの MaxPriority 属性の値よりも大きな優先度でメッセージを登録した場合、メッセージは受け付けられますが、キューマネージャの優先度の最大値に置き換えられます。この場合、MQPUT 命令および MQPUT1 命令は、完了コード MQCC\_WARNING、理由コード MQRC\_PRIORITY\_EXCEEDS\_MAXIMUM で終了します。ただし、Priority フィールドには、メッセージを登録したアプリケーションで指定した値が残ります。

応答を返す場合、アプリケーションは通常、問い合わせメッセージの優先度を応答メッセージに使用します。また、MQPRI\_PRIORITY\_AS\_Q\_DEF を指定しても、アプリケーションを変更することなく優先度を調整できます。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQPRI\_PRIORITY\_AS\_Q\_DEF です。

### ● PutApplName (MQCHAR28 型) 登録アプリケーション名

メッセージの登録元コンテキストの一部です。メッセージコンテキストについては、この章の「MQMD 構造体 - メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。このフィールドのヌル文字以降の情報は破棄されます。ヌル文字以降の文字は、キューマネージャによって空白に置き換えられます。MQPMO\_SET\_ALL\_CONTEXT を指定しなかった場合、このフィールドは入力時に無視されて出力専用フィールドとなり、このフィールドには実行中のユーザサーバ名が設定されます。

MQPUT 命令または MQPUT1 命令が成功すると、このフィールドには、転送されたメッセージの PutApplName が設定されます。メッセージにコンテキストがなかった場合、このフィールドは空白で埋められます。

MQGET 命令では、出力用フィールドとなります。

このフィールドの長さは、MQ\_PUT\_APPL\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、28 個の空白です。

### ● PutApplType (MQLONG 型) 登録アプリケーションタイプ

メッセージの登録元コンテキストの一部です。メッセージコンテキストについては、この章の「MQMD 構造体 – メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

次に示す標準のタイプを使用できます。また、ユーザが定義したタイプも使用できます。ただし、ユーザが定義できる値の範囲は、MQAT\_USER\_FIRST から MQAT\_USER\_LAST までです。

MQAT\_AIX : AIX アプリケーション

MQAT\_BROKER : ブローカ

MQAT\_CICS : CICS トランザクション

MQAT\_DOS : PC-DOS アプリケーション

MQAT\_DQM : 分散キューマネージャエージェント

MQAT\_IMS : IMS アプリケーション

MQAT\_JAVA : Java

MQAT\_MVS : MVS または TSO アプリケーション

MQAT\_OPEN\_TP1 : OpenTP1 アプリケーション

MQAT\_OS2 : OS/2 またはプレゼンテーションマネージャアプリケーション

MQAT\_OS400 : OS/400 アプリケーション

MQAT\_QMGR : キューマネージャ

MQAT\_TMS\_4V : TMS-4V/SP アプリケーション

MQAT\_WINDOWS : 16 ビット Windows アプリケーション

MQAT\_WLM : z/OS ワークロードマネージャアプリケーション

MQAT\_XDM : XDM アプリケーション

MQAT\_ZOS : z/OS アプリケーション

## MQAT\_DEFAULT

省略時のアプリケーションタイプ

アプリケーションが動作しているプラットフォームの、省略時のアプリケーションタイプです。TP1/Message Queue では、MQAT\_OPEN\_TP1 が仮定されます。

### 注意事項

この値は環境で定義する定数です。これによって、各プラットフォームに適切なヘッダ、インクルードファイル、およびコピーファイルを使用してコンパイルしたアプリケーションが実行できます。

## MQAT\_UNKNOWN

未知のアプリケーションタイプ

この値は、アプリケーションタイプがわからないときに使用できます。ほかのコンテキスト情報があってもかまいません。

## MQAT\_USER\_FIRST

ユーザが定義できるアプリケーションタイプの最小値

## MQAT\_USER\_LAST

ユーザが定義できるアプリケーションタイプの最大値

また、次の特殊な値もあります。

## MQAT\_NO\_CONTEXT

メッセージにコンテキスト情報がありません。

この値は、メッセージにコンテキスト情報がない、つまり、MQPMO\_\*コンテキストオプションが指定されなかったときに、キューマネージャによって設定されます。

取り出したメッセージにコンテキスト情報があるかどうかを、PutApplType フィールドで調べることができます。ほかのどのコンテキストフィールドも空白ではない場合、

MQPMO\_SET\_ALL\_CONTEXT を使用したアプリケーションでは、PutApplType フィールドに MQAT\_NO\_CONTEXT を指定できません。

MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。MQPMO\_SET\_ALL\_CONTEXT を指定しなかった場合、このフィールドは入力時に無視されて出力専用フィールドとなり、このフィールドには「15」が設定されま

す。「15」は、MQI ヘッドファイル (cmqc.h または CMQV.cbl) に設定されている「MQAT\_OPEN\_TP1」または「MQAT-OPEN-TP1」の値です。cmqc.h または CMQV.cbl を参照してください。

MQPUT 命令または MQPUT1 命令が成功すると、このフィールドには、転送されたメッセージの PutApplType が設定されます。メッセージにコンテキストがなかった場合、MQAT\_NO\_CONTEXT が設定されます。

MQGET 命令では、出力用フィールドとなります。

このフィールドの初期値は MQAT\_NO\_CONTEXT です。

## ● PutDate (MQCHAR8 型) 登録日付

メッセージの登録元コンテキストの一部です。メッセージコンテキストについては、この章の「MQMD 構造体 – メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

キューマネージャは、次の形式でこのフィールドを生成します。

YYYYMMDD

YYYY : 年 (4 けたの 10 進数)

MM : 月 (01~12 の 10 進数)

DD : 日 (01~31 の 10 進数)

PutDate フィールドと PutTime フィールドには、グリニッジ標準時 (GMT) を使用します。システムの時計を正しく設定してください。

トランザクション内からメッセージを登録した場合、コミットした日付ではなく、登録した日付になります。

MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。このフィールドの内容は、キューマネージャによってチェックされません。ただし、フィールド内のヌル文字以降の情報は破棄されます。ヌル文字以降の文字は、キューマネージャによって空白に置き換えられます。MQPMO\_SET\_ALL\_CONTEXT を指定しなかった場合、このフィールドは入力時に無視されて出力専用フィールドとなり、このフィールドにはメッセージを登録した日付が設定されます。

MQPUT 命令または MQPUT1 命令が成功すると、このフィールドには転送されたメッセージの PutDate が設定されます。メッセージにコンテキストがなかった場合、このフィールドは空白で埋められます。

MQGET 命令では、出力用フィールドとなります。

このフィールドの長さは、MQ\_PUT\_DATE\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、8 個の空白です。

## ● PutTime (MQCHAR8 型) 登録時刻

メッセージの登録元コンテキストの一部です。メッセージコンテキストについては、この章の「MQMD 構造体 – メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

キューマネージャは、次の形式でこのフィールドを生成します。

### HHMMSSSTH

HH：時 (00~23 の 10 進数)

MM：分 (00~59 の 10 進数)

SS：秒 (00~59 の 10 進数)

T：1/10 秒 (0~9 の 10 進数)

H：1/100 秒 (0~9 の 10 進数)

PutDate フィールドと PutTime フィールドには、グリニッジ標準時 (GMT) を使用します。システムの時計を正しく設定してください。

トランザクション内からメッセージを登録した場合、コミットした時刻ではなく登録した時刻になります。

MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。このフィールドの内容は、キューマネージャによってチェックされません。ただし、フィールド内のヌル文字以降の情報は破棄されます。ヌル文字以降の文字は、キューマネージャによって空白に置き換えられます。MQPMO\_SET\_ALL\_CONTEXT を指定しなかった場合、このフィールドは入力時に無視されて出力専用フィールドとなり、このフィールドにはメッセージを登録した時間が設定されます。

MQPUT 命令または MQPUT1 命令が成功すると、このフィールドには転送されたメッセージの PutTime が設定されます。メッセージにコンテキストがなかった場合、このフィールドは空白で埋められます。

MQGET 命令では、出力用フィールドとなります。

このフィールドの長さは、MQ\_PUT\_TIME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、8 個の空白です。

## ● ReplyToQ (MQCHAR48 型) 応答キュー名

アプリケーションが MQMT\_REPLY タイプと MQMT\_REPORT タイプのメッセージを取り出すキューの名称です。ReplyToQMgr のキューマネージャに定義されたキューのローカル名称です。

MQPUT 命令および MQPUT1 命令では、MsgType フィールドが MQMT\_REQUEST のとき、または Report フィールドで報告を要求したときに、このフィールドに空白を指定できません。ただし、ここで指定した値は、どんなタイプのメッセージの場合でも、それを取り出すアプリケーションに返されます。

ReplyToQMgr フィールドが空白の場合、ローカルキューマネージャは、ReplyToQ と同じ名称のキューの定義を参照します。この名称のリモートキューのローカル定義があれば、転送されるメッセージの ReplyToQ の名称は、リモートキューのローカル定義の RemoteQName 属性で置き換えられます。この値は、MQGET 命令でメッセージを取り出すアプリケーションに、メッセージ記述子の一部として返されます。リモートキューのローカル定義がない場合、ReplyToQ フィールドは変更されません。

名称を指定するとき、余りを空白で埋めることができます。また、最初のヌル文字以降の文字は、空白として扱われます。ただし、キューの名称の規則に従っているかどうかはチェックされません。これは、転送されるメッセージの ReplyToQ の名称が置き換えられた場合も同様です。名称を指定したかどうか、必要時にチェックされるだけです。

応答キューが必要ない場合、ReplyToQ フィールドには、次のどれかを指定してください。ただし、チェックはされません。

- 空白
- ヌル文字列 (C 言語の場合)
- ヌル文字で終了する幾つかの空白 (C 言語の場合)

このフィールドは、左詰めで指定します。

MQGET 命令では、この名称は、フィールドの余りが常に空白で埋められて返されます。

報告を要求したメッセージが配布されなかった場合で、報告メッセージも指定したキューに登録されなかったときは、登録元メッセージも報告メッセージもデッドレターキューに転送されます。詳細については、「3. オブジェクトの属性」の「[キューマネージャの属性](#)」で DeadLetterQName 属性の説明を参照してください。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● ReplyToQMgr (MQCHAR48 型) 応答キューマネージャ名

応答メッセージまたは報告メッセージを送信する先の、キューマネージャ名です。ReplyToQ フィールドの値は、このキューマネージャに定義されたローカルキューの名称です。

ReplyToQMgr フィールドが空白の場合、ローカルキューマネージャは、ReplyToQ と同じ名称のキューの定義を参照します。この名称のリモートキューのローカル定義があれば、転送されるメッセージの ReplyToQMgr の名称は、リモートキューのローカル定義の RemoteQMgrName 属性の値で置き換えられます。この値は、MQGET 命令でメッセージを取り出すアプリケーションに、メッセージ記述子の一部として返されます。リモートキューのローカル定義がない場合、ReplyToQMgr フィールドはローカルキューマネージャの名称になります。

名称を指定するとき、余りを空白で埋めることができます。また、最初のヌル文字以降の文字は、空白として扱われます。ただし、キューマネージャの名称の規則に従っているかどうかはチェックされません。また、そのキューマネージャ名が送信先のキューマネージャで認識されているかどうか、チェックされません。これは、転送されるメッセージの ReplyToQMGr の名称が置き換えられた場合も同様です。オブジェクトの命名規則については、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。

応答キューが必要ない場合、ReplyToQMGr フィールドは、次のどれかを指定してください。ただし、チェックはされません。

- 空白
- ヌル文字列 (C 言語の場合)
- ヌル文字で終了する幾つかの空白 (C 言語の場合)

このフィールドは、左詰めで指定します。

MQGET 命令では、この名称は、フィールドの余りが常に空白で埋められて返されます。

このフィールドは、MQGET 命令の出力用フィールドです。また、MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● Report (MQLONG 型) 報告オプション

報告とは、登録元メッセージに関連して発生した各種の事象についてアプリケーションに通知するためのメッセージです。Report フィールドを使用すると、登録元アプリケーションは次のことを指定できます。

- どんな種類の報告メッセージが必要かを指定します。
- 報告メッセージ内にアプリケーションデータを組み込むかどうかを指定します。
- 報告メッセージまたは応答メッセージに、どのようにしてメッセージ識別子や関連識別子を設定するかを指定します。この場合は、報告と応答の両方で、Report フィールドを使用できます。

次に示す 6 種類の報告から、一つまたは複数を指定できます。指定しなくてもかまいません。

- 例外
- 保持時間終了
- 到着確認
- 配布確認
- 肯定動作の通知
- 否定動作の通知

それぞれの種類の報告では、報告メッセージ内にアプリケーションデータを組み込むか組み込まないかの、どちらかを指定してください。複数の種類の報告メッセージが必要なとき、または、MsgId と CorrelId をどのように生成するかを制御したいときは、次のどちらかに従ってください。

- それぞれを加算します。このとき、同じ値は 2 回以上加算しないでください。
- プログラミング言語がビット演算をサポートしている場合、それぞれのビット論理和を取ります。

なお、オプションおよびその指定方法の一覧については、「付録 A 命令および引数の一覧」を参照してください。

## 例外報告オプション：

### MQRO\_EXCEPTION

例外報告を要求します。

例外報告は、メッセージをほかのキューマネージャへ送信した場合で、あて先キューに到着しなかったときに、MCA によって生成されます。例えば、次のような場合です。

- あて先キューまたは仲介する転送キューが満杯だった場合
- キューに対してメッセージが長過ぎた場合

報告メッセージが生成されるのは、到着しなかったメッセージが、デッドレターキューに転送されたときだけです。デッドレターキューに転送されなかった場合、または、デッドレターキューを定義していなかった場合は、メッセージは転送キュー内に残ります。例外報告は生成されません。

MQPUT 命令または MQPUT1 命令で返される理由コードで、直後にエラーが判明する場合は、例外報告は生成されません。

また、アプリケーションも例外報告を生成できます。この場合、取り出したメッセージを処理できないことを報告します。例えば、計算機使用料の貸方の限度額を超過したときなどに、メッセージを処理できないことを報告します。

登録元メッセージのデータは、報告メッセージ内に組み込まれません。

次に示すオプションは、MQRO\_EXCEPTION オプションと同時に指定できません。

- MQRO\_EXCEPTION\_WITH\_DATA
- MQRO\_EXCEPTION\_WITH\_FULL\_DATA

### MQRO\_EXCEPTION\_WITH\_DATA

例外報告をデータ付きで要求します。

報告メッセージ内に登録元メッセージのアプリケーションデータが組み込まれることを除けば、MQRO\_EXCEPTION と同様です。登録元メッセージに一つ以上の MQ ヘッダ構造体がある場合、それらの構造体はアプリケーションデータの 100 バイトとともに、報告メッセージに格納されます。

次に示すオプションは、MQRO\_EXCEPTION\_WITH\_DATA オプションと同時に指定できません。

- MQRO\_EXCEPTION
- MQRO\_EXCEPTION\_WITH\_FULL\_DATA

## MQRO\_EXCEPTION\_WITH\_FULL\_DATA

例外報告をデータ付きで要求します。

報告メッセージ内に、登録元メッセージのアプリケーションデータが組み込まれることを除けば、MQRO\_EXCEPTION と同様です。報告メッセージ内にアプリケーションデータをすべて組み込みます。次に示すオプションは、MQRO\_EXCEPTION\_WITH\_FULL\_DATA オプションと同時に指定できません。

- MQRO\_EXCEPTION
- MQRO\_EXCEPTION\_WITH\_DATA

## 保持時間終了報告オプション：

### MQRO\_EXPIRATION

保持時間終了報告を要求します。

保持時間終了報告は、メッセージ保持時間 (Expiry フィールド) を終了したメッセージが、アプリケーション配布前に破棄されたときに生成されます。この報告は、リモートキューマネージャによって生成されます。MQRO\_EXCEPTION オプションを指定していても、MQRO\_EXPIRATION オプションを指定しないと、メッセージ保持時間切れが原因でメッセージが破棄されても報告メッセージは生成されません。

登録元メッセージのデータは、報告メッセージ内に組み込まれません。

次に示すオプションは、MQRO\_EXPIRATION オプションと同時に指定できません。

- MQRO\_EXPIRATION\_WITH\_DATA
- MQRO\_EXPIRATION\_WITH\_FULL\_DATA

### MQRO\_EXPIRATION\_WITH\_DATA

保持時間終了報告をデータ付きで要求します。

報告メッセージ内に登録元メッセージのアプリケーションデータが組み込まれることを除けば、MQRO\_EXPIRATION と同様です。登録元メッセージに一つ以上の MQ ヘッダ構造体がある場合、それらの構造体は 100 バイトのアプリケーションデータとともに、報告メッセージに格納されます。次に示すオプションは、MQRO\_EXPIRATION\_WITH\_DATA オプションと同時に指定できません。

- MQRO\_EXPIRATION
- MQRO\_EXPIRATION\_WITH\_FULL\_DATA

### MQRO\_EXPIRATION\_WITH\_FULL\_DATA

保持時間終了報告をデータ付きで要求します。

報告メッセージ内に登録元メッセージのアプリケーションデータが組み込まれることを除けば、MQRO\_EXPIRATION と同様です。

次に示すオプションは、MQRO\_EXPIRATION\_WITH\_FULL\_DATA オプションと同時に指定できません。

- MQRO\_EXPIRATION
- MQRO\_EXPIRATION\_WITH\_DATA

## 到着確認報告オプション：

### MQRO\_COA

到着確認報告を要求します。

到着確認報告は、あて先キューにメッセージが到着したときに生成されます。この報告は、あて先キューを管理するキューマネージャによって生成されます。

トランザクション内からメッセージを登録した場合で、あて先キューがローカルキューのときは、到着確認報告メッセージはトランザクションがコミットしたときに有効になります。

メッセージ記述子の Format フィールドが MQFMT\_DEAD\_LETTER\_HEADER の場合、到着確認報告は生成されません。これによって、配布できないでデッドレターキューへ転送されたメッセージのときに、到着確認報告が生成されるのを防ぎます。

登録元メッセージのデータは、報告メッセージ内に組み込まれません。

次に示すオプションは、MQRO\_COA オプションと同時に指定できません。

- MQRO\_COA\_WITH\_DATA
- MQRO\_COA\_WITH\_FULL\_DATA

### MQRO\_COA\_WITH\_DATA

到着確認報告をデータ付きで要求します。

報告メッセージ内に登録元メッセージのアプリケーションデータが組み込まれることを除けば、MQRO\_COA と同様です。登録元メッセージに一つ以上の MQ ヘッド構造体がある場合、それらの構造体は 100 バイトのアプリケーションデータとともに、報告メッセージに格納されます。

次に示すオプションは、MQRO\_COA\_WITH\_DATA オプションと同時に指定できません。

- MQRO\_COA
- MQRO\_COA\_WITH\_FULL\_DATA

### MQRO\_COA\_WITH\_FULL\_DATA

到着確認報告をデータ付きで要求します。

報告メッセージ内に登録元メッセージのアプリケーションデータが含まれることを除けば、MQRO\_COA と同様です。

次に示すオプションは、MQRO\_COA\_WITH\_FULL\_DATA オプションと同時に指定できません。

- MQRO\_COA
- MQRO\_COA\_WITH\_DATA

## 配布確認報告オプション：

### MQRO\_COD

配布確認報告を要求します。

配布確認報告は、アプリケーションによってメッセージが取り出され、キューから削除されたときに生成されます。この報告は、キューマネージャによって生成されます。

トランザクション内からメッセージが取り出された場合、配布確認報告メッセージは、そのトランザクションの処理の一部として生成されます。このため、トランザクションがコミットするまでは、報告は有効になりません。トランザクションがロールバックした場合、報告メッセージは通知されません。

メッセージ記述子の Format フィールドが MQFMT\_DEAD\_LETTER\_HEADER の場合、配布確認報告は生成されません。これによって、配布できないでデッドレターキューへ転送されたメッセージのときに、配布確認報告が生成されるのを防ぎます。

登録元メッセージのデータは、報告メッセージ内に組み込まれません。

次に示すオプションは、MQRO\_COD オプションと同時に指定できません。

- MQRO\_COD\_WITH\_DATA
- MQRO\_COD\_WITH\_FULL\_DATA

### MQRO\_COD\_WITH\_DATA

配布確認報告をデータ付きで要求します。

報告メッセージ内に登録元メッセージのアプリケーションデータが組み込まれることを除けば、MQRO\_COD と同様です。元のメッセージに一つ以上の MQ ヘッダ構造体がある場合、それらの構造体は 100 バイトのアプリケーションデータとともに、報告メッセージに格納されます。

MQGMO\_ACCEPT\_TRUNCATED\_MSG オプションを使用して、登録元データの後部が切り捨てられていた場合、報告メッセージデータの長さは切り捨てられた長さになります。

次に示すオプションは、MQRO\_COD\_WITH\_DATA オプションと同時に指定できません。

- MQRO\_COD
- MQRO\_COD\_WITH\_FULL\_DATA

### MQRO\_COD\_WITH\_FULL\_DATA

配布確認報告をデータ付きで要求します。

報告メッセージ内に登録元メッセージのアプリケーションデータが組み込まれることを除けば、MQRO\_COD と同様です

MQGMO\_ACCEPT\_TRUNCATED\_MSG オプションを使用して登録元データの後部が切り捨てられていた場合、報告メッセージデータの長さは切り捨てられた長さになります。

次に示すオプションは、MQRO\_COD\_WITH\_FULL\_DATA オプションと同時に指定できません。

- MQRO\_COD
- MQRO\_COD\_WITH\_DATA

### 肯定動作または否定動作の報告オプション：

受信アプリケーションが肯定動作または否定動作の報告メッセージの送信を要求するために、次に示すオプションを指定できます。

### MQRO\_PAN

肯定動作を通知する報告が要求されました。

この形式の報告は、メッセージを取り出し、そのメッセージに対して動作するアプリケーションによって生成されます。メッセージで要求された動作は成功したことを示します。報告を生成するアプリケーションは、どんなデータでも報告の中に含めるかどうか決定できます。

メッセージを取り出すアプリケーションにこの要求を通知する以外に、キューマネージャはこのオプションに基づいて何もしません。報告は、取り出すアプリケーションの責任で生成してください。

## MQRO\_NAN

否定動作を通知する報告が要求されました。

この形式の報告は、メッセージを取り出し、そのメッセージに対して動作するアプリケーションによって生成されます。メッセージで要求された動作は成功しなかったことを示します。報告を生成するアプリケーションは、どんなデータでも報告の中に含めるかどうか決定できます。

メッセージを取り出すアプリケーションにこの要求を通知する以外に、キューマネージャはこのオプションに基づいて何もしません。報告は、取り出すアプリケーションの責任で生成してください。

肯定動作に対応する条件か、否定動作に対応する条件か決めるのはアプリケーションの責任です。しかし、もし要求が部分的に実行されたのであれば、PAN 報告よりも NAN 報告を生成することをお勧めします。また、すべての可能な条件は、肯定動作または否定動作の両方に対応させるのではなく、どちらかに対応させることをお勧めします。

## MsgId 設定用オプション：

報告メッセージまたは応答メッセージに、どのように MsgId を設定するかを次のオプションで指定できます。

### MQRO\_NEW\_MSG\_ID

メッセージ識別子を付け直します。

これは、メッセージ識別子を指定しないときのオプションです。このメッセージの結果で報告または応答が生成されたとき、報告メッセージまたは応答メッセージのメッセージ識別子は新しく付け直されません。

### MQRO\_PASS\_MSG\_ID

メッセージ識別子を引き渡します。

このメッセージの結果で報告または応答が生成されたとき、このメッセージの MsgId が報告メッセージまたは応答メッセージの MsgId フィールドへ複製されます。

このオプションを指定しなかった場合、MQRO\_NEW\_MSG\_ID が仮定されます。

## CorrelId 設定用オプション：

報告メッセージまたは応答メッセージにどのように CorrelId を設定するかを、次のオプションで指定できます。

### MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID

メッセージ識別子の値を複製します。

これは、相関識別子を指定しないときのオプションです。このメッセージの結果で報告または応答が生成されたとき、このメッセージのメッセージ識別子の値が報告メッセージまたは応答メッセージの CorrelId フィールドへ複写されます。

## MQRO\_PASS\_CORREL\_ID

相関識別子を引き渡します。

このメッセージの結果で報告または応答が生成されたとき、このメッセージの CorrelId が、報告メッセージまたは応答メッセージの CorrelId フィールドへ複写されます。

このオプションを指定しなかった場合、MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID が仮定されます。

応答または報告を生成するサーバでは、登録元メッセージに MQRO\_PASS\_CORREL\_ID および MQRO\_PASS\_MSG\_ID オプションが指定されているかどうか、確認する必要があります。指定されている場合は、オプションの説明の記述に従ってください。指定されていない場合は、省略時の記述に従ってください。

## 後処理オプション：

アプリケーションから登録したメッセージをあて先キューに登録できない場合、次に示すオプションのどちらかを指定して、メッセージの後処理を制御できます。これらのオプションが適用されるのは、送信側アプリケーションが要求した場合に、例外報告メッセージが生成されるときだけです。アプリケーションは後処理オプションを、例外報告の要求とは無関係に設定できます。

## MQRO\_DEAD\_LETTER\_Q

メッセージをデッドレターキューに登録します。

これはデフォルトの動作であり、あて先キューに登録できないメッセージはデッドレターキューに登録されることを示します。送信側からの要求がある場合には、例外報告メッセージが生成されます。

## MQRO\_DISCARD\_MSG

TP1/Message Queue では意味がありません。

メッセージを廃棄します。

あて先キューに登録できないメッセージを廃棄することを示します。送信側からの要求がある場合には、例外報告メッセージが生成されます。

デッドレターキューに登録しないで送信側に元のメッセージを返す必要がある場合、送信側は MQRO\_DISCARD\_MSG を MQRO\_EXCEPTION\_WITH\_FULL\_DATA と同時に指定する必要があります。

TP1/Message Queue は、送信側から MQRO\_DISCARD\_MSG オプション付きの問い合わせメッセージを受信しても、MQRO\_DEAD\_LETTER\_Q オプションの動作をします。両方指定しない場合、MQRO\_DEAD\_LETTER\_Q オプションの動作をします。

## デフォルトオプション：

次のオプションは、ほかに報告オプションを指定しない場合に指定できます。

## MQRO\_NONE

報告を要求しません。

MQRO\_NONE は、ほかのオプションが指定されていないことを示すのに使用できます。なお、MQRO\_NONE はプログラムの文書管理のために定義されています。ほかのオプションと一緒に指定しないでください。MQRO\_NONE の値は 0 なので、これ以外の用途では使用できません。

## Report フィールド全般について：

1. 指定した種類の報告だけが生成されることに注意してください。例えば、到着確認報告は必要であるが例外報告は必要ない場合、メッセージがあて先キューに格納されると到着確認報告が生成されます。しかし、メッセージが到着したときにあて先キューが満杯であっても、例外報告は生成されません。報告オプションを一つも指定しなかった場合、キューマネージャによっても MCA によっても、報告は生成されません。

報告オプションの中には、ローカルキューマネージャが認識していなくても指定できるものがあります。それは、そのオプションがあて先のキューマネージャで処理される場合です。詳細は、「[付録 D 報告オプションとメッセージフラグ](#)」を参照してください。

報告メッセージを要求する場合、報告先のキュー名を ReplyToQ フィールドに指定する必要があります。

2. 障害などで応答キューへ通知できない場合、報告メッセージはデッドレターキューへ転送されます。この転送も失敗した場合、またはデッドレターキューがなかった場合は、報告メッセージのタイプによって次のようになります。

- ・報告メッセージが例外報告の場合、例外報告を生成させたメッセージは転送キューに残っています。その場合、メッセージが消失しないことが保証されます。

- ・そのほかのタイプの報告メッセージは破棄され、通常どおりに処理が続行されます。これは、到着確認報告メッセージまたは配布確認報告メッセージがすでに配布されたことを示しています。または、保持時間終了報告は関係なくなったことを示しています。

報告メッセージがいったんキュー（通常は転送キュー）に格納されると、ほかの種類のメッセージと同様に扱われます。このため、登録元メッセージが永続メッセージだった場合、報告は永続メッセージとなり破棄されることはありません。

3. 報告が生成されるとき、登録元メッセージの UserIdentifier のユーザ識別子で ReplyToQ のキューが登録用にオープンされます。ただし、次の場合は除きます。

- ・MCA が例外報告を生成するとき、登録元メッセージが登録されたときのユーザ識別子が使用されます。

- ・キューマネージャが到着確認報告を生成するとき、登録元メッセージが登録されたときのユーザ識別子が使用されます。

報告を生成するアプリケーションは、通常、応答を生成するときと同様のユーザ識別子を使用します。つまり、登録元メッセージのユーザ識別子を使用します。

報告先がリモートキューの場合、ほかのメッセージのときと同様に、報告が必要かどうかを送信側と受信側で選択できます。

4. 報告をデータ付きで要求している場合、次のようになります。

- ・報告データは、後部を切り捨てられることはありません。応答キューに対して報告メッセージが長過ぎた場合、デッドレターキューへ転送されます。デッドレターキューに対しても長過ぎた場合、報告メッセージは破棄されます。

・登録元データの Format フィールドが MQFMT\_XMIT\_Q\_HEADER の場合、報告メッセージの 100 バイトのデータには、MQXQH 構造体は含まれません。登録元メッセージの MQXQH 構造体の後ろからの部分が、報告メッセージのデータになります。転送キューの場合も転送キュー以外の場合も同様です。

5. 到着確認報告、配布確認報告、または保持時間終了報告を応答キューに受信すると、登録元メッセージが到着したこと、配布されたこと、または保持時間が終了したことが保証されます。報告を要求しても、それらを受信しなかった場合、要因として次の現象が考えられます。
  - a. 接続が切断されたため、報告が止まっています。
  - b. 仲介する転送キューまたは応答キューに、満杯、登録禁止などのブロック状態があるため、報告が止まっています。
  - c. 報告がデッドレターキューに転送されました。
  - d. キューマネージャが報告を生成しようとしたときに、対応するキューにも、デッドレターキューにも登録できなかったため、報告を生成できませんでした。
  - e. メッセージの該当する到着、配布、または保持時間終了の状態から報告を生成するまでに、キューマネージャの障害が発生しました。ただし、次の配布確認報告のときは例外です。トランザクション内から登録元メッセージを取り出した場合、同じトランザクションの処理として配布確認報告が生成されるので、該当しません。

例外報告が届かなかった場合、上記の a., b., c. の場合が該当します。MCA が要求された例外報告を生成できなかった場合、つまり、応答キューとデッドレターキューのどちらにも登録できなかった場合は、登録元メッセージは送信側の転送キューに残り、回線はクローズされます。これは、例外報告が送信側で生成された場合も、回線の受信側で生成された場合も同様です。

6. 登録元メッセージが一時的にブロックされた場合は、例外報告が生成され、登録元メッセージはデッドレターキューへ転送されます。ブロック状態が解消されたあとに、アプリケーションがデッドレターキューから登録元メッセージを取り出し、再びあて先キューに登録したとします。この場合、次のように処理されます。

- ・例外報告が生成されても、実際にはメッセージがあて先キューに到着します。
- ・メッセージが別のブロックにぶつかると、一つの登録元メッセージに対して複数の例外報告が生成されます。

#### セグメント分割ができるメッセージへの報告メッセージ：

1. セグメントの分割ができるメッセージに、報告メッセージを要求できます。詳細は、「MsgFlags フィールド」の MQMF\_SEGMENTATION\_ALLOWED フラグの説明を参照してください。キューマネージャがメッセージのセグメント分割が必要であると判断した場合、適切な条件に合致する各セグメントに対して、報告メッセージが生成されます。したがって、アプリケーションは、要求した報告メッセージのそれぞれのタイプについて、複数の報告メッセージを受信できるようにしなければなりません。報告メッセージの GroupId フィールドは、元のメッセージのグループ識別子となるので、複数の報告を関連づけできます。Feedback フィールドは、それぞれの報告メッセージのタイプを識別するのに使用できます。
2. セグメントの報告メッセージを取り出す場合に、MQGMO\_LOGICAL\_ORDER を使用するとき、異なるタイプの報告が MQGET 命令で返却されることに注意してください。例えば、COA と COD

の両方の報告が、キューマネージャによってセグメント分割されたメッセージに要求されていた場合、報告メッセージに対する MQGET 命令では、COA と COD の報告が不定の形式で挿入されることがあります。これは、MQGMO\_COMPLETE\_MSG オプションを使用すると避けられます。このとき、任意で MQGMO\_ACCEPT\_TRUNCATED\_MSG オプションも使用できます。

MQGMO\_COMPLETE\_MSG を指定すると、キューマネージャは同じタイプの報告メッセージを組み立てます。例えば、最初の MQGET 命令では元のメッセージに関連するすべての COA メッセージを組み立て、2 番目の MQGET 命令ではすべての COD メッセージを組み立てることができません。なお、キューで最初に現れる報告メッセージが、最初に組み立てられます。

3. 自分自身でセグメントを登録するアプリケーションは、各セグメントに異なる報告オプションを指定できます。ただし、次の点に注意してください。

- ・セグメントが MQGMO\_COMPLETE\_MSG オプションで取り出されると、先頭のセグメントの報告オプションだけが、キューマネージャに認識されます。

- ・セグメントが一つ一つ取り出され、そのすべてに COD 報告オプションが設定されていない場合、1 回の MQGET 命令で報告メッセージを取り出すために MQGMO\_COMPLETE\_MSG オプションを使用したり、全報告メッセージが到着したのを確認するために MQGMO\_ALL\_SEGMENTS\_AVAILABLE オプションを使用したりすることはできません。

4. TP1/Message Queue のネットワークでは、キューマネージャはさまざまな機能を持っています。セグメント分割をサポートしていないキューマネージャや MCA によってセグメントに対する報告メッセージが生成されると、そのキューマネージャや MCA は報告メッセージに必要なセグメント情報を提供しないので、生成される報告メッセージの元となるメッセージを識別することが困難となります。これは、MQRO\*\_WITH\_DATA または MQRO\*\_WITH\_FULL\_DATA オプションを指定し、報告メッセージにデータを要求することで解決できます。ただし、MQRO\*\_WITH\_DATA を指定すると、セグメント分割をサポートしていないキューマネージャまたは MCA によって、報告メッセージが生成されます。その場合、100 バイト以下のアプリケーションデータが、報告メッセージを取り出すアプリケーションに返されることに注意してください。

### 報告メッセージのメッセージ記述子：

報告メッセージを取り出したとき、メッセージ記述子の Feedback フィールドで報告の種類を判断できます。

キューマネージャまたは MCA によって報告が生成されるとき、報告メッセージのメッセージ記述子には、次の値が設定されます。

MQMD 構造体のフィールド	値
StrucId	MQMD_STRUC_ID
Version	MQMD_VERSION_2
Report	MQRO_NONE
MsgType	MQMT_REPORT
Expiry	MQEI_UNLIMITED
Feedback	報告の種類に応じて (MQFB_COA, MQFB_COD, MQFB_EXPIRATION, または MQRC_*)

MQMD 構造体のフィールド	値
Encoding	登録元のメッセージ記述子から複写
CodedCharSetId	登録元のメッセージ記述子から複写
Format	登録元のメッセージ記述子から複写
Priority	登録元のメッセージ記述子から複写
Persistence	登録元のメッセージ記述子から複写
MsgId	登録元のメッセージ記述子の報告オプションどおり
CorrelId	登録元のメッセージ記述子の報告オプションどおり
BackoutCount	0
ReplyToQ	空白
ReplyToQMgr	空白
UserIdentifier	MQPMO_PASS_IDENTITY_CONTEXT オプションによって設定
AccountingToken	MQPMO_PASS_IDENTITY_CONTEXT オプションによって設定
ApplIdentityData	MQPMO_PASS_IDENTITY_CONTEXT オプションによって設定
PutApplType	MQAT_QMGR, または MCA に応じた値
PutApplName	キューマネージャ名または MCA 名の先頭の 28 文字
PutDate	報告メッセージが送信された日付
PutTime	報告メッセージが送信された時刻
ApplOriginData	空白
GroupId	登録元のメッセージ記述子から複写
MsgSeqNumber	登録元のメッセージ記述子から複写
Offset	登録元のメッセージ記述子から複写
MsgFlags	登録元のメッセージ記述子から複写
OriginalLength	MQOL_UNDEFINED でなければ、登録元のメッセージ記述子から複写される。それ以外の場合は、元のメッセージデータの長さが設定される。

報告を生成するアプリケーションは、ReplyToQMgr フィールドとコンテキストフィールドを除いて、同様の値を設定してください。ReplyToQMgr フィールドには空白を設定できます。また、コンテキストフィールドに設定する場合は、応答のときと同じオプションを使用します。通常、MQPMO\_PASS\_IDENTITY\_CONTEXT オプションを使用します。

#### 報告要求の有無の調査方法：

報告フィールドは、サブフィールドから構成されます。このため、ある報告が要求されているかどうかを調べる場合、次のどちらかの手順を踏んでください。

#### 1. データタイプ

ビット演算をサポートしているプログラミング言語では、次の手順を踏んでください。

- a.  
次の値から、調べたい報告の種類を一つ選択します。

報告タイプ	値
到着確認	MQRO_COA_WITH_FULL_DATA
配布確認	MQRO_COD_WITH_FULL_DATA
例外	MQRO_EXCEPTION_WITH_FULL_DATA
保持時間終了	MQRO_EXPIRATION_WITH_FULL_DATA

ここで選択したものを、値 A と呼びます。

- b.  
報告フィールドの値と値 A の論理積を取ります。この結果を、値 B と呼びます。
- c.  
値 B が、調べたい報告の種類各値と等しいかどうかを比較します。  
例えば、値 A が MQRO\_EXCEPTION\_WITH\_FULL\_DATA のとき、登録元が指定した報告タイプを調べるために、値 B を次のそれぞれと比較します。

MQRO\_NONE

MQRO\_EXCEPTION

MQRO\_EXCEPTION\_WITH\_DATA

MQRO\_EXCEPTION\_WITH\_FULL\_DATA

どんな順序で比較してもかまいません。アプリケーションで任意の順序で比較できます。

MQRO\_PASS\_MSG\_ID オプションまたは MQRO\_PASS\_CORREL\_ID オプションを調べるときも、同様の手段を利用できます。値 A として該当する定数を選択して、上記のとおり処理してください。

ビット演算をサポートしていないプログラミング言語では、次の手順を踏んでください。

- a.  
次の値から、調べたい報告の種類を一つ選択します。

報告タイプ	値
到着確認	MQRO_COA
配布確認	MQRO_COD
例外	MQRO_EXCEPTION
保持時間終了	MQRO_EXPIRATION

ここで選択したものを、値 A と呼びます。

- b.  
報告フィールドの値を、値 A で除算します。この結果を、値 B と呼びます。

- c.  
値 B を 8 で除算します。この結果を、値 C と呼びます。
- d.  
値 C を 8 で乗算し、それを B から減算します。この結果を、値 D と呼びます。
- e.  
値 D を値 A で乗算します。この結果を、値 E と呼びます。
- f.  
値 E が、調べたい報告の種類各値と等しいかどうか比較します。  
例えば、値 A が MQRO\_EXCEPTION のとき、登録元が指定した報告タイプを調べるために、値 E を次のそれぞれと比較します。

MQRO\_NONE

MQRO\_EXCEPTION

MQRO\_EXCEPTION\_WITH\_DATA

MQRO\_EXCEPTION\_WITH\_FULL\_DATA

どんな順序で比較してもかまいません。アプリケーションで任意の順序で比較できます。

この手段の擬似的なコード表現を、次に示します。

A = MQRO\_EXCEPTION

B = Report/A

C = B/8

D = B-C\*8

E = D\*A

MQRO\_PASS\_MSG\_ID オプションまたは MQRO\_PASS\_CORREL\_ID オプションを調べるときも、同様の手段を使用できます。値 A として該当する定数を選択して、上記のとおり処理してください。ただし、ステップ c および d の値 8 は、2 で置き換えてください。

Report フィールドは、MQGET 命令の出力用フィールド、および MQPUT 命令と MQPUT1 命令の入力用フィールドです。

このフィールドの初期値は MQRO\_NONE です。

## ● Struclid (MQCHAR4 型) 構造体識別子

次の値を取ります。

### MQMD\_STRUC\_ID

メッセージ記述子の構造体識別子

C 言語では、MQMD\_STRUC\_ID\_ARRAY も定義されています。これは、MQMD\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

これは入力用のフィールドです。

このフィールドの初期値は MQMD\_STRUC\_ID です。

## ● UserIdentifier (MQCHAR12 型) ユーザ識別子

メッセージの識別コンテキストの一部です。メッセージコンテキストの詳細については、この章の「MQMD 構造体 - メッセージ記述子」の「概要」、およびマニュアル「TP1/Message Queue プログラム作成の手引」を参照してください。

UserIdentifier フィールドは、メッセージを最初に登録したアプリケーションのユーザ識別子を指定します。オブジェクトをオープンする場合の ObjDesc 引数の AlternateUserId フィールドに、この情報を使用できます。

この情報は、キューマネージャでは文字データとして扱われます。ただし、データの形式は規定されません。MQOPEN 命令で MQOO\_ALTERNATE\_USER\_AUTHORITY を指定した場合、または MQPUT1 命令で MQPMO\_ALTERNATE\_USER\_AUTHORITY を指定した場合は、キューマネージャによってこの情報が生成されるときに、ObjDesc 引数の AlternateUserId が使用されます。省略した場合、環境に依存するユーザ識別子として、TP1/Message Queue では「OPENTP1」が使用されます。

通常、このフィールドはキューマネージャによって生成される出力用のフィールドです。しかし、MQPUT または MQPUT1 命令の場合、このフィールドを入出力用のフィールドにして、キューマネージャに生成させる代わりにフィールドを指定できます。

MQPUT または MQPUT1 命令の場合にキューマネージャに UserIdentifier フィールドを生成させないときは、PutMsgOpts 引数には MQPMO\_SET\_IDENTITY\_CONTEXT または MQPMO\_SET\_ALL\_CONTEXT のどちらかを指定します。また、UserIdentifier フィールドにユーザ識別子を指定します。

MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数で MQPMO\_SET\_IDENTITY\_CONTEXT または MQPMO\_SET\_ALL\_CONTEXT を指定した場合、入出力用フィールドとなります。このフィールドのヌル文字以降の情報は破棄されます。ヌル文字以降の文字は、キューマネージャによって空白に置き換えられます。MQPMO\_SET\_IDENTITY\_CONTEXT も MQPMO\_SET\_ALL\_CONTEXT も指定しなかった場合、このフィールドは入力時に無視されて出力専用フィールドとなり、このフィールドには「OPENTP1△△△△△」が設定されます。△は空白を表します。

MQPUT 命令または MQPUT1 命令が成功すると、このフィールドには、転送されたメッセージの UserIdentifier が設定されます。メッセージにコンテキストがなかった場合、このフィールドは空白で埋められます。

MQGET 命令では、出力用フィールドとなります。

このフィールドの長さは、MQ\_USER\_ID\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、12 個の空白です。

## ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

### MQMD\_VERSION\_1

バージョン 1 のメッセージ記述子の構造体バージョン番号

### MQMD\_VERSION\_2

バージョン 2 のメッセージ記述子の構造体バージョン番号

バージョン 2 構造体でだけ存在するフィールドがあります。

MQMD\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことがフィールドの説明に記載されています。次に示す定数は、現在のバージョン番号を指定します。

### MQMD\_CURRENT\_VERSION

現在のバージョンのメッセージ記述子の構造体バージョン番号

現在のバージョンのバージョン番号を指定します。

これは入力用のフィールドです。

このフィールドの初期値は MQMD\_VERSION\_1 です。

# MQMDE 構造体 – メッセージ記述子の拡張

MQMDE 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQMDE_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQMDE_VERSION_2
StrucLength (MQLONG 型)	構造体長	MQMDE_LENGTH_2
Encoding (MQLONG 型)	MQMDE 構造体に続くデータのマシンコード形式	MQENC_NATIVE
CodedCharSetId (MQLONG 型)	MQMDE 構造体に続くデータの文字セット識別子	MQCCSI_UNDEFINED
Format (MQCHAR8 型)	MQMDE 構造体に続くデータのフォーマット名	MQFMT_NONE
Flags (MQLONG 型)	フラグ	MQMDEF_NONE
GroupId (MQBYTE24 型)	グループ識別子	MQGI_NONE
MsgSeqNumber (MQLONG 型)	メッセージシーケンス番号	1
Offset (MQLONG 型)	オフセット	0
MsgFlags (MQLONG 型)	メッセージフラグ	MQMF_NONE
OriginalLength (MQLONG 型)	登録元メッセージ長	MQOL_UNDEFINED

## 概要

### 目的

MQMDE 構造体には、アプリケーションメッセージデータに先行して発生することのあるデータを記述します。MQMDE 構造体には、バージョン 2 の MQMD 構造体に含まれるが、バージョン 1 の MQMD 構造体には含まれないフィールドがあります。

### フォーマット名

MQFMT\_MD\_EXTENSION

### 文字セットおよびマシンコード

MQMDE 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれキューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。

MQMDE 構造体の文字セットとマシンコードは、次に示す構造体の CodedCharSetId および Encoding フィールドに指定します。

- MQMD 構造体 (MQMDE 構造体がメッセージデータの先頭にある場合)

- MQMDE 構造体に先行するヘッダ構造体（その他の場合）

MQMDE 構造体がキューマネージャの文字セットとマシンコードでない場合、MQMDE 構造体は保存されますが認識されません。つまり、MQMDE 構造体はメッセージデータとして扱われます。

## 使用方法

通常のアプリケーションでは、バージョン 2 の MQMD 構造体を使用してください。バージョン 2 の MQMD 構造体を使用する場合、MQMDE 構造体を使用する必要はありません。しかし、特殊な用途のアプリケーションおよびバージョン 1 の MQMD 構造体を引き続き使用するアプリケーションでは、特定の場合に MQMDE 構造体を使用することがあります。次に示す場合に使用します。

- MQPUT および MQPUT1 命令で指定する場合
- MQGET 命令で返される場合
- 転送キューのメッセージに使用する場合

それぞれについて説明します。

### MQPUT および MQPUT1 命令で MQMDE 構造体を指定する場合

MQPUT 命令および MQPUT1 命令でアプリケーションがバージョン 1 の MQMD 構造体を使用する場合について説明します。この場合、アプリケーションは、メッセージデータの前に MQMD 構造体の MQMDE 構造体が存在することを示すために、Format フィールドに MQFMT\_MD\_EXTENSION を設定します。アプリケーションが MQMDE 構造体を指定しない場合、キューマネージャは MQMDE 構造体のフィールドに初期値を仮定します。キューマネージャが使用する初期値は、構造体の初期値と同じです。

アプリケーションがバージョン 2 の MQMD 構造体を使用し、アプリケーションメッセージデータの前に MQMDE 構造体を置いた場合、MQMDE 構造体は次のように処理されます。

MQPUT または MQPUT1 命令で MQMDE 構造体が指定された場合のキューマネージャの処理を次に示します。

MQMD 構造体のバージョン	バージョン 2 のフィールドの値	MQMDE 構造体の対応するフィールドの値	キューマネージャの処理
1	—	有効	MQMDE 構造体を認識する。
2	初期値	有効	MQMDE 構造体を認識する。
2	初期値以外	有効	MQMDE 構造体をメッセージデータとして扱う。
1 または 2	任意	無効	命令が失敗し、該当する理由コードが戻る。
1 または 2	任意	MQMDE 構造体が無効な文字セットまたはマシンコードで記述されている。または、サポートされていないバージョンである。	MQMDE 構造体をメッセージデータとして扱う。

(凡例)

－：該当しません。

例外となる場合があります。アプリケーションがセグメントである場合、つまり、MQMF\_SEGMENT または MQMF\_LAST\_SEGMENT フラグが設定されている場合、かつメッセージを登録するのにバージョン 2 の MQMD 構造体を使用し、MQMD 構造体のフォーマット名が MQFMT\_DEAD\_LETTER\_HEADER である場合、キューマネージャは MQMDE 構造体を生成し、MQDLH 構造体とそれに続くデータの間に入れます。キューマネージャがメッセージに関して保持する MQMD 構造体では、バージョン 2 のフィールドは初期値に設定されます。

バージョン 2 の MQMD 構造体には存在し、バージョン 1 の MQMD 構造体には存在しない幾つかのフィールドが、MQPUT 命令および MQPUT1 命令の入出力フィールドです。ただし、キューマネージャは、MQPUT 命令および MQPUT1 命令の出力で、MQMDE 構造体の同等のフィールドに値を返却しません。アプリケーションがその出力値を必要とする場合は、バージョン 2 の MQMD 構造体を使用しなければなりません。

### MQGET 命令が返す MQMDE 構造体

MQGET 命令で、アプリケーションがバージョン 1 の MQMD 構造体を使用する場合について説明します。この場合、キューマネージャは、返却するメッセージの前に MQMDE 構造体を追加します。ただし、MQMDE 構造体の一つ以上のフィールドが初期値以外の値を持つ場合です。キューマネージャは、MQMD 構造体中の Format フィールドを MQFMT\_MD\_EXTENSION の値に設定し、MQMDE 構造体があることを示します。

アプリケーションが Buffer 引数の開始位置に MQMDE 構造体を使用する場合、MQMDE 構造体は無視されます。MQGET 命令から返却されるとき、必要であれば、メッセージの MQMDE 構造体で置き換えられます。MQMDE 構造体が不要の場合は、アプリケーションメッセージデータで上書きされます。

MQGET 命令で MQMDE 構造体が返される場合、MQMDE 構造体のデータは通常、キューマネージャの文字セットとマシンコードです。ただし次に示す場合、MQMDE 構造体が異なる文字セットとマシンコードになることがあります。

- MQMDE 構造体が MQPUT または MQPUT1 命令のデータとして扱われた場合
- TCP 接続で接続されたりリモートキューマネージャからメッセージを受け取ったが、受信側のメッセージチャンネルエージェント (MCA) が正しく設定されていなかった場合。詳細については、IBM MQ のマニュアルを参照してください。

### 転送キューにあるメッセージの MQMDE 構造体

転送キューにあるメッセージには、MQXQH 構造体が前に付いています。その MQXQH 構造体にはバージョン 1 の MQMD 構造体が含まれています。MQMDE 構造体は一つ以上のフィールドで初期値以外の値を持つときだけ、MQXQH 構造体とアプリケーションデータの間が存在します。

ほかの MQ ヘッダ構造体が、MQXQH 構造体とアプリケーションデータの間に見える場合もあります。例えば、MQDH 構造体、または MQDLH 構造体が存在し、メッセージがセグメントではない場合、現れる順序は次のとおりです。

1. MQXQH 構造体 (バージョン 1 の MQMD 構造体を含む)
2. MQDH 構造体または MQDLH 構造体
3. MQMDE 構造体
4. アプリケーションデータ

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● CodedCharSetId (MQLONG 型) 文字セット識別子

MQMDE 構造体に続くデータの文字セット識別子を指定します。MQMDE 構造体自身の文字データには適用されません。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。キューマネージャは、このフィールドが有効であることをチェックしません。

### MQCCSI\_INHERIT

この構造体の文字セット識別子を引き継ぎます。

この構造体に続くデータの文字データは、この構造体と同じ文字セットです。

メッセージで送信される構造体のこの値を、キューマネージャは構造体の実際の文字セット識別子に変更します。エラーが発生しない場合、MQGET 命令では MQCCSI\_INHERIT は返されません。

MQMD 構造体の PutApplType フィールドの値が MQAT\_BROKER の場合、MQCCSI\_INHERIT は使用できません。

このフィールドの初期値は MQCCSI\_UNDEFINED です。

### ● Encoding (MQLONG 型) マシンコード形式

MQMDE 構造体に続くデータのマシンコード形式を指定します。MQMDE 構造体自身の数値データには適用されません。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。キューマネージャは、このフィールドが有効であることをチェックしません。データのマシンコード形式については、この章の「MQMD 構造体 - メッセージ記述子」で Encoding フィールドを参照してください。

このフィールドの初期値は MQENC\_NATIVE です。

### ● Flags (MQLONG 型) フラグ

次のフラグを指定できます。

### MQMDEF\_NONE

フラグなし。

このフィールドの初期値は MQMDEF\_NONE です。

### ● Format (MQCHAR8 型) フォーマット名

MQMDE 構造体に続くデータのフォーマット名を表すフィールドです。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。キューマネージャはこのフィールドが有効であることをチェックしません。データのフォーマット名については、この章の「MQMD 構造体 – メッセージ記述子」で Format フィールドを参照してください。

このフィールドの初期値は MQFMT\_NONE です。

### ● GroupId (MQBYTE24 型) グループ識別子

グループ識別子については、この章の「MQMD 構造体 – メッセージ記述子」で GroupId フィールドを参照してください。

このフィールドの初期値は MQGI\_NONE です。

### ● MsgFlags (MQLONG 型) メッセージフラグ

メッセージフラグについては、この章の「MQMD 構造体 – メッセージ記述子」で MsgFlags フィールドを参照してください。

このフィールドの初期値は MQMF\_NONE です。

### ● MsgSeqNumber (MQLONG 型) メッセージシーケンス番号

グループ内の論理メッセージのシーケンス番号を表すフィールドです。メッセージシーケンス番号については、この章の「MQMD 構造体 – メッセージ記述子」で MsgSeqNumber フィールドを参照してください。

このフィールドの初期値は 1 です。

### ● Offset (MQLONG 型) オフセット

論理メッセージの先頭からの物理メッセージのデータオフセットを表すフィールドです。オフセットについては、この章の「MQMD 構造体 – メッセージ記述子」で Offset フィールドを参照してください。

このフィールドの初期値は 0 です。

### ● OriginalLength (MQLONG 型) 登録元メッセージ長

登録元メッセージ長については、この章の「MQMD 構造体 – メッセージ記述子」で OriginalLength フィールドを参照してください。

このフィールドの初期値は MQOL\_UNDEFINED です。

## ● StrucId (MQCHAR4 型) 構造体識別子

次の値を取ります。

### MQMDE\_STRUC\_ID

メッセージ記述子拡張構造体の識別子

C 言語では、MQMDE\_STRUC\_ID\_ARRAY も定義されています。これは、MQMDE\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの初期値は MQMDE\_STRUC\_ID です。

## ● StrucLength (MQLONG 型) 構造体長

次の値を取ります。

### MQMDE\_LENGTH\_2

バージョン 2 のメッセージ記述子拡張構造体の長さ

このフィールドの初期値は MQMDE\_LENGTH\_2 です。

## ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

### MQMDE\_VERSION\_2

バージョン 2 のメッセージ記述子拡張構造体

次に示す値には、現在のバージョンのバージョン番号を指定してください。

### MQMDE\_CURRENT\_VERSION

メッセージ記述子拡張構造体の現在のバージョン

このフィールドの初期値は MQMDE\_VERSION\_2 です。

# MQOD 構造体 – オブジェクト記述子

MQOD 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQOD_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQOD_VERSION_1
ObjectType (MQLONG 型)	オブジェクトタイプ	MQOT_Q
ObjectName (MQCHAR48 型)	オブジェクト名	ヌル文字列/空白
ObjectQMgrName (MQCHAR48 型)	オブジェクトキューマネージャ名	ヌル文字列/空白
DynamicQName (MQCHAR48 型)	動的キュー名	アスタリスク (*)
AlternateUserId (MQCHAR12 型)	代替ユーザ識別子	ヌル文字列/空白
RecsPresent (MQLONG 型) ※1	オブジェクトレコード数	0
KnownDestCount (MQLONG 型) ※1	ローカルキューの数	0
UnknownDestCount (MQLONG 型) ※1	リモートキューの数	0
InvalidDestCount (MQLONG 型) ※1	オープンに失敗したキューの数	0
ObjectRecOffset (MQLONG 型) ※1	最初のオブジェクトレコードまでのオフセット	0
ResponseRecOffset (MQLONG 型) ※1	最初の応答レコードまでのオフセット	0
ObjectRecPtr (MQPTR 型) ※1	最初のオブジェクトレコードのアドレス	ヌルポインタ/ヌル文字列
ResponseRecPtr (MQPTR 型) ※1	最初の応答レコードのアドレス	ヌルポインタ/ヌル文字列
AlternateSecurityId (MQBYTE40 型) ※2	代替セキュリティ識別子	MQSID_NONE
ResolvedQName (MQCHAR48 型) ※2	解決したキュー名称	ヌル文字列/空白
ResolvedQMgrName (MQCHAR48 型) ※2	解決したキューマネージャ名称	ヌル文字列/空白

注※1

Version フィールドの値が MQOD\_VERSION\_2 以降の場合だけ存在するフィールドです。

注※2

Version フィールドの値が MQOD\_VERSION\_3 以降の場合だけ存在するフィールドです。

## 概要

### 目的

MQOD 構造体は、オブジェクトを名前指定するために使用します。記述できるオブジェクトの種類を次に示します。

#### 1. データタイプ

- キューまたは配布リスト
- プロセス定義
- キューマネージャ

この構造体は、MQOPEN 命令および MQPUT1 命令の入出力用の引数です。

## バージョン

MQOD 構造体のバージョンには 1, 2 および 3 があります。しかし、すべての環境で MQOD\_VERSION\_3 がサポートされているわけではありません。複数の環境にアプリケーションを移植する場合は、使用するバージョンの MQOD 構造体が、すべての環境でサポートされている必要があります。MQOD\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことをフィールドの説明に記載しています。

サポートするプログラミング言語の COPY ファイルと INCLUDE ファイルでは、最新バージョンの MQOD 構造体が提供されます。ただし、Version フィールドの初期値は MQOD\_VERSION\_1 です。バージョン 1 の構造体がないフィールドを使用する場合は、アプリケーションで使用したいバージョンの番号を Version フィールドに設定してください。

配布リストをオープンするには、Version フィールドの値が MQOD\_VERSION\_2 以降でなければなりません。

## 文字セットおよびマシンコード

MQOD 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。TP1/Message Queue Access のクライアントアプリケーションの場合、MQOD 構造体はクライアントの文字セットとマシンコードに従います。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● AlternateSecurityId (MQBYTE40 型) 代替セキュリティ識別子

妥当な権限があるかどうかをチェックするために、AlternateUserId と一緒に認証サービスに渡されるセキュリティ識別子です。

AlternateSecurityId は次の場合にだけ使用されます。

- MQOO\_ALTERNATE\_USER\_AUTHORITY が MQOPEN 命令に指定されるか、MQOO\_ALTERNATE\_USER\_AUTHORITY が MQPUT1 命令に指定される場合
- AlternateUserId フィールドの最初のヌル文字まで、またはフィールドの最後までのもすべてが空白ではない場合

AlternateSecurityId は、次に示す構造体を持ちます。

- 先頭バイトは後続データの長さを含むバイナリ整数値であり、その長さは、長さを示すバイト自身も含まれます。セキュリティ識別子がない場合、長さは 0 です。
- 2 番目のバイトはセキュリティ識別子のタイプを示し、次の値を取ることができます。
  - MQSIDT\_NT\_SECURITY\_ID：Windows セキュリティ識別子
  - MQSIDT\_NONE：セキュリティ識別子ではありません。
- 3 番目以降のバイトから、先頭バイトで定義された長さまでのバイトはセキュリティ識別子自身です。
- 残りのバイトはバイナリ 0 に設定されます。

次の値を指定できます。

## MQSID\_NONE

セキュリティ識別子は指定されません。

フィールドの長さ分バイナリ 0 の値です。

C 言語では、MQSID\_NONE\_ARRAY 定数も定義されています。MQSID\_NONE と同じ値ですが、文字列ではなく、文字配列です。

これは入力用のフィールドです。

このフィールドの長さは MQ\_SECURITY\_ID\_LENGTH です。

このフィールドの初期値は MQSID\_NONE です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_3 以降の場合だけ有効です。

TP1/Message Queue ではこのフィールドは意味を持ちません。

## ● AlternateUserId (MQCHAR12 型) 代替ユーザ識別子

次のどちらかの場合、このフィールドに代替ユーザ識別子を指定します。

- MQOPEN 命令で MQOO\_ALTERNATE\_USER\_AUTHORITY を指定した場合
- MQPUT1 命令で MQPMO\_ALTERNATE\_USER\_AUTHORITY を指定した場合

キューに AlternateUserId が指定されている場合、キューマネージャは、メッセージが書き込まれるときに AlternateUserId の値を使用します。MQPUT 命令または MQPUT1 命令の呼び出しで指定された MQPMO\_\*\_CONTEXT オプションに従って識別コンテキスト情報を生成する場合、キューマネージャは、メッセージの MQMD 構造体の UserIdentifier フィールドに AlternateUserId の値を設定します。

代替ユーザ識別子は、実行中のアプリケーションのユーザ識別子の代わりとして使用されます。

MQOO\_ALTERNATE\_USER\_AUTHORITY も MQPMO\_ALTERNATE\_USER\_AUTHORITY も指定しなかった場合、このフィールドは無視されます。

これは入力用のフィールドです。

このフィールドの長さは、MQ\_USER\_ID\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、12 個の空白です。

### ● DynamicQName (MQCHAR48 型) 動的キュー名

MQOPEN 命令によって作成される動的キューの名前です。ObjectName フィールドにモデルキューの名前が指定された場合だけに関係します。その他の場合、DynamicQName フィールドは無視されます。

この名前でも有効な文字は ObjectName フィールドと同じです。さらに、アスタリスクも有効です。ObjectName フィールドがモデルキューの名前である場合は、すべて空白の名前（または最初のヌル文字の前が空白だけの名前）は無効です。

空白を除く名称の最後の文字がアスタリスク (\*) の場合、キューマネージャは、アスタリスクをユニークな文字列に置き換えます。これによって、ローカルキューマネージャ内でユニークなキュー名が保証されます。この方法で十分な個数の名称を確保するために、アスタリスクの位置は 1 文字目から 33 文字目の間にしてください。アスタリスクのあとには、空白・ヌル文字以外の文字を指定できません。

アスタリスクを先頭に指定してもかまいません。この場合、名称のすべての文字がキューマネージャによって生成されます。ObjectName フィールドにモデルキュー名を指定した場合、すべて空白の名称、または最初のヌル文字まで空白の名称は不正となります。

このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

これは、入力用のフィールドです。このフィールドの初期値は環境に依存します。TP1/Message Queue の場合の初期値は、C 言語では、ヌル文字で終わる 1 文字のアスタリスクです。そのほかのプログラミング言語では、余りを空白で埋められた 1 文字のアスタリスクです。

### ● InvalidDestCount (MQLONG 型) オープンに失敗したキューの数

配布リストの中で、オープンに失敗したキューの数です。このフィールドは、配布リストに存在しない単一のキューをオープンしたときにも設定されます。

InvalidDestCount フィールドは、MQOPEN 命令または MQPUT1 命令の CompCode 引数が MQCC\_OK または MQCC\_WARNING のときだけ設定されます。CompCode 引数が MQCC\_FAILED のときには設定されません。

これは出力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_2 以降の場合だけ有効です。

### ● KnownDestCount (MQLONG 型) ローカルキューの数

配布リストの中で、ローカルキューで受信され、オープンに成功したキューの数です。最初のメッセージ登録に使用される場合も、この値はリモートキューで受信されるキューを含みません。このフィールドは、配布リストには存在しない単一のキューをオープンしたときにも設定されます。

これは出力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_2 以降の場合だけ有効です。

### ● ObjectName (MQCHAR48 型) オブジェクト名

ObjectQMgrName のキューマネージャに定義されたオブジェクトのローカル名です。

この名前には、次に示す文字を使用できます。

- 英大文字 (A~Z)
- 英小文字 (a~z)
- 数字 (0~9)
- ピリオド (.), スラント (/), アンダスコア (\_), パーセント (%)

名前の先頭および中間には、空白を使用できません。末尾には空白を使用できます。名前の中で有効なデータの末尾を示すためにヌル文字を使用できます。ヌル文字とそれに続く文字はすべて空白として処理されます。

次に示す点は、該当するオブジェクトタイプに適用されます。

- ObjectName フィールドがモデルキューの名前である場合、キューマネージャは、モデルキューの属性で動的キューを生成し、生成されたキューの名前を ObjectName フィールドに返します。モデルキューは MQOPEN 命令にだけ指定できます。MQPUT1 命令では無効です。
- オープンされているオブジェクトが配布リストの場合 (つまり、RecsPresent フィールドがあり、ゼロより大きい場合)、ObjectName フィールドは空白またはヌル文字列にする必要があります。条件が満たされない場合、命令は失敗し、理由コード MQRC\_OBJECT\_NAME\_ERROR が返されます。
- ObjectType フィールドに MQOT\_Q\_MGR を指定したときは、最初のヌル文字までまたはフィールドの最後までを、空白で埋める必要があります。

このフィールドは、MQOPEN 命令でモデルキュー名を指定した場合は、入出力用のフィールドとなります。そのほかの場合は、入力専用のフィールドです。

このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

### ● ObjectQMgrName (MQCHAR48 型) オブジェクトキューマネージャ名

ObjectName のオブジェクトを定義している、キューマネージャの名称です。

この名前で有効な文字は、ObjectName フィールドと同じです。最初のヌル文字またはフィールドの最後まで、全体が空白である名前は、アプリケーションが接続されるキューマネージャ（ローカルキューマネージャ）を表します。

次に示す項目は、該当するオブジェクトタイプに適用されます。

- ObjectType フィールドが MQOT\_PROCESS または MQOT\_Q\_MGR である場合は、ObjectQMgrName フィールドは空白、またはローカルキューマネージャの名前にする必要があります。
- ObjectName フィールドにモデルキュー名を指定したときは、キューマネージャによってモデルキューの属性でキューが生成されます。このとき、生成されたキューを定義しているキューマネージャの名称が ObjectQMgrName フィールドに返されます。モデルキューは、MQOPEN 命令のときだけ指定できます。MQPUT1 命令では無効です。
- ObjectName がクラスタキューの名称で、かつ ObjectQMgrName が空白の場合、MQOPEN 命令で返却されたキューハンドルを使用して送信されたメッセージの実際のあて先はキューマネージャによって選択されます。
  - MQOO\_BIND\_ON\_OPEN が指定された場合、キューマネージャは MQOPEN 命令の処理でクラスタキューの中にある特定のキューを選択し、そのキューハンドルを使用して登録したすべてのメッセージはそのキューに送信されます。
  - MQOO\_BIND\_NOT\_FIXED が指定された場合、そのキューハンドルを使用した後続の MQPUT 命令では、クラスタ内の異なるキューマネージャ上に存在するあて先キューの中の異なるキューが選択される場合もあります。

アプリケーションが特定のクラスタキュー、つまり、クラスタ内の特定のキューマネージャ上に存在するキューにメッセージを送信する必要がある場合、アプリケーションは ObjectQMgrName フィールドに、そのキューマネージャ名称を指定する必要があります。これによって、ローカルキューマネージャは指定されたあて先キューマネージャにメッセージを送信します。

- 配布リストがオープンされている場合、つまり、RecsPresent が存在し、かつ 0 より大きい場合は、ObjectName は空白かヌル文字列でなければなりません。それ以外では、理由コード MQRC\_OBJECT\_Q\_MGR\_NAME\_ERROR で失敗します。

このフィールドは、MQOPEN 命令でモデルキュー名を指定した場合は、入出力用のフィールドとなります。そのほかの場合は、入力専用のフィールドです。

ObjectQMgrName フィールドの長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● ObjectRecOffset (MQLONG 型) 最初のオブジェクトレコードまでのオフセット

MQOD 構造体の先頭から最初の MQOR 構造体のオブジェクトレコードのバイトオフセットです。

オフセットの値は正の場合も負の場合もあります。ObjectRecOffset フィールドは、配布リストがオープンされているときだけ使用されます。RecsPresent フィールドが 0 であれば、無視されます。

配布リストがオープンされているとき、一つ以上の MQOR 構造体のオブジェクトレコードの配列が配布リストのあて先キューの名称を指定するために指定されなければなりません。次に示すどちらかの方法で指定してください。

- ObjectRecOffset フィールドを使用して指定する方法

アプリケーションは、必要な要素のある MQOR 構造体のレコード配列が続く MQOD 構造体を含む構造体を宣言しなければなりません。また、ObjectRecOffset フィールドを MQOD 構造体の開始からその配列の最初の要素までのオフセットに指定しなければなりません。このオフセットが正しく、値が MQLONG に収まることを確認してください（最も制限の強いプログラミング言語は COBOL 言語です。有効範囲は-999999999~+999999999 です）。

ObjectRecOffset フィールドは、ポインタのデータ型をサポートしていない、または異なる環境に移植しない場合にポインタ型のデータ型を実装するプログラム言語（例：COBOL 言語）でプログラミングするときに使用することをお勧めします。

- ObjectRecPtr フィールドを使用する方法

アプリケーションは、MQOR 構造体を MQOD 構造体とは別に宣言できます。その後、ObjectRecPtr フィールドを配列のアドレスに設定します。

ObjectRecPtr フィールドは、異なる環境に移植する場合にポインタのデータ型を実装するプログラム言語（例：C 言語）でプログラミングするときに使用することをお勧めします。

どちらの場合も、ObjectRecOffset フィールドまたは ObjectRecPtr フィールドのどちらか一つを使用しなければなりません。両方のフィールドが 0、または両方のフィールドとも 0 以外の場合、MQRC\_OBJECT\_RECORDS\_ERROR の理由コードで失敗します。

これは入力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

## ● ObjectRecPtr (MQPTR 型) 最初のオブジェクトレコードのアドレス

最初のオブジェクトレコードのアドレスです。ObjectRecPtr フィールドは配布リストがオープンされたときだけ使用されます。RecsPresent フィールドが 0 の場合は無視されます。

ObjectRecPtr フィールドまたは ObjectRecOffset フィールドのどちらかをオブジェクトレコードに指定するのに使用できます。指定方法については、「ObjectRecOffset フィールド」を参照してください。

ObjectRecPtr フィールドを使用しない場合は、ヌルポインタかヌル文字列を設定しなければなりません。

これは入力用のフィールドです。

このフィールドの初期値は、ポインタをサポートしているプログラミング言語ではヌルポインタです。文字列の場合はヌル文字列となります。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

## 注意事項

プログラミング言語がポインタのデータ型をサポートしていないオペレーティングシステムでは、このフィールドは、初期値がヌル文字列の適切な長さのバイト文字列として宣言されます。

### ● ObjectType (MQLONG 型) オブジェクトタイプ

ObjectName の名称を持つオブジェクトの種類です。次の値を指定できます。

MQOT\_Q：キュー

MQOT\_PROCESS：プロセス定義

MQOT\_Q\_MGR：キューマネージャ

これは入力用のフィールドです。

このフィールドの初期値は MQOT\_Q です。

### ● RecsPresent (MQLONG 型) オブジェクトレコード数

アプリケーションによって指定された MQOR 構造体のオブジェクトレコードの数です。この数が 0 より大きい場合、配布リストが、そのリスト中のあて先キューと同じ数だけ存在する RecsPresent フィールドに対してオープンされていることを示しています。ただし、配布リストがただ一つのあて先だけを含んでいる場合もあります。

RecsPresent フィールドの値は 0 より大きくなければなりません。また、その場合、ObjectType フィールドは MQOT\_Q でなければなりません。この条件が満たされない場合、MQRC\_RECS\_PRESENT\_ERROR の理由コードで失敗します。

これは入力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQGMO\_VERSION\_2 以降の場合だけ有効です。

### ● ResolvedQMgrName (MQCHAR48 型) 解決したキューマネージャ名称

ローカルキューマネージャに定義される最終のあて先キューマネージャの名称です。

ローカルキューマネージャが名称解決を実行したあとのあて先キューマネージャの名前です。返される名前は、ResolvedQName フィールドに指定されたキューを保持するキューマネージャの名前です。

ResolvedQMgrName フィールドは、ローカルキューマネージャの名前になることがあります。

空白でない値が返されるのは、オブジェクトが検索、入力または出力（またはこれらの組み合わせ）を目的にオープンされた単一のキューである場合だけです。オープンされているオブジェクトが次に示すどれかである場合、ResolvedQMgrName フィールドは空白に設定されます。

- キューではない場合

- キューであるが、オープンが検索、入力および出力のどれでもない場合
- MQOO\_BIND\_NOT\_FIXED が指定されたクラスタキュー（または、キューの DefBind 属性が MQBND\_BIND\_NOT\_FIXED のときの MQOO\_BIND\_AS\_Q\_DEF)
- 配布リストである場合

これは出力用のフィールドです。

このフィールドの長さは MQ\_Q\_NAME\_LENGTH です。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_3 以降の場合だけ有効です。

### ● ResolvedQName (MQCHAR48 型) 解決したキュー名称

ローカルキューマネージャに知らされる最終のあて先キューの名称です。

ローカルキューマネージャが名称解決を実行したあとのあて先キューの名前です。返される名前は、ResolvedQMmgrName フィールドに指定されたキューマネージャが保持するキューの名前です。

空白でない値が返されるのは、オブジェクトが検索、入力または出力（またはこれらの組み合わせ）を目的にオープンされた単一のキューである場合だけです。オープンされているオブジェクトが次に示すどれかである場合、ResolvedQName フィールドは空白に設定されます。

- キューではない場合
- キューであるが、オープンが検索、入力および出力のどれでもない場合
- 配布リストである場合

これは出力用のフィールドです。

このフィールドの長さは MQ\_Q\_NAME\_LENGTH です。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_3 以降の場合だけ有効です。

### ● ResponseRecOffset (MQLONG 型) 最初の応答レコードまでのオフセット

MQOD 構造体の先頭から最初の MQRR 構造体の応答レコードまでのバイトオフセットです。オフセットは正の場合も負の場合もあります。ResponseRecOffset フィールドは、配布リストがオープンされているときだけ使用できます。RecsPresent フィールドが 0 の場合は無視されます。

配布リストがオープンされているとき、一つ以上の MQRR 構造体の応答レコードの配列がオープンに失敗したキューと、その失敗の理由を特定するために指定されなければなりません。なお、オープンに失敗

したキューは MQRR 構造体の CompCode フィールドに記述され、その失敗の理由は MQRR 構造体の Reason フィールドに記述されます。

データは、キュー名称がオブジェクトレコードの配列に現れるのと同じ順序で応答レコードの配列に返されます。キューマネージャが応答レコードを設定するのは、命令の結果が混在しているときだけです。例えば、次のような場合です。

- あるキューはオープンに成功しているが、ほかのキューが失敗していた場合
- すべてのキューが失敗し、その理由が異なる場合

この場合、MQRC\_MULTIPLE\_REASONS の理由コードが出現します。同じ理由コードがすべてのキューに適用される場合、応答レコードは設定されません。応答レコードは任意で指定できます。ただし、指定する場合は、RecsPresent フィールドと同じ数だけ指定しなければなりません。

応答レコードはオブジェクトレコードと同じように、ResponseRecOffset フィールドのオフセットまたは ResponseRecPtr のアドレスを指定できます。指定方法については、「ObjectRecOffset フィールド」を参照してください。ただし、ResponseRecOffset フィールドと ResponseRecPtr フィールドは一つ以上指定できません。どちらのフィールドも 0 以外の場合は、MQRC\_RESPONSE\_RECORDS\_ERROR の理由コードで失敗します。

キューをオープンしたときのエラーと同様に、応答レコードは、MQPUT1 命令でメッセージが配布リストのキューに送信されるときに発生したエラーに関する情報を返すのに使用されます。キューに対するオープン処理の完了コードが MQCC\_OK または MQCC\_WARNING の場合だけ、キューに対する登録操作の完了コードと理由コードに置き換えます。

これは入力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_2 以降の場合だけ有効です。

## ● ResponseRecPtr (MQPTR 型) 最初の応答レコードのアドレス

最初の応答レコードのアドレスです。このフィールドは、配布リストがオープンされたときだけ使用されます。RecsPresent フィールドが 0 の場合は無視されます。

応答レコードの指定には、ResponseRecPtr フィールドまたは ResponseRecOffset フィールドを同時に使用できません。詳細については、「ResponseRecOffset フィールド」を参照してください。

ResponseRecPtr フィールドが使用されない場合、ヌルポインタかヌル文字列を設定しなければなりません。

これは入力用のフィールドです。

このフィールドの初期値は、ポインタをサポートしているプログラミング言語ではヌルポインタです。文字列の場合はヌル文字列です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_2 以降の場合だけ有効です。

## 注意事項

プログラミング言語がポインタのデータ型をサポートしていないオペレーティングシステムでは、このフィールドは、初期値がすべてヌル文字列の、適切な長さのバイト文字列として宣言されます。

### ● StruId (MQCHAR4 型) 構造体識別子

次の値を取ります。

#### MQOD\_STRUC\_ID

オブジェクト記述子の構造体識別子

C 言語では、MQOD\_STRUC\_ID\_ARRAY も定義されています。これは、MQOD\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

これは入力用のフィールドです。

このフィールドの初期値は MQOD\_STRUC\_ID です。

### ● UnknownDestCount (MQLONG 型) リモートキューの数

配布リストの中で、リモートキューで受信され、オープンに成功したキューの数です。このフィールドは、配布リストには存在しない単一のキューをオープンしたときにも設定されます。

これは出力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQOD\_VERSION\_2 以降の場合だけ有効です。

### ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

#### MQOD\_VERSION\_1

バージョン 1 のオブジェクト記述子の構造体バージョン番号

#### MQOD\_VERSION\_2

バージョン 2 のオブジェクト記述子の構造体バージョン番号

#### MQOD\_VERSION\_3

バージョン 3 のオブジェクト記述子の構造体バージョン番号

MQOD\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことがフィールドの説明に記載されています。次に示す定数は、現在のバージョン番号を指定します。

#### MQOD\_CURRENT\_VERSION

現在のバージョンのオブジェクト記述子の構造体バージョン番号

これは入力用のフィールドです。

このフィールドの初期値は MQOD\_VERSION\_1 です。

# MQOR 構造体 – オブジェクトレコード

MQOR 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
ObjectName (MQCHAR48 型)	オブジェクト名	ヌル文字列/空白
ObjectQMgrName (MQCHAR48 型)	オブジェクトキューマネージャ名	ヌル文字列/空白

## 概要

### 目的

MQOR 構造体は、単一のあて先キューのキュー名およびキューマネージャ名を指定するのに使用されます。MQOR 構造体は、MQOPEN および MQPUT1 命令の入力用の構造体です。

### 文字セットおよびマシンコード

MQOR 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。TP1/Message Queue Access のクライアントアプリケーションの場合、MQOD 構造体はクライアントの文字セットとマシンコードに従います。

### 使用方法

MQOPEN 命令に構造体の配列を提供することによって、キューのリストをオープンできます。このリストを配布リストといいます。キューのオープンに成功し、MQOPEN 命令で返されたキューハンドルを使用してメッセージを登録すると、各メッセージはリストの各キューに格納されます。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● ObjectName (MQCHAR48 型) オブジェクト名

次に示す項目を除いて、MQOD 構造体の ObjectName フィールドと同じです。

- キュー名である。
- モデルキュー名ではない。

これは常に入力用のフィールドです。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかの言語では、48 個の空白です。

### ● ObjectQMgrName (MQCHAR48 型) オブジェクトキューマネージャ名

MQOD 構造体の ObjectQMgrName フィールドと同じです。

これは常に入力用のフィールドです。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかの言語では、48 個の空白です。

# MQPMO 構造体 — メッセージ登録オプション

MQPMO 構造体は、MQPUT 命令および MQPUT1 命令の入出力用の引数です。次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQPMO_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQPMO_VERSION_1
Options (MQLONG 型)	登録オプション	MQPMO_NONE
Timeout (MQLONG 型)	(予備)	-1
Context (MQHOBJ 型)	コンテキスト	0
KnownDestCount (MQLONG 型)	ローカルキューの数	0
UnknownDestCount (MQLONG 型)	リモートキューの数	0
InvalidDestCount (MQLONG 型)	送信に失敗したキューの数	0
ResolvedQName (MQCHAR48 型)	受信キュー名	ヌル文字列/空白
ResolvedQMgrName (MQCHAR48 型)	受信キューマネージャ名	ヌル文字列/空白
RecsPresent (MQLONG 型) ※	登録メッセージレコード数または応答レコード数	0
PutMsgRecFields (MQLONG 型) ※	MQPMR 構造体フィールドの存在	MQPMRF_NONE
PutMsgRecOffset (MQLONG 型) ※	最初の登録メッセージレコードまでのオフセット	0
ResponseRecOffset (MQLONG 型) ※	最初の応答レコードまでのオフセット	0
PutMsgRecPtr (MQPTR 型) ※	最初の登録メッセージレコードのアドレス	ヌルポインタ/ヌル文字列
ResponseRecPtr (MQPTR 型) ※	最初の応答レコードのアドレス	ヌルポインタ/ヌル文字列

注※

Version フィールドの値が MQPMO\_VERSION\_2 以降の場合だけ存在するフィールドです。

## 概要

### 目的

MQPMO 構造体によって、アプリケーションはメッセージのキューへの格納方法を制御できます。この構造体は、MQPUT および MQPUT1 命令の入出力用の引数です。

## バージョン

MQPMO 構造体のバージョンには 1 および 2 があります。しかし、すべての環境で MQPMO\_VERSION\_2 がサポートされているわけではありません。複数の環境にアプリケーションを移植する場合は、使用するバージョンの MQPMO 構造体が、すべての環境でサポートされている必要があります。MQPMO\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことをフィールドの説明に記載しています。

サポートするプログラミング言語の COPY ファイルと INCLUDE ファイルでは、最新バージョンの MQPMO 構造体が提供されます。しかし、Version フィールドの初期値は MQPMO\_VERSION\_1 です。バージョン 1 の構造体がないフィールドを使用する場合は、アプリケーションで使用したいバージョンの番号を Version フィールドに設定してください。

## 文字セットおよびマシンコード

MQPMO 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。TP1/Message Queue Access のクライアントアプリケーションの場合、MQPMO 構造体はクライアントの文字セットとマシンコードに従います。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● Context (MQHOBJ 型) コンテキスト

入力キューのオブジェクトハンドルです。

MQPMO\_PASS\_IDENTITY\_CONTEXT または MQPMO\_PASS\_ALL\_CONTEXT を指定した場合、このフィールドに入力キューハンドルを指定します。登録するメッセージのコンテキスト情報は、そのキューハンドルから取得されます。

MQPMO\_PASS\_IDENTITY\_CONTEXT も MQPMO\_PASS\_ALL\_CONTEXT も指定しなかった場合、このフィールドは無視されます。

これは入力用のフィールドです。

このフィールドの初期値は 0 です。

### ● InvalidDestCount (MQLONG 型) 送信に失敗したキューの数

配布リスト中のキューの中で、送信できなかったメッセージの数です。オープンに成功したが登録に失敗したキューや、オープンに失敗したキューの数も含まれます。このフィールドは配布リストではない単一のキューへの登録時にも設定されます。

## 注意事項

このフィールドは、MQPUT 命令または MQPUT1 命令の CompCode 引数が MQCC\_OK または MQCC\_WARNING の場合だけ設定されます。CompCode 引数が MQCC\_FAILED の場合は設定されません。

これは出力用のフィールドです。

このフィールドの初期値は 0 です。

なお、このフィールドは、Version フィールドが MQPMO\_VERSION\_2 より前の場合は設定されません。

### ● KnownDestCount (MQLONG 型) ローカルキューの数

配布リストの中で、MQPUT 命令または MQPUT1 命令による、ローカルキューであるキューへの送信が成功したキューの数を示しています。ローカルな転送キューが最初のメッセージ登録に使用された場合も、値はリモートキューで受信されるキューは含みません。このフィールドは配布リストではない単一のキューへの登録時にも設定されます。

これは出力用のフィールドです。

このフィールドの初期値は 0 です。

なお、このフィールドは、Version フィールドが MQPMO\_VERSION\_2 より前の場合は設定されません。

### ● Options (MQLONG 型) 登録オプション

MQPUT 命令および MQPUT1 命令の動作を制御します。

指定できるオプションを説明します。なお、指定しなくてもかまいません。二つ以上の値を指定する場合は、次のどちらかに従ってください。

- それぞれを加算します。ただし、同じ値は 2 回以上加算しないでください。
- プログラミング言語がビット演算をサポートしている場合に、それぞれのビット論理和を取ります。

同時に指定できない組み合わせは、そのつど示します。それ以外の組み合わせは自由です。オプションおよびその指定方法の一覧については、「付録 A 命令および引数の一覧」を参照してください。

同期点オプション：

MQPUT または MQPUT1 命令のトランザクションへの参加に関連するオプションを次に示します。

#### MQPMO\_SYNCPOINT

同期点でメッセージを登録します。

通常のトランザクションの範囲内で登録します。トランザクションがコミットするまでは、トランザクション内から登録したメッセージは有効になりません。トランザクションがロールバックすると、メッセージは取り消されます。

このオプションは、ローカルキューマネージャが同期点処理をサポートしているときだけ指定できます。同期点処理をサポートしていないキューマネージャで MQPMO\_SYNCPOINT を指定した場合、完了コード MQCC\_FAILED、理由コード MQRC\_SYNCPOINT\_NOT\_AVAILABLE で命令は終了します。同期点の取得とトランザクションについては、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。

このオプションと MQPMO\_NO\_SYNCPOINT オプションのどちらも指定しなかった場合、登録命令がトランザクション内に含まれるかどうかは、環境に依存します。TP1/Message Queue では、この命令をトランザクション内から呼び出したかどうかで、省略時にトランザクション内に含まれるかが決まります。省略時の動作が環境によって異なるため、移植を考慮するアプリケーションでは、オプションを省略しないでください。つまり、MQPMO\_SYNCPOINT オプションか MQPMO\_NO\_SYNCPOINT オプションを、明確に指定してください。

MQPMO\_NO\_SYNCPOINT オプションと同時に MQPMO\_SYNCPOINT オプションを指定できません。

## MQPMO\_NO\_SYNCPOINT

同期点の処理なしでメッセージを登録します。

通常のトランザクションの範囲外で登録します。メッセージは登録直後から有効になります。トランザクションがロールバックしても、メッセージは取り消されません。

このオプションと MQPMO\_SYNCPOINT オプションのどちらも指定しなかった場合、登録命令がトランザクション内に含まれるかどうかは、環境に依存します。TP1/Message Queue では、この命令をトランザクション内から呼び出したかどうかで、省略時にトランザクション内に含まれるかが決まります。省略時の動作が環境によって異なるため、移植を考慮するアプリケーションでは、オプションを省略しないでください。つまり、MQPMO\_SYNCPOINT オプションか MQPMO\_NO\_SYNCPOINT オプションのどちらかを、明確に指定してください。

MQPMO\_NO\_SYNCPOINT オプションと同時に MQPMO\_SYNCPOINT オプションを指定できません。

同期点のオプションとキューマネージャの SyncPoint 属性の関係を、次の表に示します。

キューマネージャの SyncPoint 属性	MQPUT/MQPUT1 命令の呼び出し位置	MQPUT/MQPUT1 命令の同期点のオプション		
		MQPMO_SYNCPOINT	MQPMO_NO_SYNCPOINT	省略
MQSP_AVAILABLE	トランザクション内	1	2	1
	トランザクション外	×	2	2
MQSP_NOT_AVAILABLE	トランザクション内	×	2	×
	トランザクション外	×	2	2

(凡例)

- 1：命令はトランザクション内に含まれます。
- 2：命令はトランザクション内に含まれません。
- ×：命令は理由コード MQRC\_SYNCPOINT\_NOT\_AVAILABLE で失敗します。

## メッセージ識別子および関連識別子オプション：

キューマネージャに新しいメッセージ識別子または関連識別子を生成させるオプションを次に示します。

### MQPMO\_NEW\_MSG\_ID

新しいメッセージ識別子を生成します。

MQMD 構造体の MsgId フィールドの内容を新しいメッセージ識別子に置き換えるように、キューマネージャに指定します。メッセージ識別子はメッセージとともに送信され、MQPUT 命令または MQPUT1 命令の出力でアプリケーションに返されます。

このオプションは、メッセージが配布リストに登録されるときにも指定できます。この章の「[MQPMR 構造体 - 登録メッセージレコード](#)」の MsgId フィールド参照してください。

MQPMO\_NEW\_MSG\_ID を指定すると、アプリケーションは MQPUT 命令または MQPUT1 命令に先立って、MsgId フィールドを MQML\_NONE にリセットする必要がなくなります。

### MQPMO\_NEW\_CORREL\_ID

新しい関連識別子を生成します。

キューマネージャに MQMD 構造体の CorrelId フィールドの内容を、新しい関連識別子に置き換えるよう指定できます。関連識別子はメッセージとともに送信され、MQPUT 命令または MQPUT1 命令の出力でアプリケーションに返されます。

このオプションは、メッセージが配布リストに登録されるときにも指定できます。この章の「[MQPMR 構造体 - 登録メッセージレコード](#)」で CorrelId フィールドを参照してください。

MQPMO\_NEW\_CORREL\_ID オプションは、アプリケーションがユニークな関連識別子を必要とする場合に有効です。

## グループとセグメントに関するオプション：

グループ内のメッセージおよび論理メッセージのセグメントの処理に関連するオプションを次に示します。オプションを理解するために、次の用語について説明します。

### 物理メッセージ

キューに登録したり取り出したりすることのできる最も小さい情報単位です。通常、単一の MQPUT 命令、MQPUT1 命令または MQGET 命令で指定したり取り出したりする情報に該当します。

それぞれの物理メッセージはメッセージ記述子 (MQMD 構造体) を持っています。

キューマネージャによって定義されているわけではありませんが、一般的に、物理メッセージはメッセージ記述子 (MQMD 構造体の MsgId フィールド) の値で区別されます。

### 論理メッセージ

アプリケーション情報の一単位です。システムの強制がない場合、論理メッセージは物理メッセージと同じである場合があります。ただし、論理メッセージが極端に大きい場合、論理メッセージを二つまたはそれ以上のセグメントと呼ばれる物理メッセージに分割することをお勧めします。また、分割が必要となる場合もあります。

セグメント分割された論理メッセージは、同一のヌルではないグループ記述子 (MQMD 構造体の GroupId フィールド) と同一のメッセージシーケンス番号 (MQMD 構造体の MsgSeqNumber フィー

ルド) を持つ二つ以上の物理メッセージによって構成されます。セグメントはセグメントオフセット (MQMD 構造体の Offset フィールド) の値で区別されます。セグメントオフセットは、論理メッセージのデータ先頭からの、物理メッセージのデータの位置を表します。それぞれのセグメントは物理メッセージであるため、論理メッセージのセグメントは、通常異なるメッセージ記述子を持ちます。

セグメント分割はされていないメッセージのうち、送信アプリケーションによってセグメント分割が許可されている論理メッセージも、ヌルではないグループ記述子を持ちます。この場合、論理メッセージがメッセージグループに属していないときは、そのグループ記述子を持つ物理メッセージはただ一つしか存在しないことになります。送信アプリケーションによってセグメント分割が禁止されている論理メッセージがメッセージグループに属していない場合は、ヌルのグループ記述子 (MQGI\_NONE) を持っています。

## メッセージグループ

ヌルではない同一のグループ記述子を持つ一つ以上の物理メッセージの集まりです。

グループ内の論理メッセージは、メッセージシーケンス番号の値で区別されます。メッセージシーケンス番号は、1~N の整数範囲となります。なお、N はグループ内の論理メッセージの数を表します。一つ以上の論理メッセージがセグメント分割されている場合、グループ内には N 個以上の物理メッセージが存在します。

次に、メッセージが、MQGET 命令で返却される方法を制御する個々のオプションについて説明します。

## MQPMO\_LOGICAL\_ORDER

グループ内のメッセージと論理メッセージのセグメントを論理的な順序で登録します。

このオプションを指定すると、アプリケーションがキューマネージャにグループ内のメッセージと論理メッセージのセグメントの登録方法を通知します。MQPUT 命令でだけ指定できます。MQPUT1 命令で指定すると不正となります。

MQPMO\_LOGICAL\_ORDER が指定されると、アプリケーションは MQPUT 命令を次のように発行します。

- 各論理メッセージのセグメントを、セグメントオフセットが 0 から空きがなくなるまで、昇順で登録します。
- 論理メッセージの全セグメントを、次の論理メッセージのセグメント登録の前に登録します。
- 各メッセージグループの論理メッセージを、メッセージシーケンス番号が 1 から空きがなくなるまで、昇順で登録します。
- メッセージグループの全論理メッセージを次のメッセージグループの論理メッセージ登録の前に登録します。

上記の順序を論理的順序といいます。

アプリケーションは、キューマネージャにグループ内のメッセージと論理メッセージのセグメントの登録方法を通知しているため、おのこの MQPUT 命令でグループとセグメントの情報を保持したり、更新したりする必要がありません。特に、MQMD 構造体の GroupId フィールド、MsgSeqNumber フィールド、および Offset フィールドは、キューマネージャが適当な値に設定するため、アプリケーションは設定する必要がありません。アプリケーションは、次に示す内容を指示するため、MQMD 構造体の MsgFlags フィールドだけ指定する必要があります。

- メッセージがグループに属する，または論理メッセージのセグメントであることを指示する
- グループ内の最終メッセージ，または論理メッセージの最終セグメントを指示する

メッセージグループまたは論理メッセージが開始すると，次の MQPUT 命令は MQMD 構造体の MsgFlags フィールドに適切な MQMF\_\*フラグを指定する必要があります。アプリケーションが，メッセージグループが終了していないときにグループに属さないメッセージを登録しようとしたり，論理メッセージが終了していないときにセグメントではないメッセージを登録しようとした場合，命令は MQRC\_INCOMPLETE\_GROUP または MQRC\_INCOMPLETE\_MSG の理由コードで失敗します。

ただし，キューマネージャは，現在のメッセージグループおよび現在の論理メッセージに関する情報を保持しています。アプリケーションは MQMF\_LAST\_MSG\_IN\_GROUP または MQMF\_LAST\_SEGMENT を指定してメッセージを送信することで，MQPUT 命令を再発行する前にグループに属していない，かつセグメントでもないメッセージを登録するために保持されている情報を終了できます。ただし，この場合アプリケーションが送信するメッセージには，メッセージデータがないときもあります。

オプションとフラグの正しい組み合わせ，およびキューマネージャが使用する GroupId フィールド，MsgSeqNumber フィールド，および Offset フィールドの値を次の表に示します。表に示されていないオプションとフラグの組み合わせは誤りです。

指定するオプション				命令に先行するグループと論理メッセージの状態		キューマネージャが使用する値		
MQPMO LOGICAL_ORDER	MQMF_ MSG_ IN_ GROUP または MQMF_ LAST_ MSG_ IN_ GROUP	MQMF_ SEG MENT または MQMF_ LAST_ SEG MENT	MQMF_ SEG MENT ATION_ ALLO WED	命令に先行する現在のメッセージグループがある	命令に先行する現在の論理メッセージがある	Group Id	MsgSeq Number	Offset
○	×	×	×	×	×	MQGL_NONE	1	0
○	×	×	○	×	×	新グループ識別子	1	0
○	×	○	—	×	×	新グループ識別子	1	0
○	×	○	—	×	○	前グループ識別子	1	前オフセット値と前セグメント長の和
○	○	—	—	×	×	新グループ識別子	1	0

指定するオプション				命令に先行するグループと論理メッセージの状態		キューマネージャが使用する値		
MQPMO - LOG ICAL_ ORDER	MQMF_ MSG_ IN_ GROUP または MQMF_ LAST_ MSG_ IN_ GROUP	MQMF_ SEG MENT または MQMF_ LAST_ SEG MENT	MQMF_ SEG MENT ATION_ ALLO WED	命令に先行する現在のメッセージグループがある	命令に先行する現在の論理メッセージがある	Group Id	MsgSeq Number	Offset
○	○	-	-	○	×	前グループ識別子	前シーケンス番号に1を加算	0
○	○	○	-	○	○	前グループ識別子	前シーケンス番号	前オフセット値と前セグメント長の和
×	×	×	×	-	-	MQGL_ NONE	1	0
×	×	×	○	-	-	<ul style="list-style-type: none"> <li>MQGL_ NONE の場合： 新グループ識別子</li> <li>それ以外の場合： フィールドの値</li> </ul>	1	0
×	×	○	-	-	-	<ul style="list-style-type: none"> <li>MQGL_ NONE の場合： 新グループ識別子</li> <li>それ以外の場合： フィールドの値</li> </ul>	1	フィールドの値
×	○	×	-	-	-	<ul style="list-style-type: none"> <li>MQGL_ NONE の場合： 新グループ識別子</li> </ul>	フィールドの値	0

指定するオプション				命令に先行するグループと論理メッセージの状態		キューマネージャが使用する値		
MQPMO _LOGICAL_ORDER	MQMF_ MSG_ IN_ GROUP または MQMF_ LAST_ MSG_ IN_ GROUP	MQMF_ SEG MENT または MQMF_ LAST_ SEG MENT	MQMF_ SEG MENT ATION_ ALLO WED	命令に先行する現在のメッセージグループがある	命令に先行する現在の論理メッセージがある	Group Id	MsgSeq Number	Offset
×	○	×	—	—	—	<ul style="list-style-type: none"> <li>それ以外の場合：フィールドの値</li> </ul>	フィールドの値	0
×	○	○	—	—	—	<ul style="list-style-type: none"> <li>MQGL_NONEの場合：新グループ識別子</li> <li>それ以外の場合：フィールドの値</li> </ul>	フィールドの値	フィールドの値

(凡例)

- ：該当します。
- ×
- ：有無に関係ありません。

## 注意事項

MQPMO\_LOGICAL\_ORDER は、MQPUT1 命令に対しては無効です。

MsgId フィールドに対して、MQPMO\_NEW\_MSG\_ID または MQMI\_NONE が指定されたとき、キューマネージャは新しいメッセージ識別子を生成します。その場合は、ほかのフィールドの値を使用します。

CorrelId フィールドに対して MQPMO\_NEW\_CORREL\_ID が指定されたとき、キューマネージャは新しい相関識別子を生成します。その場合は、ほかのフィールドの値を使用します。

MQPMO\_LOGICAL\_ORDER が指定されたとき、グループ内の全メッセージと論理メッセージの全セグメントの MQMD 構造体の Persistence フィールドに、同一の値を指定してキューマネージャに登録しなければなりません。つまり、すべてが永続的であるか、または非永続であるかのどちらかでなければなりません。

ばなりません。この条件が満たされない場合、MQPUT 命令は理由コード MQRC\_INCONSISTENT\_PERSISTENCE で失敗します。

MQPMO\_LOGICAL\_ORDER オプションが同期点に及ぼす影響を次に示します。

- グループの最初の論理メッセージまたはセグメントがトランザクション内で登録された場合、同じキューハンドルを使うときは、グループ内にあるほかのすべての論理メッセージとセグメントはトランザクション内で登録されなければなりません。ただし、メッセージが同じトランザクション内で登録される必要はありません。したがって、多くの物理メッセージで構成されるメッセージグループは、キューハンドルで二つ以上の連続したトランザクションに分割することが可能になります。
- グループの最初の論理メッセージ、またはセグメントがトランザクション内で登録されなかった場合、同じキューハンドルを使うときは、グループ内のほかの論理メッセージとセグメントを同じトランザクション内で登録することはできません。

上記の条件を満たさなかった場合、MQPUT 命令は MQRC\_INCONSISTENT\_UOW の理由コードで失敗します。

MQPMO\_LOGICAL\_ORDER が指定されたとき、MQPUT 命令で指定された MQMD 構造体は MQMD\_VERSION\_2 より前のバージョンではいけません。この条件が満足されない場合は、MQRC\_WRONG\_MD\_VERSION の理由コードで失敗します。

MQPUT 命令に MQPMO\_LOGICAL\_ORDER を指定しない場合、グループのメッセージおよび論理メッセージのセグメントは、どのような順序で登録してもかまいません。また、完全なメッセージグループや完全な論理メッセージを登録する必要もありません。GroupId フィールド、MsgSeqNumber フィールド、Offset フィールド、および MsgFlags フィールドには、アプリケーションによって正しい値が設定されます。

システムに障害が起きた場合、メッセージグループや論理メッセージを途中から再開する方法を示します。システム再開時、アプリケーションは GroupId フィールド、MsgSeqNumber フィールド、Offset フィールド、MsgFlags フィールド、および Persistence フィールドに適切な値を設定します。その後、MQPMO\_SYNCPOINT または MQPMO\_NO\_SYNCPOINT に適切な値を設定します。すると、MQPMO\_LOGICAL\_ORDER を指定しないで MQPUT 命令を発行できます。この MQPUT 命令が成功したとき、キューマネージャは、グループとセグメントの情報を保持します。そのキューハンドルを使用した後続の MQPUT 命令では、通常の MQPMO\_LOGICAL\_ORDER を指定できます。

キューマネージャが MQPUT 命令用に保持しているグループ情報およびセグメント情報は、MQGET 命令用に保持しているグループ情報およびセグメント情報とは別です。

どのキューハンドルについても、アプリケーションでは MQPMO\_LOGICAL\_ORDER を指定した MQPUT 命令と指定しない MQPUT 命令を混在してもかまいませんが、次の点に注意してください。

- MQPMO\_LOGICAL\_ORDER を指定していない場合、MQPUT 命令が成功するたびに、キューマネージャはアプリケーションに指定された値を、キューハンドルのグループおよびセグメント情報に設定します。その情報で、キューハンドルに対してキューマネージャが保持していたグループとセグメントの情報を置き換えます。
- MQPMO\_LOGICAL\_ORDER が指定されない場合、現在のメッセージグループまたは論理メッセージが存在するとき、その命令は失敗しません。命令は MQCC\_WARNING 完了コードで成功します。MQPUT 命令と MQCLOSE 命令が、グループ情報およびセグメント情報で矛盾している場合の結果を次の表に示します。

実行時の命令	直前の命令が MQPMO_LOGICAL_ORDER 指定ありの MQPUT 命令	直前の命令が MQPMO_LOGICAL_ORDER 指定なしの MQPUT 命令
MQPMO_LOGICAL_ORDER 指定ありの MQPUT 命令	MQCC_FAILED	MQCC_FAILED
MQPMO_LOGICAL_ORDER 指定なしの MQPUT 命令	MQCC_WARNING	MQCC_OK
非永続グループと論理メッセージありの MQCLOSE 命令	MQCC_WARNING	MQCC_OK

上記の場合で完了コードが MQCC\_OK ではないときは、理由コードは次のどれかになります。

MQRC\_INCOMPLETE\_GROUP  
MQRC\_INCOMPLETE\_MSG  
MQRC\_INCONSISTENT\_PERSISTENCE  
MQRC\_INCONSISTENT\_UOW

### 注意事項

キューマネージャは、MQPUT1 命令でグループとセグメントの情報をチェックしません。

論理的な順序でメッセージとセグメントを単純に登録したいアプリケーションでは、MQPMO\_LOGICAL\_ORDER を指定することをお勧めします。このオプションによってキューマネージャがグループとセグメントの情報を管理することを指定できるため、アプリケーションはそれらの情報を管理する必要はありません。しかし、アプリケーションによっては、MQPMO\_LOGICAL\_ORDER オプションが提供する機能より高い制御機能を必要とする場合があります。その場合、このオプションを指定しないでください。MQPMO\_LOGICAL\_ORDER を指定しなかったとき、アプリケーションは、MQPUT 命令または MQPUT1 命令に先立って、MQMD 構造体の GroupId フィールド、MsgSeqNumber フィールド、Offset フィールド、および MsgFlags フィールドを正しく設定しなければなりません。

例えば、受信した物理メッセージをグループまたは論理メッセージのセグメントに属するかどうか関係なく転送するアプリケーションには、次に示す理由のため、MQPMO\_LOGICAL\_ORDER を指定しないでください。

- メッセージが順番に取り出されたり登録されたりする場合、MQPMO\_LOGICAL\_ORDER を指定すると、新しいグループ識別子がメッセージに割り当てられます。その結果、メッセージの生成元がメッセージグループの結果の応答や報告メッセージを関連づけることが困難になります。
- 送信側と受信側のキューマネージャ間に複数の通信路がある複雑なネットワークの場合、物理メッセージは順序どおりに到着しません。MQPMO\_LOGICAL\_ORDER および MQGET 命令で対応する MQGMO\_LOGICAL\_ORDER を指定しないことで、転送するアプリケーションは各メッセージが到着すると、次の論理的な順序のメッセージを待たないで、そのメッセージを取り出したり転送したりできます。

グループのメッセージや、論理メッセージのセグメントに対する報告メッセージを生成するアプリケーションにも、MQPMO\_LOGICAL\_ORDER を指定しないでください。

MQPMO\_LOGICAL\_ORDER は、ほかの MQPMO\_\*オプションのどれとも同時に指定できます。

## コンテキストオプション：

メッセージコンテキストの処理を制御するオプションを次に示します。

### MQPMO\_NO\_CONTEXT

メッセージコンテキストを使用しません。

識別コンテキストも登録元コンテキストも使用しません。文字フィールドは空白、バイトフィールドはヌル、数値フィールドは0となります。

### MQPMO\_DEFAULT\_CONTEXT

省略時のメッセージコンテキストを使用します。

識別コンテキストも登録元コンテキストも、省略時の仮定値を使用します。キューマネージャは、メッセージ記述子のコンテキストフィールドを次のように設定します。

MQMD 構造体のフィールド	値
UserIdentifier	"OPENTP1"または AlternateUserId (環境に依存する値)
AccountingToken	MQACT_NONE (環境に依存する値)
ApplIdentityData	空白
PutApplType	MQAT_OPEN_TP1 (環境に依存する値)
PutApplName	ユーザサーバ名 (SUP 名, SPP 名) (環境に依存する値)
PutDate	メッセージを登録した日付
PutTime	メッセージを登録した時刻
ApplOriginData	空白

メッセージコンテキストおよびコンテキスト情報の制御については、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。

このほかのコンテキストオプションを指定しなかった場合、このオプションが仮定されます。

### MQPMO\_PASS\_IDENTITY\_CONTEXT

入力キューハンドルの識別コンテキストを引き渡します。

関連するコンテキスト情報をメッセージに使用します。識別コンテキストとして、Context フィールドに指定したキューハンドルの情報を使用します。登録元コンテキストは、キューマネージャによって、MQPMO\_DEFAULT\_CONTEXT と同様の方法で設定されます。メッセージコンテキストおよびコンテキスト情報の制御については、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。

MQPUT 命令では、MQOO\_PASS\_IDENTITY\_CONTEXT またはこれに相当するオプションを指定してキューをオープンしておく必要があります。

### MQPMO\_PASS\_ALL\_CONTEXT

入力キューハンドルの全コンテキストを引き渡します。

関連するコンテキスト情報をメッセージに使用します。識別コンテキストおよび登録元コンテキストとして、Context フィールドに指定したキューハンドルの情報を使用します。メッセージコンテキストおよびコンテキスト情報の制御については、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。

MQPUT 命令では、MQOO\_PASS\_ALL\_CONTEXT またはこれに相当するオプションを指定してキューをオープンしておく必要があります。

### MQPMO\_SET\_IDENTITY\_CONTEXT

アプリケーションが識別コンテキストを設定します。

関連するコンテキスト情報をメッセージに使用します。識別コンテキストとして、アプリケーションが指定した MQMD 構造体の情報を設定します。登録元コンテキストは、キューマネージャによって、MQPMO\_DEFAULT\_CONTEXT と同様の方法で設定されます。メッセージコンテキストおよびコンテキスト情報の制御については、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。

MQPUT 命令では、MQOO\_SET\_IDENTITY\_CONTEXT またはこれに相当するオプションを指定してキューをオープンしておく必要があります。

### MQPMO\_SET\_ALL\_CONTEXT

アプリケーションが全コンテキストを設定します。

関連するコンテキスト情報をメッセージに使用します。識別コンテキストおよび登録元コンテキストとして、アプリケーションが指定した MQMD 構造体の情報を設定します。メッセージコンテキストおよびコンテキスト情報の制御については、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。

MQPUT 命令では、MQOO\_SET\_ALL\_CONTEXT を指定してキューをオープンしておく必要があります。

MQPMO\*\_CONTEXT オプションは、一つだけ指定できます。これらのどれも指定しなかった場合、MQPMO\_DEFAULT\_CONTEXT オプションが仮定されます。

### その他のオプション：

権限チェックおよびキューマネージャ停止時の動作を制御するオプションを次に示します。

### MQPMO\_ALTERNATE\_USER\_AUTHORITY

代替ユーザ識別子を使用します。

キューを利用するユーザの識別子として、アプリケーションを実行しているユーザ識別子ではなく、MQPUT1 命令の ObjDesc 引数の AlternateUserId の代替ユーザ識別子が使用されます。

これは、MQPUT1 命令のときだけ指定できます。

### MQPMO\_FAIL\_IF QUIESCING

TP1/Message Queue では意味がありません。

このオプションを指定しても、TP1/Message Queue の動作には影響しません。これは、アプリケーションの移植性を向上させるために定義されています。

## デフォルトオプション：

ここまでで説明されたオプションを指定しない場合、次に示すオプションを使用できます。

### MQPMO\_NONE

オプションを指定しません。

このオプションは、ほかにオプションを指定しない場合に指定します。このほかのオプションには、どれも省略時の仮定値があります。MQPMO\_NONE オプションは、プログラムのコーディングをわかりやすくするために定義されています。値としては0が定義されていますが、コーディングをわかりやすくする以外の目的はありません。

このフィールドの初期値はMQPMO\_NONEです。

### ● PutMsgRecFields (MQLONG 型) MQPMR 構造体フィールドの存在

アプリケーションによって指定された登録メッセージレコードに、どんなMQPMR 構造体フィールドが存在するか指示するように指定します。PutMsgRecFields フィールドは、メッセージが配布リストに登録されたときだけ使用されます。RecsPresent フィールドが0、またはPutMsgRecOffset フィールドとPutMsgRecPtr フィールドが両方とも0の場合、このフィールドは無視されます。

存在するフィールドに対して、キューマネージャはそれぞれのあて先に対応する登録メッセージレコードのフィールドの値を使用します。存在しないフィールドに対しては、キューマネージャはMQMD 構造体の初期値を使用します。

登録メッセージレコードにどんなフィールドがあるか示すために、フラグを一つ以上指定できます。指定できるフラグを次に示します。

### MQPMRF\_MSG\_ID

メッセージ識別子フィールドが存在することを示します。

### MQPMRF\_CORREL\_ID

相関識別子フィールドが存在することを示します。

### MQPMRF\_GROUP\_ID

グループ識別子フィールドが存在することを示します。

### MQPMRF\_FEEDBACK

フィードバックフィールドが存在することを示します。

### MQPMRF\_ACCOUNTING\_TOKEN

課金トークンフィールドが存在することを示します。

このフラグが指定された場合、MQPMO\_SET\_IDENTITY\_CONTEXT または MQPMO\_SET\_ALL\_CONTEXT が Options フィールドに指定されなければなりません。この条件が満たされない場合、MQRC\_PMO\_RECORD\_FLAGS\_ERROR の理由コードで失敗します。

MQPMR 構造体のフィールドが存在しない場合、次のフラグを指定できます。

## MQPMRF\_NONE

登録メッセージレコードが存在しないことを示します。

この値が指定されると、RecsPresent フィールドは 0 でなければなりません。同時に、PutMsgRecOffset フィールドおよび PutMsgRecPtr フィールドは、どちらも 0 でなければなりません。MQPMRF\_NONE はプログラムの文書化のために定義されています。値としては 0 が定義されていますが、プログラムを文書化する以外の目的はありません。

次の場合、MQRC\_PMO\_RECORD\_FLAGS\_ERROR の理由コードで失敗します。

- PutMsgRecFields フィールドに不正なフラグが含まれている場合
- 登録メッセージレコードが指定されているが、PutMsgRecFields フィールドが MQPMRF\_NONE である場合

これは入力用のフィールドです。

このフィールドの初期値は MQPMRF\_NONE です。

このフィールドは、Version フィールドの値が MQPMO\_VERSION\_2 以降の場合だけ有効です。

### ● PutMsgRecOffset (MQLONG 型) 最初の登録メッセージレコードまでのオフセット

MQPMO 構造体の先頭から最初の MQPMR 構造体登録メッセージレコードまでのバイトオフセットです。オフセットは正の場合も負の場合もあります。PutMsgRecOffset フィールドは、配布リストにメッセージを登録するときだけ使用されます。RecsPresent フィールドが 0 の場合は無視されます。

配布リストにメッセージが登録される時、一つ以上の MQPMR 構造体登録メッセージレコードの配列があて先ごとのメッセージ属性を指定するために指定されなければなりません。指定する属性を次に示します。

- メッセージ識別子
- 相関識別子
- グループ識別子
- フィードバック値
- 課金トークン

この属性をすべて指定する必要はありません。ただし、どの属性を選択しても、フィールドは正しい順序に指定されなければなりません。詳細については、この章の「[MQPMR 構造体](#) - [登録メッセージレコード](#)」を参照してください。

配布リストのオープン時に、MQOD 構造体によって指定されたオブジェクトレコードと同じ数の登録メッセージレコードが存在しなければなりません。登録メッセージレコードは、対応するオブジェクトレコードによって指定されるキューに対してメッセージ属性を与えます。配布リストのオープンに失敗したキューには、配列の適当な位置に配置される登録メッセージレコードが登録されなければなりません。その場合、メッセージ属性は無視されます。

登録メッセージレコードの数が、オブジェクトレコードの数と異なることがあります。オブジェクトレコードより登録メッセージレコードが少ない場合、登録メッセージレコードのないあて先のメッセージ属性は、メッセージ記述子 MQMD 構造体の対応するフィールドの属性が指定されます。オブジェクトレコードより登録メッセージレコードが多い場合、必要数以上のレコードは使用されません。ただし、使用されないレコードにもアクセスできなければなりません。

登録メッセージレコードの指定は任意です。ただし、RecsPresent フィールドがある場合は、指定しなければなりません。

登録メッセージレコードは MQOD 構造体のオブジェクトレコードと同様の方法で指定できます。PutMsgRecOffset フィールドのオフセットを指定するか、または PutMsgRecPtr フィールドのアドレスを使用して指定してください。詳細は、この章の「MQDH 構造体 - 配布ヘッダ」で ObjectRecOffset フィールドを参照してください。

PutMsgRecOffset フィールドと PutMsgRecPtr フィールドは同時に使用できません。両方のフィールドが 0 以外の場合、MQRC\_PUT\_MSG\_RECORDS\_ERROR の理由コードで失敗します。

これは入力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQPMO\_VERSION\_2 以降の場合だけ有効です。

### ● PutMsgRecPtr (MQPTR 型) 最初の登録メッセージレコードのアドレス

最初の登録メッセージレコードのアドレスです。PutMsgRecPtr フィールドは、配布リストにメッセージが登録されたときだけ使用されます。RecsPresent フィールドが 0 の場合は無視されます。

PutMsgRecPtr フィールドまたは PutMsgRecOffset フィールドのどちらかを登録メッセージレコードを指定するのに使用できます。ただし、同時に使用できません。詳細については、「PutMsgRecOffset フィールド」を参照してください。PutMsgRecPtr フィールドが使用されない場合は、ヌルポインタか 0 を設定しなければなりません。

これは入力用のフィールドです。

このフィールドの初期値は、ポインタをサポートしているプログラミング言語ではヌルポインタです。文字列の場合はヌル文字列です。

このフィールドは、Version フィールドの値が MQPMO\_VERSION\_2 以降の場合だけ有効です。

#### ■ 注意事項

プログラミング言語がポインタのデータ型をサポートしていないオペレーティングシステムでは、このフィールドは、初期値がすべてヌル文字列の、適当な長さのバイト文字列として宣言されます。

## ● RecsPresent (MQLONG 型) 登録メッセージレコード数または応答レコード数

アプリケーションによって指定された MQPMR 構造体の登録メッセージレコード数または MQRR 構造体の応答レコード数です。メッセージが配布リストに登録されたときだけ、この数が 0 より大きくなります。登録メッセージレコードまたは応答レコードは任意に選択できます。アプリケーションはどちらのレコードも指定しなかったり、どちらか一つのレコードだけ選択したりできます。ただし、アプリケーションが両方のレコードを指定する場合は、それぞれの RecsPresent フィールドのレコードを指定しなければなりません。

RecsPresent フィールドの値は、配布リストのあて先の数と同じである必要はありません。指定された値が必要数より大きい場合、必要数以上のレコードは使用されません。必要数より少ないレコードが指定された場合、登録メッセージレコードがないあて先のメッセージのプロパティ値は初期値を取ります。詳細については、「PutMsgRecOffset フィールド」を参照してください。

RecsPresent フィールドの値が 0 より小さい、または 0 より大きいメッセージは配布リストに登録されない場合、理由コード MQRC\_RECS\_PRESENT\_ERROR で失敗します。

これは入力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQPMO\_VERSION\_2 以降の場合だけ有効です。

## ● ResolvedQMgrName (MQCHAR48 型) 受信キューマネージャ名

ResolvedQName のキューを定義しているキューマネージャの名称 (実名) です。

ローカルキューマネージャが名称解決を実行したあとのあて先キューマネージャの名前です。返される名前は、ResolvedQName フィールドに指定されたキューを保持するキューマネージャの名前です。

ResolvedQMgrName フィールドは、ローカルキューマネージャの名前になることがあります。

空白でない値が返されるのは、オブジェクトが単一のキューである場合だけです。オブジェクトが配布リストである場合、返される値は未定義です。

これは出力用のフィールドです。

このフィールドの長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● ResolvedQName (MQCHAR48 型) 受信キュー名

キューマネージャによって設定される出力用のフィールドです。

ローカルキューマネージャが名称解決を実行したあとのあて先キューの名前です。返される名前は、ResolvedQMgrName フィールドに指定されたキューマネージャが保持するキューの名前です。キューがモデルキューでオープンされた場合、生成された動的ローカルキューの名称が返されます。

空白でない値が返されるのは、オブジェクトが単一のキューである場合だけです。オブジェクトが配布リストである場合、返される値は未定義です。

これは出力用のフィールドです。

このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● ResponseRecOffset (MQLONG 型) 最初の応答レコードまでのオフセット

MQPMO 構造体の先頭から最初の MQRR 構造体の応答レコードまでのバイトオフセットです。オフセットは、正の場合も負の場合もあります。ResponseRecOffset フィールドは、メッセージが配布リストに登録されるときだけ使用されます。RecsPresent フィールドが 0 の場合は無視されます。

メッセージが配布リストに登録される時、一つ以上の MQRR 構造体の応答レコードの配列が、メッセージの送信が成功しなかったキューと、その失敗の理由を特定するために指定されなければなりません。なお、メッセージの送信が成功しなかったキューは MQRR 構造体の CompCode フィールドに記述され、その失敗の理由は MQRR 構造体の Reason フィールドに記述されます。

メッセージは、キューのオープンに失敗したか、登録操作が失敗したかのどちらかの理由で送信されていない可能性があります。キューマネージャが応答レコードを設定するのは、命令の結果が混在している場合だけです。例えば、次のような場合です。

- あるメッセージは送信に成功したが、ほかのメッセージが失敗している場合
- すべての送信が失敗し、その理由が異なる場合

この場合、MQRC\_MULTIPLE\_REASONS の理由コードが出現します。同じ理由コードがすべてのキューに適用される場合、応答レコードは設定されません。

配布リストがオープンされたとき、MQOD 構造体によって指定されたオブジェクトレコードと同じ数の応答レコードが存在しなければなりません。応答レコードが必要であれば、対応するオブジェクトレコードによって指定されるキューに対する登録の完了コードと理由コードに設定されます。配布リストのオープンに失敗したキューには、配列の適当な位置に配置される応答レコードがなければなりません。その場合、登録操作ではなく、オープン操作の完了コードと理由コードが設定されます。

応答レコードの数がオブジェクトレコードの数と異なることがあります。

オブジェクトレコードよりも応答レコードが少ない場合、アプリケーションは登録に失敗したあて先や、失敗した理由のすべてを特定できなくなります。オブジェクトレコードよりも応答レコードが多い場合、必要数以上のレコードは使用されません。ただし、使用されないレコードにもアクセスできなければなりません。応答レコードは任意で指定できますが、指定された場合は RecsPresent フィールドがなければなりません。

応答レコードはオブジェクトレコードと同様に、ResponseRecOffset フィールドのオフセットを指定するか、または ResponseRecPtr フィールドのアドレスを指定できます。指定方法については、この章の

「MQOD 構造体 - オブジェクト記述子」で ObjectRecOffset フィールドを参照してください。ただし、ResponseRecOffset フィールドと ResponseRecPtr フィールドは同時に指定できません。どちらのフィールドも 0 以外の場合は、MQRC\_RESPONSE\_RECORDS\_ERROR の理由コードで失敗します。

MQPUT1 命令では、このフィールドがヌルポインタまたは 0 でなければなりません。応答情報を要求すると、オブジェクト記述子である MQOD 構造体によって指定される応答レコードに返されます。

これは入力用のフィールドです。

このフィールドの初期値は 0 です。

このフィールドは、Version フィールドの値が MQPMO\_VERSION\_2 以降の場合だけ有効です。

### ● ResponseRecPtr (MQPTR 型) 最初の応答レコードのアドレス

最初の MQRR 構造体の応答レコードのアドレスです。ResponseRecPtr フィールドは、配布リストにメッセージが登録されたときだけ使用されます。RecsPresent フィールドが 0 の場合は無視されます。

ResponseRecPtr フィールドまたは ResponseRecOffset フィールドのどちらかを、応答レコードとして指定するために使用できます。ただし、同時に使用できません。詳細については、「ResponseRecOffset フィールド」を参照してください。ResponseRecPtr フィールドを使用しない場合は、ヌルポインタかヌル文字列を設定しなければなりません。

MQPUT1 命令では、このフィールドはヌルポインタまたは 0 でなければなりません。応答情報を要求すると、オブジェクト記述子 MQOD 構造体によって指定される応答レコードに返されます。

これは入力用のフィールドです。

このフィールドの初期値は、ポインタをサポートしているプログラミング言語ではヌルポインタです。文字列の場合はヌル文字列となります。

このフィールドは、Version フィールドの値が MQPMO\_VERSION\_2 以降の場合だけ有効です。

### ■ 注意事項

プログラミング言語がポインタのデータ型をサポートしていないオペレーティングシステムでは、このフィールドは、初期値がヌル文字列の、適当な長さのバイト文字列として宣言されます。

### ● StrucId (MQCHAR4 型) 構造体識別子

次の値を取ります。

#### MQPMO\_STRUC\_ID

メッセージ登録オプションの構造体識別子

C 言語では、MQPMO\_STRUC\_ID\_ARRAY も定義されています。これは、MQPMO\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

これは常に入力用のフィールドです。

このフィールドの初期値は、MQPMO\_STRUC\_ID です。

### ● Timeout (MQLONG 型) 予備

予備のフィールドです。キューマネージャはこの内容を特に使用しません。

このフィールドの初期値は-1 です。

### ● UnknownDestCount (MQLONG 型) リモートキューの数

配布リストの中で、リモートキューで受信されて、MQPUT 命令または MQPUT1 命令による送信が成功したキューの数を示しています。

キューマネージャが配布リストに一時的に保持しているメッセージは、その配布リストに含まれるそれぞれのあて先の数としてカウントされます。このフィールドは配布リストではない単一のキューへの登録時にも設定されます。

これは出力用のフィールドです。

このフィールドの初期値は 0 です。

なお、このフィールドは、Version フィールドが MQPMO\_VERSION\_2 より前の場合は設定されません。

### ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

#### MQPMO\_VERSION\_1

バージョン 1 のメッセージ登録オプションの構造体バージョン番号

このバージョンはすべての環境でサポートされています。

#### MQPMO\_VERSION\_2

バージョン 2 のメッセージ登録オプションの構造体バージョン番号

MQPMO\_VERSION\_2 以降の構造体にだけ存在するフィールドについては、そのことがフィールドの説明に記載されています。次に示す定数は、現在のバージョンのバージョン番号を指定します。

#### MQPMO\_CURRENT\_VERSION

現在のバージョンのメッセージ登録オプションの構造体バージョン番号

これは常に入力用のフィールドです。

このフィールドの初期値は MQPMO\_VERSION\_1 です。

# MQPMR 構造体 — 登録メッセージレコード

MQPMR 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
MsgId (MQBYTE24 型)	メッセージ識別子	なし
CorrelId (MQBYTE24 型)	相関識別子	なし
GroupId (MQBYTE24 型)	グループ識別子	なし
Feedback (MQLONG 型)	報告メッセージ返答コード	なし
AccountingToken (MQBYTE32 型)	課金トークン	なし

## 概要

### 目的

MQPMR 構造体は、メッセージが配布リストに登録されるときに、単一のあて先にさまざまなメッセージ属性を指定するために使用されます。MQPMR 構造体は、MQPUT および MQPUT1 命令の入出力用の構造体です。

### 文字セットおよびマシンコード

MQPMR 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。TP1/Message Queue Access のクライアントアプリケーションの場合、MQPMR 構造体はクライアントの文字セットとマシンコードに従います。

### 使用方法

MQPUT または MQPUT1 命令に構造体の配列を提供することによって、配布リスト中のあて先キューごとに異なる値を指定できます。入力だけのフィールドや入出力の可能なフィールドがあります。

#### 注意事項

この構造体は、固定のレイアウトがない点で特殊です。構造体中のフィールドはオプションであり、各フィールドの有無は MQPMO 構造体の PutMsgRecFields フィールドのフラグによって示されます。存在するフィールドは次に示す順序にする必要があります。また、ここにはないフィールドは、レコードにありません。

- MsgId
- CorrelId
- GroupId
- Feedback

- AccountingToken

MQPMR 構造体には固定のレイアウトがないので、サポートするプログラミング言語の COPY ファイルと INCLUDE ファイルでの定義の提供はありません。プログラマは、アプリケーションに必要なフィールドを含む宣言をする必要があります。また、PutMsgRecFields フィールドのフラグを指定してフィールドがあることを示す必要があります。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● AccountingToken (MQBYTE32 型) 課金トークン

キューに送信されるメッセージに使用される課金トークンです。メッセージの送信先となるキューの名称は、MQOPEN 命令または MQPUT1 命令の MQOR 構造体の配列中の対応する要素が指定しています。単一のキューに登録するときの MQMD 構造体の AccountingToken フィールドと同様に処理されます。

このフィールドが存在しないとき、MQMD 構造体の値が使用されます。

これは入力用のフィールドです。

### ● CorrelId (MQBYTE24 型) 相関識別子

キューに送信されるメッセージに使用される相関識別子です。メッセージの送信先となるキューの名称は、MQOPEN 命令または MQPUT1 命令の MQOR 構造体の配列中の対応する要素が指定しています。単一のキューに登録するときの MQMD 構造体の CorrelId フィールドと同様に処理されます。

このフィールドが MQPMR 構造体レコードに存在しない場合、またはあて先の数より MQPMR 構造体レコードの数が少ない場合、MQMD 構造体の値は、CorrelId フィールドを含む MQPMR 構造体レコードを持たないあて先に使用されます。

MQPMO\_NEW\_CORREL\_ID が指定されると、MQPMR 構造体レコードの有無に関係なく、一つの新しい相関識別子が生成されます。生成された相関識別子は、配布リストのすべてのあて先に使用されます。これは MQPMO\_NEW\_MSG\_ID が指定されたときとは異なる方法です。詳細については、「MsgId フィールド」を参照してください。

これは入出力用のフィールドです。

### ● Feedback (MQLONG 型) 報告メッセージ返答コード

キューに送信されるメッセージに使用されるフィードバックコードです。メッセージの送信先となるキューの名称は、MQOPEN 命令または MQPUT1 命令の MQOR 構造体の配列中の対応する要素が指定しています。単一のキューに登録するときの MQMD 構造体の Feedback フィールドと同様に処理されます。

このフィールドが存在しないときは、MQMD 構造体の値が使用されます。

これは入力用のフィールドです。

## ● GroupId (MQBYTE24 型) グループ識別子

キューに送信されるメッセージに使用されるグループ識別子です。メッセージの送信先となるキューの名称は、MQOPEN 命令または MQPUT1 命令の MQOR 構造体の配列中の対応する要素が指定しています。単一のキューに登録するときの MQMD 構造体の GroupId フィールドと同様に処理されます。

このフィールドが MQPMR 構造体レコードに存在しない場合、またはあて先の数より MQPMR 構造体レコードの数が少ない場合、MQMD 構造体の値は、GroupId フィールドを含む MQPMR 構造体レコードを持たないあて先に使用されます。この値は、MQPMO 構造体の MQPMO\_LOGICAL\_ORDER 中の表で示される処理と同様の処理がされますが、次の相違点があります。

- 新しいグループ識別子が使用されるとき、キューマネージャは、それぞれのあて先に異なるグループ識別子を生成します。したがって、同じグループ識別子を持つあて先はありません。
- このフィールドの値が使用されるとき、MQRC\_GROUP\_ID\_ERROR の理由コードで失敗します。

これは入出力用のフィールドです。

## ● MsgId (MQBYTE24 型) メッセージ識別子

キューに送信されるメッセージに使用されるメッセージ識別子です。メッセージの送信先となるキューの名称は、MQOPEN 命令または MQPUT1 命令の MQOR 構造体の配列中の対応する要素が指定しています。単一のキューに登録するときの MQMD 構造体の MsgId フィールドと同様に処理されます。

このフィールドが MQPMR 構造体レコードに存在しない場合、またはあて先の数よりも MQPMR 構造体レコードの数が少ない場合、MQMD 構造体の値は、MsgId フィールドを含む MQPMR 構造体レコードを持たないあて先に使用されます。その値が MQMI\_NONE の場合、新しいメッセージ識別子がそれぞれのあて先に対して生成されます。つまり、おのおののあて先は異なるメッセージ識別子を持っています。

MQPMO\_NEW\_MSG\_ID が指定されると、MQPMR 構造体レコードの有無に関係なく、新しいメッセージ識別子が配布リストのすべてのあて先に生成されます。これは、MQPMO\_NEW\_CORREL\_ID が指定されたときとは異なる方法です。詳細については、「CorrelId フィールド」を参照してください。

これは入出力用のフィールドです。

# MQRMH 構造体 – メッセージ参照ヘッダ

MQRMH 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQRMH_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQRMH_VERSION_1
StrucLength (MQLONG 型)	構造体長	0
Encoding (MQLONG 型)	長大データのマシンコード形式	MQENC_NATIVE
CodedCharSetId (MQLONG 型)	長大データの文字セット識別子	MQCCSI_UNDEFINED
Format (MQCHAR8 型)	長大データのフォーマット名	MQFMT_NONE
Flags (MQLONG 型)	フラグ	MQRMHF_NOT_LAST
ObjectType (MQCHAR8 型)	オブジェクトタイプ	空白
ObjectInstanceId (MQBYTE24 型)	オブジェクトインスタンス識別子	MQOIL_NONE
SrcEnvLength (MQLONG 型)	送信元環境データ長	0
SrcEnvOffset (MQLONG 型)	送信元環境データオフセット	0
SrcNameLength (MQLONG 型)	送信元オブジェクト名長	0
SrcNameOffset (MQLONG 型)	送信元オブジェクトのオフセット	0
DestEnvLength (MQLONG 型)	あて先環境データ長	0
DestEnvOffset (MQLONG 型)	あて先環境データオフセット	0
DestNameLength (MQLONG 型)	あて先オブジェクト名長	0
DestNameOffset (MQLONG 型)	あて先オブジェクトのオフセット	0
DataLogicalLength (MQLONG 型)	長大データ長	0
DataLogicalOffset (MQLONG 型)	長大データ低位オフセット	0
DataLogicalOffset2 (MQLONG 型)	長大データ高位オフセット	0

## 概要

### 目的

MQRMH 構造体はメッセージ参照ヘッダの形式を定義します。このヘッダは、キューマネージャ間で非常に大きなデータ（長大データといいます）を送信するために、メッセージ編集出口 UOC とともに使用します。通常のメッセージ通信と異なり、長大データはキューに格納されません。代わりに、長大データへの参照だけがキューに格納されます。こうして、少数の極端に大きなメッセージによって、MQ リソースが消費される可能性を減らします。

### 1. データタイプ

## フォーマット名

MQFMT\_REF\_MSG\_HEADER

## 文字セットおよびマシンコード

MQRMH 構造体の文字データ、およびオフセットフィールドが示す先の文字列は、ローカルキューマネージャの文字セットに従います。これはキューマネージャの CodedCharSetId 属性で指定します。MQRMH 構造体の数値データは、計算機固有のマシンコードです。これは MQENC\_NATIVE で指定します。

MQRMH 構造体の文字セットおよびマシンコードは、次に示す構造体中の CodedCharSetId および Encoding フィールドに設定する必要があります。

- MQMD 構造体 (MQRMH 構造体がメッセージデータの先頭にある場合)
- MQRMH 構造体に先行するヘッダ構造体 (その他のすべての場合)

## 使用方法

アプリケーションは長大データを省略して、MQRMH 構造体で構成されるメッセージを登録します。メッセージチャンネルエージェント (MCA) が転送キューからメッセージを取り出すとき、ユーザが作成したメッセージ編集出口 UOC が呼び出されてメッセージ参照ヘッダを処理します。出口は MQRMH 構造体 が示す長大データを参照メッセージに追加できます。その後、MCA はチャンネルを介して次のキューマネージャにメッセージを送信します。

受信側では、参照メッセージを待つメッセージの出口が存在しなければなりません。参照メッセージを受信すると、出口はメッセージの MQRMH 構造体に続くデータからオブジェクトを生成し、データなしの参照メッセージを返します。その後、参照メッセージはアプリケーションによって取り出されます。

通常、MQRMH 構造体が、メッセージのすべてです。ただし、メッセージが転送キューにある場合は、一つ以上のヘッダが MQRMH 構造体の前に付きます。

参照メッセージは配布リストに送信されることも可能です。この場合、メッセージが転送キューにあるときは、MQDH 構造体および MQDH 構造体に関連するレコードが、MQRMH 構造体の前にあります。

### 注意事項

参照メッセージはセグメント分割されたメッセージとして送信されてはいけません。メッセージの出口は、セグメント分割されたメッセージを正しく処理できません。

## データ変換

データ変換時、MQRMH 構造体では、送信元の環境データ、送信元オブジェクト名称、あて先環境変数、およびあて先オブジェクト名がデータ変換されます。構造体の先頭の StrucLength 外にあるバイトは、データ変換後に廃棄されるか、未定義の値になります。長大データは、次に示すすべての条件が満たされる場合に変換されます。

- データが変換されるときに長大データがメッセージに存在する場合

- MQRMH 構造体の Format フィールドが MQFMT\_NONE 以外の値である場合
- ユーザの記述したデータ変換出口が、指定されたフォーマット名で存在する場合

ただし、通常、メッセージがキューに存在するとき、長大データはメッセージには存在しません。したがって、MQGMO\_CONVERT オプションでは変換されないことに注意してください。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● CodedCharSetId (MQLONG 型) 文字セット識別子

長大データの文字セット識別子です。

これは、長大データの文字セット識別子を指定します。MQRMH 構造体自身の文字データには適用されません。

MQPUT または MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。

### MQCCSI\_INHERIT

この構造体の文字セット識別子を引き継ぎます。

この構造体に続くデータの文字データは、この構造体と同じ文字セットです。

メッセージで送信される構造体のこの値を、キューマネージャは構造体の実際の文字セット識別子に変更します。エラーが発生しない場合、MQGET 命令では MQCCSI\_INHERIT は返されません。

MQMD 構造体の PutApplType フィールドの値が MQAT\_BROKER の場合、MQCCSI\_INHERIT は使用できません。

このフィールドの初期値は MQCCSI\_UNDEFINED です。

### ● DataLogicalLength (MQLONG 型) 長大データ長

DataLogicalLength フィールドは MQRMH 構造体で参照される長大データの長さを指定します。

長大データが実際にメッセージ中に存在するときは、MQRMH 構造体のすぐあとに続きます。全メッセージ長から、MQRMH 構造体の終端のメッセージを含むオフセットを減算すると、存在する長大データの長さが求められます。

データがメッセージ内に存在すると、DataLogicalLength フィールドは、関連するデータ量を指定します。通常、DataLogicalLength フィールドは、実際にメッセージに存在するデータ長と同じ値となります。

MQRMH 構造体が、指定された論理オフセットからのオブジェクト内にある残りのデータを表す場合、DataLogicalLength フィールドには 0 が指定できます。つまり、長大データはメッセージ内に存在しません。

データが存在しないとき、MQRMH 構造体の終端がメッセージの終端と一致します。

このフィールドの初期値は0です。なお、長大データのデータ変換が正常に行われた場合、変換後の長大データ長が DataLogicalLength フィールドに返却されます。

### ● DataLogicalOffset (MQLONG 型) 長大データ低位オフセット

一部が長大データによって形成されたオブジェクトの開始位置からの、長大データの低位オフセットを指定します。オブジェクトの開始位置から長大データまでのオフセットは論理オフセットといいます。これは、MQRMH 構造体の先頭から長大データまでの物理的オフセットではありません。物理的なオフセットは StrucLength フィールドで与えられます。

参照メッセージを使用して大きなオブジェクトを送信するために、論理オフセットは二つのフィールドに分割されます。実際の論理オフセットは、これら二つのフィールドの合計で与えられます。

- DataLogicalOffset フィールドは、論理オフセットを 1000000000 で除算したときの余りを表します。0 から 999999999 の範囲で値を取ります。
- DataLogicalOffset2 フィールドは、論理オフセットを 1000000000 で除算したときの結果を表します。結果は、論理オフセットに存在する 1000000000 の倍数となります。倍数の数は 0 から 999999999 の範囲で値を取ります。

このフィールドの初期値は0です。

### ● DataLogicalOffset2 (MQLONG 型) 長大データ高位オフセット

一部が長大データによって形成されたオブジェクトの開始位置からの、長大データの高位オフセットを指定します。0 から 999999999 の範囲の値を取ります。詳細については、「DataLogicalOffset フィールド」を参照してください。

このフィールドの初期値は0です。

### ● DestEnvLength (MQLONG 型) あて先環境データ長

このフィールドが0の場合、あて先環境データは存在しません。したがって、DestEnvOffset フィールドは無視されます。

このフィールドの初期値は0です。

### ● DestEnvOffset (MQLONG 型) あて先環境データオフセット

MQRMH 構造体の先頭からあて先環境データまでのオフセットを指定します。あて先環境データは、参照メッセージの作成者が知っている場合は、作成者によって指定されます。作成者があて先環境データを知らない場合は、ユーザ作成メッセージ出口が、必要な環境情報を特定しなければなりません。

あて先環境データの長さは DestEnvLength フィールドによって与えられます。この長さが0の場合、あて先環境データは存在しません。したがって DestEnvOffset フィールドは無視されます。存在する場合、あて先環境データは構造体の先頭から StrucLength フィールドの間に存在しなければなりません。

アプリケーションは、あて先環境データが構造体の最後の固定フィールドのあとからすぐに開始したり、SrcEnvOffset フィールド、SrcNameOffset フィールド、および DestNameOffset フィールドで指定されるデータとあて先環境データが隣接したりすることを仮定しないでください。

このフィールドの初期値は 0 です。

### ● DestNameLength (MQLONG 型) あて先オブジェクト名長

このフィールドが 0 の場合、あて先オブジェクト名はありません。したがって、DestNameOffset フィールドは無視されます。

このフィールドの初期値は 0 です。

### ● DestNameOffset (MQLONG 型) あて先オブジェクトのオフセット

MQRMH 構造体の先頭からあて先オブジェクト名までのオフセットを指定します。あて先オブジェクト名は、参照メッセージの作成者が知っている場合は、作成者によって指定されます。作成者があて先オブジェクト名を知らない場合は、ユーザ作成メッセージ出口が作成されたり変更されたりするオブジェクトを識別します。

あて先オブジェクト名の長さは、DestNameLength フィールドによって与えられます。この長さが 0 の場合は、あて先オブジェクト名は存在しません。したがって、DestNameOffset フィールドは無視されます。存在する場合、あて先オブジェクト名は、構造体の先頭から StrucLength フィールドの間に存在しなければなりません。

アプリケーションは、SrcEnvOffset フィールド、SrcNameOffset フィールド、および DestEnvOffset フィールドで指定されるデータとあて先オブジェクト名が隣接することを仮定しないでください。

このフィールドの初期値は 0 です。

### ● Encoding (MQLONG 型) マシンコード形式

長大データのマシンコード形式です。

長大データのマシンコードを指定します。MQRMH 構造体自身の数値データには適用されません。

MQPUT および MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。

このフィールドの初期値は MQENC\_NATIVE です。

### ● Flags (MQLONG 型) フラグ

メッセージを参照するフラグです。

次のフラグが定義されています。

#### MQRMHF\_LAST

オブジェクトの最終部分を表します。

このフラグは、参照メッセージが参照オブジェクトの最終部分を含んでいる、または最終部分であることを示しています。

## MQRMHF\_NOT\_LAST

オブジェクトの最終部分ではないことを表します。

MQRMHF\_NOT\_LAST はプログラムの文書化の目的で定義されています。値としては 0 が定義されていますが、プログラムを文書化する以外の目的はありません。

このフィールドの初期値は MQRMHF\_NOT\_LAST です。

### ● Format (MQCHAR8 型) フォーマット名

長大データのフォーマット名です。

長大データのフォーマット名を指定します。

MQPUT および MQPUT1 命令では、アプリケーションはデータに合った値をこのフィールドに設定する必要があります。このフィールドの設定規則は MQMD 構造体の Format フィールドと同じです。

このフィールドの初期値は、MQFMT\_NONE です。

### ● ObjectInstanceld (MQBYTE24 型) オブジェクトインスタンス識別子

オブジェクトの特別なインスタンスを識別するために使用します。使用しない場合は、次の値を設定してください。

## MQOII\_NONE

オブジェクトインスタンス識別子は指定されません

この値はフィールドの長さと同じ 2 進数の 0 です。

C 言語では、MQOII\_NONE\_ARRAY も定義されています。これは、MQOII\_NONE と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの長さは、MQ\_OBJECT\_INSTANCE\_ID\_LENGTH に定義されています。

このフィールドの初期値は MQOII\_NONE です。

### ● ObjectType (MQCHAR8 型) オブジェクトタイプ

メッセージ出口でサポートする参照メッセージのタイプを識別するために使用できる名称です。この名称を指定する場合、次に示す文字を使用することをお勧めします。

- 英大文字の A~Z
- 数値の 0~9

このフィールドの初期値は 8 個の空白です。

## ● SrcEnvLength (MQLONG 型) 送信元環境データ長

このフィールドが 0 の場合、送信元メッセージの環境データは存在しません。したがって、SrcEnvOffset フィールドは無視されます。

このフィールドの初期値は 0 です。

## ● SrcEnvOffset (MQLONG 型) 送信元環境データオフセット

MQRMH 構造体の先頭から送信元環境データまでのオフセットを指定します。送信元環境データは、参照メッセージの作成者が知っている場合は、作成者によって指定されます。作成者が送信元環境データを知らない場合は、ユーザ作成メッセージ出口が必要な環境情報を特定しなければなりません。

送信元環境データの長さは、SrcEnvLength フィールドによって与えられます。この長さが 0 の場合、送信元環境データは存在しません。したがって、SrcEnvOffset フィールドは無視されます。送信元環境データが存在する場合、構造体の先頭から StrucLength フィールドの間に存在しなければなりません。

アプリケーションは、送信元環境データが構造体の最後の固定フィールドのあとからすぐに開始したり、SrcNameOffset フィールド、DestEnvOffset フィールド、および DestNameOffset フィールドで指定されるデータと送信元環境データが隣接したりすることを仮定しないでください。

このフィールドの初期値は 0 です。

## ● SrcNameLength (MQLONG 型) 送信元オブジェクト名長

このフィールドが 0 の場合、送信元オブジェクト名はありません。したがって、SrcNameOffset フィールドは無視されます。

このフィールドの初期値は 0 です。

## ● SrcNameOffset (MQLONG 型) 送信元オブジェクトのオフセット

MQRMH 構造体の先頭から送信元オブジェクト名までのオフセットを指定します。送信元オブジェクト名は、参照メッセージの作成者が知っている場合は、作成者によって指定されます。作成者が送信元オブジェクト名を知らない場合は、ユーザ作成メッセージ出口がアクセスするオブジェクトを識別しなければなりません。

送信元オブジェクト名の長さは SrcNameLength フィールドによって与えられます。この長さが 0 の場合、送信元オブジェクト名は存在しません。したがって、SrcNameOffset フィールドは無視されます。存在する場合、送信元オブジェクト名は構造体の先頭から StrucLength フィールドの間に存在しなければなりません。

アプリケーションは、SrcEnvOffset フィールド、DestEnvOffset フィールド、および DestNameOffset フィールドで指定されるデータと送信元オブジェクト名が隣接することを仮定しないでください。

このフィールドの初期値は 0 です。

## ● StrucId (MQCHAR4 型) 構造体識別子

次の値を取ります。

### MQRMH\_STRUC\_ID

参照メッセージヘッダ構造体の識別子

C 言語では、MQRMH\_STRUC\_ID\_ARRAY も定義されています。これは、MQRMH\_STRUC\_ID と同じ値です。ただし文字列ではなく文字の配列として定義されています。

このフィールドの初期値は MQRMH\_STRUC\_ID です。

## ● StrucLength (MQLONG 型) 構造体長

MQRMH 構造体の構造体長を表します。固定フィールドの終端の文字列を含みます。ただし、長大データは含んでいません。

このフィールドの初期値は 0 です。

## ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

### MQRMH\_VERSION\_1

バージョン 1 の参照メッセージヘッダの構造体バージョン番号

次の値には、現在のバージョンのバージョン番号を指定します。

### MQRMH\_CURRENT\_VERSION

現在のバージョンの参照メッセージヘッダの構造体バージョン番号

このフィールドの初期値は MQRMH\_VERSION\_1 です。

# MQRR 構造体 - 応答レコード

MQRR 構造体は、次のフィールドで構成されます。

フィールド (データタイプ)	内容	初期値
CompCode (MQLONG 型)	完了コード	MQCC_OK
Reason (MQLONG 型)	理由コード	MQRC_NONE

## 概要

### 目的

MQRR 構造体は、あて先が配布リストである場合に、単一のあて先キューに対するオープンまたは登録操作の結果である完了コードと理由コードを受け取るために使用します。MQRR 構造体は、MQOPEN、MQPUT および MQPUT1 命令の出力用の構造体です。

### 文字セットおよびマシンコード

MQRR 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれ、キューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。TP1/Message Queue Access のクライアントアプリケーションの場合、MQOD 構造体はクライアントの文字セットとマシンコードに従います。

### 使用方法

MQOPEN、MQPUT または MQPUT1 命令に構造体の配列を提供することによって、命令の結果が混在する場合、つまり配布リストの幾つかのキューで命令が成功し、ほかのキューで失敗する場合に、配布リストのすべてのキューについて完了コードと理由コードを判別できます。命令が理由コード MQRC\_MULTIPLE\_REASONS を返す場合、(アプリケーションが提供したとき) 応答レコードがキューマネージャによって設定されたことを示します。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● CompCode (MQLONG 型) 完了コード

MQOPEN 命令または MQPUT1 命令の MQOR 構造体の配列中の対応する要素によって指定された名称のキューに対する、オープンまたは登録操作の結果を表す完了コードです。

これは常に入力用のフィールドです。このフィールドの初期値は MQCC\_OK です。

### ● Reason (MQLONG 型) 理由コード

MQOPEN 命令または MQPUT1 命令の MQOR 構造体の配列中の対応する要素によって指定された名称のキューに対する、オープンまたは登録操作の結果を表す理由コードです。

これは常に入力用のフィールドです。このフィールドの初期値は MQRC\_NONE です。

# MQTM 構造体 – トリガメッセージ

MQTM 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQTM_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQTM_VERSION_1
QName (MQCHAR48 型)	イベント発生キュー名	ヌル文字列/空白
ProcessName (MQCHAR48 型)	プロセス名	ヌル文字列/空白
TriggerData (MQCHAR64 型)	トリガデータ	ヌル文字列/空白
ApplType (MQLONG 型)	アプリケーションタイプ	0
ApplId (MQCHAR256 型)	アプリケーション識別子	ヌル文字列/空白
EnvData (MQCHAR128 型)	環境データ	ヌル文字列/空白
UserData (MQCHAR128 型)	ユーザデータ	ヌル文字列/空白

## 概要

### 目的

この構造体には、トリガメッセージのデータが記述されます。キューマネージャは、トリガイベントが発生したときに、トリガモニタアプリケーションへトリガメッセージを通知します。

### フォーマット名

MQFMT\_TRIGGER

### 文字セットおよびマシンコード

MQTM 構造体の文字データは、MQTM 構造体を生成するキューマネージャの文字セットに従います。MQTM 構造体の数値データは、MQTM 構造体を生成するキューマネージャのマシンコードに従います。

MQTM 構造体の文字セットおよびマシンコードは、次に示す構造体中の CodedCharSetId および Encoding フィールドで指定されます。

- MQMD 構造体 (MQTM 構造体がメッセージデータの先頭にある場合)
- MQTM 構造体に先行するヘッダ構造体 (その他のすべての場合)

### 使用方法

トリガモニタアプリケーションがほかのアプリケーションを起動するとき、トリガメッセージ内の幾つかの情報を、そのアプリケーションに渡す必要があります。そのアプリケーションで必要となる情報には、QName、TriggerData、UserData などがあります。トリガによるアプリケーション開始については、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。

## トリガメッセージの MQMD 構造体

トリガメッセージの MQMD 構造体のフィールドには、次の値が設定されます。これは、トリガメッセージを取り出した MQGET 命令の、MsgDesc 引数で返される値です。

MQMD 構造体のフィールド	値
StrucId	MQMD_STRUC_ID
Version	MQMD_VERSION_1
Report	MQRO_NONE
MsgType	MQMT_DATAGRAM
Expiry	MQEI_UNLIMITED
Feedback	MQFB_NONE
Encoding	MQENC_NATIVE
CodedCharSetId	キューマネージャの CodedCharSetId 属性の値
Format	MQFMT_TRIGGER
Priority	MQPRI_PRIORITY_AS_Q_DEF (イニシエーションキューに定義された、省略時のメッセージ優先度)
Persistence	MQPER_NOT_PERSISTENT
MsgId	ユニークな値
CorrelId	MQCI_NONE
BackoutCount	0
ReplyToQ	空白
ReplyToQMGr	空白
UserIdentifier	空白
AccountingToken	MQACT_NONE
ApplIdentityData	空白
PutApplType	MQAT_QMGR
PutApplName	キューマネージャ名の先頭から 28 文字
PutDate	トリガメッセージを通知した日付
PutTime	トリガメッセージを通知した時刻
ApplOriginData	空白

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### 1. データタイプ

## ● ApplId (MQCHAR256 型) アプリケーション識別子

起動するアプリケーションを識別する文字列です。トリガメッセージを取り出したトリガモニタアプリケーションで使用します。キューマネージャはこのフィールドに、ProcessName のプロセスオブジェクトの ApplId 属性の値を設定します。キューマネージャはこの内容を特に使用しません。

ApplId の意味は、トリガモニタアプリケーションによって決まります。IBM MQ が提供するトリガモニタでは、ApplId を実行プログラムの名前にする必要があります。

このフィールドの長さは、MQ\_PROCESS\_APPL\_ID\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、256 個の空白です。

## ● ApplType (MQLONG 型) アプリケーションタイプ

起動するアプリケーションの種類です。トリガメッセージを取り出したトリガモニタアプリケーションで使用します。キューマネージャはこのフィールドに、ProcessName のプロセスオブジェクトの ApplType 属性の値を設定します。キューマネージャはこの内容を特に使用しません。

ApplType フィールドには、次の標準の値があります。また、ユーザが定義した値も使用できます。ただし、ユーザが定義できる値の範囲は MQAT\_USER\_FIRST から MQAT\_USER\_LAST までです。

MQAT\_AIX : AIX アプリケーション

MQAT\_CICS : CICS トランザクション

MQAT\_DOS : PC-DOS 上の IBM MQ クライアントアプリケーション

MQAT\_IMS : IMS アプリケーション

MQAT\_MVS : MVS または TSO アプリケーション

MQAT\_OPEN\_TP1 : OpenTP1 アプリケーション

MQAT\_OS2 : OS/2 またはプレゼンテーションマネージャアプリケーション

MQAT\_OS400 : OS/400 アプリケーション

MQAT\_TMS\_4V : TMS-4V/SP アプリケーション

MQAT\_UNIX : UNIX アプリケーション

MQAT\_WINDOWS : 16 ビット Windows アプリケーション

MQAT\_WINDOWS\_NT : 32 ビット Windows アプリケーション

MQAT\_XDM : XDM アプリケーション

MQAT\_USER\_FIRST : ユーザが定義できるアプリケーションタイプの最小値

MQAT\_USER\_LAST：ユーザが定義できるアプリケーションタイプの最大値

このフィールドの初期値は0です。

### ● EnvData (MQCHAR128型) 環境データ

起動するアプリケーションに関連する環境についての文字データです。トリガメッセージを取り出したトリガモニタアプリケーションで使用します。キューマネージャは、このフィールドに ProcessName のプロセスオブジェクトの EnvData 属性の値を設定します。キューマネージャはこの内容を特に使用しません。

このフィールドの長さは、MQ\_PROCESS\_ENV\_DATA\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、128 個の空白です。

### ● ProcessName (MQCHAR48型) プロセス名

トリガイベントが発生したキューのプロセスオブジェクトの名称です。トリガメッセージを取り出したトリガモニタアプリケーションで使用します。キューマネージャはこのフィールドに、QName のキューの ProcessName 属性の値を設定します。

フィールドの長さより名称が短い場合、余りが空白で埋められます。ヌル文字は使用されません。このフィールドの長さは、MQ\_PROCESS\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

### ● QName (MQCHAR48型) イベント発生キュー名

トリガイベントが発生したキューの名称です。このキューの名称は、トリガモニタアプリケーションによって起動されたアプリケーションで使用します。キューマネージャは、このフィールドにトリガイベントが発生したキューの QName 属性の値を設定します。

フィールドの長さより名称が短い場合、余りが空白で埋められます。ヌル文字は使用されません。このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

### ● StrucId (MQCHAR4型) 構造体識別子

次の値を取ります。

#### MQTM\_STRUC\_ID

トリガメッセージの構造体識別子

C 言語では、MQTM\_STRUC\_ID\_ARRAY も定義されています。これは、MQTM\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの初期値は MQTM\_STRUC\_ID です。

## ● TriggerData (MQCHAR64 型) トリガデータ

任意の形式のデータです。トリガメッセージを取り出したトリガモニタアプリケーションで使用します。キューマネージャはこのフィールドに、QName のキューの TriggerData 属性の値を設定します。キューマネージャはこの内容を特に使用しません。

このフィールドの長さは、MQ\_TRIGGER\_DATA\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、64 個の空白です。

## ● UserData (MQCHAR128 型) ユーザデータ

起動するアプリケーションに関するユーザ任意の情報です。トリガメッセージを取り出したトリガモニタアプリケーションで使用します。キューマネージャは、このフィールドに ProcessName のプロセスオブジェクトの UserData 属性の値を設定します。キューマネージャはこの内容を特に使用しません。

このフィールドの長さは、MQ\_PROCESS\_USER\_DATA\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、128 個の空白です。

## ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

`MQTM_VERSION_1`

トリガメッセージの構造体バージョン番号

次の定数には、現在のバージョン番号を指定します。

`MQTM_CURRENT_VERSION`

現在のバージョンを指定したトリガメッセージの構造体バージョン番号

このフィールドの初期値は `MQTM_VERSION_1` です。

# MQXQH 構造体 – 転送キューヘッダ

MQXQH 構造体は、次のフィールドから構成されます。

フィールド (データタイプ)	内容	初期値
StrucId (MQCHAR4 型)	構造体識別子	MQXQH_STRUC_ID
Version (MQLONG 型)	構造体バージョン番号	MQXQH_VERSION_1
RemoteQName (MQCHAR48 型)	リモートキュー名	ヌル文字列/空白
RemoteQMgrName (MQCHAR48 型)	リモートキューマネージャ名	ヌル文字列/空白
MsgDesc (MQMD 構造体)	登録元メッセージ記述子	該当しない

## 概要

### 目的

MQXQH 構造体は、転送キューにメッセージを格納するときに使用されます。そのメッセージの先頭には、この構造体が付けられます。転送キューは、リモートキューへ送信するときに、一時的にメッセージを格納する特別なローカルキューです。したがって、ローカルキューマネージャに属するキューへの転送には使用しません。転送キューは、Usage 属性が MQUS\_TRANSMISSION のキューです。

### フォーマット名

MQFMT\_XMIT\_Q\_HEADER

### 文字セットおよびマシンコード

MQXQH 構造体のデータは、ローカルキューマネージャの文字セットおよびマシンコードに従います。それぞれキューマネージャの CodedCharSetId 属性および MQENC\_NATIVE で指定します。

MQXQH 構造体の文字セットおよびマシンコードは、次に示す構造体中の CodedCharSetId および Encoding フィールドに設定する必要があります。

- 分離 MQMD 構造体 (MQXQH 構造体がメッセージデータの先頭にある場合)
- MQXQH 構造体に先行するヘッダ構造体 (その他のすべての場合)

### 使用方法

転送キューにあるメッセージには、次に示す二つのメッセージ記述子があります。

- メッセージデータから分離して格納されるメッセージ記述子  
これは分離メッセージ記述子と呼ばれます。メッセージが転送キューに登録されるときに、キューマネージャによって生成されます。分離メッセージ記述子のフィールドの幾つかは、アプリケーションが MQPUT または MQPUT1 命令に指定するメッセージ記述子からコピーされます。詳細については、「分離メッセージ記述子のフィールド」を参照してください。

分離メッセージ記述子は、メッセージが転送キューから削除されたときに、MQGET 命令の MsgDesc 引数でアプリケーションに返されます。

- MQXQH 構造体にメッセージデータの一部として格納されているメッセージ記述子

これは埋め込みメッセージ記述子と呼ばれます。このメッセージ記述子は、アプリケーションが MQPUT または MQPUT1 命令に指定するメッセージ記述子とほぼ同じコピーです。詳細については、「埋め込みメッセージ記述子のフィールド」を参照してください。

埋め込みメッセージ記述子は、常にバージョン 1 の MQMD 構造体です。アプリケーションによって登録されたメッセージが、MQMD 構造体のバージョン 2 フィールドで初期値ではない値を一つ以上持つ場合、MQMDE 構造体が MQXQH 構造体に続きます。その後、アプリケーションメッセージデータが存在する場合は、そのアプリケーションメッセージデータが続きます。このとき、MQMDE 構造体は、次のどちらかです。

- アプリケーションがメッセージを登録するためにバージョン 2 の MQMD 構造体を使用したときに、キューマネージャによって生成された MQMDE 構造体
- すでにアプリケーションメッセージデータの先頭に存在する MQMDE 構造体

アプリケーションがメッセージを登録するためにバージョン 1 の MQMD 構造体を使用すると、メッセージが最終のあと先キューから削除されたときに、埋め込みメッセージ記述子は MQGET 命令の MsgDesc 引数でアプリケーションに返されます。

## 分離メッセージ記述子のフィールド

分離メッセージ記述子は、キューマネージャによって次のように設定されます。キューマネージャがバージョン 2 をサポートしていない場合は、バージョン 1 の MQMD 構造体フィールドがすべて使用されます。

MQMD 構造体のフィールド	使用される値
StrucId	MQMD_STRUC_ID
Version	MQMD_VERSION_2
Report	埋め込みメッセージ記述子から複写されますが、MQRO_ACCEPT_UNSUP_IF_XMIT_MASK で識別されるビットが 0 に設定されます。これによって、メッセージが転送キューに到着したとき、または転送キューから取り出されたとき、COA または COD 報告が生成されません。
MsgType	埋め込みメッセージ記述子から複写
Expiry	埋め込みメッセージ記述子から複写
Feedback	埋め込みメッセージ記述子から複写
Encoding	MQENC_NATIVE
CodedCharSetId	キューマネージャの CodedCharSetId
Format	MQFMT_XMIT_Q_HEADER
Priority	埋め込みメッセージ記述子から複写
Persistence	埋め込みメッセージ記述子から複写

MQMD 構造体のフィールド	使用される値
MsgId	キューマネージャによって生成される新しい値が使用されます。このメッセージ記述子は、キューマネージャが埋め込みメッセージ記述子に生成する MsgId フィールドとは異なります。
CorrelId	埋め込みメッセージ記述子の CorrelId フィールド
BackoutCount	0
ReplyToQ	埋め込みメッセージ記述子から複写
ReplyToQMGr	埋め込みメッセージ記述子から複写
UserIdentifier	埋め込みメッセージ記述子から複写
AccountingToken	埋め込みメッセージ記述子から複写
ApplIdentityData	埋め込みメッセージ記述子から複写
PutApplType	MQAT_QMGR
PutApplName	キューマネージャ名称の先頭 28 バイト
PutDate	メッセージが転送キューに登録された日
PutTime	メッセージが転送キューに登録された時間
ApplOriginData	空白
GroupId	MQGI_NONE
MsgSeqNumber	1
Offset	0
MsgFlags	MQMF_NONE
OriginalLength	MQOL_UNDEFINED

## 埋め込みメッセージ記述子のフィールド

埋め込みメッセージ記述子のフィールドは、次に示す場合を除いて、MQPUT 命令または MQPUT1 命令の MsgDesc 引数と同じ値です。

- Version フィールドは、常に MQMD\_VERSION\_1 の値となります。  
Priority フィールドが MQPRI\_PRIORITY\_AS\_Q\_DEF の場合、キューの DefPriority 属性の値に変換されます。Persistence フィールドが MQPER\_PERSISTENCE\_AS\_Q\_DEF の場合、キューの DefPersistence 属性の値に変換されます。
- MsgId フィールドは、次に示す場合、キューマネージャによって生成された新しいメッセージ識別子に変換されます。
  - MsgId フィールドが MQMI\_NONE の場合
  - MQPMO\_NEW\_MSG\_ID オプションが指定された場合
  - メッセージが配布リストメッセージの場合

配布リストメッセージが、異なる転送キューに登録するために分割されることがあります。そのとき、新しい埋め込みメッセージ記述子のそれぞれの MsgId フィールドは、元の配布リストメッセージと同じです。

- MQPMO\_NEW\_CORREL\_ID オプションが指定されると、CorrelId フィールドは、キューマネージャによって生成された新しい相関識別子に変換されます。

コンテキストフィールドは、PutMsgOpts 引数で指定された MQPMO\_\*\_CONTEXT オプションによって設定されます。コンテキストフィールドとは、次に示すフィールドのことです。

- UserIdentifier
- AccountingToken
- ApplIdentityData
- PutApplType
- PutApplName
- PutDate
- PutTime
- ApplOriginData

バージョン 2 のフィールドがある場合は、MQMD 構造体から削除され、MQMDE 構造体に移動されます。これは、バージョン 2 フィールドの一つ以上の値が初期値以外の場合です。

### リモートキューへのメッセージの登録

アプリケーションが、リモートキューに直接指定してメッセージを登録する場合、またはリモートキューのローカル定義を使用してメッセージを登録する場合、ローカルキューマネージャは、次の処理をします。

- 埋め込みメッセージ記述子を含む MQXQH 構造体の生成
- MQMDE 構造体が必要であるが、存在しない場合は、MQMDE 構造体の付加
- アプリケーションメッセージデータの付加
- メッセージの適切な転送キューへの登録

### 転送キューに直接メッセージを登録する方法

アプリケーションは、転送キューに直接メッセージを登録することもできます。この場合、アプリケーションは MQXQH 構造体をアプリケーションメッセージデータの前に設定し、フィールドを適切な値で初期化しなければなりません。また、アプリケーションが、MQPUT 命令または MQPUT1 命令で指定したメッセージ記述子の MQMD 構造体の Format フィールドに、MQFMT\_XMIT\_Q\_HEADER を設定しなければなりません。

アプリケーションによって作成された MQXQH 構造体の文字データは、MQPUT 命令または MQPUT1 命令で指定された MsgDesc 引数の CodedCharSetId フィールドで定義された文字集合に含まれなければなりません。整数データも、そのメッセージ記述子の Encoding フィールドで定義されたマシンコード変換形式に含まれなければなりません。また、MQXQH 構造体の文字データは、定義されたフィールドの長

さまで空白で埋められなければなりません。キューマネージャは、MQXQH 構造体の中のヌル文字とそれに続く文字を空白に変換しません。したがって、データ文字列の終わりにヌル文字を使用してなりません。

キューマネージャは MQXQH 構造体が存在するかどうか、およびフィールドに有効な値が指定されているかどうかをチェックしません。注意してください。

## 転送キューからメッセージを取り出す方法

転送キューからメッセージを取り出すアプリケーションは、MQXQH 構造体の情報を適切な方法で処理しなければなりません。アプリケーションメッセージデータの始めに MQXQH 構造体が存在するかどうかは、MQGET 命令の MsgDesc 引数の Format フィールドに返される値である MQFMT\_XMIT\_Q\_HEADER で示されます。

MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドに返される値は、それぞれ MQXQH 構造体の文字データの文字集合と、整数データのコード変換形式を示します。アプリケーションメッセージデータの文字集合とコード変換形式は、埋め込みメッセージ記述子の CodedCharSetId フィールドと Encoding フィールドで定義されます。

## フィールド

構造体を構成するフィールドについて、アルファベット順に説明します。

### ● MsgDesc (MQMD 構造体) 登録元メッセージ記述子

アプリケーションがメッセージをリモートキューへ送信したときの MQPUT 命令または MQPUT1 命令で、MsgDesc 引数に指定されたメッセージ記述子 MQMD 構造体の複写です。

#### 注意事項

これはバージョン 1 の MQMD 構造体です。

この構造体の各フィールドの初期値は、MQMD 構造体の各フィールドの初期値と同じです。

### ● RemoteQMgrName (MQCHAR48 型) リモートキューマネージャ名

メッセージの表面上の最終あて先であるキューを保持するキューマネージャの名前です。

メッセージが配布リストメッセージの場合、RemoteQMgrName フィールドは空白となります。

このフィールドの長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● RemoteQName (MQCHAR48 型) リモートキュー名

メッセージの表面上のあて先キューの名称です。これは、キューの実名と異なる場合があります。例えば、ほかのリモートキューのローカル定義として、RemoteQMgrName のリモートキューマネージャに定義された場合などです。

メッセージが配布リストメッセージのとき、つまり、MQMD 構造体の Format フィールドが MQFMT\_DIST\_HEADER であるとき、RemoteQName フィールドは空白です。

このフィールドの長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

このフィールドの初期値は、C 言語ではヌル文字列です。そのほかのプログラミング言語では、48 個の空白です。

## ● StrucId (MQCHAR4 型) 構造体識別子

次の値を取ります。

### MQXQH\_STRUC\_ID

転送キューヘッダの構造体識別子

C 言語では、MQXQH\_STRUC\_ID\_ARRAY も定義されています。これは、MQXQH\_STRUC\_ID と同じ値です。ただし、文字列ではなく文字の配列として定義されています。

このフィールドの初期値は MQXQH\_STRUC\_ID です。

## ● Version (MQLONG 型) 構造体バージョン番号

次の値を取ります。

### MQXQH\_VERSION\_1

転送キューヘッダの構造体バージョン番号

次の定数には、現在のバージョンのバージョン番号を指定します。

### MQXQH\_CURRENT\_VERSION

現在のバージョンの転送キューヘッダの構造体バージョン番号

このフィールドの初期値は MQXQH\_VERSION\_1 です。

# 2

## メッセージキューイング機能の命令

この章では、メッセージキューイング機能で使用する、命令について説明します。

# 命令の説明形式

この章では、メッセージキューイング機能を使用するための命令について、アルファベット順に説明します。

それぞれの命令では、引数とその説明を示します。各引数の後ろの括弧には、その引数のデータタイプを示します。データタイプについては、「1. データタイプ」を参照してください。

データタイプの次に、その引数の「向き」を次の値で示します。引数の向きは、その引数を何が使用し、何が設定するかを表します。

## input

キューマネージャで使用する値をアプリケーションが設定します。

## output

アプリケーションに返す値をキューマネージャが設定します。

## input/output

キューマネージャで使用する値をアプリケーションが設定し、そのあとにキューマネージャが修正してアプリケーションに返します。

あるフィールドが input で、ほかのフィールドが output の構造体のときも input/output で示します。詳しくは、「1. データタイプ」を参照してください。

また、この章では、キュー、キューマネージャなどの命令で利用する対象をオブジェクトと呼びます。オブジェクトの詳細については、「3. オブジェクトの属性」を参照してください。

関連する幾つかの定数を示すときに、アスタリスク「\*」を使用します。「\*」は、一つ以上の文字を表します。例えば、MQFB\_\*は、MQFB\_で始まるすべての定数を示します。

命令の使用方法は、すべてのプログラミング言語でほぼ同じです。ただし、サポートするプログラミング言語が機能的に異なるため、各プログラミング言語によって多少の書き換えがあります。この書き換えについては、「付録 E 各プログラミング言語での実現例」を参照してください。

## 注意事項

### 引数の表記に関する注意

C 言語の形式では、引数が大文字と小文字で表記され、COBOL 言語の形式では、引数が大文字だけで表記されます。以降の説明では、C 言語の形式で示した引数、つまり、大文字と小文字の混在で示します。COBOL 言語を使用する場合は、引数の小文字を大文字で置き換えてお読みください。

なお、C 言語の場合の形式で、引数の先頭に付けられたアスタリスク (\*) は、アドレス渡しかどうかを示します。

各命令を説明する形式を次に示します。

## MQXXXXX命令 — 日本語名称

### 機能

命令の目的の概要

### 形式

#### C言語の場合

MQXXXX (データタイプ 引数1, ……., データタイプ 引数n)

#### COBOL言語の場合

CALL 'MQXXXX' using 引数1, ……., 引数n.

### 引数

#### ● 引数1 (データタイプ) ー向き

引数の説明  
設定できる値

#### 値1

値の説明

⋮

#### 値N

値の説明

⋮

#### ● 引数n (データタイプ) ー向き

引数の説明

⋮

### 注意事項

命令の使用方法に関連する注意事項

# MQCLOSE 命令 – オブジェクトのクローズ

## 機能

MQCLOSE 命令で、オブジェクトの利用を終了します。MQOPEN 命令の反対の命令です。

## 形式

### C 言語の場合

```
MQCLOSE(MQHCONN Hconn, PMQHOBJ Hobj, MQLONG Options,  
        PMQLONG CompCode, PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQCLOSE' USING HCONN, HOBJ, OPTIONS, COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。

### ● Hobj (MQHOBJ 型) –input/output

オブジェクトハンドルです。

クローズするオブジェクトを示すハンドルです。どんなオブジェクトタイプでもかまいません。MQOPEN 命令の戻り値を指定してください。

命令が成功すると、この引数には次の値が返されます。

#### MQHO\_UNUSABLE\_HOBJ

使用できないオブジェクトハンドル

### ● Options (MQLONG 型) –input

MQCLOSE 命令の動作を制御するクローズオプションです。

オブジェクトのクローズの方法を示します。永続的動的キューの場合だけ、キューを残す方法または削除する方法のどちらかでクローズできます。なお、永続的動的キューは、DefinitionType 属性が MQQDT\_PERMANENT\_DYNAMIC のキューです。クローズオプションについては、表 2-1 に示します。

次のどれかの値を取れます。

#### MQCO\_NONE

特別なクローズオプションを要求しません。

次の場合に指定します。

- キュー以外のオブジェクト
- 定義済みキュー
- 一時的動的キュー（ただし、Hobj のハンドルが、そのキューを生成した MQOPEN 関数で返されたハンドルと異なる場合だけ指定できます）
- 配布リスト

上記の場合、オブジェクトは削除されないで残されます。

このオプションを一時的動的キューに指定した場合で、Hobj のハンドルがそのキューを生成した MQOPEN 命令で返されたハンドルのとき、キューは削除されます。そのほかのハンドルのときは、キューは残ります。

このオプションを永続的動的キューに指定した場合、キューは削除されないで残ります。

## MQCO\_DELETE

キューを削除します。

次のどちらかの場合、キューは削除されます。

- 永続的動的キューで、メッセージを格納しないで、かつコミットしていないトランザクションからの登録・取り出しがない場合  
ほかのアプリケーションによる登録・取り出しがない場合も含まれます。
- 一時的動的キューで、Hobj のハンドルがそのキューを生成した MQOPEN 命令で返されたハンドルの場合

上記以外の場合、理由コード MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE で命令は失敗します。このとき、キューは削除されません。

## MQCO\_DELETE\_PURGE

キューを強制的に削除します。

次のどちらかの場合、キューは削除されます。

- 永続的動的キューで、コミットしていないトランザクションからの登録・取り出しがない場合  
ほかのアプリケーションによる登録・取り出しがない場合も含まれます。
- 一時的動的キューで、Hobj のハンドルがそのキューを生成した MQOPEN 命令で返されたハンドルの場合

上記以外の場合、理由コード MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE で命令は失敗します。この場合、キューは削除されません。

クローズオプションとオブジェクトの関係を次の表に示します。

表 2-1 クローズオプションとオブジェクトの関係

オブジェクト	クローズオプション		
	MQCO_NONE	MQCO_DELETE	MQCO_DELETE_PURGE
キュー以外	1	2	2
定義済みキュー	1	2	2
永続的動的キュー	1	3	4
一時的動的キュー (作成者)	5	5	5
一時的動的キュー (作成者以外)	1	2	2
配布リスト	1	2	2

(凡例)

- 1：オブジェクトは削除されません。
- 2：不正な組み合わせです。
- 3：キューが空であり、かつ未決着の登録・取り出しがないときだけ削除されます。
- 4：未決着の登録・取り出しがないときだけ削除されます。
- 5：オブジェクトは削除されます。

### ● CompCode (MQLONG 型) -output

完了コードです。

次のどれかが返されます。

- MQCC\_OK：成功
- MQCC\_WARNING：警告（一部成功）
- MQCC\_FAILED：失敗

### ● Reason (MQLONG 型) -output

理由コードです。

CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

CompCode 引数が MQCC\_WARNING の場合

理由コード	数値	意味
MQRC_INCOMPLETE_GROUP	2241	メッセージグループが完全ではありません。

理由コード	数値	意味
MQRC_INCOMPLETE_MSG	2242	論理メッセージが完全ではありません。

## CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_HOBJ_ERROR	2019	オブジェクトハンドルが不正です。
MQRC_OBJECT_DAMAGED	2101	オブジェクトが破損しています。
MQRC_OPTION_NOT_VALID_FOR_TYPE	2045	オブジェクトタイプに適合しないオプションが指定されました。
MQRC_OPTIONS_ERROR	2046	オプションが不正です。または、指定されていません。
MQRC_Q_NOT_EMPTY	2055	キューに一つ以上のメッセージがあります。または、キューに対する登録・取り出し要求がコミットしていません。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。

詳細は、「[付録 B.2 理由コード](#)」を参照してください。

## 注意事項

- アプリケーションが MQDISC 命令を呼び出したとき、または正常終了・異常終了したときに、オープンしたままのオブジェクトは MQCO\_NONE オプションで自動的にクローズされます。
- キューをクローズするときは、次の点に注意してください。
  - トランザクション内で処理するときでも、同期点の結果に影響しません。同期点の前でもあとでもキューをクローズできます。
  - MQOO\_BROWSE オプションでキューをオープンしていた場合、検索カーソルは消滅します。MQOO\_BROWSE オプションで再びオープンしたときは、新しい検索カーソルが生成されます。MQOO\_BROWSE オプションについては、この章の「[MQOPEN 命令 - オブジェクトのオープン](#)」で Options 引数の説明を参照してください。
  - MQCLOSE 命令を呼び出した時点で、そのハンドルで排他状態にしたメッセージがある場合、排他状態は解除されます。「1. データタイプ」の「[MQGMO 構造体 - メッセージ取り出しオプション](#)」で Options フィールドの MQGMO\_LOCK オプションの説明を参照してください。
- クローズされるオブジェクトが動的キュー（永続的動的キューまたは一時的動的キュー）であるとき、次の点に注意してください。
  - 一時的動的キューの場合で、Hobj のハンドルがそのキューを生成した MQOPEN 命令で返されたハンドルのとき、キュー内にメッセージが格納されていてもキューは削除されます。このとき、

Options 引数に指定したオプションは関係ありません。これは、コミットしていない MQGET 命令、MQPUT 命令または MQPUT1 命令が、このハンドルを使用した場合、使用しなかった場合、およびそのキューに対して未決着のままの場合でも同様です。コミットしていない処理での更新が失われた場合も、トランザクションの失敗にはなりません。

- 動的キューが削除されるときにキューに対して未解決の MQGMO\_WAIT オプションを指定する MQGET 命令は取り消され、理由コード MQRC\_Q\_DELETED が返されます。

MQGMO\_WAIT オプションについては、「1. データタイプ」の「MQGMO 構造体 - メッセージ取り出しオプション」で Options フィールドの説明を参照してください。

動的キューが削除されたあとで、取得済みの Hobj ハンドルを使用して MQCLOSE 命令以外の命令がキューを参照しようとする、理由コード MQRC\_Q\_DELETED で失敗します。

削除されたキューにアプリケーションからアクセスできなくても、キューを参照するすべてのハンドルがクローズされ、キューに影響するすべてのトランザクションがコミットされるかロールバックされるまでは、キューはシステムから除去されず、関連リソースも解放されないことに注意してください。

- 一時的動的キューをクローズするとき MQCLOSE 命令に指定した Hobj ハンドルが、キューを作成した MQOPEN 命令が返したハンドルである場合、キューは削除されます。これは MQCLOSE 命令のクローズオプションに関係なく発生します。キューにメッセージがある場合は破棄されます。報告メッセージは生成されません。

キューに影響を与える未コミットのトランザクションがあるときでも、キューとメッセージは削除されます。しかし、これによってトランザクションは失敗しません。ただし、上記のとおり、各トランザクションがコミットされるかロールバックされるまでは、トランザクションに関連するリソースは解放されません。

4. クローズされるオブジェクトが配布リストの場合、次の点に注意してください。

- 配布リストに対する適切なオプションは MQCO\_NONE だけです。ほかのオプションが指定されると、理由コード MQRC\_OPTIONS\_ERROR または MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE で失敗します。
- 配布リストがクローズされる時、リスト中のキューについてのおのおのの完了コードと理由コードは、返却されません。CompCode 引数と Reason 引数は結果を確認するためだけに利用できます。キューのうち一つのクローズに失敗した場合も、キューマネージャは処理を続行し、配布リストの残りのキューをクローズしようとします。CompCode 引数と Reason 引数は失敗を記述した情報を返します。しかし、ほとんどのキューのクローズに成功しても、完了コードが MQCC\_FAILED になる場合があります。なお、エラーの発生したキューは識別できません。一つ以上のキューでクローズに失敗しても、どんな失敗が CompCode 引数と Reason 引数に報告されるかは定義されていません。

# MQCONN 命令 – キューマネージャへの接続

## 機能

MQCONN 命令で、アプリケーションをキューマネージャへ接続できます。この命令では、キューマネージャの接続ハンドルが返されます。接続ハンドルは、以降のそのキューマネージャへの命令で使用します。

## 形式

### C 言語の場合

```
MQCONN(PMQCHAR Name, PMQHCONN Hconn, PMQLONG CompCode,  
        PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQCONN' USING NAME, HCONN, COMPCODE, REASON.
```

## 引数

### ● Name (MQCHAR48 型) –input

キューマネージャ名です。

アプリケーションが接続するキューマネージャの名前です。名前には、次に示す文字を使用できます。

- 英大文字 (A~Z)
- 英小文字 (a~z)
- 数字 (0~9)
- ピリオド (.), スラント (/), アンダスコア (\_), パーセント (%)

名前の先頭および中間には、空白を使用できません。末尾には空白を使用できます。名前の中で有効なデータの末尾を示すためにヌル文字を使用できます。ヌル文字とそれに続く文字はすべて空白として処理されます。

すべて空白の名前を指定した場合、省略時のキューマネージャ名が使用されます。

指定するキューマネージャ名は接続可能なキューマネージャの名前にする必要があります。

接続できるキューマネージャは、環境に依存します。TP1/Message Queue では、アプリケーションと同じ OpenTP1 で動作するキューマネージャにだけ接続できます。

この引数の長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

## ● Hconn (MQHCONN 型) -output

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。以降に呼び出すメッセージキューイング機能の命令で指定してください。MQDISC 命令を呼び出すと、またはこのハンドルの範囲の処理が終了すると、このハンドルは無効になります。

このハンドルの範囲は、環境の平行処理の最小単位に制限されます。MQCONN 命令を呼び出した平行処理の単位以外では、このハンドルは無効です。TP1/Message Queue では、平行処理の最小単位はプロセスです。

## ● CompCode (MQLONG 型) -output

完了コードです。

次のどれかが返されます。

- MQCC\_OK : 成功
- MQCC\_WARNING : 警告 (一部成功)
- MQCC\_FAILED : 失敗

## ● Reason (MQLONG 型) -output

理由コードです。

CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

CompCode 引数が MQCC\_WARNING の場合

理由コード	数値	意味
MQRC_ALREADY_CONNECTED	2002	アプリケーションはすでに接続されています。

CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_Q_MGR_NAME_ERROR	2058	キューマネージャ名が不正です。または、キューマネージャ名を認識できません。
MQRC_Q_MGR_NOT_AVAILABLE	2059	キューマネージャが接続できる状態ではありません。
MQRC_Q_MGR_STOPPING	2162	キューマネージャが終了処理中です。

理由コード	数値	意味
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。

詳細は、「付録 B.2 理由コード」を参照してください。

## 注意事項

- MQCONN 命令で接続したキューマネージャを、ローカルキューマネージャとといいます。
- ローカルキューマネージャに属するキューを、ローカルキューとといいます。アプリケーションはローカルキューに対して、メッセージの登録・取り出しができます。  
リモートキューマネージャに属するキューを、リモートキューとといいます。アプリケーションはリモートキューに対して、メッセージを登録できます。しかし、取り出しはできません。
- アプリケーションの動作中にキューマネージャに障害が発生する場合、アプリケーションは再度 MQCONN 命令を発行し、以降の MQ 命令で使用するための新しいコネクションハンドルを取得する必要があります。命令が成功するまでアプリケーションは定期的に MQCONN 命令を発行できます。  
アプリケーションがキューマネージャに接続されているかわからない場合、アプリケーションは MQCONN 命令を問題なく発行してコネクションハンドルを取得できます。アプリケーションがすでに接続されているとき、返されるハンドルは以前に発行した MQCONN 命令で返されたハンドルと同じです。ただし、完了コードは MQCC\_WARNING、理由コードは MQRC\_ALREADY\_CONNECTED です。
- MQ 命令を使用する処理の最後には、アプリケーションは MQDISC 命令を使用してキューマネージャから切り離す必要があります。

# MQDISC 命令 – キューマネージャからの切り離し

## 機能

MQDISC 命令で、キューマネージャとアプリケーションプログラムとの接続を切り離します。これは、MQCONN 命令の反対の命令です。

## 形式

### C 言語の場合

```
MQDISC(PMQHCONN Hconn, PMQLONG CompCode, PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQDISC' USING HCONN, COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input/output

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。命令が成功すると、この引数には次の値が返されます。

MQHC\_UNUSABLE\_HCONN：使用できないコネクションハンドル

### ● CompCode (MQLONG 型) –output

完了コードです。

次のどちらかが返されます。

- MQCC\_OK：成功
- MQCC\_FAILED：失敗

### ● Reason (MQLONG 型) –output

理由コードです。

CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。

詳細は、「[付録 B.2 理由コード](#)」を参照してください。

## 注意事項

接続下でオブジェクトがオープンされているときに MQDISC 命令を発行すると、オブジェクトは、キューマネージャによってクローズされます。このときのクローズオプションは MQCO\_NONE です。

# MQGET 命令 – メッセージの取り出し

---

## 機能

MQGET 命令で、ローカルキューからメッセージを取り出せます。ローカルキューは、MQOPEN 命令でオープンしておく必要があります。

## 形式

### C 言語の場合

```
MQGET(MQHCONN Hconn, MQHOBJ Hobj, PMQVOID MsgDesc,  
      PMQVOID GetMsgOpts, MQLONG BufferLength,  
      PMQVOID Buffer, PMQLONG DataLength,  
      PMQLONG CompCode, PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQGET' USING HCONN, HOBJ, MSGDESC, GETMSGOPTS,  
                  BUFFERLENGTH, BUFFER, DATALENGTH, COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。

### ● Hobj (MQHOBJ 型) –input

オブジェクトハンドルです。

メッセージを取り出すキューのハンドルです。MQOPEN 命令の戻り値を指定してください。そのキューは、次に示すオプションを一つ以上使用してオープンしておく必要があります。詳細については、この章の「MQOPEN 命令 – オブジェクトのオープン」を参照してください。

MQOO\_INPUT\_SHARED

MQOO\_INPUT\_EXCLUSIVE

MQOO\_INPUT\_AS\_Q\_DEF

MQOO\_BROWSE

### ● MsgDesc (MQMD 構造体) –input/output

メッセージ記述子です。

この構造体で、取り出したいメッセージの属性を記述します。また、取り出したメッセージの属性が返されます。詳細については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」を参照してください。

BufferLength 引数の値がメッセージ長より短い場合でも、キューマネージャによって MsgDesc 引数に値が設定されます。これは、GetMsgOpts 引数で MQGMO\_ACCEPT\_TRUNCATED\_MSG を指定しても指定しなくても同様です。詳細については、「1. データタイプ」の「MQGMO 構造体 – メッセージ取り出しオプション」で Options フィールドの説明を参照してください。

アプリケーションがバージョン 1 の MQMD 構造体を指定すると、返却されるメッセージには、アプリケーションメッセージデータの前に MQMDE 構造体が付加されます。これは、MQMDE 構造体の一つ以上のフィールドが初期値以外の場合だけです。MQMDE 構造体の Format フィールドにある MQFMT\_MD\_EXTENSION のフォーマット名は、MQMDE 構造体が存在することを示します。

### ● GetMsgOpts (MQGMO 構造体) –input/output

メッセージ取り出しオプションです。

詳細については、「1. データタイプ」の「MQGMO 構造体 – メッセージ取り出しオプション」を参照してください。

### ● BufferLength (MQLONG 型) –input

メッセージデータを格納するバッファのバイト長です。

0 も指定できます。0 を指定した場合は、データのないメッセージを取り出すものとみなされるか、メッセージがキューから削除されて、破棄されなければなりません。メッセージを削除する場合は、MQGMO\_ACCEPT\_TRUNCATED\_MSG の指定が必要です。

#### ■ 注意事項

キューから取り出せるメッセージの最大長は、ローカルキューの MaxMsgLength 属性に定義されています。

### ● Buffer (MQBYTE 型×BufferLength) –output

メッセージデータを格納するバッファです。

バッファには、メッセージ中のデータに適したバウンダリ調整が必要です。多くのメッセージ (MQ ヘッダ構造体のあるメッセージを含む) には 4 バイト単位の調整が適していますが、より厳しい調整が必要なメッセージもあります。例えば、64 ビット 2 進整数を含めるメッセージには 8 バイト単位のバウンダリ調整が必要です。

BufferLength 引数の値がメッセージ長より短い場合、Buffer の領域に格納できるだけのメッセージが返されます。これは、GetMsgOpts 引数に MQGMO\_ACCEPT\_TRUNCATED\_MSG を指定しても指定しなくても同様です。詳細については、「1. データタイプ」の「MQGMO 構造体 – メッセージ取り出しオプション」で Options フィールドの説明を参照してください。

Buffer の領域の文字セットとマシンコード形式は、MsgDesc 引数の CodedCharSetId フィールドと Encoding フィールドに返されます。要求される値と異なる場合、アプリケーションメッセージデータを、その文字セットとマシンコード形式に変換してください。アプリケーションメッセージデータを変換するため、

MQGMO\_CONVERT オプションを使用できます。詳細については、「1. データタイプ」の「MQGMO 構造体 - メッセージ取り出しオプション」で Options フィールドの説明を参照してください。

## 注意事項

MQGET 命令のほかの引数には、ローカルキューマネージャの文字セットとマシンコード形式、つまり、キューマネージャの CodedCharSetId 属性と MQENC\_NATIVE を使用します。

命令が失敗した場合でも、Buffer の領域の内容が変わることがあります。

C 言語では、void 型のポインタとしてこの引数を宣言します。したがって、どんなデータタイプのアドレスでも引数として指定できます。

BufferLength 引数に 0 を指定した場合、Buffer 引数は参照されません。したがって、C 言語のプログラムでは、この引数のアドレスとしてヌルを指定できます。

### ● DataLength (MQLONG 型) -output

メッセージ長です。

メッセージのアプリケーションデータのバイト長です。データ長が BufferLength 引数の値より大きい場合、BufferLength のバイト数だけが Buffer 引数に返されます。この場合、領域を超えたメッセージの部分は切り捨てられます。この値が 0 の場合、メッセージにアプリケーションデータがなかったことを意味します。

BufferLength 属性の値がメッセージ長より短い場合でも、キューマネージャによって DataLength 引数に実際のメッセージの長さが設定されます。これは、GetMsgOpts 引数に MQGMO\_ACCEPT\_TRUNCATED\_MSG オプションを指定しても指定しなくても同様です。詳細については、「1. データタイプ」の「MQGMO 構造体 - メッセージ取り出しオプション」で Options フィールドの説明を参照してください。これによってアプリケーションは、メッセージ全体を格納できるバッファを確保して、再び命令を呼び出せます。

ただし、MQGMO\_CONVERT オプションが指定され、変換されたメッセージデータが大き過ぎて Buffer 引数に設定できない場合、DataLength 引数には、キューマネージャ定義のフォーマットに対応する未変換データの長さが設定されます。

データの性質上、変換中のデータが拡大することがあるため、アプリケーションは、キューマネージャが DataLength 引数に設定した値より大きいバッファを割り当ててください。

### ● CompCode (MQLONG 型) -output

完了コードです。

次のどれかが返されます。

- MQCC\_OK：成功
- MQCC\_WARNING：警告（一部成功）
- MQCC\_FAILED：失敗

### ● Reason (MQLONG 型) -output

理由コードです。

#### CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

#### CompCode 引数が MQCC\_WARNING の場合

理由コード	数値	意味
MQRC_CONVERTED_MSG_TOO_BIG	2120	変換したメッセージがアプリケーションバッファに対して大き過ぎます。
MQRC_FORMAT_ERROR	2110	メッセージのフォーマットが不正です。
MQRC_INCONSISTENT_CCSDS	2243	メッセージセグメントが異なる CCSID を持っています。
MQRC_INCONSISTENT_ENCODINGS	2244	メッセージセグメントが異なるマシンコード形式を持っています。
MQRC_INCONSISTENT_UOW	2245	コミット単位の指定が矛盾しています。
MQRC_NO_MSG_LOCKED	2209	排他状態のメッセージはありません。
MQRC_NOT_CONVERTED	2119	メッセージデータは変換されません。
MQRC_SOURCE_CCSDS_ERROR	2111	生成元コード化文字セット識別子が無効です。
MQRC_TARGET_CCSDS_ERROR	2115	受信側コード化文字セット識別子が無効です。
MQRC_TRUNCATED_MSG_ACCEPTED	2079	バッファ不足のため、メッセージの後部を切り捨てました（処理は完了しました）。
MQRC_TRUNCATED_MSG_FAILED	2080	バッファ不足のため、メッセージの後部を切り捨てました（処理は完了していません）。

#### CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_BUFFER_ERROR	2004	バッファの引数が不正です。
MQRC_BUFFER_LENGTH_ERROR	2005	バッファ長の引数が不正です。
MQRC_DATA_LENGTH_ERROR	2010	データ長の引数が不正です。

理由コード	数値	意味
MQRC_GET_INHIBITED	2016	指定されたキューからの取り出しが禁止されています。
MQRC_GMO_ERROR	2186	メッセージ取り出しオプションの構造体が不正です。
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_HOBJ_ERROR	2019	オブジェクトハンドルが不正です。
MQRC_INCONSISTENT_BROWSE	2259	ブラウザ指定が矛盾しています。
MQRC_INCONSISTENT_UOW	2245	コミット単位の指定が矛盾しています。
MQRC_INVALID_MSG_UNDER_CURSOR	2246	メッセージ下のカーソルは取り出せません。
MQRC_MATCH_OPTIONS_ERROR	2247	一致オプションが不正です。
MQRC_MD_ERROR	2026	メッセージ記述子が不正です。
MQRC_MSG_SEQ_NUMBER_ERROR	2250	メッセージシーケンス番号が無効です。
MQRC_NO_MSG_AVAILABLE	2033	該当するメッセージがありません。
MQRC_NO_MSG_UNDER_CURSOR	2034	検索カーソルの位置にメッセージがありません。
MQRC_NOT_OPEN_FOR_BROWSE	2036	キューが検索用にオープンされていません。
MQRC_NOT_OPEN_FOR_INPUT	2037	キューが取り出し用にオープンされていません。
MQRC_OBJECT_DAMAGED	2101	オブジェクトが破損しています。
MQRC_OPTIONS_ERROR	2046	オプションが不正です。または、指定されていません。
MQRC_Q_DELETED	2052	キューが削除されています。
MQRC_Q_MGR_STOPPING	2162	キューマネージャが終了処理中です。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_SYNCPOINT_NOT_AVAILABLE	2072	同期点処理はできません。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。
MQRC_WAIT_INTERVAL_ERROR	2090	MQGMO 構造体の待ち合わせ最大時間が不正です。
MQRC_WRONG_GMO_VERSION	2256	MQGMO 構造体で指定されたバージョンは不正です。
MQRC_WRONG_MD_VERSION	2257	MQMD 構造体で指定されたバージョンは不正です。

詳細は、「付録 B.2 理由コード」を参照してください。

## 注意事項

1. 取り出したメッセージは、通常、キューから削除されます。この削除は、MQGET 命令の処理の一部、または同期点の処理の一部として起こります。

GetMsgOpts 引数に検索オプションが指定されている場合、メッセージは削除されません。検索オプションを次に示します。

- MQGMO\_BROWSE\_FIRST
- MQGMO\_BROWSE\_NEXT
- MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR

これらのオプションの詳細については、「1. データタイプ」の「MQGMO 構造体 – メッセージ取り出しオプション」で Options フィールドの説明を参照してください。

2. 検索オプションと同時に MQGMO\_LOCK オプションを指定した場合、検索したメッセージは排他状態になり、このハンドルでだけ利用できるようになります。

MQGMO\_UNLOCK オプションを指定した場合、以前に排他状態であったメッセージの排他状態が解除されます。この場合、どんなメッセージも取り出せません。また、MsgDesc 引数、BufferLength 引数、Buffer 引数、および DataLength 引数は、参照・変更されません。

3. アプリケーションがキューにメッセージを登録した順序は、次の条件がすべて満たされる場合だけ保たれます。

- すべてのメッセージが同じ優先度の場合
- すべての MQPUT 命令で、同じオブジェクトハンドル Hobj を使用した場合
- どの MQPUT 命令も、トランザクション内から呼び出していない場合
- そのキューが、MQPUT 命令を呼び出したキューマネージャのローカルキューである場合（詳細は、後述の「注意 b.」を参照してください）

これらの条件を満たす場合、かつ次の方法でメッセージを取り出した場合、メッセージは登録された順序で取り出せます。

- メッセージを取り出すアプリケーションが、MsgId を指定するなどして、取り出し順序を変更しなかった場合
- そのキューからメッセージを取り出すのが、一つのアプリケーションだけの場合

複数のアプリケーションがそのキューからメッセージを取り出す場合、それらのアプリケーションは、メッセージが一連のものであることを識別する取り決めをしておく必要があります。例えば、登録元のアプリケーションは、メッセージの CorrelId フィールドに独自の値を設定できます。

### 注意

a.

ほかのアプリケーションから、またはアプリケーション内の異なるトランザクションからのメッセージは、一つのトランザクションの観点からは登録した順序である場合も、間隔をおいてキュー内に格納されることがあります。

b.

リモートキューでは、送信側のキューマネージャから受信側のキューマネージャまでに一つの回線しかない構成のときに、メッセージの順序が保たれます。メッセージの幾つかが異なる回線で送信されると、メッセージの到着順序は保証されません。例えば、再構成、トラフィック調整、メッセージ長による回線選択などでは、到着順序は保証されません。また、あて先キューが満杯であるなどの理由で、メッセージの幾つかがデッドレターキューに転送された場合にも、順序が乱れることになります。

登録した順序を保証する条件を満たせない場合も、アプリケーションは、登録した順序を次の方法で保証できます。

- 順序を示す独自の情報をアプリケーションメッセージデータ内に組み込みます。
  - 次のメッセージを登録する前に、相手が一つ前のメッセージを受け取ったことを確認します。
4. 単一のトランザクション内で特定のキューに連続するメッセージをアプリケーションが登録し、トランザクションのコミットに成功する場合、メッセージは次に示す要領で取り出し可能になります。
- キューが非共有キュー（つまりローカルキュー）である場合、トランザクション内のメッセージはすべて同時に利用できるようになります。
5. アプリケーションがメッセージグループを使用しないで、連続するメッセージを同じキューに登録する場合、特定の条件が満たされれば、メッセージの順序が保持されます。詳細については、この章の「MQPUT 命令 – メッセージの登録」の「注意事項」を参照してください。条件が満たされると、受信側アプリケーションは送信された順序でメッセージを受信できます。ただし、次に示す制約があります。
- キューからメッセージを取り出す受信側は一つだけであること  
キューからメッセージを取り出すアプリケーションが二つ以上ある場合は、各メッセージを識別するためのメカニズムについて、送信側と申し合わせる必要があります。例えば、送信側アプリケーションで、連続するメッセージのすべての CorrelId フィールドに順序に応じたユニークな値を設定できます。
  - 特定の MsgId や CorrelId を指定することなどによって、受信側は取り出しの順序を故意に変更しないこと
- 送信側のアプリケーションがメッセージグループとしてメッセージを登録する場合、受信側のアプリケーションが MQGET 命令に MQGMO\_LOGICAL\_ORDER オプションを指定するとき、メッセージは受信側のアプリケーションによって正しい順序で受信されます。メッセージグループの詳細については、次に示す項目を参照してください。
- MQMD 構造体の MsgFlags フィールド
  - MQPMO 構造体の MQPMO\_LOGICAL\_ORDER オプション
  - MQGMO 構造体の MQGMO\_LOGICAL\_ORDER オプション
6. アプリケーションは、MsgDesc 引数の Feedback フィールドが MQFB\_QUIT かどうかを調べる必要があります。この値の場合は、アプリケーションを終了してください。詳細については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で Feedback フィールドの説明を参照してください。

7. Hobj に該当するキューを MQOO\_SAVE\_ALL\_CONTEXT オプションでオープンしている場合で、MQGET 命令の完了コードが MQCC\_OK または MQCC\_WARNING のとき、キューハンドル Hobj に関連するコンテキストは、取り出したメッセージのコンテキストで置き換えられます。ただし、MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_NEXT, または MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR オプションを指定した場合を除きます。この場合、この注意は該当しません。

保存されたコンテキストは MQPMO\_PASS\_IDENTITY\_CONTEXT または MQPMO\_PASS\_ALL\_CONTEXT オプションを指定することによって、以降の MQPUT または MQPUT1 命令で使用できます。これによって、受信されたメッセージのコンテキストが、その全体または一部を（メッセージが異なるキューに転送される時などに）異なるメッセージに伝達することが可能になります。メッセージコンテキストの詳細については、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。

8. GetMessageOptions 引数に MQGMO\_CONVERT オプションが指定されている場合、アプリケーションメッセージデータは、Buffer 引数に設定される前に、受信側アプリケーションが要求する表現に変換されます。

- メッセージの制御情報の中の Format フィールドがアプリケーションデータの構造を識別し、CodedCharSetId フィールドと Encoding フィールドがデータの文字セット識別子とマシンコード形式を指定しています。
- MQGET 命令を呼び出すアプリケーションでは、MsgDesc 引数の CodedCharSetId フィールドと Encoding フィールドに、アプリケーションデータ中の変換させる文字セット識別子とマシンコード形式を指定します。

メッセージデータは、変換が必要なとき、メッセージの制御情報の Format フィールドの値に依存し、キューマネージャ自身によって変換されます。

- 次に示すフォーマット名称に対しては、キューマネージャが変換します。このフォーマット名称は、組み込みフォーマットと呼ばれます。

MQFMT\_ADMIN  
MQFMT\_COMMAND1  
MQFMT\_COMMAND2  
MQFMT\_DEAD\_LETTER\_HEADER  
MQFMT\_EVENT バージョン 1  
MQFMT\_MD\_EXTENSION  
MQFMT\_PCF  
MQFMT\_REF\_MSG\_HEADER  
MQFMT\_STRING  
MQFMT\_TRIGGER  
MQFMT\_XMIT\_Q\_HEADER

- フォーマット名称 MQFMT\_NONE は、メッセージデータの性質が未定義の特別な値です。したがって、メッセージがキューから取り出されたときも、キューマネージャによってその値が変換されることはありません。

### ■ 注意事項

フォーマット名称 MQFMT\_NONE のメッセージに MQGET 命令で MQGMO\_CONVERT が指定されたとき、文字セットまたはマシンコード形式が MsgDesc 引数に指定されたものと異なる場合、かつほかのエラーがない場合は、メッセージは Buffer 引数に返されます。ただし、命令は完了コード MQCC\_WARNING および理由コード MRC\_FORMAT\_ERROR で完了します。

- MQGET 命令から返された値で、次に示す理由コードはメッセージの変換が成功したことを示します。

#### MQRC\_NONE

- 次に示す理由コードはメッセージの変換が成功した可能性があることを示します。成功したかどうかを調べるには、アプリケーションは MsgDesc 引数の CodedCharSetId フィールドと Encoding フィールドをチェックしなければなりません。

#### MQRC\_TRUNCATED\_MSG\_ACCEPTED

そのほかのすべての理由コードは、メッセージが変換されなかったことを示します。

9. メッセージの順序性については、マニュアル「TP1/Message Queue プログラム作成の手引」の「12.3 注意事項」を参照してください。

# MQINQ 命令 – オブジェクトの属性の照会

## 機能

MQINQ 命令で、オブジェクトの属性を示す整数の配列と文字列が返されます。次に示す種類のオブジェクトの属性を照会できます。

- キュー
- プロセス定義
- キューマネージャ

## 形式

### C 言語の場合

```
MQINQ(MQHCONN Hconn, MQHOBJ Hobj, MQLONG SelectorCount,  
      PMQLONG Selectors, MQLONG IntAttrCount,  
      PMQLONG IntAttrs, MQLONG CharAttrLength,  
      PMQCHAR CharAttrs, PMQLONG CompCode,  
      PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQINQ' USING HCONN, HOBJ, SELECTORCOUNT, SELECTORS,  
                  INTATTRCOUNT, INTATTRS, CHARATTRLENGTH, CHARATTRS,  
                  COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。

### ● Hobj (MQHOBJ 型) –input

オブジェクトハンドルです。

属性を照会するオブジェクトを示すハンドルです。MQOO\_INQUIRE オプションを指定した MQOPEN 命令の戻り値を指定してください。

### ● SelectorCount (MQLONG 型) –input

セレクトタ数です。

Selectors の配列に指定するセレクトタの数を指定します。ここで指定した数の属性が返されます。セレクトタ数は、0～256 の範囲で指定できます。

## ● Selectors (MQLONG 型×SelectorCount) –input

セレクタです。

SelectorCount の要素数の配列でセレクタを指定します。それぞれのセレクタは、照会する整数型属性または文字型属性の種類を示します。

それぞれのセレクタは、Hobj のオブジェクトの種類に合ったものを指定してください。適合しないセレクタを指定した場合、完了コード MQCC\_FAILED、理由コード MQRC\_SELECTOR\_ERROR で命令は終了します。

キューの場合は次のようになります。

- どんなキュータイプにも適合しないセレクタのときは、完了コード MQCC\_FAILED、理由コード MQRC\_SELECTOR\_ERROR で命令は終了します。
- そのオブジェクトとは異なるタイプのキューに適合するセレクタのときは、完了コード MQCC\_WARNING、理由コード MQRC\_SELECTOR\_NOT\_FOR\_TYPE で命令は終了します。
- 照会するキューがクラスタキューの場合、有効なセレクタはキューの解決方法に依存します。詳細については「注意事項」の 4 を参照してください。

セレクタはどの順序で指定してもかまいません。整数型属性のセレクタ (MQIA\_\*セレクタ) の属性値は、Selectors 引数に指定した順序で、IntAttrs 引数に返されます。文字型属性のセレクタ (MQCA\_\*セレクタ) の属性値は、セレクタで指定した順序で、CharAttrs 引数に返されます。MQIA\_\*セレクタは、MQCA\_\*セレクタと交互に指定できます。各種類での順序だけが重要です。

### ■ 注意事項

1. 整数型属性と文字型属性のセレクタは、二つの異なる範囲で割り当てられています。MQIA\_\*セレクタは、MQIA\_FIRST から MQIA\_LAST の範囲にあります。また、MQCA\_\*セレクタは、MQCA\_FIRST から MQCA\_LAST の範囲にあります。  
それぞれの範囲に、キューマネージャで受け付けられる最大値として、定数 MQIA\_LAST\_USED と MQCA\_LAST\_USED が定義されています。
2. すべての MQIA\_\*セレクタを始めに指定した場合、要素の位置として、その要素と同じ番号を Selectors と IntAttrs の配列で使用できます。
3. SelectorCount 引数に 0 を指定した場合、Selectors 引数は参照されません。この場合、C 言語のプログラムでは Selectors のアドレスにヌルを指定できます。

照会できる属性について、次の表に示します。括弧内の値は、MQCA\_\*セレクタで CharAttrs 引数に設定する文字列の長さを定義した定数です。

表 2-2 MQINQ 命令の属性セレクタ (キュー用)

セレクタ	説明
MQCA_BASE_Q_NAME	ベースキュー名 (MQ_Q_NAME_LENGTH)
MQCA_CLUSTER_NAME	クラスタ名称 (MQ_CLUSTER_NAME_LENGTH)
MQCA_CREATION_DATE	作成日付 (MQ_CREATION_DATE_LENGTH)
MQCA_CREATION_TIME	作成時刻 (MQ_CREATION_TIME_LENGTH)
MQCA_INITIATION_Q_NAME	イニシエーションキュー名 (MQ_Q_NAME_LENGTH)
MQCA_PROCESS_NAME	プロセス名 (MQ_PROCESS_NAME_LENGTH)
MQCA_Q_DESC	キュー記述子 (MQ_Q_DESC_LENGTH)
MQCA_Q_NAME	キュー名 (MQ_Q_NAME_LENGTH)
MQCA_REMOTE_Q_MGR_NAME	リモートキューマネージャ名 (MQ_Q_MGR_NAME_LENGTH)
MQCA_REMOTE_Q_NAME	リモートキュー名 (MQ_Q_NAME_LENGTH)
MQCA_TRIGGER_DATA	トリガデータ (MQ_TRIGGER_DATA_LENGTH)
MQCA_XMIT_Q_NAME	転送キュー名 (MQ_Q_NAME_LENGTH)
MQIA_CURRENT_Q_DEPTH	メッセージ登録数
MQIA_DEFINITION_TYPE	定義タイプ
MQIA_DEF_BIND	デフォルトバインディング
MQIA_DEF_INPUT_OPEN_OPTION	省略時の取り出しオプション
MQIA_DEF_PERSISTENCE	省略時のメッセージ永続性
MQIA_DEF_PRIORITY	省略時のメッセージ優先度
MQIA_DIST_LISTS	配布リストサポート
MQIA_HARDEN_GET_BACKOUT	ロールバック回数記録方法
MQIA_INHIBIT_GET	取り出し許可

セレクタ	説明
MQIA_INHIBIT_PUT	登録許可
MQIA_MAX_MSG_LENGTH	最大メッセージ長
MQIA_MAX_Q_DEPTH	最大メッセージ登録数
MQIA_MSG_DELIVERY_SEQUENCE	メッセージ配布順序
MQIA_OPEN_INPUT_COUNT	入力オープン数
MQIA_OPEN_OUTPUT_COUNT	出力オープン数
MQIA_Q_TYPE	キュータイプ
MQIA_RETENTION_INTERVAL	キュー保持時間
MQIA_SCOPE	キューの有効範囲
MQIA_SHAREABILITY	共用性
MQIA_TRIGGER_CONTROL	トリガ制御
MQIA_TRIGGER_DEPTH	トリガのためのメッセージ登録数の下限値
MQIA_TRIGGER_MSG_PRIORITY	トリガのためのメッセージ優先度の下限値
MQIA_TRIGGER_TYPE	トリガタイプ
MQIA_USAGE	使用種別

表 2-3 MQINQ 命令の属性セレクタ (プロセス定義用)

セレクタ	説明
MQCA_APPL_ID	アプリケーション識別子 (MQ_PROCESS_APPL_ID_LENGTH)
MQCA_ENV_DATA	環境データ (MQ_PROCESS_ENV_DATA_LENGTH)
MQCA_PROCESS_DESC	プロセス記述子 (MQ_PROCESS_DESC_LENGTH)
MQCA_PROCESS_NAME	プロセス名 (MQ_PROCESS_NAME_LENGTH)
MQCA_USER_DATA	ユーザデータ (MQ_PROCESS_USER_DATA_LENGTH)
MQIA_APPL_TYPE	アプリケーションタイプ

表 2-4 MQINQ 命令の属性セレクタ (キューマネージャ用)

セレクタ	説明
MQCA_DEAD_LETTER_Q_NAME	デッドレターキュー名

セレクタ	説明
MQCA_DEAD_LETTER_Q_NAME	(MQ_Q_NAME_LENGTH)
MQCA_DEF_XMIT_Q_NAME	省略時の転送キュー名 (MQ_Q_NAME_LENGTH)
MQIA_DIST_LISTS	配布リストサポート
MQCA_Q_MGR_DESC	キューマネージャ記述子 (MQ_Q_MGR_DESC_LENGTH)
MQCA_Q_MGR_NAME	キューマネージャ名 (MQ_Q_MGR_NAME_LENGTH)
MQIA_CODED_CHAR_SET_ID	文字セット識別子
MQIA_MAX_HANDLES	最大ハンドル数
MQIA_MAX_MSG_LENGTH	最大メッセージ長
MQIA_MAX_PRIORITY	最大メッセージ優先度
MQIA_MAX_UNCOMMITTED_MSGS	トランザクション完了待ち最大メッセージ数*
MQIA_PLATFORM	プラットフォーム
MQIA_SYNCPOINT	トランザクション有効性
MQIA_TRIGGER_INTERVAL	トリガ間隔
MQCA_REPOSITORY_NAME	キューマネージャがリポジトリサービスを提供するクラスタの名前

#### 注※

TP1/Message Queue ではこの属性は使用されないため、返された値に意味はありません。

#### ● IntAttrCount (MQLONG 型) -input

整数型属性値の数です。

IntAttrs の配列の要素数を指定します。0 も指定できます。

これが Selectors 引数の MQIA\_\*セレクタの数と等しいとき、すべて整数型属性値が返されます。

#### ● IntAttrs (MQLONG 型×IntAttrCount) -output

整数型属性値の配列です。

IntAttrCount の要素数の配列で整数型属性値が返されます。

整数型属性値は、Selectors 引数の MQIA\_\*セレクタで指定した順序で返されます。MQIA\_\*セレクタの数よりも大きい配列を用意した場合、不要な部分に変更されません。

Hobj 引数がキューを示す場合で、属性のセレクタがそのキュータイプに適合しないとき、IntAttrs の配列の該当する要素として、MQIAV\_NOT\_APPLICABLE が返されます。

IntAttrCount 引数または SelectorCount 引数の値が 0 の場合、IntAttrs 引数は参照されません。この場合、C 言語のプログラムでは引数のアドレスとしてヌルを指定できます。

### ● CharAttrLength (MQLONG 型) -input

文字型属性値の格納領域長です。

CharAttrs の格納領域長をバイト単位で指定します。

照会する文字型属性値の長さの合計を指定してください。文字型属性値の長さについては、「Selectors 引数」の説明を参照してください。0 も指定できます。

### ● CharAttrs (MQCHAR 型×CharAttrLength) -output

文字型属性値の格納領域です。

この格納領域に、文字型の属性がつながれて返されます。バッファの長さは CharAttrLength 引数の値です。

文字型属性値は、Selectors 引数の MQCA\_\*セクタで指定した順序で返されます。属性の文字列の長さは、各属性に固有です。文字型属性値の長さについては、「Selectors 引数」の説明を参照してください。値の余りは空白で埋められます。空白による埋めを含めて、照会する属性の文字列よりも格納領域が大きい場合、不要な部分は変更されません。

Hobj 引数がキューを示す場合で、属性のセクタがそのキュータイプに適合しないとき、CharAttrs 引数の格納領域の該当する部分にアスタリスク (\*) が設定されます。

CharAttrLength 引数または SelectorCount 引数の値が 0 の場合、CharAttrs 引数は参照されません。この場合、C 言語のプログラムでは引数のアドレスとしてヌルを指定できます。

### ● CompCode (MQLONG 型) -output

完了コードです。

次のどれかが返されます。

- MQCC\_OK : 成功
- MQCC\_WARNING : 警告 (一部成功)
- MQCC\_FAILED : 失敗

### ● Reason (MQLONG 型) -output

理由コードです。

CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

## CompCode 引数が MQCC\_WARNING の場合

理由コード	数値	意味
MQRC_CHAR_ATTRS_TOO_SHORT	2008	文字型属性値に対して十分な領域がありません。
MQRC_INT_ATTR_COUNT_TOO_SMALL	2022	整数型属性値に対して十分な領域がありません。
MQRC_SELECTOR_NOT_FOR_TYPE	2068	セレクタがキュータイプに適合しません。

## CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_CHAR_ATTR_LENGTH_ERROR	2006	文字型属性長が不正です。
MQRC_CHAR_ATTRS_ERROR	2007	文字型属性値が不正です。
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_HOBJ_ERROR	2019	オブジェクトハンドルが不正です。
MQRC_INT_ATTR_COUNT_ERROR	2021	整数型属性数が不正です。
MQRC_INT_ATTRS_ARRAY_ERROR	2023	整数型属性値が不正です。
MQRC_NOT_OPEN_FOR_INQUIRE	2038	キューが照会用にオープンされていません。
MQRC_OBJECT_DAMAGED	2101	オブジェクトが破損しています。
MQRC_Q_DELETED	2052	キューが削除されています。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_SELECTOR_COUNT_ERROR	2065	セレクタ数が不正です。
MQRC_SELECTOR_ERROR	2067	セレクタが不正です。
MQRC_SELECTOR_LIMIT_EXCEEDED	2066	セレクタ数が最大値を超えています。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。

詳細は、「付録 B.2 理由コード」を参照してください。

## 注意事項

1. 返された情報は、指定した属性のその時点での値です。返された値をアプリケーションが使用する前に、その属性値が変更されていない保証はありません。
2. モデルキューをオープンするとき、ローカルの動的キューが生成されます。属性を照会するためにモデルキューをオープンする場合にも該当します。

動的キューの属性は（幾つか例外はありますが）、動的キューが生成されるときモデルキューの属性と同じです。その後、MQINQ 命令をキューに使用すると、キューマネージャはモデルキューではなく、動的キューの属性を返します。

動的キューがモデルキューから継承する属性については、「3. オブジェクトの属性」の「[キューの属性](#)」を参照してください。
3. 別名キューに対する MQINQ 命令で返される属性は、そのベースキューではなく、別名キューの属性です。
4. 照会されたキューがクラスタキューである場合、照会できる属性はキューのオープンの方法に依存します。
  - クラスタキューが照会と、入力、ブラウズ、または設定のどれか一つと一緒にオープンされた場合、オープンが成功するためにはクラスタキューのローカルなインスタンスがなければいけません。
  - クラスタキューが照会だけ、または照会と出力でオープンされた場合、次に示す属性だけを照会できます。なお、QType 属性はこの場合 MQQT\_CLUSTER となります。

MQCA\_Q\_DESC

MQCA\_Q\_NAME

MQIA\_DEF\_BIND

MQIA\_DEF\_PERSISTENCE

MQIA\_DEF\_PRIORITY

MQIA\_INHIBIT\_PUT

MQIA\_Q\_TYPE

クラスタキューが固定バイディングなしでオープンされた場合、つまり、MQOPEN 命令で MQOO\_BIND\_NOT\_FIXED が指定された場合、または DefBind 属性が MQBND\_BIND\_NOT\_FIXED の場合に MQOO\_BIND\_AS\_Q\_DEF が指定されたとき、そのキューに対する後続の MQINQ 命令では、クラスタキューの異なるインスタンスを照会することもあります。通常はすべてのインスタンスは同じ属性値を持ちます。

5. 属性を照会したあとにそれらの幾つかを MQSET 命令で設定する場合、始めにセレクタの配列を作成すると便利です。こうすることで、要素数を減らして、同じ配列を MQSET 命令でも使用できます。
6. 一つ以上の警告がある場合、次のどれかの理由コードが返されます。警告については、「[CompCode 引数](#)」を参照してください。

MQRC\_SELECTOR\_NOT\_FOR\_TYPE

MQRC\_INT\_ATTR\_COUNT\_TOO\_SMALL

MQRC\_CHAR\_ATTRS\_TOO\_SHORT

7. 各オブジェクトタイプの属性については、「[3. オブジェクトの属性](#)」を参照してください。
8. TP1/Message Queue では、MaxUncommittedMsgs 属性は使用されません。このため、MQIA\_MAX\_UNCOMMITTED\_MSGS セレクタで返される属性値には、意味がありません。なお、このセレクタを使用すると、常に 999999999 が返されます。

# MQOPEN 命令 – オブジェクトのオープン

## 機能

MQOPEN 命令で、そのオブジェクトの利用を確立します。次の種類のオブジェクトをオープンできます。

- キュー（配布リストを含む）
- プロセス定義
- キューマネージャ

## 形式

### C 言語の場合

```
MQOPEN(MQHCONN Hconn, PMQVOID ObjDesc, MQLONG Options,  
        PMQHOBj Hobj, PMQLONG CompCode, PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQOPEN' USING HCONN, OBJDESC, OPTIONS, HOBJ,  
                   COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。

ただし、MQOPEN 命令では、Hconn 引数の値として MQCONN 命令の戻り値以外を指定してもエラーにはなりません。それでも、MQCONN 命令は必ず発行しなければならないため、MQCONN 命令の戻り値を必ず指定してください。また、MQOPEN 命令以降に発行する命令でも、同様に MQCONN 命令の戻り値を指定してください。

### ● ObjDesc (MQOD 構造体) –input/output

オブジェクト記述子です。

オープンするオブジェクトを識別する構造体です。詳細は、「1. データタイプ」の「MQOD 構造体 – オブジェクト記述子」を参照してください。

ObjDesc 引数の ObjectName フィールドがモデルキューの名称の場合、モデルキューの属性で動的キューが生成されます。これは、Options 引数でどのオープンオプションを指定しても同様です。この MQOPEN 命令で返された Hobj を使用すると、モデルキューではなく、生成された動的キューを利用することになります。これは、MQINQ 命令と MQSET 命令のときも同様です。ObjDesc 引数のモデルキューの名称は、生成された動的キューの名称に置き換えられます。動的キューのタイプは、モデルキューの

DefinitionType 属性で決まります。動的キューに対するクローズオプションについては、この章の「MQCLOSE 命令 - オブジェクトのクローズ」を参照してください。

## ● Options (MQLONG 型) -input

MQOPEN 命令の動作を制御するオープンオプションです。

少なくとも、次の一つのオプションを指定してください。

- MQOO\_BROWSE
- MQOO\_INPUT\_\* (これらのオプションのうち一つだけ)
- MQOO\_INQUIRE
- MQOO\_OUTPUT
- MQOO\_SET

これらのオプションの意味を次に示します。また、ほかのオプションも指定できます。複数のオプションを指定する場合は、次のどちらかに従ってください。

- それぞれを加算します。ただし、同じ値は 2 回以上加算しないでください。
- プログラミング言語がビット演算をサポートしている場合に、それぞれのビット論理和を取ります。

同時に指定できない組み合わせは、そのつど示します。それ以外の組み合わせは自由です。ObjDesc のオブジェクトに合ったオプションを指定してください。オプションおよびその指定方法の一覧については、「付録 A 命令および引数の一覧」を参照してください。

### アクセスオプション：

次に示すオプションはオブジェクトに実行される操作の種類を制御します。

#### MQOO\_INPUT\_AS\_Q\_DEF

キューの定義を省略するときの取り出しオプションで、取り出し用にオープンします。

以降の MQGET 命令で使用するために、キューをオープンします。共用アクセスで取り出すか排他アクセスで取り出すかは、キューの DefInputOpenOption 属性の値で決まります。

このオプションは、ローカルキュー、別名キュー、およびモデルキューのときだけ有効です。リモートキュー、配布リスト、キューではないオブジェクトについては不正となります。

#### MQOO\_INPUT\_SHARED

共用アクセスの取り出し用にオープンします。

以降の MQGET 命令で使用するために、キューをオープンします。そのキューがほかのアプリケーションによってすでにオープンされていても、MQOO\_INPUT\_SHARED オプションでオープンされている場合は、命令は成功します。MQOO\_INPUT\_EXCLUSIVE オプションでオープンされている場合は、理由コード MQRC\_OBJECT\_IN\_USE で命令は失敗します。

このオプションは、ローカルキュー、別名キュー、およびモデルキューのときだけ有効です。リモートキュー、配布リスト、キューではないオブジェクトについては不正となります。

## MQOO\_INPUT\_EXCLUSIVE

排他アクセスの取り出し用にオープンします。

以降の MQGET 命令で使用するために、キューをオープンします。そのキューがほかのアプリケーションによってすでに取り出し用にオープンされていると、理由コード MQRC\_OBJECT\_IN\_USE で命令は失敗します。

このオプションは、ローカルキュー、別名キュー、およびモデルキューのときだけ有効です。リモートキュー、配布リスト、キューではないオブジェクトについては不正となります。

これらのオプションに対しては、次の注意があります。

- これらのオプションのうち、一つだけ指定できます。
- キューの InhibitGet 属性が MQQA\_GET\_INHIBITED の場合でも、MQOPEN 命令はこれらのオプションで成功します。ただし、この属性値が設定されている間は、MQGET 命令は失敗します。
- キューが共用アクセス用に定義されていない場合、つまり、ローカルキューの Shareability 属性が MQQA\_NOT\_SHAREABLE の場合、それに対して共用アクセス用にオープンしようとする、排他アクセス用にオープンするときと同様に扱われます。
- 別名キューをこれらのオプションでオープンすると、排他アクセスできるかどうか、つまり、ほかのアプリケーションが排他アクセスで使っていないかどうかの確認は、別名キューのベースキューに対してされます。
- ObjectQMgrName フィールドがキューマネージャの別名の場合、これらのオプションは指定できません。また、キューマネージャの別名として使用されるリモートキューのローカル定義の RemoteQMgrName 属性がローカルキューマネージャ名の場合も同様です。名称解決については、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。

## MQOO\_BROWSE

検索用にオープンします。

次のどれかのオプションで MQGET 命令を使用するために、キューをオープンします。

**MQGMO\_BROWSE\_FIRST**

**MQGMO\_BROWSE\_NEXT**

**MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR**

これは、すでに MQOO\_INPUT\_EXCLUSIVE オプションでオープンされているキューに対しても、指定できます。MQOO\_BROWSE オプションで MQOPEN 命令を呼び出すと、検索カーソルが生成されます。検索カーソルの論理的な位置は、キューにある最初のメッセージの前になります。詳細については、「1. データタイプ」の「[MQGMO 構造体 - メッセージ取り出しオプション](#)」で Options フィールドの説明を参照してください。

このオプションは、ローカルキュー、別名キュー、およびモデルキューのときだけ有効です。リモートキュー、配布リスト、キューではないオブジェクトについては不正となります。ObjectQMgrName フィールドがキューマネージャの別名の場合、このオプションは指定できません。また、キューマネージャの別名として使用されるリモートキューのローカル定義の RemoteQMgrName 属性がローカルキューマネージャ名の場合も同様です。名称解決については、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。

## MQOO\_OUTPUT

登録用にオープンします。

以降の MQPUT 命令で使用するために、キューをオープンします。

キューの InhibitPut 属性が MQQA\_PUT\_INHIBITED の場合でも、MQOPEN 命令はこのオプションで成功します。ただし、この属性値が設定されている間は、MQPUT 命令は失敗します。

このオプションは、配布リストを含むすべてのキューに有効です。

## MQOO\_INQUIRE

属性の照会用にオープンします。

以降の MQINQ 命令で使用するために、キュー、プロセス定義、またはキューマネージャをオープンします。

このオプションは、配布リストを除くすべてのオブジェクトタイプに有効です。ObjectQMgrName フィールドがキューマネージャの別名の場合は、このオプションは指定できません。これは、キューマネージャの別名として使用されるリモートキューのローカル定義の RemoteQMgrName 属性がローカルキューマネージャ名の場合も同様です。名称解決については、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。

## MQOO\_SET

属性の設定用にオープンします。

以降の MQSET 命令で使用するために、キューをオープンします。

このオプションは、配布リストを除くすべてのキューに有効です。ObjectQMgrName フィールドがキューマネージャの別名の場合、このオプションは指定できません。また、キューマネージャの別名として使用されるリモートキューのローカル定義の RemoteQMgrName 属性がローカルキューマネージャ名の場合も同様です。名称解決については、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。

## バインディングオプション：

次に示すオプションは、オープンされたオブジェクトがクラスタキューである場合に適用されます。これらのオプションは、キューハンドルのバインディングを、クラスタキューの特定のインスタンスへと制御します。

## MQOO\_BIND\_ON\_OPEN

キューがオープンされたときにハンドルをあて先にバインドします。

このオプションを指定すると、キューがオープンされたとき、ローカルキューマネージャはキューハンドルを特定のあて先キューに固定します。その結果、このハンドルを使用して登録されるすべてのメッセージは、あて先キューの同じインスタンスへ同じルートで送信されます。

このオプションはキューにだけ使用でき、クラスタキューにだけ意味があります。

クラスタキューではないキューに指定された場合、このオプションは無視されます。

## MQOO\_BIND\_NOT\_FIXED

特定のあて先にバインドしません。

このオプションを指定すると、ローカルキューマネージャがキューハンドルを特定のあて先キューに固定しないようにします。その結果、このハンドルを使用する後続の MQPUT 命令では、メッセージはあて先キューの異なるインスタンスに送信されるか、同じインスタンスであっても異なるルートで送信されます。ローカルキューマネージャ、リモートキューマネージャ、または MCA によって、ネットワークの条件に従って、あて先がそのつど変更されます。

## 注意事項

トランザクションを完了させるために、連続するメッセージを交換する必要のあるクライアントおよびサーバアプリケーションは、MQOO\_BIND\_NOT\_FIXED、または DefBind 属性が MQBND\_BIND\_NOT\_FIXED の場合の MQOO\_BIND\_AS\_Q\_DEF を指定しないでください。なぜなら、連続している後続メッセージはサーバアプリケーションの異なるあて先に送信される可能性があるためです。

MQOO\_BROWSE または MQOO\_INPUT\_\* オプションの一つがクラスタキューに指定された場合、キューマネージャはクラスタキューの中でローカルなインスタンスを選択します。その結果、MQOO\_BIND\_NOT\_FIXED が指定されていても、キューハンドルのバインディングは固定されます。MQOO\_INQUIRE が MQOO\_BIND\_NOT\_FIXED とともに指定された場合、このハンドルを使用した後続の MQINQ では、クラスタキューの異なるインスタンスを照会することがあります。通常はすべてのインスタンスは同じ属性値を持ちます。

MQOO\_BIND\_NOT\_FIXED はキューに対してだけ使用でき、クラスタキューに対して意味がありません。クラスタキューではないキューに指定された場合、オプションは無視されます。

## MQOO\_BIND\_AS\_Q\_DEF

キューについてのデフォルトバインディングを使用します。

このオプションを指定すると、ローカルキューマネージャは、DefBind キュー属性で定義された方法でキューハンドルをバインドします。この属性の値は、MQBND\_BIND\_ON\_OPEN または MQBND\_BIND\_NOT\_FIXED のどちらかです。

MQOO\_BIND\_ON\_OPEN および MQOO\_BIND\_NOT\_FIXED のどちらも指定されていない場合、MQOO\_BIND\_AS\_Q\_DEF がデフォルト値となります。

MQOO\_BIND\_AS\_Q\_DEF は、プログラムの文書化のために定義されています。

このオプションは、ほかに二つあるバインドオプションのどちらかと一緒に指定されることは意図していません。しかし、値が 0 であるため、そのような指定を検知できません。

## コンテキストオプション：

次に示すオプションは、メッセージコンテキストの処理を制御します。

## MQOO\_SAVE\_ALL\_CONTEXT

取り出し時にコンテキストを保存します。

コンテキスト情報は、このキューハンドルに関連づけられます。このハンドルを使用して取り出したメッセージのコンテキストから、この情報が取り出されます。メッセージコンテキストについては、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。

このコンテキスト情報は、以降の MQPUT 命令および MQPUT1 命令で登録するメッセージに引き渡されます。コンテキスト情報の引き渡しについては、「1. データタイプ」の「MQPMO 構造体 – メッセージ登録オプション」で、Options フィールドの MQPMO\_PASS\_IDENTITY\_CONTEXT オプションと MQPMO\_PASS\_ALL\_CONTEXT オプションの説明を参照してください。

一度もメッセージを取り出さないうちは、登録するメッセージにコンテキストを引き渡せません。

MQGMO\_\*検索オプションを使用した取り出しでは、コンテキスト情報は保存されません。ただし、MsgDesc 引数のコンテキストフィールドには返されます。

このオプションは、ローカルキュー、別名キュー、およびモデルキューのときだけ有効です。リモートキュー、配布リスト、キューではないオブジェクトについては不正となります。

MQOO\_INPUT\_\*オプションのどれかと同時に指定してください。

## MQOO\_PASS\_IDENTITY\_CONTEXT

識別コンテキストを引き渡せるようにします。

このオプションを指定することで、メッセージ登録時の PutMsgOpts 引数に MQPMO\_PASS\_IDENTITY\_CONTEXT オプションを指定できるようになります。具体的には、MQOO\_SAVE\_ALL\_CONTEXT オプションでオープンした入力キューからの識別コンテキストの情報をメッセージに組み込みます。メッセージコンテキストについては、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。

MQOO\_OUTPUT オプションと同時に指定してください。

このオプションは、配布リストを含むすべてのキューに有効です。

## MQOO\_PASS\_ALL\_CONTEXT

全コンテキストを引き渡せるようにします。

このオプションを指定することで、メッセージ登録時の PutMsgOpts 引数に MQPMO\_PASS\_ALL\_CONTEXT オプションを指定できるようになります。具体的には、MQOO\_SAVE\_ALL\_CONTEXT オプションでオープンした入力キューからの識別コンテキストと登録元コンテキストの情報をメッセージに組み込みます。メッセージコンテキストについては、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。

このオプションは、MQOO\_PASS\_IDENTITY\_CONTEXT オプションの内容を含みます。このため、同時に MQOO\_PASS\_IDENTITY\_CONTEXT オプションを指定する必要はありません。

MQOO\_OUTPUT オプションと同時に指定してください。

このオプションは、配布リストを含むすべてのキューに有効です。

## MQOO\_SET\_IDENTITY\_CONTEXT

識別コンテキストを設定できるようにします。

このオプションを指定することで、メッセージ登録時の PutMsgOpts 引数に MQPMO\_SET\_IDENTITY\_CONTEXT オプションを指定できるようになります。具体的には、MQPUT 命令または MQPUT1 命令の MsgDesc 引数に指定した識別コンテキストの情報をメッセージに組み込みます。メッセージコンテキストについては、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。

このオプションは、MQOO\_PASS\_IDENTITY\_CONTEXT オプションの内容を含みます。このため、同時に MQOO\_PASS\_IDENTITY\_CONTEXT オプションを指定する必要はありません。MQOO\_OUTPUT オプションと同時に指定してください。

このオプションは、配布リストを含むすべてのキューに有効です。

## MQOO\_SET\_ALL\_CONTEXT

全コンテキストを設定できるようにします。

このオプションを指定することで、メッセージ登録時の PutMsgOpts 引数に MQPMO\_SET\_ALL\_CONTEXT オプションを指定できるようになります。具体的には、MQPUT 命令または MQPUT1 命令の MsgDesc 引数に指定した識別コンテキストと登録元コンテキストの情報をメッセージに組み込みます。メッセージコンテキストについては、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。

このオプションは、次のオプションの内容を含みます。このため、同時にこれらのオプションを指定する必要はありません。

MQOO\_PASS\_IDENTITY\_CONTEXT

MQOO\_PASS\_ALL\_CONTEXT

MQOO\_SET\_IDENTITY\_CONTEXT

MQOO\_OUTPUT オプションと同時に指定してください。

このオプションは、配布リストを含むすべてのキューに有効です。

## その他のオプション：

次に示すオプションは権限チェックおよびキューマネージャ停止時の動作を制御します。

## MQOO\_ALTERNATE\_USER\_AUTHORITY

代替ユーザ識別子を使用します。

オブジェクトを利用するユーザの識別子として、アプリケーションを実行しているユーザ識別子ではなく、ObjDesc 引数の AlternateUserId の代替ユーザ識別子が使用されます。

このオプションは、すべてのオブジェクトタイプに有効です。

## MQOO\_FAIL\_IF\_QUIESCING

TP1/Message Queue では意味がありません。

このオプションを指定しても、TP1/Message Queue の動作には影響しません。これは、アプリケーションの移植性を向上させるために定義されています。

次に、MQOPEN 命令のオプションとキュータイプとの関係を示します。

オプション	別名キュー ※1	ローカル キューおよびモデル キュー	リモート キュー	非ローカル クラスタ キュー	配布リスト
MQOO_INPUT_AS_Q_DEF	○	○	×	×	×
MQOO_INPUT_SHARED	○	○	×	×	×

オプション	別名キュー ※1	ローカル キューおよびモデル キュー	リモート キュー	非ローカル クラスタ キュー	配布リスト
MQOO_INPUT_EXCLUSIVE	○	○	×	×	×
MQOO_BROWSE	○	○	×	×	×
MQOO_OUTPUT	○	○	○	○	○
MQOO_INQUIRE	○	○	△	○	×
MQOO_SET	○	○	△	×	×
MQOO_BIND_ON_OPEN※2	○	○	○	○	○
MQOO_BIND_NOT_FIXED※2	○	○	○	○	○
MQOO_BIND_AS_Q_DEF※2	○	○	○	○	○
MQOO_SAVE_ALL_CONTEXT	○	○	×	×	×
MQOO_PASS_IDENTITY_CONTEXT	○	○	○	○	○
MQOO_PASS_ALL_CONTEXT	○	○	○	○	○
MQOO_SET_IDENTITY_CONTEXT	○	○	○	○	○
MQOO_SET_ALL_CONTEXT	○	○	○	○	○
MQOO_ALTERNATE_USER_AUTHORITY	○	○	○	○	○
MQOO_FAIL_IF QUIESCING	○	○	○	○	○

(凡例)

○：有効

×：無効

△：リモートキューのローカル定義にだけ有効

注※1

別名キューに対するオプションの有効性は、別名が解決されるキューでのオプションの有効性に依存します。

注※2

このオプションは任意のキュータイプに指定できますが、キューがクラスタキューではない場合は無視されます。

## ● Hobj (MQHOBJ型) -output

オブジェクトハンドルです。

利用するオブジェクトを示すハンドルです。以降の命令でこのオブジェクトを利用するときに、このハンドルを指定します。MQCLOSE 命令を呼び出すと、またはこのハンドルの範囲の処理が終了すると、このハンドルは無効になります。

返されるオブジェクトハンドルの有効範囲は、命令に指定された接続ハンドルと同じです。接続ハンドルの有効範囲については、この章の「MQCONN 命令 - キューマネージャへの接続」で Hconn 引数を参照してください。

## ● CompCode (MQLONG 型) -output

完了コードです。

次のどれかが返されます。

- MQCC\_OK : 成功
- MQCC\_WARNING : 警告 (部分的完了)
- MQCC\_FAILED : 失敗

## ● Reason (MQLONG 型) -output

理由コードです。

### CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

### CompCode 引数が MQCC\_WARNING の場合

理由コード	数値	意味
MQRC_MULTIPLE_REASONS	2136	複数のリターンコードが返されました。

### CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_CLUSTER_PUT_INHIBITED	2268	クラスタ内のすべてのキューに対して登録が禁止されています。
MQRC_CLUSTER_RESOLUTION_ERROR	2189	クラスタ名称の解決に失敗しました。
MQRC_CLUSTER_RESOURCE_ERROR	2269	クラスタリソースのエラーです。
MQRC_DEF_XMIT_Q_USAGE_ERROR	2199	省略時の転送キューの使用方法に誤りがあります。
MQRC_DYNAMIC_Q_NAME_ERROR	2011	動的キュー名が不正です。
MQRC_HANDLE_NOT_AVAILABLE	2017	現在そのハンドルは有効ではありません。
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_HOBJ_ERROR	2019	オブジェクトハンドルが不正です。
MQRC_MULTIPLE_REASONS	2136	複数の理由コードが返されました。

理由コード	数値	意味
MQRC_NAME_NOT_VALID_FOR_TYPE	2194	オブジェクトタイプに対してオブジェクト名が不正です。
MQRC_OBJECT_ALREADY_EXISTS	2100	オブジェクトがすでにあります。
MQRC_OBJECT_DAMAGED	2101	オブジェクトが破損しています。
MQRC_OBJECT_IN_USE	2042	オブジェクトは衝突するオプションですすでにオープンされています。
MQRC_OBJECT_NAME_ERROR	2152	オブジェクト名が不正です。
MQRC_OBJECT_Q_MGR_NAME_ERROR	2153	オブジェクトキューマネージャ名が不正です。
MQRC_OBJECT_RECORDS_ERROR	2155	オブジェクトレコードが不正です。
MQRC_OBJECT_TYPE_ERROR	2043	オブジェクトタイプが不正です。
MQRC_OD_ERROR	2044	オブジェクト記述子の構造体が不正です。
MQRC_OPTION_NOT_VALID_FOR_TYPE	2045	オブジェクトタイプに対してオプションが不正です。
MQRC_OPTIONS_ERROR	2046	オプションが不正です。または、指定されていません。
MQRC_Q_TYPE_ERROR	2057	キュータイプが不正です。
MQRC_RECS_PRESENT_ERROR	2154	存在するレコードの数が不正です。
MQRC_REMOTE_Q_NAME_ERROR	2184	リモートキュー名が不正です。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_RESPONSE_RECORDS_ERROR	2156	応答レコードが不正です。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。
MQRC_UNKNOWN_ALIAS_BASE_Q	2082	別名キューのベースキューを認識できません。
MQRC_UNKNOWN_DEF_XMIT_Q	2197	省略時の転送キューを認識できません。
MQRC_UNKNOWN_OBJECT_NAME	2085	オブジェクト名を認識できません。
MQRC_UNKNOWN_OBJECT_Q_MGR	2086	オブジェクトのキューマネージャを認識できません。
MQRC_UNKNOWN_REMOTE_Q_MGR	2087	リモートキューマネージャを認識できません。

理由コード	数値	意味
MQRC_UNKNOWN_XMIT_Q	2196	転送キューを認識できません。
MQRC_XMIT_Q_USAGE_ERROR	2092	転送キューの使用方法に誤りがあります。

詳細は、「付録 B.2 理由コード」を参照してください。

## 注意事項

1. オブジェクトをオープンする用途は、次のとおりです。

- キューをオープンする場合
  - メッセージの取り出し、または検索 (MQGET 命令)
  - メッセージの登録 (MQPUT 命令)
  - キューの属性の照会 (MQINQ 命令)
  - キューの属性の設定 (MQSET 命令)

モデルキューの名称を指定した場合、動的キューが生成されます。詳細については「ObjDesc 引数」の説明を参照してください。

配布リストはキューのリストを含む特別なキューオブジェクトです。メッセージ登録のためにだけオープンできます。次に示す場合はオープンできません。詳細については、「注意事項」の 6. を参照してください。

- メッセージの取り出し、または検索
  - キューの属性の照会、または設定
  - プロセス定義をオープンする場合
    - プロセス定義の属性の照会 (MQINQ 命令)
  - キューマネージャをオープンする場合
    - ローカルキューマネージャの属性の照会 (MQINQ 命令)
2. 一つのアプリケーションが、同じオブジェクトを 2 回以上オープンすることもできます。それぞれのオープンに対して、異なるハンドルが返されます。そのハンドルは、それをオープンした目的の機能でそれぞれ使用できます。
3. MQOPEN 命令を呼び出したときに、ローカルキューマネージャ内の実名で置き換えられます。つまり、次のようになります。
- 別名キューの実名は、ベースキューの名称になります。
  - リモートキューのローカル定義の実名は、リモートキューマネージャ名とそれに定義されるキューの名称になります。
  - リモートキューマネージャ名の実名は、転送キューの名称になります。

ただし、以降にそのハンドルを使用して MQINQ 命令または MQSET 命令を呼び出す場合でも、オープンしたときの名称を照会することに注意してください。実名に置き換えられた名称を照会するわけではありません。例えば、別名キューをオープンした場合、MQINQ 命令で返される属性は、別名キュー

の属性です。別名キューのベースキューの属性ではありません。実名のチェックはされますが、MQOPEN 命令の Options 引数の指定内容とは関係ありません。

オープンされるオブジェクトがクラスタキューである場合、名称解決は MQOPEN 命令発行時、またはあとで解決されます。名称解決が発生する場合は、MQOPEN 命令の次に示す MQOO\_BIND\_\* オプションで制御されます。

MQOO\_BIND\_ON\_OPEN

MQOO\_BIND\_NOT\_FIXED

MQOO\_BIND\_AS\_Q\_DEF

4. この命令の ObjDesc 引数にリモートキューを指定する方法に、次の 2 とおりがあります。詳細については、「1. データタイプ」の「MQOD 構造体 - オブジェクト記述子」で ObjectName フィールドと ObjectQMgrName フィールドの説明を参照してください。

- リモートキューのローカル定義の名称を ObjectName フィールドに指定します。この場合、ローカルキューマネージャを参照するための ObjectQMgrName フィールドには、空白または C 言語の場合はヌル文字列を指定できます。
- リモートキューマネージャが認識できるリモートキュー名を、ObjectName フィールドに指定します。この場合、ObjectQMgrName 属性にリモートキューマネージャの名称を指定します。

メッセージがリモートキューマネージャに到着したとき、登録元ユーザのユーザ識別子によっては、リモートキューマネージャがメッセージを拒否することがあります。

5. MQOO\_BROWSE オプションを指定して MQOPEN 命令を呼び出すと、検索カーソルが生成されます。検索カーソルは、オブジェクトハンドルと検索オプションを指定した MQGET 命令で使用します。これによって、内容を変更しないでキューを検索できます。メッセージを検索したあとに、MQGMO\_MSG\_UNDER\_CURSOR オプションを使用して、キューから削除できます。同じキューに対して何回か MQOPEN 命令を実行することで、一つのアプリケーションで複数の検索カーソルを持てます。

6. 配布リスト使用時には、次の点に注意してください。

- 配布リストをオープンするとき、MQOD 構造体のフィールドの値は、次の条件を満たさなければなりません。
  - Version フィールドの値は MQOD\_VERSION\_2 以上である。
  - ObjectType フィールドの値は MQOT\_Q である。
  - ObjectName フィールドは空白またはヌル文字列である。
  - ObjectQMgrName フィールドは空白またはヌル文字列である。
  - RecsPresent フィールドは 0 より大きい。
  - ObjectRecOffset フィールドおよび ObjectRecPtr フィールドのどちらかが 0 で、他方は 0 以外である。
  - ResponseRecOffset フィールドおよび ResponseRecPtr フィールドは両方とも 0 か、どちらかが 0 で他方は 0 以外である。

・ ObjectRecOffset フィールドまたは ObjectRecPtr フィールドのどちらかで指定される RecsPresent フィールドのオブジェクトレコードが存在しなければならない。オブジェクトレコードは、オープンするあて先キューの名称を設定しなければならない。

・ ResponseRecOffset フィールドおよび ResponseRecPtr フィールドのどちらかが 0 以外の場合、RecsPresent フィールドの応答レコードが存在しなければならない。これは、命令が MQRC\_MULTIPLE\_REASONS の理由コードで完了すると、キューマネージャによって設定される。バージョン 2 の MQOD 構造体は、RecsPresent フィールドが 0 であることを保証すると、配布リストではない単一のキューをオープンするときも使用できます。

- Options 引数には、次のオプションだけが指定できます。

MQOO\_OUTPUT

MQOO\_PASS\_\*\_CONTEXT

MQOO\_SET\_\*\_CONTEXT

MQOO\_ALTERNATE\_USER\_AUTHORITY

MQOO\_FAIL\_IF\_QUIESCING

- 配布リストのあて先キューは、ローカルキュー、別名キュー、またはリモートキューです。モデルキューは指定できません。モデルキューが指定されると、キューのオープンが理由コード MQRC\_Q\_TYPE\_ERROR で失敗します。ただし、リストにあるほかのキューのオープンを妨げることはしません。

- 完了コードおよび理由コードの引数は、次のように設定されます。

・ 配布リストのキューに対する操作が同じ方法で、すべて成功またはすべて失敗した場合、完了コードと理由コードの引数は、共通の結果を記述するように設定されます。アプリケーションによって指定されている場合も、MQRR 構造体の応答レコードは設定されません。

例えば、オープンが成功すると、完了コードに MQCC\_OK、理由コードに MQRC\_NONE がそれぞれ設定されます。キューが存在しないで、オープンに失敗した場合、引数は MQCC\_FAILED および MQRC\_UNKNOWN\_OBJECT\_NAME に設定されます。

・ 配布リストのキューに対するオープン操作が、同じ方法で部分的に成功または失敗した場合は次のとおりです。

— 少なくとも一つのオープンに成功すると、完了コードの引数には MQCC\_WARNING が設定されます。すべてが失敗すると MQCC\_FAILED が設定されます。

— 理由コードの引数は、MQRC\_MULTIPLE\_REASONS に設定されます。

— 応答レコードがアプリケーションによって指定されている場合は、配布リストのキューに対して個別の完了コードと理由コードを設定します。

- 配布リストのオープンが成功すると、返却された Hobj ハンドルは、配布リストのキューにメッセージを登録するための次の MQPUT 命令で使用できます。また、配布リストへのアクセスを廃止するために、MQCLOSE 命令でも使用できます。配布リストをクローズするのに適当なオプションは MQCO\_NONE だけです。

MQPUT1 命令は、配布リストにメッセージを登録するときも使用できます。リスト中のキューを定義する MQOD 構造体は、MQPUT1 命令の引数として指定されます。

- 配布リスト中のオープンに成功したあて先は、アプリケーションが一つのタスクでオープンできる最大のハンドル数を越えたかどうかチェックするとき、別々のハンドルとしてカウントされます。オープンできる最大のハンドル数については「3. オブジェクトの属性」の「キューマネージャの属性」で MaxHandles 属性を参照してください。

配布リスト中の二つ以上のあて先が、同じ物理キューに受信されるときも同様に処理されます。

オープンに成功したあて先は、OpenOutputCount 属性の値を 1 ずつ増加させます。

7. クラスタキューを使用するときには、次の点に注意してください。

- クラスタキューが最初にオープンされた場合で、かつローカルキューマネージャがフルリポジトリキューマネージャではない場合、ローカル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クラスタキューについての情報を取得します。ネットワークがビジー状態の場合、ローカルキューマネージャがリポジトリキューマネージャから必要な情報を受信するために数秒掛かるときがあります。その結果、MQOPEN 命令を発行したアプリケーションが、MQOPEN 命令がリターンするまでに 10 秒以上待つことがあります。ローカルキューマネージャがこの時間内にクラスタキューについての必要な情報を受信しないと、命令は MQRC\_CLUSTER\_RESOLUTION\_ERROR で失敗します。
- クラスタキューがオープンされ、クラスタ内に複数のキューインスタンスがある場合、実際にオープンされるインスタンスは MQOPEN 命令で指定されるオプションに依存します。
  - ・次に示すどれかのオプションを指定したとき、オープンされるクラスタキューインスタンスは、ローカルインスタンスでなければいけません。キューのローカルインスタンスがないとき、MQOPEN 命令は失敗します。

MQOO\_BROWSE

MQOO\_INPUT\_AS\_Q\_DEF

MQOO\_INPUT\_EXCLUSIVE

MQOO\_INPUT\_SHARED

MQOO\_SET

・上記のオプションは指定されていない場合で、MQOO\_INQUIRE と MQOO\_OUTPUT の一方または両方を指定するとき、ローカルインスタンスがオープンされます。ローカルインスタンスがなければリモートインスタンスがオープンされます。

# MQPUT 命令 – メッセージの登録

---

## 機能

MQPUT 命令で、メッセージをキューまたは配布リストへ登録できます。このとき、キューはオープンされている必要があります。

## 形式

### C 言語の場合

```
MQPUT(MQHCONN Hconn, MQHOBJ Hobj, PMQVOID MsgDesc,  
      PMQVOID PutMsgOpts, MQLONG BufferLength,  
      PMQVOID Buffer, PMQLONG CompCode, PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQPUT' USING HCONN, HOBJ, MSGDESC, PUTMSGOPTS,  
                  BUFFERLENGTH, BUFFER, COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。

### ● Hobj (MQHOBJ 型) –input

オブジェクトハンドルです。

メッセージを登録するキューを示すハンドルです。MQOO\_OUTPUT オプションを指定した MQOPEN 命令の戻り値を指定してください。

### ● MsgDesc (MQMD 構造体) –input/output

メッセージ記述子です。

送信するメッセージの属性を記述する構造体です。また、登録後のメッセージに関する情報が返されます。

詳細については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」を参照してください。

アプリケーションがバージョン 1 の MQMD 構造体を指定すると、返却されるメッセージは、バージョン 2 の MQMD 構造体にだけ存在するフィールドの値を指定するために、アプリケーションメッセージデータの前に MQMDE 構造体を付加できます。そのとき、MQMDE 構造体が存在することを示すために、MQMD 構造体の Format フィールドには、MQFMT\_MD\_EXTENSION を設定しなければなりません。

詳細については、「1. データタイプ」の「MQMDE 構造体 – メッセージ記述子の拡張」を参照してください。

### ● PutMsgOpts (MQPMO 構造体) –input/output

メッセージ登録オプションです。

詳細については、「1. データタイプ」の「MQPMO 構造体 – メッセージ登録オプション」を参照してください。

### ● BufferLength (MQLONG 型) –input

メッセージデータを格納したバッファのバイト長です。

0 も指定できます。このとき、アプリケーションデータのないメッセージを登録するものとみなされます。BufferLength の上限は状況によって異なります。

- あて先キューが共有キューの場合、上限は 63 キロバイト (64512 バイト) です。
- あて先キューがローカルキューの場合、またはローカルキューの別名を使用している場合は、ローカルキューまたはキューマネージャの MaxMsgLength 属性のうち、小さい方の値を超えて指定できません。
- あて先キューがリモートキューの場合、またはリモートキューの別名を使用している場合は、次に示すローカルキューまたはキューマネージャの MaxMsgLength 属性のうち、小さい方の値を超えて指定できません。

1. ローカルキューマネージャで一時的に格納するために使用される転送キュー
2. ローカルキューからあて先キューまでの経路で仲介するキューマネージャで使用される転送キュー
3. あて先キューマネージャで使用されるあて先キュー

なお、メッセージが転送キューに格納されるとき、メッセージに付加情報が追加されます。このため、送信できるアプリケーションデータの量が少なくなります。BufferLength 引数の値を決めるときは、転送キューの MaxMsgLength 属性の値から MQ\_MSG\_HEADER\_LENGTH のバイト数を引いた値を参考にしてください。

#### ■ 注意事項

上記の 1. の場合だけ、メッセージの登録直後にバッファ長のエラー（理由コード MQRC\_MSG\_TOO\_BIG\_FOR\_Q または MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR）が判明します。2., 3. の場合は、登録直後はエラーが判明しません。2., 3. でエラーが判明した場合、仲介するキューマネージャまたはあて先キューマネージャのデッドレターキューに、メッセージが転送されます。送信側で要求している場合は、エラー発生時に報告メッセージが生成されます。

### ● Buffer (MQBYTE 型×BufferLength) –input

メッセージデータを格納したバッファです。バッファには、メッセージ中のデータに適したバウンダリ調整が必要です。多くのメッセージ（MQ ヘッダ構造体のあるメッセージを含む）には 4 バイト単位の調整

が適していますが、より厳しい調整が必要なメッセージもあります。例えば、64ビット2進整数を含めるメッセージには8バイト単位のバウンダリ調整が必要です。

バッファ内に文字または数値がある場合、MsgDesc 引数の CodedCharSetId フィールドと Encoding フィールドに、適切な値を指定してください。これらの値によって、受信側でデータを適切な文字セットとマシンコード形式に変換できます。

## 注意事項

MQPUT 命令のそのほかの引数は、ローカルキューマネージャの文字セットとマシンコード形式、つまり、ローカルキューの CodedCharSetId 属性と MQENC\_NATIVE で指定してください。

C 言語では、この引数を void 型のポインタとして指定します。したがって、どんなデータタイプのアドレスでも、引数として指定できます。

BufferLength 引数に 0 を指定した場合、Buffer 引数は参照されません。したがって、C 言語のプログラムでは、この引数のアドレスとしてヌルを指定できます。

### ● CompCode (MQLONG 型) -output

完了コードです。

次のどれかが返されます。

- MQCC\_OK : 成功
- MQCC\_WARNING : 警告 (一部成功)
- MQCC\_FAILED : 失敗

### ● Reason (MQLONG 型) -output

理由コードです。

CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

CompCode 引数が MQCC\_WARNING の場合

理由コード	数値	意味
MQRC_INCOMPLETE_GROUP	2241	メッセージグループが完全ではありません。
MQRC_INCOMPLETE_MSG	2242	論理メッセージが完全ではありません。
MQRC_INCONSISTENT_PERSISTENCE	2185	永続性指定が矛盾しています。

理由コード	数値	意味
MQRC_INCONSISTENT_UOW	2245	コミット単位の指定が矛盾しています。
MQRC_MULTIPLE_REASONS	2136	複数の理由コードが返されました。
MQRC_PRIORITY_EXCEEDS_MAXIMUM	2049	メッセージ優先度が最大値を超えています。
MQRC_UNKNOWN_REPORT_OPTION	2104	メッセージ記述子中の報告オプションを認識できません。

## CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_BUFFER_ERROR	2004	バッファの引数が不正です。
MQRC_BUFFER_LENGTH_ERROR	2005	バッファ長の引数が不正です。
MQRC_CLUSTER_RESOLUTION_ERROR	2189	クラスタ名称の解決に失敗しました。
MQRC_CLUSTER_RESOURCE_ERROR	2269	クラスタリソースのエラーです。
MQRC_CONTEXT_HANDLE_ERROR	2097	参照先のキューハンドルにコンテキストが保存されていません。
MQRC_CONTEXT_NOT_AVAILABLE	2098	参照先のキューハンドルのコンテキストが不正です。
MQRC_DH_ERROR	2135	配布リストの構造体が不正です。
MQRC_EXPIRY_ERROR	2013	メッセージ保持時間が不正です。
MQRC_FEEDBACK_ERROR	2014	報告メッセージ返答コードが不正です。
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_HOBJ_ERROR	2019	オブジェクトハンドルが不正です。
MQRC_INCOMPLETE_GROUP	2241	メッセージグループが完全ではありません。
MQRC_INCOMPLETE_MSG	2242	論理メッセージが完全ではありません。
MQRC_INCONSISTENT_PERSISTENCE	2185	永続性指定が矛盾しています。
MQRC_INCONSISTENT_UOW	2245	コミット単位の指定が矛盾しています。
MQRC_MD_ERROR	2026	メッセージ記述子が不正です。
MQRC_MDE_ERROR	2248	メッセージ記述子の拡張子が不正です。
MQRC_MISSING_REPLY_TO_Q	2027	応答キューがありません。
MQRC_MSG_FLAGS_ERROR	2249	メッセージフラグが不正です。

理由コード	数値	意味
MQRC_MSG_SEQ_NUMBER_ERROR	2250	メッセージシーケンス番号が不正です。
MQRC_MSG_TOO_BIG_FOR_Q	2030	メッセージ長がキューに定義された最大値を超えています。
MQRC_MSG_TYPE_ERROR	2029	メッセージ記述子中のメッセージタイプが不正です。
MQRC_MULTIPLE_REASONS	2136	複数の理由コードが返されています。
MQRC_NO_DESTINATIONS_AVAILABLE	2270	利用可能なあて先キューが存在しません。
MQRC_NOT_OPEN_FOR_OUTPUT	2039	キューが登録用にオープンされていません。
MQRC_NOT_OPEN_FOR_PASS_ALL	2093	キューが全コンテキスト引き渡し用にオープンされていません。
MQRC_NOT_OPEN_FOR_PASS_IDENT	2094	キューが識別コンテキスト引き渡し用にオープンされていません。
MQRC_NOT_OPEN_FOR_SET_ALL	2095	キューが全コンテキスト設定用にオープンされていません。
MQRC_NOT_OPEN_FOR_SET_IDENT	2096	キューが識別コンテキスト設定用にオープンされていません。
MQRC_OBJECT_DAMAGED	2101	オブジェクトが破損しています。
MQRC_OFFSET_ERROR	2251	メッセージセグメントのオフセットが不正です。
MQRC_OPEN_FAILED	2137	キューのオープンに失敗しました。
MQRC_OPTIONS_ERROR	2046	オプションが不正です。または、指定されていません。
MQRC_ORIGINAL_LENGTH_ERROR	2252	元のメッセージの長さが不正です。
MQRC_PCF_ERROR	2149	PCF 構造体が不正です。
MQRC_PERSISTENCE_ERROR	2047	メッセージ永続性が不正です。
MQRC_PERSISTENT_NOT_ALLOWED	2048	キューは永続メッセージをサポートしていません。
MQRC_PMO_ERROR	2173	メッセージ登録オプションの構造体が不正です。
MQRC_PMO_RECORD_FLAGS_ERROR	2158	メッセージ登録レコードのフラグが不正です
MQRC_PRIORITY_ERROR	2050	メッセージ優先度が不正です。
MQRC_PUT_INHIBITED	2051	指定されたキューへの登録は禁止されています。
MQRC_PUT_MSG_RECORDS_ERROR	2159	メッセージ登録レコードが不正です。
MQRC_Q_DELETED	2052	キューが削除されています。

理由コード	数値	意味
MQRC_Q_FULL	2053	キューが満杯です。
MQRC_Q_SPACE_NOT_AVAILABLE	2056	キューに対応するディスクに空き領域がありません。
MQRC_RECS_PRESENT_ERROR	2154	不正なレコードが存在します。
MQRC_REPORT_OPTIONS_ERROR	2061	メッセージ記述子中の報告オプションが不正です。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_RESPONSE_RECORDS_ERROR	2156	応答レコードが不正です。
MQRC_SEGMENT_LENGTH_ZERO	2253	メッセージセグメント長が0です。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_SYNCPOINT_NOT_AVAILABLE	2072	同期点処理はできません。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。
MQRC_UOW_NOT_AVAILABLE	2255	コミット単位ではキューマネージャを使用できません。
MQRC_WRONG_MD_VERSION	2257	MQMD 構造体で指定されたバージョンが誤っています。

詳細は、「付録 B.2 理由コード」を参照してください。

## 注意事項

- キューへメッセージを登録する命令には、MQPUT 命令と MQPUT1 命令の二つがあります。これらの命令は、状況に応じて使い分けてください。
  - MQPUT 命令は、同じキューへ複数のメッセージを登録するときに使用します。  
この方法では、始めに MQOO\_OUTPUT オプションを指定して MQOPEN 命令を呼び出します。次に、MQPUT 命令を一回またはそれ以上呼び出してメッセージを登録してください。最後に MQCLOSE 命令を呼び出して、キューをクローズします。この方法でメッセージを登録すると、MQPUT1 命令を繰り返して呼び出すよりも効率的です。
  - MQPUT1 命令は、あるキューへメッセージを一つだけ登録するときに使用します。  
この方法では、一回の呼び出しで MQOPEN 命令、MQPUT 命令、および MQCLOSE 命令を兼ねます。このため、命令の呼び出しを少なくできます。
- アプリケーションがメッセージグループを使用しないで、連続するメッセージを同じキューに登録する場合、特定の条件が満たされれば、メッセージの順序が保持されます。ローカルあて先キューとリモートあて先キューの両方に適用される条件と、リモートあて先キューだけに適用される条件があります。  
**ローカルあて先キューおよびリモートあて先キューに適用される条件**

- すべての MQPUT 命令が同じトランザクション内にある。またはすべてトランザクション内でないこと  
メッセージが一つのトランザクション内で特定のキューに登録される時、他アプリケーションからのメッセージがキュー上の連続するメッセージに混じることがあるので注意してください。
- すべての MQPUT 命令が同じオブジェクトハンドル Hobj を使用すること  
環境によっては、異なるオブジェクトハンドルを使用しても、メッセージの順序が保持されることがあります。ただし、命令が同じアプリケーションから発行されるものとします。
- メッセージがすべて同じメッセージ優先度であること

#### リモートあて先キューに適用される追加の条件

- 送信側のキューマネージャからあて先キューマネージャへの送信経路が一つしかないこと  
連続するメッセージの一部のメッセージが異なる送信経路を使用する可能性がある場合（例えば、再構成、トラフィック分散またはメッセージサイズに基づく経路選択の場合）、あて先キューマネージャでのメッセージの順序は保証されません。
- 送信側、中間またはあて先キューマネージャで、メッセージが一時的にデッドレターキューに登録されないこと  
一つ以上のメッセージが一時的にデッドレターキューに登録される場合（例えば、転送キューまたはあて先キューが一時的に満杯などが原因の場合）、あて先キューに到着するメッセージは順序とおりでないことがあります。
- メッセージがすべて永続的メッセージであるか、またはすべて非永続メッセージであること  
送信側キューマネージャとあて先キューマネージャ間の送信経路にあるチャンネルの NonPersistentMsgSpeed 属性が MQNPMS\_FAST に設定される場合、非永続メッセージが永続メッセージに先行することがあります。その結果、非永続メッセージに対する永続メッセージの順序が保持されないことがあります。しかし、永続メッセージ間および非永続メッセージ間の順序は保持されます。

これらの条件が満たされない場合、メッセージグループを使用してメッセージの順序を保持できます。しかし、送信側アプリケーションと受信側アプリケーションの両方について、メッセージグループを使用できる必要があることに注意してください。メッセージグループについては、次に示す項目を参照してください。

- MQMD 構造体の MsgFlags フィールド
- MQPMO 構造体の MQPMO\_LOGICAL\_ORDER オプション
- MQGMO 構造体の MQGMO\_LOGICAL\_ORDER オプション

#### 3. 配布リストを使用するときは、次の点に注意してください。

- メッセージは、バージョン 1 またはバージョン 2 の MQPMO 構造体を使用して配布リストに登録できます。バージョン 1 の MQPMO 構造体を使用する場合、または RecsPresent フィールドが 0 であるバージョン 2 の MQPMO 構造体を使用する場合、アプリケーションは登録メッセージレコードや、応答レコードを指定できません。この場合、メッセージが配布リストのあるキューに送信が成功して、ほかのキューへの送信が失敗したとき、エラーとなったキューを特定できません。アプ

リケーションによって登録メッセージレコードや応答レコードが指定される場合、Version フィールドは MQPMO\_VERSION\_2 でなければなりません。

バージョン 2 の MQPMO 構造体は、RecsPresent フィールドが 0 であれば、配布リストではない単一のキューにメッセージを送信する場合も使用できます。

- CompCode 引数および Reason 引数は、次のように設定されます。
  - ・配布リストのキューへ登録が、同じ方法で、すべて成功、または失敗した場合、完了コードおよび理由コードは、共通の結果を記述するように設定されます。MQRR 構造体がアプリケーションで指定されている場合は、応答レコードは設定されません。

例えば、すべての登録が成功した場合、完了コードは MQCC\_OK、理由コードは MQRC\_NONE に設定されます。すべてのキューが登録禁止になっているため、すべての登録が失敗する場合、引数は MQCC\_FAILED と MQRC\_PUT\_INHIBITED に設定されます。

- ・配布リストのキューへの登録が同じ方法で、部分的に成功または失敗した場合は次のとおりです。
  - －少なくとも一つの登録が成功していれば完了コードは MQCC\_WARNING が設定されます。すべてが失敗すると MQCC\_FAILED が設定されます。

- －理由コードの引数は、MQRC\_MULTIPLE\_REASONS に設定されます。

- －応答レコードがアプリケーションによって指定されている場合は、配布リストのキューに対して個別の完了コードと理由コードを設定します。

あて先のオープンに失敗したために登録に失敗した場合、応答レコードのフィールドは MQCC\_FAILED および MQRC\_OPEN\_FAILED に設定されます。そのあて先は InvalidDestCount フィールドに記述されます。

- 配布リストのあて先がローカルキューに受信される場合、メッセージは通常の形式でキューに登録されます。配布リストメッセージではありません。複数のあて先が同じローカルキューに受信される場合、一つのメッセージがあて先キューに登録されます。

配布リストのあて先がリモートキューに受信される場合、メッセージは転送キューに登録されます。幾つかのあて先が同じ転送キューに受信される場合、あて先を含んだ配布リストメッセージは、そのあて先がアプリケーションに指定されたあて先リストと隣接していなくても、転送キューに登録されます。ただし、これは転送キューが配布リストメッセージをサポートしているときだけ可能となります。詳細は、「3. オブジェクトの属性」の「[キューの属性](#)」の DistLists 属性を参照してください。

転送キューが配布リストをサポートしていないとき、通常の形式のメッセージを複製したものが、その転送キューを使用するそれぞれのあて先に対する転送キューに登録されます。

アプリケーションメッセージを含んだ配布リストが転送キューよりも大きい場合、配布リストメッセージはさらに小さな配布リストメッセージに分割されます。分割された配布リストメッセージは、それぞれ幾つかのあて先を持っています。アプリケーションデータだけがキューに格納できれば、キューマネージャは配布リストメッセージをまったく使用しないで、その転送キューを使用するあて先に、通常の形式でメッセージの複製を生成します。

例えば、アプリケーションが MQPRI\_PRIORITY\_AS\_Q\_DEF、MQPER\_PERSISTENCE\_AS\_Q\_DEF を指定している場合など、異なるあて先が異なるメッセージ優先度やメッセージ永続性を持っている場合、メッセージは同じ配布リストメッセージに必要なあ

りません。その代わりに、キューマネージャは、異なる優先度および永続性の値に対応するために必要な数の配布リストメッセージを生成します。

- 配布リストへ登録した結果は、次のどれかです。
  - 単一の配布リストメッセージ
  - 分割された配布リストメッセージの数
  - 配布リストメッセージと通常のメッセージの混在
  - 通常のメッセージだけ

上記のどの結果になるかは次に示す理由によります。

- リストのあて先が、ローカル、リモート、または混在している。
- あて先が、同じメッセージ優先度とメッセージ永続性となっているかどうか。
- 転送キューが配布リストメッセージを保持できるかどうか。
- 転送キューの最大メッセージ長が、配布リスト形式のメッセージに対応する大きさかどうか。

上記のどの原因が発生したとしても、物理メッセージ、つまり登録したメッセージまたは配布リストメッセージは、次に示す場合は、ただ一つのメッセージとしてカウントされます。

- アプリケーションがトランザクションの最大メッセージ数を超過している、または最大メッセージ数を超過しているかどうかをチェックする場合

最大メッセージ数の詳細については「3. オブジェクトの属性」の「キューマネージャの属性」の MaxUncommittedMsgs 属性を参照してください。

- トリガ条件を満たしているかどうかをチェックする場合
- キューの登録数を加算し、その値がキューの最大登録数を超過するかどうかをチェックする場合
- 例えば、解決パスが変更された場合など、キューが個別にオープンされるとハンドルが不正になるようにキュー定義が変更されても、配布リストのハンドルは不正になりません。しかし、次の MQPUT 命令によって、その配布リストのハンドルが使用されたキューのオープンは失敗します。

4. アプリケーションメッセージデータの開始位置で、メッセージが一つ以上の MQ ヘッダ構造体と同時に登録された場合、キューマネージャはヘッダ構造体が正しいかどうかをチェックします。キューマネージャがエラーを検出すると、命令は適当な理由コードで失敗します。実行されるチェックの方法は、存在する構造体によって異なります。

- バージョン 2 以降の MQMD 構造体が MQPUT または MQPUT1 命令で使用される場合にだけ、チェックが実行されます。メッセージデータの開始位置に MQMDE 構造体があっても、バージョン 1 の MQMD 構造体を使用されている場合、チェックは実行されません。
- ローカルキューマネージャによってサポートされない構造体、およびメッセージ中で最初の MQDLH 構造体に続く構造体は、有効性を確認されません。
- MQDH および MQMDE 構造体の有効性は、キューマネージャによって完全に確認されます。
- ほかの構造体の有効性は、キューマネージャによって部分的に確認されます（すべてのフィールドが確認されるわけではありません）。

キューマネージャによって実行される一般的なチェックには、次に示す項目が含まれます。

- StrucId フィールドが有効であること

- Version フィールドが有効であること
- StrucLength フィールドには、構造体およびその一部である可変長データの格納に十分な大きさの値が指定されていること
- CodedCharSetId フィールドは、ゼロまたは無効な負の値でないこと  
MQCCSI\_DEFAULT, MQCCSI\_EMBEDDED, MQCCSI\_Q\_MGR および  
MQCCSI\_UNDEFINED は、ほとんどの MQ ヘッダ構造体で無効です。
- 命令の BufferLength 引数には、構造体の格納に十分な大きさの値が指定されていること  
メッセージの末尾を超えて構造体を拡張できません。

構造体についての一般的なチェックに加えて、次に示す条件が満たされている必要があります。

- PCF メッセージの構造体での長さの和は、MQPUT 命令または MQPUT1 命令の BufferLength 引数によって指定された長さと等しくなければなりません。PCF メッセージは、次のフォーマット名称のどれかを持つメッセージです。

MQFMT\_ADMIN

MQFMT\_EVENT

MQFMT\_PCF

- MQ 構造体は、省略された構造体が許可される次の場合を除いて、省略しないでください。  
報告メッセージであるメッセージ  
PCF メッセージ  
MQDLH 構造体を含むメッセージ  
なお、MQDLH 構造体を含むメッセージでは、最初の MQDLH 構造体が続く構造体は省略できますが、MQDLH 構造体の前にある構造体は省略できません。
- MQ ヘッダ構造体は、二つ以上のセグメントに分割されてはなりません。つまり、構造体は一つのセグメントに含まれなければなりません。

# MQPUT1 命令 – 1 メッセージの登録

## 機能

MQPUT1 命令で、一つのメッセージをキューへ登録できます。キューはオープンされている必要があります。

## 形式

### C 言語の場合

```
MQPUT1(MQHCONN Hconn, PMQVOID ObjDesc, PMQVOID MsgDesc,  
        PMQVOID PutMsgOpts, MQLONG BufferLength,  
        PMQVOID Buffer, PMQLONG CompCode,  
        PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQPUT1' USING HCONN, OBJDESC, MSGDESC, PUTMSGOPTS,  
                   BUFFERLENGTH, BUFFER, COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。

### ● ObjDesc (MQOD 構造体) –input

オブジェクト記述子です。

登録するキューを識別する構造体です。詳細については、「1. データタイプ」の「MQOD 構造体 – オブジェクト記述子」を参照してください。

モデルキューは指定できません。

### ● MsgDesc (MQMD 構造体) –input/output

メッセージ記述子です。

送信するメッセージの属性を記述する構造体です。また、登録後のメッセージに関する情報が返されます。

詳細については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」を参照してください。

アプリケーションがバージョン 1 の MQMD 構造体を指定すると、返却されるメッセージは、バージョン 2 の MQMD 構造体にだけ存在するフィールドの値を指定するために、アプリケーションメッセージデー

タの前に MQMDE 構造体を付加できます。そのとき、MQMDE 構造体が存在することを示すために、MQMD 構造体の Format フィールドには、MQFMT\_MD\_EXTENSION を設定しなければなりません。

詳細については、「1. データタイプ」の「MQMDE 構造体 – メッセージ記述子の拡張」を参照してください。

### ● PutMsgOpts (MQPMO 構造体) –input/output

メッセージ登録オプションです。

詳細については、「1. データタイプ」の「MQPMO 構造体 – メッセージ登録オプション」を参照してください。

### ● BufferLength (MQLONG 型) –input

メッセージデータを格納したバッファのバイト長です。

0 も指定できます。このとき、アプリケーションデータのないメッセージを登録するものとみなされます。

あて先キューがローカルキューの場合、またはローカルキューの別名を使用している場合は、ローカルキューまたはキューマネージャの MaxMsgLength 属性のうち、小さい方の値を超えて指定できません。

あて先キューがリモートキューの場合、またはリモートキューの別名を使用している場合は、次に示すローカルキューまたはキューマネージャの MaxMsgLength 属性のうち、小さい方の値を超えて指定できません。

1. ローカルキューマネージャで一時的に格納するために使用される転送キュー
2. ローカルキューからあて先キューまでの経路で仲介するキューマネージャで使用される転送キュー
3. あて先キューマネージャで使用されるあて先キュー

なお、メッセージが転送キューに格納されるとき、メッセージに付加情報が追加されます。このため、送信できるアプリケーションデータの量が少なくなります。BufferLength 引数の値を決めるときは、転送キューの MaxMsgLength 属性の値から MQ\_MSG\_HEADER\_LENGTH のバイト数を引いた値を参考にしてください。

#### ■ 注意事項

上記の 1. の場合だけ、メッセージの登録直後にバッファ長のエラー（理由コード MQRC\_MSG\_TOO\_BIG\_FOR\_Q または MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR）が判明します。2., 3. の場合は、登録直後はエラーが判明しません。2., 3. でエラーが判明した場合、仲介するキューマネージャまたはあて先キューマネージャのデッドレターキューに、メッセージが転送されず。送信側で要求している場合は、エラー発生時に報告メッセージが生成されます。

### ● Buffer (MQBYTE 型×BufferLength) –input

メッセージデータを格納したバッファです。バッファには、メッセージ中のデータに適したバウンダリ調整が必要です。多くのメッセージ（MQ ヘッダ構造体のあるメッセージを含む）には 4 バイト単位の調整

が適していますが、より厳しい調整が必要なメッセージもあります。例えば、64ビット2進整数を含めるメッセージには8バイト単位のバウンダリ調整が必要です。

バッファ内に文字または数値がある場合、MsgDesc 引数の CodedCharSetId フィールドと Encoding フィールドに、適切な値を指定してください。これらの値によって、受信側でデータを適切な文字セットとマシンコード形式に変換できます。

## 注意事項

MQPUT1 命令のそのほかの引数は、ローカルキューマネージャの文字セットとマシンコード形式、つまり、ローカルキューの CodedCharSetId 属性と MQENC\_NATIVE で指定してください。

C 言語では、この引数を void 型のポインタとして指定します。したがって、どんなデータタイプのアドレスでも、引数として指定できます。

BufferLength 引数で 0 を指定した場合、Buffer 引数は参照されません。したがって、C 言語のプログラムでは、この引数のアドレスとしてヌルを指定できます。

### ● CompCode (MQLONG 型) -output

完了コードです。

次のどれかが返されます。

- MQCC\_OK : 成功
- MQCC\_WARNING : 警告 (一部成功)
- MQCC\_FAILED : 失敗

### ● Reason (MQLONG 型) -output

理由コードです。

CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

CompCode 引数が MQCC\_WARNING の場合

理由コード	数値	意味
MQRC_MULTIPLE_REASONS	2136	複数の理由コードが返されました。
MQRC_PRIORITY_EXCEEDS_MAXIMUM	2049	メッセージ優先度が最大値を超えています。
MQRC_UNKNOWN_REPORT_OPTION	2104	メッセージ記述子中の報告オプションを認識できません。

## CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_ALIAS_BASE_Q_TYPE_ERROR	2001	別名キューのベースキューのタイプが不正です。
MQRC_BUFFER_ERROR	2004	バッファの引数が不正です。
MQRC_BUFFER_LENGTH_ERROR	2005	バッファ長の引数が不正です。
MQRC_CLUSTER_PUT_INHIBITED	2268	クラスタ内のすべてのキューに対して登録が禁止されています。
MQRC_CLUSTER_RESOLUTION_ERROR	2189	クラスタ名称の解決に失敗しました。
MQRC_CLUSTER_RESOURCE_ERROR	2269	クラスタリソースのエラーです。
MQRC_CONTEXT_HANDLE_ERROR	2097	参照先のキューハンドルにコンテキストが保存されていません。
MQRC_CONTEXT_NOT_AVAILABLE	2098	参照先のキューハンドルのコンテキストが不正です。
MQRC_DEF_XMIT_Q_TYPE_ERROR	2198	省略時の転送キューがローカルキューではありません。
MQRC_DEF_XMIT_Q_USAGE_ERROR	2199	省略時の転送キューの使用方法に誤りがあります。
MQRC_DH_ERROR	2135	配布リスト構造体が不正です。
MQRC_EXPIRY_ERROR	2013	メッセージ保持時間が不正です。
MQRC_FEEDBACK_ERROR	2014	報告メッセージ返答コードが不正です。
MQRC_HANDLE_NOT_AVAILABLE	2017	現在そのハンドルは有効ではありません。
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_MD_ERROR	2026	メッセージ記述子が不正です。
MQRC_MDE_ERROR	2248	メッセージ記述子の拡張子が不正です。
MQRC_MISSING_REPLY_TO_Q	2027	応答キューがありません。
MQRC_MSG_FLAGS_ERROR	2249	メッセージフラグが不正です。
MQRC_MSG_SEQ_NUMBER_ERROR	2250	メッセージシーケンス番号が不正です。
MQRC_MSG_TOO_BIG_FOR_Q	2030	メッセージ長がキューに定義された最大値を超えています。
MQRC_MSG_TYPE_ERROR	2029	メッセージ記述子中のメッセージタイプが不正です。
MQRC_MULTIPLE_REASONS	2136	複数の理由コードが返されます。

理由コード	数値	意味
MQRC_NO_DESTINATIONS_AVAILABLE	2270	利用可能なあて先キューが存在しません。
MQRC_OBJECT_DAMAGED	2101	オブジェクトが破損しています。
MQRC_OBJECT_NAME_ERROR	2152	オブジェクト名が不正です。
MQRC_OBJECT_Q_MGR_NAME_ERROR	2153	オブジェクトキューマネージャ名が不正です。
MQRC_OBJECT_RECORDS_ERROR	2155	オブジェクトレコードが不正です。
MQRC_OBJECT_TYPE_ERROR	2043	オブジェクトタイプが不正です。
MQRC_OD_ERROR	2044	オブジェクト記述子の構造体が不正です。
MQRC_OFFSET_ERROR	2251	メッセージセグメントのオフセットが不正です。
MQRC_OPTIONS_ERROR	2046	オプションが不正です。または、指定されていません。
MQRC_ORIGINAL_LENGTH_ERROR	2252	元の長さが不正です。
MQRC_PERSISTENCE_ERROR	2047	メッセージ永続性が不正です。
MQRC_PERSISTENT_NOT_ALLOWED	2048	キューは永続メッセージをサポートしていません。
MQRC_PMO_ERROR	2173	メッセージ登録オプションの構造体が不正です。
MQRC_PMO_RECORD_FLAGS_ERROR	2158	登録メッセージレコードのフラグが不正です。
MQRC_PRIORITY_ERROR	2050	メッセージ優先度が不正です。
MQRC_PUT_INHIBITED	2051	指定されたキューへの登録は禁止されています。
MQRC_PUT_MSG_RECORDS_ERROR	2159	登録メッセージレコードが不正です。
MQRC_Q_DELETED	2052	キューが削除されています。
MQRC_Q_FULL	2053	キューが満杯です。
MQRC_Q_SPACE_NOT_AVAILABLE	2056	キューに対応するディスクに空き領域がありません。
MQRC_Q_TYPE_ERROR	2057	キュータイプが不正です。
MQRC_RECS_PRESENT_ERROR	2154	存在するレコード数が不正です。
MQRC_REMOTE_Q_NAME_ERROR	2184	リモートキュー名が不正です。
MQRC_REPORT_OPTIONS_ERROR	2061	メッセージ記述子中の報告オプションが不正です。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。

理由コード	数値	意味
MQRC_RESPONSE_RECORDS_ERROR	2156	応答レコードが不正です。
MQRC_SEGMENT_LENGTH_ZERO	2253	メッセージセグメントのデータ長が0です。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_SYNCPOINT_NOT_AVAILABLE	2072	同期点処理はできません。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。
MQRC_UNKNOWN_ALIAS_BASE_Q	2082	別名キューのベースキューを認識できません。
MQRC_UNKNOWN_DEF_XMIT_Q	2197	省略時の転送キューを認識できません。
MQRC_UNKNOWN_OBJECT_NAME	2085	オブジェクト名を認識できません。
MQRC_UNKNOWN_OBJECT_Q_MGR	2086	オブジェクトのキューマネージャを認識できません。
MQRC_UNKNOWN_REMOTE_Q_MGR	2087	リモートキューマネージャを認識できません。
MQRC_UNKNOWN_XMIT_Q	2196	転送キューを認識できません。
MQRC_UOW_NOT_AVAILABLE	2255	コミット単位でキューマネージャを使用できません。
MQRC_WRONG_MD_VERSION	2257	MQMD 構造体に指定されたバージョンが不正です。
MQRC_XMIT_Q_TYPE_ERROR	2091	転送キューがローカルキューではありません。
MQRC_XMIT_Q_USAGE_ERROR	2092	転送キューの使用方法に誤りがあります。

詳細は、「付録 B.2 理由コード」を参照してください。

## 注意事項

- キューへメッセージを登録する命令には、MQPUT 命令と MQPUT1 命令の二つがあります。これらの命令は、状況に応じて使い分けてください。
  - MQPUT 命令は、同じキューへ複数のメッセージを登録するときに使用します。  
この方法では、始めに MQOO\_OUTPUT オプションを指定して MQOPEN 命令を呼び出します。次に、MQPUT 命令を一回またはそれ以上呼び出してメッセージを登録してください。最後に MQCLOSE 命令を呼び出して、キューをクローズします。この方法でメッセージを登録すると、MQPUT1 命令を繰り返して呼び出すよりも効率的です。
  - MQPUT1 命令は、あるキューへメッセージを一つだけ登録するときに使用します。

この方法では、一回の呼び出しで MQOPEN 命令、MQPUT 命令、および MQCLOSE 命令を兼ねます。このため、命令の呼び出しを少なくできます。

2. アプリケーションがメッセージグループを使用しないで、連続するメッセージを同じキューに登録する場合、特定の条件が満たされれば、メッセージの順序が保持されます。しかし、ほとんどの環境では MQPUT1 命令はこれらの条件を満たさず、メッセージの順序は保持されません。これらの環境では、代わりに MQPUT 命令を使用する必要があります。詳細については、この章の「MQPUT 命令 – メッセージの登録」の「注意事項」を参照してください。
3. MQPUT1 命令は、メッセージを配布リストに登録するのに使用できます。配布リストの登録については、この章の「MQOPEN 命令 – オブジェクトのオープン」および「MQPUT 命令 – メッセージの登録」の注意事項を参照してください。

次に、MQPUT1 命令に特有の注意事項を示します。

- MQRR 構造体の応答レコードがアプリケーションによって指定されるとき、その応答レコードは MQOD 構造体を使用して指定されなければなりません。なお、MQPMO 構造体によって指定することはできません。
- 理由コード MQRC\_OPEN\_FAILED は、MQPUT1 命令では応答レコードに返されません。キューのオープンに失敗すると、そのキューの応答レコードはオープン操作から結果までの理由コードを保持します。キューのオープン操作が MQCC\_WARNING の完了コードで成功すると、そのキューの応答レコードの完了コードと理由コードは、登録操作の結果である完了コードと理由コードで置き換えられます。MQOPEN 命令および MQPUT 命令と同様に、キューマネージャは結果が配布リストのすべてのキューで異なるときだけ、応答レコードが指定されている場合は応答レコードを設定します。これは、MQRC\_MULTIPLE\_REASONS の理由コードで完了することで示されます。
- MQPUT1 命令がメッセージを配布リストに登録するのに使用されると、アプリケーションがハンドルの最大数、つまり、キューマネージャの MaxHandles 属性を超えているかチェックするときに、一つのハンドルとしてカウントされます。これは、配布リストのあて先数とは関係ありません。

ただし、MQOPEN 命令が使用されると、配布リストのおおののあて先が、別々のハンドルとしてカウントされます。

4. MQPUT1 命令がクラスタキューにメッセージを登録するために使用された場合、MQOPEN 命令で MQOO\_BIND\_NOT\_FIXED が指定されたように動作します。
5. メッセージがアプリケーションメッセージデータの先頭に、一つ以上の MQ ヘッダ構造体を付けて登録されると、キューマネージャはこれらが正しいか確認するために、ヘッダ構造体をチェックします。詳細については、この章の「MQPUT 命令 – メッセージの登録」の注意事項を参照してください。
6. CompCode 引数を参照したときに一つ以上の警告が発生していると、返却される理由コードの最初のものは次に示すコードのどれかになります。

MQRC\_MULTIPLE\_REASONS  
MQRC\_INCOMPLETE\_MSG  
MQRC\_INCOMPLETE\_GROUP  
MQRC\_PRIORITY\_EXCEEDS\_MAXIMUM  
MQRC\_UNKNOWN\_REPORT\_OPTION

# MQSET 命令 – オブジェクトの属性の設定

## 機能

MQSET 命令で、指定したハンドルのオブジェクトの属性を変更できます。属性を変更できるオブジェクトはキューだけです。

## 形式

### C 言語の場合

```
MQSET(MQHCONN Hconn, MQHOBJ Hobj, MQLONG SelectorCount,  
      PMQLONG Selectors, MQLONG IntAttrCount,  
      PMQLONG IntAttrs, MQLONG CharAttrLength,  
      PMQCHAR CharAttrs, PMQLONG CompCode,  
      PMQLONG Reason)
```

### COBOL 言語の場合

```
CALL 'MQSET' USING HCONN, HOBJ, SELECTORCOUNT, SELECTORS,  
                  INTATTRCOUNT, INTATTRS, CHARATTRLENGTH,  
                  CHARATTRS, COMPCODE, REASON.
```

## 引数

### ● Hconn (MQHCONN 型) –input

コネクションハンドルです。

キューマネージャへの接続を示すハンドルです。MQCONN 命令の戻り値を指定してください。

### ● Hobj (MQHOBJ 型) –input

オブジェクトハンドルです。

属性を設定するオブジェクトを示すハンドルです。MQOO\_SET オプションを指定した MQOPEN 命令の戻り値を指定してください。

### ● SelectorCount (MQLONG 型) –input

セクタ数です。

Selectors の配列に指定するセクタの数を指定します。ここで指定した数の属性を設定します。セクタ数は、0～256 の範囲で指定できます。

### ● Selectors (MQLONG 型×SelectorCount) –input

セクタです。

SelectorCount の要素数の配列でセレクタを指定します。それぞれのセレクタは、設定する整数型属性または文字型属性の種類を示します。

それぞれのセレクタは、Hobj のキューの種類に合ったものを指定してください。MQIA\_\*セレクタまたはMQCA\_\*セレクタの一部だけ指定できます。使用できるセレクタは、表 2-5 に示します。

セレクタはどの順序で指定してもかまいません。整数型属性のセレクタ (MQIA\_\*セレクタ) の属性値は、Selectors 引数に指定した順序で IntAttrs 引数に指定してください。文字型属性のセレクタ (MQCA\_\*セレクタ) の属性値は、セレクタで指定した順序で CharAttrs 引数に指定してください。MQIA\_\*セレクタは、MQCA\_\*セレクタと交互に指定できます。各種類での順序だけが重要です。

同じセレクタを 2 回以上指定してもかまいません。この場合、そのセレクタで最後に指定した値が設定されます。

## 注意事項

1. 整数型属性と文字型属性のセレクタは、二つの異なる範囲で割り当てられています。MQIA\_\*セレクタは、MQIA\_FIRST から MQIA\_LAST の範囲にあります。また、MQCA\_\*セレクタは、MQCA\_FIRST から MQCA\_LAST の範囲にあります。  
それぞれの範囲に、キューマネージャで受け付けられる最大値として、定数 MQIA\_LAST\_USED と MQCA\_LAST\_USED が定義されています。
2. すべての MQIA\_\*セレクタを始めに指定した場合、要素の位置として、その要素と同じ番号を Selectors と IntAttrs の配列で使用できます。
3. SelectorCount 引数に 0 を指定した場合、Selectors 引数は参照されません。この場合、C 言語のプログラムでは、Selectors のアドレスにヌルを指定できます。

設定できる属性について、次の表に示します。ほかの属性はこの命令で設定できません。括弧内の値は、MQCA\_\*セレクタで CharAttrs 引数に設定する文字列の長さを定義した定数です。

表 2-5 MQSET 命令の属性セレクタ (キュー用)

セレクタ	説明
MQCA_TRIGGER_DATA	トリガデータ (MQ_TRIGGER_DATA_LENGTH)
MQIA_DIST_LISTS	配布リストサポート
MQIA_INHIBIT_GET	取り出し許可
MQIA_INHIBIT_PUT	登録許可
MQIA_TRIGGER_CONTROL	トリガ制御
MQIA_TRIGGER_DEPTH	トリガのためのメッセージ登録数の下限値
MQIA_TRIGGER_MSG_PRIORITY	トリガのためのメッセージ優先度の下限値
MQIA_TRIGGER_TYPE	トリガタイプ

このほかの属性は、この命令では設定できません。

### ● IntAttrCount (MQLONG 型) -input

整数型属性値の数です。

IntAttrs の配列の要素数を指定します。Selectors 引数の MQIA\_\*セクタの数を指定してください。MQIA\_\*セクタがないときは 0 を指定します。

### ● IntAttrs (MQLONG 型×IntAttrCount) -input

整数型属性値の配列です。

IntAttrCount の要素数の配列で整数型属性値を設定します。

整数型属性値は、Selectors 引数の MQIA\_\*セクタで指定した順序で指定してください。

IntAttrCount 引数または SelectorCount 引数の値が 0 の場合、IntAttrs 引数は参照されません。この場合、C 言語のプログラムでは引数のアドレスとしてヌルを指定できます。

### ● CharAttrLength (MQLONG 型) -input

文字型属性値の格納領域長です。

CharAttrs の格納領域長をバイト単位で指定します。CharAttrs 引数の MQCA\_\*セクタで指定する文字型属性値の合計の長さを指定してください。MQCA\_\*セクタがないときは 0 を指定します。

### ● CharAttrs (MQCHAR 型×CharAttrLength) -input

文字型属性値の格納領域です。

この格納領域に文字型の属性値を並べて指定します。格納領域の長さは CharAttrLength 引数に指定します。

文字型属性値は、Selectors 引数の MQCA\_\*セクタで指定した順序で指定してください。属性の文字列の長さは各属性に固有です。文字型属性値の長さについては、「Selectors 引数」を参照してください。属性値の文字列がその属性の固有の長さより短い場合、固有の長さになるまで後ろを空白で埋めて、CharAttrs の格納領域に設定してください。

CharAttrLength 引数または SelectorCount 引数の値が 0 の場合、CharAttrs 引数は参照されません。この場合、C 言語のプログラムでは引数のアドレスとしてヌルを指定できます。

### ● CompCode (MQLONG 型) -output

完了コードです。

次のどちらかが返されます。

- MQCC\_OK : 成功
- MQCC\_FAILED : 失敗

## ● Reason (MQLONG 型) -output

理由コードです。

### CompCode 引数が MQCC\_OK の場合

理由コード	数値	意味
MQRC_NONE	0	理由コードはありません。

### CompCode 引数が MQCC\_FAILED の場合

理由コード	数値	意味
MQRC_CHAR_ATTR_LENGTH_ERROR	2006	文字型属性長が不正です。
MQRC_CHAR_ATTRS_ERROR	2007	文字型属性値が不正です。
MQRC_HCONN_ERROR	2018	コネクションハンドルが不正です。
MQRC_HOBJ_ERROR	2019	オブジェクトハンドルが不正です。
MQRC_INHIBIT_VALUE_ERROR	2020	取り出し許可または登録許可の属性値が不正です。
MQRC_INT_ATTR_COUNT_ERROR	2021	整数型属性数が不正です。
MQRC_INT_ATTRS_ARRAY_ERROR	2023	整数型属性値が不正です。
MQRC_NOT_OPEN_FOR_SET	2040	キューが設定用にオープンされていません。
MQRC_OBJECT_DAMAGED	2101	オブジェクトが破損しています。
MQRC_Q_DELETED	2052	キューが削除されています。
MQRC_RESOURCE_PROBLEM	2102	システム資源が不足しています。
MQRC_SELECTOR_COUNT_ERROR	2065	セレクタ数が不正です。
MQRC_SELECTOR_ERROR	2067	セレクタが不正です。
MQRC_SELECTOR_LIMIT_EXCEEDED	2066	セレクタ数が最大値を超えています。
MQRC_STORAGE_NOT_AVAILABLE	2071	記憶容量が不足しています。
MQRC_TRIGGER_CONTROL_ERROR	2075	MQIA_TRIGGER_CONTROL セレクタに対する属性値が不正です。
MQRC_TRIGGER_DEPTH_ERROR	2076	MQIA_TRIGGER_DEPTH セレクタに対する属性値が不正です。
MQRC_TRIGGER_MSG_PRIORITY_ERR	2077	MQIA_TRIGGER_MSG_PRIORITY セレクタに対する属性値が不正です。

理由コード	数値	意味
MQRC_TRIGGER_TYPE_ERROR	2078	MQIA_TRIGGER_TYPE セレクタに対する属性値が不正です。
MQRC_UNEXPECTED_ERROR	2195	予期しないエラーが発生しました。

詳細は、「[付録 B.2 理由コード](#)」を参照してください。

## 注意事項

- この命令では、整数型属性値、文字型属性値、またはその両方を指定できます。エラーが発生しない限り、指定した属性値は同時に更新されます。例えばセレクタの不正や、設定できない属性を設定しようとしたときなどにエラーが発生した場合は、一つの属性値も設定されないで命令は失敗します。
- 属性の値は MQINQ 命令を使用して照会できます。属性の値の照会については、この章の「[MQINQ 命令 - オブジェクトの属性の照会](#)」の説明を参照してください。

### 注意事項

MQINQ 命令で照会できるすべての属性を、MQSET 命令で変更できるわけではありません。例えば、プロセス定義およびキューマネージャの属性は MQSET 命令では設定できません。

- 変更した属性値は、キューマネージャを再開始したあとでも保たれます。ただし、一時的動的キューの属性の変更は除きます。一時的動的キューは、キューマネージャの再開始時には失われます。
- モデルキューの属性は、MQSET 命令を使用して変更できません。ただし、MQOO\_SET オプションを指定した MQOPEN 命令でモデルキューをオープンした場合、その MQOPEN 命令で生成された動的キューの属性を MQSET 命令で設定できます。
- 設定されるオブジェクトがクラスタキューの場合、オープンが成功するためにはクラスタキューのローカルインスタンスが存在しなければなりません。
- MQSET 命令の使用による属性の変更は、AlterationDate および AlterationTime 属性の値には影響しません。
- 各オブジェクトタイプの属性については、「[3. オブジェクトの属性](#)」を参照してください。

# 3

## オブジェクトの属性

TP1/Message Queue のオブジェクトには、キュー、プロセス、およびキューマネージャがあります。

この章では、API でアクセスできる TP1/Message Queue オブジェクトの属性について説明します。

## キューの属性

---

キューマネージャは次に示すキュータイプをサポートします。

### ローカルキュー

メッセージを保存する物理的なキューです。ローカルキューマネージャにあります。

ローカルキューマネージャに接続したアプリケーションは、このタイプのキューにメッセージを登録したり取り出したりできます。キューの QType 属性値は MQQT\_LOCAL です。

### クラスタキュー

メッセージを保存する物理的なキューです。ローカルキューマネージャにあつたり、ローカルキューマネージャと同じクラスタ内の一つ以上のキューマネージャにあつたりします。

ローカルキューマネージャに接続したアプリケーションは、キューの場所に関係なく、このタイプのキューにメッセージを登録できます。ローカルキューマネージャにクラスタキューがある場合は、ローカルキューと同じ動作をし、ローカルキューマネージャに接続したアプリケーションはクラスタキューからメッセージを取り出せます。キューの QType 属性値は MQQT\_CLUSTER です。

### リモートキューのローカル定義

これは物理的なキューではなく、リモートキューマネージャにあるキューのローカル定義です。リモートキューのローカル定義にある情報は、リモートキューマネージャへのメッセージの送信経路をローカルキューマネージャに知らせます。

ローカルキューマネージャに接続したアプリケーションは、このタイプのキューにメッセージを登録できます。メッセージは実際にはローカルの転送キューに登録されます。転送キューはリモートキューマネージャへのメッセージの転送に使用されるキューです。キューの QType 属性値は MQQT\_REMOTE です。

また、リモートキューのローカル定義は次に示す目的でも使用できます。

- 応答キューの別名  
定義名が応答キュー名の場合です。名称解決については、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。
- キューマネージャの別名  
定義名がキューマネージャの別名の場合です。名称解決については、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。

### 別名キュー

これは物理的なキューではなく、ローカルキュー、クラスタ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。別名が解決するキューの名前は、別名キューの属性定義にあります。

ローカルキューマネージャに接続したアプリケーションは、このタイプのキューにメッセージを登録できます。メッセージは実際には別名が解決されたキューに登録されます。別名がローカルキューまたはローカルのクラスタキューに解決される場合、アプリケーションは別名キューからメッセージを取り出せます。キューの QType 属性値は MQQT\_ALIAS です。

## モデルキュー

これは物理的なキューではなく、ローカルキューを生成するための属性の定義です。

このタイプのキューにメッセージは登録できません。

キュー属性には、すべてのキューに適用できる属性と、特定のキュータイプに適用できる属性があります。各タイプに適用できる属性を次の表に示します。属性をアルファベット順に示します。

表 3-1 キューの属性

属性	説明	ローカル キュー	モデル キュー	別名 キュー	リモート キューの ローカル 定義	クラス タ キュー
BaseQName	ベースキュー名	×	×	○	×	×
ClusterName	キューが属するクラスタの名称	○	×	○	○	×
CreationDate	作成日付	○	×	×	×	×
CreationTime	作成時刻	○	×	×	×	×
CurrentQDepth	メッセージ登録数	○	×	×	×	×
DefBind	デフォルトバインディング	○	×	○	○	○
DefinitionType	定義タイプ	○	○	×	×	×
DefInputOpenOption	省略時の取り出しオプション	○	○	×	×	×
DefPersistence	省略時のメッセージ永続性	○	○	○	○	○
DefPriority	省略時のメッセージ優先度	○	○	○	○	○
DistLists	配布リストのサポート	○	○	×	×	×
HardenGetBackout	ロールバック回数記録方法	○	○	×	×	×
InhibitGet	取り出し許可	○	○	○	×	×
InhibitPut	登録許可	○	○	○	○	○
InitiationQName	イニシエーションキュー名	○	○	×	×	×
MaxMsgLength	最大メッセージ長	○	○	×	×	×
MaxQDepth	最大メッセージ登録数	○	○	×	×	×
MsgDeliverySequence	メッセージ配布順序	○	○	×	×	×
OpenInputCount	入力オープン数	○	×	×	×	×
OpenOutputCount	出力オープン数	○	×	×	×	×
ProcessName	プロセス名	○	○	×	×	×
QDesc	キュー記述子	○	○	○	○	○

属性	説明	ローカル キュー	モデル キュー	別名 キュー	リモート キューの ローカル 定義	クラス タ キュー
QName	キュー名	○	×	○	○	○
QType	キュータイプ	○	×	○	○	○
RemoteQMgrName	リモートキューマネージャ名	×	×	×	○	×
RemoteQName	リモートキュー名	×	×	×	○	×
RetentionInterval	キュー保持時間	○	○	×	×	×
Scope	キューの有効範囲	○	×	○	○	×
Shareability	取り出し時の共用性	○	○	×	×	×
TriggerControl	トリガ制御	○	○	×	×	×
TriggerData	トリガデータ	○	○	×	×	×
TriggerDepth	トリガのためのメッセージ登録数の下限値	○	○	×	×	×
TriggerMsgPriority	トリガのためのメッセージ優先度の下限	○	○	×	×	×
TriggerType	トリガタイプ	○	○	×	×	×
Usage	使用種別	○	○	×	×	×
XmitQName	転送キュー名	×	×	×	○	×

(凡例)

- ：該当します。
- ×：該当しません。

## 注

- モデルキューの欄は、モデルキューから生成されるローカルキューがどの属性を継承するかを示します。
- クラスタキューの欄は、クラスタキューが照会用、または照会と登録用にオープンされるときに照会可能な属性を示します。クラスタキューが照会とあわせて、取り出し、検索、または設定の一つ以上のためにオープンされる場合は、代わりにローカルキューの欄を参照してください。

## 属性

### ● BaseQName (MQCHAR48 型)

ベースキュー名

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
×	×	○	×	×

(凡例)

○：該当します。

×：該当しません。

別名キューの実体を示します。ローカルキューマネージャに定義されたキューの名称です。キューを含むオブジェクトの命名規則については、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。このキューのキュータイプは、次のどれかです。

MQQT\_LOCAL：ローカルキュー

MQQT\_REMOTE：リモートキューのローカル定義

MQQT\_CLUSTER：クラスタキュー

この属性の値を照会するときは、MQINQ 命令で MQCA\_BASE\_Q\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

### ● ClusterName (MQCHAR48 型)

キューが属するクラスタの名称

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	○	○	×

キューが所属しているクラスタの名称です。この属性はモデルキューには適用されません。

この属性の値を照会するときは、MQINQ 命令で MQCA\_CLUSTER\_NAME セレクタを使用してください。この属性の長さは MQ\_CLUSTER\_NAME\_LENGTH に定義されています。

### ● CreationDate (MQCHAR12 型)

作成日付

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	×	×	×

キューを作成した日付です。形式は次のとおりです。

YYYY-MM-DD△△

△△は 2 文字の空白を表します。この 2 文字の空白を埋めた 12 バイトです。

例：1995-09-30△△

例は、1995 年 9 月 30 日を示します。

この属性の値を照会するときは、MQINQ 命令で MQCA\_CREATION\_DATE セレクタを使用してください。

この属性の長さは、MQ\_CREATION\_DATE\_LENGTH に定義されています。

### ● CreationTime (MQCHAR8 型)

作成時刻

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	×	×	×

キューを作成した時刻です。形式は次のとおりです。

HH.MM.SS

24 時間時計を使用し、10 時より前の時刻は先頭が 0 となります。

例：21.10.20

この属性は 8 文字列を使用します。また、現地時間を使用します。

この属性の値を照会するときは、MQINQ 命令で MQCA\_CREATION\_TIME セレクタを使用してください。

この属性の長さは、MQ\_CREATION\_TIME\_LENGTH に定義されています。

### ● CurrentQDepth (MQLONG 型)

メッセージ登録数

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	×	×	×

現時点でキューに格納しているメッセージの個数です。MQPUT 命令のときと MQGET 命令のロールバックのときに、数が増加します。検索用ではない MQGET 命令のときと MQPUT 命令のロールバックのときに数が減少します。つまり、まだトランザクションがコミットしていないときでも、その中から登録したメッセージは、この数に算入されます。ただし、コミットしていないときのメッセージは MQGET 命令で取り出せません。同様に、トランザクションはコミットしてなくても MQGET 命令で取り出された場合はこの数から除かれます。

保持時間を経過したが破棄されていないメッセージも、取り出しはできませんが数に含まれます。MQMD 構造体の Expiry フィールドについては、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」を参照してください。

この属性の値は、キューマネージャの操作で変動します。

この属性の値を照会するときは、MQINQ 命令で MQIA\_CURRENT\_Q\_DEPTH セレクタを使用してください。

## ● DefBind (MQLONG 型)

デフォルトバインディング

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	○	○	○

MQOPEN 命令で MQOO\_BIND\_AS\_Q\_DEF が指定され、キューがクラスタキューである場合に使用されるデフォルトのバインディングです。次のどちらかの値を取ります。

MQBND\_BIND\_ON\_OPEN：バインディングは MQOPEN 命令で固定されます。

MQBND\_BIND\_NOT\_FIXED：バインディングは固定されません。

この属性の値を照会するときは、MQINQ 命令で MQIA\_DEF\_BIND セレクタを使用してください。

## ● DefinitionType (MQLONG 型)

定義タイプ

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

キューを定義する種類を示します。次のどれかの値を取ります。

### MQQDT\_PREDEFINED

定義済みキュー

システム管理者が作成する永続的キューです。システム管理者だけが削除できます。

### MQQDT\_PERMANENT\_DYNAMIC

永続的動的キュー

アプリケーションが呼び出す MQOPEN 命令によって、モデルキューの名称で作成される永続的キューです。キューの名称は、オブジェクト記述子である MQOD 構造体に指定します。そのモデルキューの DefinitionType 属性は MQQDT\_PERMANENT\_DYNAMIC です。永続的動的キューは、MQCLOSE 命令で削除できます。詳細については、「2. メッセージキューイング機能の命令」の「MQCLOSE 命令 - オブジェクトのクローズ」の説明を参照してください。

### MQQDT\_TEMPORARY\_DYNAMIC

一時的動的キュー

アプリケーションが呼び出す MQOPEN 命令によって、モデルキューの名称で作成される一時的キューです。キューの名称は、オブジェクト記述子に指定します。そのモデルキューの DefinitionType 属性

は MQQDT\_TEMPORARY\_DYNAMIC です。一時的動的キューは、キューを作成したアプリケーションが MQCLOSE 命令を呼び出したときに、自動的に削除されます。

モデルキューは常に定義済みであるため、モデルキュー定義のこの属性は MQQDT\_PREDEFINED です。MQINQ 命令でモデルキューの属性を照会した場合、このモデルキューを基にして生成された動的キューの属性が照会されます。

この属性の値を照会するときは、MQINQ 命令で MQIA\_DEFINITION\_TYPE セレクタを使用してください。

### ● DefInputOpenOption (MQLONG 型)

省略時の取り出しオプション

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

取り出し用にキューをオープンしたときの排他方法の省略時解釈値です。MQOPEN 命令で MQOO\_INPUT\_AS\_Q\_DEF を指定したときに、この属性値が使用されます。次のどちらかの値を取ります。

#### MQOO\_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。

そのキューは、以降の MQGET 命令で使用します。そのキューが MQOO\_INPUT\_SHARED オプションまたは MQOO\_INPUT\_EXCLUSIVE オプションのどちらの場合も、ほかのアプリケーションによってすでに取り出し用にオープンされていると、理由コード MQRC\_OBJECT\_IN\_USE で MQOPEN 命令は失敗します。

#### MQOO\_INPUT\_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。

そのキューは、以降の MQGET 命令で使用します。そのキューがほかのアプリケーションによってすでにオープンされている場合、MQOO\_INPUT\_SHARED オプションでオープンされている場合は、MQOPEN 命令は成功します。MQOO\_INPUT\_EXCLUSIVE でオープンされていると、理由コード MQRC\_OBJECT\_IN\_USE で MQOPEN 命令は失敗します。

この属性の値を照会するときは、MQINQ 命令で MQIA\_DEF\_INPUT\_OPEN\_OPTION セレクタを使用してください。

### ● DefPersistence (MQLONG 型)

省略時のメッセージ永続性

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	○	○	○

キュー内のメッセージに対する省略時の永続性です。登録時のメッセージ記述子で MQPER\_PERSISTENCE\_AS\_Q\_DEF を指定したときに、この属性値が使用されます。

キューの実名をたどるまでに複数の定義がある場合、最初に登録したキューの定義のこの属性が、MQPUT または MQPUT1 命令の発行時に省略時のメッセージ永続性になります。これは、次に示すオブジェクトに登録する場合に発生します。

- 別名キュー
- ローカルキュー
- リモートキューのローカル定義
- キューマネージャの別名
- 転送キュー（例えば DefXmitQName キュー）

次のどちらかの値を取ります。

### MQPER\_PERSISTENT

永続メッセージにします。

メッセージは、システム障害が発生したりキューマネージャを再開始したりしても失われません。一時的動的キューは再開始時に失われるので、永続メッセージは一時的動的キューに格納できません。永続メッセージは、永続的動的キューおよび定義済みキューに登録できます。

### MQPER\_NOT\_PERSISTENT

永続メッセージにしません。

メッセージは、システム障害が発生したりキューマネージャを再開始したりすると失われます。再開始の時点で補助記憶装置にメッセージが残っている場合でも、同様です。

永続メッセージと非永続メッセージを、同じキューに格納できます。

この属性の値を照会するときは、MQINQ 命令で MQIA\_DEF\_PERSISTENCE セレクタを使用してください。

## ● DefPriority (MQLONG 型)

省略時のメッセージ優先度

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	○	○	○

キュー内のメッセージに対する省略時の優先度です。登録時のメッセージ記述子で MQPRI\_PRIORITY\_AS\_Q\_DEF を指定したときに、この属性値が使用されます。

キューの実名をたどるまでに複数の定義がある場合、最初に登録したキューの定義のこの属性が、省略時のメッセージ優先度になります。これは、次に示すキューに登録する場合に発生します。

- 別名キュー
- ローカルキュー
- リモートキューのローカル定義
- キューマネージャの別名
- 転送キュー（例えば DefXmitQName キュー）

MsgDeliverySequence 属性が MQMDS\_PRIORITY のキューでは、メッセージのキュー内の位置は、その優先度に依存します。MsgDeliverySequence 属性が MQMDS\_FIFO のキューでは、すべてのメッセージは、実体のキューの DefPriority 属性の優先度で登録されます。このときは、メッセージ記述子に指定した優先度は関係ありません。詳細については、MsgDeliverySequence 属性の説明を参照してください。

メッセージ優先度の範囲は、0 から MaxPriority 属性の値までです。詳細については、この章の「[キューマネージャの属性](#)」で MaxPriority 属性の説明を参照してください。

この属性の値を照会するときは、MQINQ 命令で MQIA\_DEF\_PRIORITY セレクタを使用してください。

## ● DistLists (MQLONG 型)

配布リストのサポート

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

配布リストメッセージを、キューに登録できるかどうか示します。

属性は、チャンネルのもう一方のキューマネージャが配布リストをサポートしているか、MCA によってローカルキューに通知するために設定されます。チャンネルの向こう側のキューマネージャ（相手キューマネージャ）は、送信 MCA によってローカル転送キューから取り出されたあと、メッセージを受け取ります。

DistLists 属性は、送信 MCA によって、相手キューマネージャ上の受信 MCA と接続する時に、常に設定されます。送信 MCA は、ローカルキューマネージャが相手キューマネージャが正しく処理できるメッセージだけを転送キューに登録するようにします。

この属性は主に転送キューに使用されますが、記述された処理は、キューに対して定義された使用方法（Usage 属性）とは関係なく実行されます。

次の値を取ります。

### MQDL\_SUPPORTED

配布リストはサポートされています。

配布リストメッセージはキューに格納され、そのままの形式で相手キューマネージャに転送されます。したがって、メッセージを複数のあて先に送信するときの処理を削減します。

## MQDL\_NOT\_SUPPORTED

配布リストはサポートされていません。

配布リストメッセージは、キューに格納されていません。これは、相手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です。アプリケーションが配布リストメッセージを登録し、メッセージがこのキューに登録されると、キューマネージャは配布リストを分割します。そして、代わりにキューにメッセージを登録します。したがって、メッセージを複数のあて先に送信するときの処理が増加します。ただし、メッセージが相手キューマネージャによって正しく処理されることは保証されます。

DistLists 属性の値を確定するためには、MQINQ 命令で、MQIA\_DIST\_LISTS セレクタを使用してください。

この属性の値を変更するには MQSET 命令を使用します。

### ● HardenGetBackout (MQLONG 型)

ロールバック回数記録方法

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

メッセージ記述子 (MQMD 構造体) の BackoutCount フィールドのロールバック回数に関連する情報です。

TP1/Message Queue では、この属性は常に次の値になります。

## MQQA\_BACKOUT\_NOT\_HARDENED

バックアウト回数がカウントされない場合があります。このキューにあるメッセージのバックアウト回数が正確であることを保証するような記録はしません。回数は実際の値よりも小さい場合があります。

この属性の値を照会するときは、MQINQ 命令で MQIA\_HARDEN\_GET\_BACKOUT セレクタを使用してください。

### ● InhibitGet (MQLONG 型)

取り出し許可

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	○	×	×

このキューからの取り出しを許可するかどうかを制御します。

この属性は、ローカルキュー、モデルキュー、または別名キューのときだけ有効です。

別名キューで MQGET 命令が成功するためには、別名キューとそのベースキューの両方で取り出しが許可されている必要があります。

次のどちらかの値を取ります。

### MQQA\_GET\_INHIBITED

取り出しを禁止します。

MQGET 命令は、理由コード MQRC\_GET\_INHIBITED で失敗します。MQGET 命令で MQGMO\_BROWSE\_FIRST または MQGMO\_BROWSE\_NEXT オプションを指定した場合も含まれます。

#### 注意事項

トランザクション内の MQGET 命令が成功したあとに、InhibitGet 属性が MQQA\_GET\_INHIBITED に変更された場合でも、そのトランザクションはコミットできます。

### MQQA\_GET\_ALLOWED

取り出しを許可します。

この属性の値を照会するときは、MQINQ 命令で MQIA\_INHIBIT\_GET セレクタを使用してください。この属性を変更するときは、MQSET 命令を使用してください。

## ● InhibitPut (MQLONG 型)

登録許可

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
<input type="radio"/>				

このキューへの登録を許可するかどうかを制御します。

キューの実名をたどるまでに複数の定義がある場合、MQPUT 命令および MQPUT1 が成功するためには、キューマネージャの別名の定義も含む、すべての定義で登録が許可されている必要があります。

次のどちらかの値を取ります。

### MQQA\_PUT\_INHIBITED

登録を禁止します。

MQPUT 命令および MQPUT1 命令は、理由コード MQRC\_PUT\_INHIBITED で失敗します。

#### 注意事項

トランザクション内の MQPUT 命令が成功したあとに、InhibitPut 属性が MQQA\_PUT\_INHIBITED に変更された場合でも、そのトランザクションはコミットできます。

## MQQA\_PUT\_ALLOWED

登録を許可します。

この属性の値を照会するときは、MQINQ 命令で MQIA\_INHIBIT\_PUT セレクタを使用してください。  
この属性を変更するときは、MQSET 命令を使用してください。

### ● InitiationQName (MQCHAR48 型)

イニシエーションキュー名

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

ローカルキューマネージャに定義されたキューの名称です。つまり、キュータイプが MQQT\_LOCAL のキューです。イニシエーションキューは、キューマネージャがトリガメッセージを通知するあて先のキューです。トリガメッセージは、メッセージの到着を契機にアプリケーションを起動したいときに、それを要求すると通知されます。トリガメッセージの受信を契機として適切なアプリケーションを起動するために、トリガモニタアプリケーションでイニシエーションキューを監視する必要があります。

この属性の値を照会するときは、MQINQ 命令で MQCA\_INITIATION\_Q\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

### ● MaxMsgLength (MQLONG 型)

最大メッセージ長

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

このキューに格納するメッセージの、アプリケーションデータの最大長です。

キューの MaxMsgLength 属性は、キューマネージャの MaxMsgLength 属性とは異なる値を設定できます。また、キューに登録できる最も長い物理メッセージは、ローカルキューおよびメッセージキューマネージャの MaxMsgLength 属性の値より小さくなります。

MaxMsgLength の長さを超えるメッセージを登録しようとした場合、命令は、次のどちらかで失敗します。

- メッセージがキューに対して大き過ぎる場合、理由コード MQRC\_MSG\_TOO\_BIG\_FOR\_Q で失敗します。
- メッセージがキューマネージャに対しては大きいですが、キューに対しては大きくない場合は、理由コード MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR で失敗します。

この属性値の範囲は、0 からキューマネージャの MaxMsgLength 属性の値までです。

詳細については、「2. メッセージキューイング機能の命令」の「MQPUT 命令 - メッセージの登録」で BufferLength 引数の説明を参照してください。

この属性の値を照会するときは、MQINQ 命令で MQIA\_MAX\_MSG\_LENGTH セレクタを使用してください。

## ● MaxQDepth (MQLONG 型)

最大メッセージ登録数

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

キューに同時に格納できるメッセージの最大数です。MaxQDepth の個数を格納しているキューにさらに登録しようとした場合、理由コード MQRC\_Q\_FULL で命令は失敗します。

コミット単位の処理およびメッセージのフラグメンテーションによって、キューにある物理メッセージ数が MaxQDepth を超える数になります。ただし、これはメッセージの取り出しには影響しません。キューにあるすべてのメッセージは、MQGET 命令で取り出し可能です。

この属性の値は、0 以上です。

この属性の値を照会するときは、MQINQ 命令で MQIA\_MAX\_Q\_DEPTH セレクタを使用してください。

## ● MsgDeliverySequence (MQLONG 型)

メッセージ配布順序

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

MQGET 命令でアプリケーションに返されるメッセージの順序を示します。

### MQMDS\_FIFO

FIFO 順序（先入れ先出し）でメッセージが返されます。

該当するメッセージの中で最も早く到着したものが、MQGET 命令で返されます。このとき、優先度は無視されます。

### MQMDS\_PRIORITY

優先度の順序でメッセージが返されます。

該当するメッセージの中で最も優先度の高いものが、MQGET 命令で返されます。同じ優先度のメッセージでは、FIFO 順序（先入れ先出し）で返されます。

キューにメッセージが格納されているときに関連する属性が変更されると、配布順序は次のようになります。

MQGET 命令で返されるメッセージの順序は、メッセージを格納したときの MsgDeliverySequence 属性と DefPriority 属性の値で決まります。

- メッセージ到着時の MsgDeliverySequence 属性が MQMDS\_FIFO の場合、DefPriority 属性のメッセージ優先度で格納されます。
- メッセージ到着時の MsgDeliverySequence 属性が MQMDS\_PRIORITY の場合、メッセージ記述子の Priority フィールドの優先度で格納されます。また、MQMD 構造体の Priority フィールドの値に影響することはありません。メッセージが最初に登録されたときの優先度を保ちます。

キューにメッセージが格納されているときに MsgDeliverySequence 属性が変更されても、キュー内のメッセージの順序は変更されません。

DefPriority 属性が変更されると、MsgDeliverySequence 属性が MQMDS\_FIFO の場合でも、必ずしもメッセージが FIFO 順序で配布されるとは限りません。優先度の高いメッセージが、先に配布されることがあります。

この属性の値を照会するときは、MQINQ 命令で MQIA\_MSG\_DELIVERY\_SEQUENCE セレクタを使用してください。

### ● OpenInputCount (MQLONG 型)

入力オープン数

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	×	×	×

MQGET 命令の取り出し用にそのキューを使用している、現時点のハンドルの個数です。このハンドルの総数は、ローカルキューマネージャがつかんでいます。

このキューの別名キューが取り出し用にオープンされた場合のハンドルも、この数に含まれます。検索など、取り出し以外の目的でオープンされた場合は、この数には含まれません。

この属性の値は、キューマネージャの操作で変動します。

この属性の値を照会するときは、MQINQ 命令で MQIA\_OPEN\_INPUT\_COUNT セレクタを使用してください。

### ● OpenOutputCount (MQLONG 型)

出力オープン数

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	×	×	×

MQPUT 命令の登録用にそのキューを使用している、現時点のハンドルの個数です。このハンドルの総数は、ローカルキューマネージャがつかんでいます。リモートキューマネージャのキューを登録用にオープンした場合は、この数に含まれません。

このキューの別名キューが登録用にオープンされた場合のハンドルも、この数に含まれます。照会など、登録以外の目的でオープンされた場合は、この数に含まれません。

この属性の値は、キューマネージャの操作で変動します。

この属性の値を照会するときは、MQINQ 命令で MQIA\_OPEN\_OUTPUT\_COUNT セレクタを使用してください。

### ● ProcessName (MQCHAR48 型)

プロセス名

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

ローカルキューマネージャに定義されたプロセスオブジェクトの名称です。プロセスオブジェクトは、キューに対してサービスするプログラムを識別します。トリガについては、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。

この属性の値を照会するときは、MQINQ 命令で MQCA\_PROCESS\_NAME セレクタを使用してください。

この属性の長さは、MQ\_PROCESS\_NAME\_LENGTH に定義されています。

### ● QDesc (MQCHAR64 型)

キュー記述子

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	○	○	○

補足説明を記述するフィールドです。キューマネージャはこの内容を特に使用しません。このフィールドには、表示できる文字だけを使用してください。ヌル文字は使用できません。必要に応じて余りを空白で埋めてください。

#### ■ 注意事項

キューマネージャの文字セットにない、つまりキューマネージャの CodedCharSetId 属性で定義されていない文字は、ほかのキューマネージャへ送信したとき、正しく変換されないことがあります。

この属性の値を照会するときは、MQINQ 命令で MQCA\_Q\_DESC セレクタを使用してください。

この属性の長さは、MQ\_Q\_DESC\_LENGTH に定義されています。

### ● QName (MQCHAR48 型)

キュー名

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	○	○	○

ローカルキューマネージャに定義されたキューの名称です。キューを含むオブジェクトの命名規則については、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。同じキューマネージャに定義されたキューの場合、キュー名の中に同じ文字列を含みます。このため、ローカルキューと別名キューが同じ名称になることはありません。

この属性の値を照会するときは、MQINQ 命令で MQCA\_Q\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

### ● QType (MQLONG 型)

キュータイプ

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	○	○	○

次のどれかの値を取ります。

MQQT\_ALIAS：別名キュー定義

MQQT\_CLUSTER：クラスタキュー

MQQT\_LOCAL：ローカルキュー

MQQT\_MODEL：モデルキュー定義

MQQT\_REMOTE：リモートキューのローカル定義

この属性の値を照会するときは、MQINQ 命令で MQIA\_Q\_TYPE セレクタを使用してください。

### ● RemoteQMgrName (MQCHAR48 型)

リモートキューマネージャ名

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
×	×	×	○	×

RemoteQName 属性に示すキューを定義しているリモートキューマネージャの名称です。

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、オープンした時点で RemoteQMgrName 属性が空白ではない必要があります。また、ローカルキューマネージャ名でもいけません。オープンした時点で XmitQName 属性が空白の場合は、RemoteQMgrName 属性と同じ名称のローカルキューが必要です。このキューは、転送キューとして使用されます。

この定義をキューマネージャの別名として使用する場合、RemoteQMgrName 属性は、別名を付けられるキューマネージャの名称です。ローカルキューマネージャの名称でもかまいません。オープンした時点で XmitQName 属性が空白の場合は、RemoteQMgrName 属性と同じ名称のローカルキューが必要です。このキューは、転送キューとして使用されます。

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 RemoteQMgrName 属性のキューマネージャ名が、ReplyToQMgr フィールドに設定されます。

## 注意事項

キューの定義を作成・修正したときには、この属性の内容が正しいかどうかはチェックされません。

この属性の値を照会するときは、MQINQ 命令で MQCA\_REMOTE\_Q\_MGR\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

## ● RemoteQName (MQCHAR48 型)

リモートキュー名

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
×	×	×	○	×

RemoteQMgrName 属性で示すリモートキューマネージャに定義されているキューの名称です。

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、オープンした時点で RemoteQName 属性が空白ではない必要があります。

この定義をキューマネージャの別名として使用する場合、オープンした時点で RemoteQName 属性が空白である必要があります。

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 RemoteQName 属性のキュー名が、ReplyToQ フィールドに設定されます。

## 注意事項

キューの定義を作成・修正したときには、この属性の内容が正しいかどうかはチェックされません。

この属性の値を照会するときは、MQINQ 命令で MQCA\_REMOTE\_Q\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

### ● RetentionInterval (MQLONG 型)

キュー保持時間

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

キューを保持する時間です。この時間が経過すると、キューは破棄できる状態になります。

これは、キューが作成された日付および時刻からの時間で、単位は時間です。キューが作成された日付と時刻は、CreationDate 属性と CreationTime 属性にそれぞれ記録されています。

アプリケーションまたはオペレータは、必要がなくなったキューを RetentionInterval 属性の値で判断して削除できます。

## 注意事項

キューマネージャがこの属性に従ってキューを削除することはありません。また、キュー保持時間が経過していないキューの削除を抑制することはありません。キューはユーザの判断で削除してください。

キュー保持期間は、不要になった永続的動的キューの増加を避けるために使用してください。なお、定義済みキューの場合でも使用できます。

この属性の値を照会するときは、MQINQ 命令で MQIA\_RETENTION\_INTERVAL セレクタを使用してください。

### ● Scope (MQLONG 型)

キューの有効範囲

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	×	○	○	×

セルディレクトリの範囲でこのキューに登録するかどうかを制御します。

次の値を取ります。

## MQSCO\_Q\_MGR

キューマネージャの範囲で使用します。

キューの定義は、キューマネージャの範囲で有効です。つまり、定義したキューマネージャ以外には認識されません。ほかのキューマネージャが登録用にこのキューをオープンするには、キューマネージャ名を指定するか、キューのローカル定義を使用する必要があります。

この属性の値を照会するときは、MQINQ 命令で MQIA\_SCOPE セレクタを使用してください。

### ● Shareability (MQLONG 型)

取り出し時の共用性

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

このキューに対して、取り出しのために同時に複数のオープンができるかどうかを示します。次のどちらかの値を取ります。

## MQQA\_SHAREABLE

キューを共用できます。

MQOO\_INPUT\_SHARED オプションでの複数のオープンができます。

## MQQA\_NOT\_SHAREABLE

キューを共用できません。

MQOO\_INPUT\_SHARED オプションでの MQOPEN 命令は、MQOO\_INPUT\_EXCLUSIVE と同様に扱われます。

この属性の値を照会するときは、MQINQ 命令で MQIA\_SHAREABILITY セレクタを使用してください。

### ● TriggerControl (MQLONG 型)

トリガ制御

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

トリガメッセージをイニシエーションキューに通知させるかどうかを示します。トリガメッセージは、キューに対応するサービスのアプリケーションを起動（開始）する契機となります。トリガによるアプリケーション開始については、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。

次のどちらかの値を取ります。

## MQTC\_OFF

トリガメッセージを要求しません。

このキューのイベントでは、トリガメッセージは通知されません。この場合、TriggerType 属性は無効です。

## MQTC\_ON

トリガメッセージを要求します。

該当するトリガイベントがこのキューに発生したときに、トリガメッセージが通知されます。

この属性の値を照会するときは、MQINQ 命令で MQIA\_TRIGGER\_CONTROL セレクタを使用してください。この属性を変更するときは、MQSET 命令を使用してください。

### ● TriggerData (MQCHAR64 型)

トリガデータ

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

任意の形式で設定できるトリガメッセージのデータです。キューマネージャはトリガメッセージを通知するときに、メッセージ内にこのデータを組み込みます。

キューマネージャはこの内容を特に使用しません。イニシエーションキューを監視するトリガモニタアプリケーション、またはトリガモニタアプリケーションによって起動（開始）されたアプリケーションで使用します。トリガによるアプリケーション開始については、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。

このデータ内では、ヌル文字を使用できません。必要なら、余りを空白で埋めてください。

この属性の値を照会するときは、MQINQ 命令で MQCA\_TRIGGER\_DATA セレクタを使用してください。この属性を変更するときは、MQSET 命令を使用してください。

この属性の長さは、MQ\_TRIGGER\_DATA\_LENGTH に定義されています。

### ● TriggerDepth (MQLONG 型)

トリガのためのメッセージ登録数の下限値

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

トリガメッセージが通知されるまでの、キューのメッセージ登録数です。TriggerType 属性が MQTT\_DEPTH のときに使用されます。TriggerDepth 属性の値は、1 以上です。TriggerType 属性が

MQTT\_DEPTH 以外のおときは、この属性は使用されません。トリガについては、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。

この属性の値を照会するときは、MQINQ 命令で MQIA\_TRIGGER\_DEPTH セレクタを使用してください。この属性を変更するときは、MQSET 命令を使用してください。

### ● TriggerMsgPriority (MQLONG 型)

トリガのためのメッセージ優先度の下限値

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

この属性値より低い優先度のメッセージでは、トリガメッセージは通知されません。つまり、キューマネージャがトリガメッセージを通知するかどうかを判断するとき、優先度の低いメッセージは無視されます。この属性値の範囲は、0 から MaxPriority 属性の値までです。この属性値が 0 の場合、すべてのメッセージについて、トリガメッセージを通知するかどうか判断されます。トリガについては、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。

この属性の値を照会するときは、MQINQ 命令で MQIA\_TRIGGER\_MSG\_PRIORITY セレクタを使用してください。この属性を変更するときは、MQSET 命令を使用してください。

### ● TriggerType (MQLONG 型)

トリガタイプ

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

トリガメッセージを通知させる条件を示します。トリガについては、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。

次のどれかの値を取ります。

#### MQTT\_NONE

トリガメッセージは通知されません。

このキューにメッセージが到着しても、トリガメッセージは通知されません。つまり、TriggerControl 属性が MQTT\_OFF の場合と同様です。

#### MQTT\_FIRST

空ではないときに通知されます。

このキューが空、つまり、メッセージがない状態から、空ではない、つまり、メッセージがある状態に変化したときに、トリガメッセージが通知されます。

## MQTT\_EVERY

メッセージごとに通知されます。

メッセージがこのキューに到着するごとに、トリガメッセージが通知されます。

## MQTT\_DEPTH

一定の格納数を越えたときに通知されます。

このキューに一定数 (TriggerDepth 属性の値) を超えてメッセージが格納されたときに、トリガメッセージが通知されます。トリガメッセージが通知されると、TriggerControl 属性が MQTC\_OFF になります。これによって、通知が連続することがなくなります。TriggerControl 属性は、設定し直すまで MQTC\_OFF のままです。

この属性の値を照会するときは、MQINQ 命令で MQIA\_TRIGGER\_TYPE セレクタを使用してください。この属性を変更するときは、MQSET 命令を使用してください。

## ● Usage (MQLONG 型)

使用種別

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
○	○	×	×	×

キューの使用種別を示します。次のどちらかの値を取ります。

## MQUS\_NORMAL

通常キュー

このキューは、通常のアプリケーションがメッセージを登録・取り出しするために使用されます。転送キューではありません。

## MQUS\_TRANSMISSION

転送キュー

このキューは、リモートキューマネージャへ送信するメッセージを格納するために使用されます。通常のアプリケーションがメッセージをリモートキューへ送信するとき、メッセージはローカルキューマネージャによって一時的に転送キューに特別な形式で格納されます。その後、MCA が、転送キューからメッセージを取り出してリモートキューマネージャへ送信します。転送キューについては、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。

アプリケーションは、転送キューを MQOO\_OUTPUT オプションでオープンして、メッセージを直接それに登録できます。ユティリティアプリケーションがこれに当たります。そのアプリケーションは、メッセージデータの形式が正しいかどうか、注意する必要があります。詳細については、「1. データタイプ」の「MQXQH 構造体 - 転送キューヘッダ」を参照してください。メッセージデータの形式が正しくない場合、転送処理中に障害が発生することがあります。コンテキストオプション MQPMO\_\* のどれかを指定しないと、コンテキストは付けられません。

この属性の値を照会するときは、MQINQ 命令で MQIA\_USAGE セレクタを使用してください。

## ● XmitQName (MQCHAR48 型)

転送キュー名

ローカルキュー	モデルキュー	別名キュー	リモートキューのローカル定義	クラスタキュー
×	×	×	○	×

この属性は、リモートキューのローカル定義またはキューマネージャの別名の場合に使用できます。オープンした時点でこの属性が空白ではない場合、メッセージを送信するための転送キューを示します。

XmitQName 属性が空白の場合、RemoteQMgrName 属性と同じ名称のキューが、代わりに転送キューとして使用されます。

キューマネージャの別名の場合で、RemoteQMgrName 属性がローカルキューマネージャ名るとき、この属性は無視されます。

また、応答キューの別名の場合も、この属性は無視されます。

この属性の値を照会するときは、MQINQ 命令で MQCA\_XMIT\_Q\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

## プロセス定義の属性

---

プロセス定義には、次の属性があります。

### 属性

#### ● ApplId (MQCHAR256 型) アプリケーション識別子

起動させるアプリケーションを識別する文字列です。

この情報は、イニシエーションキューのメッセージを処理するトリガモニタアプリケーションで使用します。トリガメッセージに組み込まれてイニシエーションキューへ送られます。

この情報の意味は、トリガモニタアプリケーションで決めます。例えば、ApplId 属性で次を示すことができます。

- プログラム名 (MQAT\_MVS アプリケーションの場合)
- CICS トランザクション識別子 (MQAT\_CICS アプリケーションの場合)

この文字列にはヌル文字は使用できません。必要に応じて余りを空白で埋めてください。

この属性の値を照会するときは、MQINQ 命令で MQCA\_APPL\_ID セレクタを使用してください。

この属性の長さは、MQ\_PROCESS\_APPL\_ID\_LENGTH に定義されています。

#### ● ApplType (MQLONG 型) アプリケーションタイプ

トリガメッセージの受信に応じて起動させるプログラムの種類を示します。

この情報は、イニシエーションキューのメッセージを処理するトリガモニタアプリケーションで使用します。トリガメッセージに組み込まれてイニシエーションキューへ送られます。

ApplType 属性はどんな値でもとれます。ただし、通常は、次の標準のアプリケーションタイプです。ユーザが定義するアプリケーションタイプは、MQAT\_USER\_FIRST から MQAT\_USER\_LAST の範囲の値にしてください。

MQAT\_AIX : AIX アプリケーション

MQAT\_CICS : CICS トランザクション

MQAT\_DOS : PC-DOS 上の IBM MQ クライアントアプリケーション

MQAT\_IMS : IMS アプリケーション

MQAT\_MVS : MVS または TSO アプリケーション

MQAT\_OPEN\_TP1 : OpenTP1 アプリケーション

MQAT\_OS2 : OS/2 またはプレゼンテーションマネージャアプリケーション

MQAT\_OS400 : OS/400 アプリケーション

MQAT\_UNIX : UNIX アプリケーション

MQAT\_TMS\_4V : TMS-4V/SP アプリケーション

MQAT\_WINDOWS : 16 ビット Windows アプリケーション

MQAT\_WINDOWS\_NT : 32 ビット Windows アプリケーション

MQAT\_XDM : XDM アプリケーション

MQAT\_USER\_FIRST : ユーザ定義のアプリケーションタイプの最小値

MQAT\_USER\_LAST : ユーザ定義のアプリケーションタイプの最大値

この属性の値を照会するときは、MQINQ 命令で MQIA\_APPL\_TYPE セレクタを使用してください。

### ● EnvData (MQCHAR128 型) 環境データ

起動させるアプリケーションに関連する環境についてのデータです。

この情報は、イニシエーションキューのメッセージを処理するトリガモニタアプリケーションで使用します。トリガメッセージに組み込まれてイニシエーションキューへ送られます。

EnvData の意味は、トリガモニタアプリケーションで決めます。

この文字列にはヌル文字は使用できません。必要に応じて余りを空白で埋めてください。

この属性の値を照会するときは、MQINQ 命令で MQCA\_ENV\_DATA セレクタを使用してください。

この属性の長さは、MQ\_PROCESS\_ENV\_DATA\_LENGTH に定義されています。

### ● ProcessDesc (MQCHAR64 型) プロセス記述子

補足説明を記述するフィールドです。キューマネージャはこの内容を特に使用しません。このフィールドには、表示できる文字だけを使用してください。ヌル文字は使用できません。必要に応じて余りを空白で埋めてください。

#### ■ 注意事項

キューマネージャの文字セットにない、つまり、キューマネージャの CodedCharSetId 属性で定義されていない文字は、ほかのキューマネージャへ送信したとき、正しく変換されないことがあります。

この属性の値を照会するときは、MQINQ 命令で MQCA\_PROCESS\_DESC セレクタを使用してください。

この属性の長さは、MQ\_PROCESS\_DESC\_LENGTH に定義されています。

### ● ProcessName (MQCHAR48 型) プロセス名

ローカルキューマネージャに定義されたプロセス定義の名称です。

一つのキューマネージャに属するプロセス定義は、互いに名称が異なります。ただし、キューなど、ほかのオブジェクトの定義と同じ名称の場合があります。

この属性の値を照会するときは、MQINQ 命令で MQCA\_PROCESS\_NAME セレクタを使用してください。

この属性の長さは、MQ\_PROCESS\_NAME\_LENGTH に定義されています。

### ● UserData (MQCHAR128 型) ユーザデータ

起動させるアプリケーションに関連するユーザ任意の情報です。

この情報は、イニシエーションキューのメッセージを処理するトリガモニタアプリケーションで使用します。トリガメッセージに組み込まれてイニシエーションキューへ送られます。

UserData の意味は、トリガモニタアプリケーションで決めます。

この文字列にはヌル文字は使用できません。必要に応じて余りを空白で埋めてください。

この属性の値を照会するときは、MQINQ 属性で MQCA\_USER\_DATA セレクタを使用してください。

この属性の長さは、MQ\_PROCESS\_USER\_DATA\_LENGTH に定義されています。

## キューマネージャの属性

キューマネージャ自身にも、多数の属性があります。これらは、MQOT\_Q\_MGR オブジェクトをオープンするとき、オープン時に返されたハンドルを使用して MQINQ 命令を呼び出すときに必要です。

### 属性

#### ● CodedCharSetId (MQLONG 型) 文字セット識別子

キューマネージャで使用される文字セットを定義します。この文字セットは、オブジェクトの名称、キューの作成日付・作成時刻などの、MQI で定義するすべての文字列で使用されます。文字セットは、オブジェクト名として有効な 1 バイト文字を持つ必要があります。この識別子は、メッセージ内のアプリケーションデータには適用されません。

この属性の値を照会するときは、MQINQ 命令で MQIA\_CODED\_CHAR\_SET\_ID セレクタを使用してください。

#### ● DeadLetterQName (MQCHAR48 型) デッドレターキュー名

ローカルキューマネージャに定義されたキューの名称です。メッセージがあて先キューに届かなかったときに、そのメッセージがこのキューへ転送されます。

例えば、次の場合にこのキューへ転送されます。

- メッセージがキューマネージャに到着したが、キューマネージャにあて先キュー（受信キュー）が定義されていない場合
- メッセージがキューマネージャに到着したが、次の理由などであて先キュー（受信キュー）に格納できなかった場合
  - － キューが満杯のとき
  - － 登録が禁止されていたとき
  - － 送信側のノードが、キューへ登録する権限を持っていなかったとき

また、アプリケーションから直接メッセージをデッドレターキューへ登録することもできます。

報告メッセージも、通常のメッセージと同様に扱われます。つまり、報告メッセージが目的のキューに格納できなかった場合も、デッドレターキューへ転送されます。なお、この場合の目的のキューは、通常、登録元メッセージのメッセージ記述子の ReplyToQ フィールドに指定されたキューになります。

### 注意事項

保持時間が終了したメッセージが破棄されるときは、このキューへは転送されません。ただし、登録元のアプリケーションが要求していれば、保持時間終了報告メッセージ (MQRO\_EXPIRATION) が生成され、ReplyToQ フィールドのキューに登録されます。

MQPUT 命令または MQPUT1 命令で返される理由コードで、直後にエラーが判明する場合は、メッセージはデッドレターキューへ転送されません。例えば、登録が禁止されているローカルキューへ登録した場合などは、理由コードで直後にエラーが判明します。

デッドレターキューに格納されたメッセージの中には、そのアプリケーションデータの先頭に、MQDLH 構造体が付けられているものがあります。この構造体には、デッドレターキューへメッセージが転送された理由が示されます。MQDLH 構造体については、「1. データタイプ」の「MQDLH 構造体 - デッドレターヘッダ」を参照してください。

このキューは、Usage 属性が MQUS\_NORMAL のローカルキューである必要があります。

キューマネージャがデッドレターキューをサポートしていない場合、または定義していない場合は、この名称はすべて空白です。

TP1/Message Queue のすべてのキューマネージャは、デッドレターキューをサポートしていますが、初期値としては定義されていません。

デッドレターキューが未定義、満杯、そのほかの理由で使用できないとき、MCA によって転送されたメッセージは、代わりに転送キューで保持されます。

この属性の値を照会するときは、MQINQ 命令で MQCA\_DEAD\_LETTER\_Q\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

### ● DefXmitQName (MQCHAR48 型) 省略時の転送キュー名

リモートキューマネージャへメッセージを送信するときに使用される転送キューの名称です。ほかに転送キューの指定がないときに使用されます。

省略時の転送キューがない場合、この名称は空白です。

この属性の値を照会するときは、MQINQ 命令で MQCA\_DEF\_XMIT\_Q\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_NAME\_LENGTH に定義されています。

### ● DistLists (MQLONG 型) 配布リストのサポート

MQPUT 命令および MQPUT1 命令で、ローカルキューマネージャが配布リストをサポートしているかどうかを示します。値は次のどちらかです。

#### MQDL\_SUPPORTED

配布リストはサポートされています。

#### MQDL\_NOT\_SUPPORTED

配布リストはサポートされていません

この属性の値を確定するためには、MQINQ 命令で、MQIA\_DIST\_LISTS セレクタを使用してください。値を変更する場合は、MQSET 命令を使用してください。

### ● MaxHandles (MQLONG 型) 最大ハンドル数

一つのタスクで同時にオープンできるハンドルの最大数です。

この属性値の範囲は、1~999999999 です。

この属性の値を照会するときは、MQINQ 命令で MQIA\_MAX\_HANDLES セレクタを使用してください。

### ● MaxMsgLength (MQLONG 型) 最大メッセージ長

キューマネージャで扱えるメッセージの最大長です。キューの MaxMsgLength 属性は、これより大きい値をとれません。

この属性の値を照会するときは、MQINQ 命令で MQIA\_MAX\_MSG\_LENGTH セレクタを使用してください。

### ● MaxPriority (MQLONG 型) 最大メッセージ優先度

キューマネージャがサポートするメッセージ優先度の最大値です。優先度の範囲は、0 から MaxPriority 属性の値までです。

この属性の値を照会するときは、MQINQ 命令で MQIA\_MAX\_PRIORITY セレクタを使用してください。

### ● MaxUncommittedMsgs (MQLONG 型) トランザクション完了待ち最大メッセージ数

TP1/Message Queue では、この属性は使用されません。

この属性値は、常に 0 です。

この属性の値を照会するときは、MQINQ 命令で MQIA\_MAX\_UNCOMMITTED\_MSGS セレクタを使用してください。

### ● Platform (MQLONG 型) プラットフォーム

キューマネージャが動作しているプラットフォームのアーキテクチャを示します。

MQPL\_AIX : AIX

MQPL\_MVS : MVS/ESA

MQPL\_OPEN\_TP1 : OpenTP1

MQPL\_OS2 : OS/2

MQPL\_OS400 : OS/400

MQPL\_UNIX : UNIX

MQPL\_WINDOWS : Windows 3.1

MQPL\_WINDOWS\_NT : Windows

MQPL\_XDM : XDM

MQPL\_TMS\_4V : TMS-4V/SP/Message Queue

この属性の値を照会するときは、MQINQ 命令で MQIA\_PLATFORM セレクタを使用してください。

### ● QMgrDesc (MQCHAR64 型) キューマネージャ記述子

補足説明を記述するフィールドです。キューマネージャはこの内容を特に使用しません。このフィールドには、表示できる文字だけを使用してください。ヌル文字は使用できません。必要に応じて余りを空白で埋めてください。

#### 注意事項

キューマネージャの文字セットにない、つまり、キューマネージャの CodedCharSetId 属性で定義されていない文字は、ほかのキューマネージャへ送信したとき、正しく変換されないことがあります。

この属性の値を照会するときは、MQINQ 命令で MQCA\_Q\_MGR\_DESC セレクタを使用してください。

この属性の長さは、MQ\_Q\_MGR\_DESC\_LENGTH に定義されています。

### ● QMgrName (MQCHAR48 型) キューマネージャ名

ローカルキューマネージャの名称です。これは、アプリケーションがキューマネージャに接続するために使用する名称です。

名称の先頭の 12 文字は、ユニークなメッセージ識別子が生成されるときに使用されます。詳細については、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」で MsgId フィールドの説明を参照してください。通信するキューマネージャ同士では、名称の先頭の 12 文字が互いに異なっている必要があります。これによって、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になります。

この属性の値を照会するときは、MQINQ 命令で MQCA\_Q\_MGR\_NAME セレクタを使用してください。

この属性の長さは、MQ\_Q\_MGR\_NAME\_LENGTH に定義されています。

### ● SyncPoint (MQLONG 型) トランザクション有効性

MQGET 命令、MQPUT 命令、および MQPUT1 命令の同期点処理をローカルキューマネージャでサポートするかどうかを示します。

MQSP\_AVAILABLE : トランザクションの同期点処理をサポートします。

MQSP\_NOT\_AVAILABLE : トランザクションの同期点処理をサポートしません。

この属性の値を照会するときは、MQINQ 命令で MQIA\_SYNCPOINT セレクタを使用してください。

## ● TriggerInterval (MQLONG 型) トリガ間隔

トリガメッセージの数を抑制するために使用される時間間隔で、単位はミリ秒です。この属性は、TriggerType 属性が MQTT\_FIRST のときだけ有効です。MQTT\_FIRST のときは、通常、空のキューに適切なメッセージが到着したときだけ、トリガメッセージが通知されます。ただし、MQTT\_FIRST の場合でもキューが空ではない場合に特別なトリガメッセージを通知させることができます。これらの特別なトリガメッセージは、TriggerInterval 属性の時間間隔より頻繁に通知されることはありません。

トリガについては、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。

この属性値の範囲は、0~999999999 です。

この属性の値を照会するときは、MQINQ 命令で MQIA\_TRIGGER\_INTERVAL セレクタを使用してください。

# 付録

## 付録 A 命令および引数の一覧

メッセージキューイング機能で使用する命令ごとに、引数の一覧を示します。また、オプションを指定する引数については、指定できるオプションの一覧を示します。

### 付録 A.1 MQCLOSE 命令の引数一覧

MQCLOSE 命令で指定できる引数の一覧を次の表に示します。

表 A-1 MQCLOSE 命令の引数一覧

引数	データタイプ	向き	内容
Hconn	MQHCONN	i	コネクションハンドル
Hobj	MQHOBJ	i/o	オブジェクトハンドル
Options	MQLONG	i	クローズオプション
CompCode	MQLONG	o	完了コード
Reason	MQLONG	o	理由コード

(凡例)

i：入力用

i/o：入出力用

o：出力用

MQCLOSE 命令の Options 引数で指定できるクローズオプションの一覧を次の表に示します。

表 A-2 クローズオプション一覧 (MQCLOSE 命令の Options 引数)

オプション	指定方法	オープン時に必要なオプション
MQCO_NONE	• どれか一つだけ指定 • 省略不可	なし
MQCO_DELETE		
MQCO_DELETE_PURGE		

### 付録 A.2 MQCONN 命令の引数一覧

MQCONN 命令で指定できる引数の一覧を次の表に示します。

表 A-3 MQCONN 命令の引数一覧

引数	データタイプ	向き	内容
Name	MQHCHAR48	i	キューマネージャ名

引数	データタイプ	向き	内容
Hconn	MQHCONN	o	コネクションハンドル
CompCode	MQLONG	o	完了コード
Reason	MQLONG	o	理由コード

(凡例)

i : 入力用

o : 出力用

## 付録 A.3 MQDISC 命令の引数一覧

MQDISC 命令で指定できる引数の一覧を次の表に示します。

表 A-4 MQDISC 命令の引数一覧

引数	データタイプ	向き	内容
Hconn	MQHCONN	i/o	コネクションハンドル
CompCode	MQLONG	o	完了コード
Reason	MQLONG	o	理由コード

(凡例)

i/o : 入出力用

o : 出力用

## 付録 A.4 MQGET 命令の引数一覧

MQGET 命令で指定できる引数の一覧を次の表に示します。

表 A-5 MQGET 命令の引数一覧

引数	データタイプ	向き	内容	
Hconn	MQHCONN	i	コネクションハンドル	
Hobj	MQHOBJ	i	オブジェクトハンドル	
MsgDesc	MQMD	i/o	メッセージ記述子	
フィールド	StrucId	MQCHAR4	i	構造体識別子
	Version	MQLONG	i	構造体バージョン番号
	Report	MQLONG	o	報告オプション
	MsgType	MQLONG	o	メッセージタイプ

引数		データタイプ	向き	内容
フィールド	Expiry	MLONG	o	メッセージ保持時間
	Feedback	MLONG	o	報告メッセージ返答コード
	Encoding	MLONG	o	マシンコード形式
	CodedCharSetId	MLONG	o	文字セット識別子
	Format	MQCHAR8	o	フォーマット名
	Priority	MLONG	o	メッセージ優先度
	Persistence	MLONG	o	メッセージ永続性
	MsgId	MQBYTE24	i/o	メッセージ識別子
	CorrelId	MQBYTE24	i/o	相関識別子
	BackoutCount	MLONG	o	ロールバック回数
	ReplyToQ	MQCHAR48	o	応答キュー名
	ReplyToQMgr	MQCHAR48	o	応答キューマネージャ名
	UserIdentifier	MQCHAR12	o	ユーザ識別子
	AccountingToken	MQBYTE32	o	課金トークン
	ApplIdentityData	MQCHAR32	o	アプリケーション識別データ
	PutApplType	MLONG	o	登録アプリケーションタイプ
	PutApplName	MQCHAR28	o	登録アプリケーション名
	PutDate	MQCHAR8	o	登録日付
	PutTime	MQCHAR8	o	登録時刻
	ApplOriginData	MQCHAR4	o	登録元データ
	GroupId <sup>*1</sup>	MQBYTE24	i/o	グループ識別子
MsgSeqNumber <sup>*1</sup>	MLONG	i/o	メッセージシーケンス番号	
Offset <sup>*1</sup>	MLONG	i/o	オフセット	
MsgFlags <sup>*1</sup>	MLONG	i/o	フラグ	
OriginalLength <sup>*1</sup>	MLONG	o	登録元メッセージ長	
GetMsgOpts		MQGMO	i/o	メッセージ取り出しオプション
フィールド	StrucId	MQCHAR4	i	構造体識別子
	Version	MLONG	i	構造体バージョン番号
	Options	MLONG	i	取り出しオプション
	WaitInterval	MLONG	i	待ち合わせ最大時間

引数		データタイプ	向き	内容
フィールド	Signal1 <sup>※2</sup>	MLONG	—	シグナル
	Signal2	MLONG	—	予備
	ResolvedQName	MQCHAR48	o	受信キュー名
	MatchOptions <sup>※1</sup>	MLONG	i	一致オプション
	GroupStatus <sup>※1</sup>	MQCHAR	o	メッセージグループフラグ
	SegmentStatus <sup>※1</sup>	MQCHAR	o	論理メッセージフラグ
	Segmentation <sup>※1</sup>	MQCHAR	o	セグメント分割フラグ
	Reserved1 <sup>※1</sup>	MQCHAR	—	予備
BufferLength		MLONG	i	バッファ長
Buffer		MQBYTE× BufferLength	o	バッファ
DataLength		MLONG	o	データ長
CompCode		MLONG	o	完了コード
Reason		MLONG	o	理由コード

(凡例)

- i：入力用
- i/o：入出力用
- o：出力用
- ：該当しません。

注※1

Version フィールドが、MQ\*\_VERSION\_2 の場合だけ存在するフィールドです。

注※2

TP1/Message Queue では操作しません。

MQGET 命令の引数に対応するオプションには、取り出しオプション (GetMsgOpts 引数の Options フィールド) と報告オプション (MsgDesc 引数の Report フィールド) の 2 種類があります。ただし、MQGET 命令では、MsgDesc 引数の Report フィールドは出力用のフィールドとなるため、報告オプションの指定はできません。MsgDesc 引数の Report フィールドには、MQPUT 命令または MQPUT1 命令でオプションを指定します。Report フィールドで指定できる報告オプションの一覧については、「付録 A.7 MQPUT 命令および MQPUT1 命令の引数一覧」を参照してください。

GetMsgOpts 引数の Options フィールドで指定できる取り出しオプションの一覧を次の表に示します。

表 A-6 取り出しオプション一覧 (MQGET 命令の GetMsgOpts 引数の Options フィールド)

オプション	指定方法	オープン時に必要なオプション
MQGMO_WAIT <sup>※1</sup>	<ul style="list-style-type: none"> <li>• どちらか一つだけ指定</li> </ul>	MQOO_INPUT_*

オプション	指定方法	オープン時に必要なオプション
MQGMO_NO_WAIT (省略時解釈値)	<ul style="list-style-type: none"> <li>省略可</li> </ul>	MQOO_INPUT_*
MQGMO_SYNCPOINT※2	<ul style="list-style-type: none"> <li>どれか一つだけ指定</li> <li>省略可※4</li> </ul>	
MQGMO_NO_SYNCPOINT		
MQGMO_SYNCPOINT_IF_PERSISTENT※3		
MQGMO_BROWSE_FIRST	必要時にどれか一つだけ指定	<ul style="list-style-type: none"> <li>MQOO_INPUT_*</li> <li>MQOO_BROWSE</li> </ul>
MQGMO_BROWSE_NEXT		
MQGMO_BROWSE_MSG_UNDER_CURSOR		
MQGMO_MSG_UNDER_CURSOR		
MQGMO_LOCK	必要時に、MQGMO_BROWSE_FIRST、MQGMO_BROWSE_NEXT、またはMQGMO_BROWSE_MSG_UNDER_CURSORと同時に指定	MQOO_INPUT_*
MQGMO_UNLOCK※5	必要時に指定	
MQGMO_ACCEPT_TRUNCATED_MSG		
MQGMO_FAIL_IF_QUIESCING※6		
MQGMO_CONVERT		
MQGMO_COMPLETE_MSG		
MQGMO_LOGICAL_ORDER	必要時に指定	<ul style="list-style-type: none"> <li>MQOO_INPUT_*</li> <li>MQOO_BROWSE</li> </ul>
MQGMO_ALL_MSGS_AVAILABLE		
MQGMO_ALL_SEGMENTS_AVAILABLE		
MQGMO_NONE	ほかの取り出しオプションを指定しない場合に指定	

注※1

MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR、またはMQGMO\_MSG\_UNDER\_CURSORと同時に指定すると、MQGMO\_WAITは無視されます。

注※2

MQGMO\_SYNCPOINTは、MQGMO\_BROWSE\_FIRST、MQGMO\_BROWSE\_NEXT、またはMQGMO\_BROWSE\_MSG\_UNDER\_CURSORと同時に指定できません。

注※3

MQGMO\_SYNCPOINT\_IF\_PERSISTENT は、MQGMO\_BROWSE\_FIRST, MQGMO\_BROWSE\_NEXT, MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR, MQGMO\_COMPLETE\_MSG, または MQGMO\_UNLOCK と同時に指定できません。

注※4

省略時解釈値は環境に依存します。TP1/Message Queue では、トランザクションの範囲内から呼び出したかどうかで決まります。ほかの IBM MQ の製品では、省略時解釈値が異なる場合があります。

注※5

MQGMO\_NO\_WAIT および MQGMO\_NO\_SYNCPOINT 以外の取り出しオプションと同時に指定できません。

注※6

TP1/Message Queue では、指定しても動作に影響がありません。

## 付録 A.5 MQINQ 命令の引数一覧

MQINQ 命令で指定できる引数の一覧を次の表に示します。

表 A-7 MQINQ 命令の引数一覧

引数	データタイプ	向き	内容
Hconn	MQHCONN	i	コネクションハンドル
Hobj	MQHOBJ	i	オブジェクトハンドル
SelectorCount	MQLONG	i	セレクタ数
Selectors	MQLONG× SelectorCount	i	セレクタ
IntAttrCount	MQLONG	i	整数型属性数
IntAttrs	MQLONG× IntAttrCount	o	整数型属性値
CharAttrLength	MQLONG	i	文字型属性長
CharAttrs	MQCHAR× CharAttrLength	o	文字型属性値
CompCode	MQLONG	o	完了コード
Reason	MQLONG	o	理由コード

(凡例)

i : 入力用

o : 出力用

## 付録 A.6 MQOPEN 命令の引数一覧

MQOPEN 命令で指定できる引数の一覧を次の表に示します。

表 A-8 MQOPEN 命令の引数一覧

引数	データタイプ	向き	内容	
Hconn	MQHCONN	i	コネクションハンドル	
ObjDesc	MQOD	i/o	オブジェクト記述子	
フィールド	StrucId	MLONG	i	構造体識別子
	Version	MLONG	i	構造体バージョン番号
	ObjectType	MLONG	i	オブジェクトタイプ
	ObjectName	MQCHAR48	i*1	オブジェクト名
	ObjectQMgrName	MQCHAR48	i*1	オブジェクトキューマネージャ名
	DynamicQName	MQCHAR48	i	動的キュー名
	AlternateUserId	MQCHAR12	i	代替ユーザ識別子
	RecsPresent*2	MLONG	i	オブジェクトレコードの数
	KnownDestCount*2	MLONG	o	ローカルキューの数
	UnknownDestCount*2	MLONG	o	リモートキューの数
	InvalidDestCount*2	MLONG	o	オープンに失敗したキューの数
	ObjectRecOffset*2	MLONG	i	最初のオブジェクトレコードまでのオフセット
	ResponseRecOffset*2	MLONG	i	最初の応答レコードまでのオフセット
	ObjectRecPtr*2	MQPTR	i	オブジェクトレコードのアドレス
	ResponseRecPtr*2	MQPTR	i	応答レコードのアドレス
AlternateSecurityId*3 *4	MQBYTE40	i	代替セキュリティ識別子	
ResolvedQName*4	MQCHAR48	o	解決したキュー名称	
ResolvedQMgrName* 4	MQCHAR48	o	解決したキューマネージャ名称	
Options	MLONG	i	オープンオプション	
Hobj	MQHOBJ	o	オブジェクトハンドル	
CompCode	MLONG	o	完了コード	
Reason	MLONG	o	理由コード	

(凡例)

i：入力用

i/o：入出力用

o：出力用

注※1

MQOPEN 命令でモデルキュー名を指定した場合、入出力用のフィールドとなります。

注※2

Version フィールドが、MQ\*\_VERSION\_2 の場合だけ存在するフィールドです。

注※3

TP1/Message Queue では操作しません。

注※4

Version フィールドが、MQ\*\_VERSION\_3 の場合だけ存在するフィールドです。

MQOPEN 命令の Options 引数で指定できるオープンオプションの一覧を次の表に示します。

表 A-9 オープンオプション一覧 (MQOPEN 命令の Options 引数)

オプション	指定方法	オープン時に必要なオプション
MQOO_INPUT_AS_Q_DEF	このうち一つ以上指定 (ただし、MQOO_INPUT_* は重複不可)	なし
MQOO_INPUT_SHARED		
MQOO_INPUT_EXCLUSIVE		
MQOO_BROWSE		
MQOO_OUTPUT		
MQOO_INQUIRE		
MQOO_SET		
MQOO_BIND_ON_OPEN	必要時にどれか一つだけ指定	
MQOO_BIND_NOT_FIXED		
MQOO_BIND_AS_Q_DEF		
MQOO_SAVE_ALL_CONTEXT	必要時に、MQOO_INPUT と同時に指定	
MQOO_PASS_IDENTITY_CONTEXT	必要時に、 MQOO_OUTPUT と同時に 指定 (オプションの機能に包含関係 があるため、 重複して指定する必要がない 場合があります)	
MQOO_PASS_ALL_CONTEXT		
MQOO_SET_IDENTITY_CONTEXT		
MQOO_SET_ALL_CONTEXT		
MQOO_ALTERNATE_USER_AUTHORITY	必要時に指定	
MQOO_FAIL_IF_QUIESCING*		

注※

TP1/Message Queue では、指定しても動作に影響がありません。

## 付録 A.7 MQPUT 命令および MQPUT1 命令の引数一覧

MQPUT 命令および MQPUT1 命令で指定できる引数の一覧を次の表に示します。

表 A-10 MQPUT 命令および MQPUT1 命令の引数一覧

引数	データタイプ	向き	内容	
Hconn	MQHCONN	i	コネクションハンドル	
Hobj※1	MQHOBJ	i	オブジェクトハンドル	
ObjDesc※2	MQOD	i	オブジェクト記述子	
フィールド	StrucId	MQCHAR4	i	構造体識別子
	Version	MQLONG	i	構造体バージョン番号
	ObjectType	MQLONG	i	オブジェクトタイプ
	ObjectName	MQCHAR48	i	オブジェクト名
	ObjectQMgrName	MQCHAR48	i	オブジェクトキューマネージャ名
	DynamicQName	MQCHAR48	—	動的キュー名
	AlternateUserId	MQCHAR12	i	代替ユーザ識別子
	RecsPresent※3	MQLONG	i	オブジェクトレコードの数
	KnownDestCount※3	MQLONG	o	ローカルキューの数
	UnknownDestCount※3	MQLONG	o	リモートキューの数
	InvalidDestCount※3	MQLONG	o	オープンに失敗したキューの数
	ObjectRecOffset※3	MQLONG	i	最初のオブジェクトレコードまでのオフセット
	ResponseRecOffset※3	MQLONG	i	最初の応答レコードまでのオフセット
	ObjectRecPtr※3	MQPTR	i	オブジェクトレコードのアドレス
	ResponseRecPtr※3	MQPTR	i	応答レコードのアドレス
	AlternateSecurityId※4	MQBYTE40	o	代替セキュリティ識別子
ResolvedQName※4	MQCHAR48	o	解決したキュー名称	
ResolvedQMgrName※4	MQCHAR48	o	解決したキューマネージャ名称	
MsgDesc	MQMD	—	メッセージ記述子	

引数		データタイプ	向き	内容
フィールド	StrucId	MQCHAR4	i	構造体識別子
	Version	MQLONG	i	構造体バージョン番号
	Report	MQLONG	i	報告オプション
	MsgType	MQLONG	i	メッセージタイプ
	Expiry	MQLONG	i	メッセージ保持時間
	Feedback	MQLONG	i	報告メッセージ返答コード
	Encoding	MQLONG	i	マシンコード形式
	CodedCharSetId	MQLONG	i	文字セット識別子
	Format	MQCHAR8	i	フォーマット名
	Priority	MQLONG	i	メッセージ優先度
	Persistence	MQLONG	i	メッセージ永続性
	MsgId	MQBYTE24	i/o	メッセージ識別子
	CorrelId	MQBYTE24	i	相関識別子
	BackoutCount	MQLONG	—	ロールバック回数
	ReplyToQ	MQCHAR48	i	応答キュー名
	ReplyToQMgr	MQCHAR48	i	応答キューマネージャ名
	UserIdentifier	MQCHAR12	o*5	ユーザ識別子
	AccountingToken	MQBYTE32	o*5	課金トークン
	ApplIdentityData	MQCHAR32	o*5	アプリケーション識別データ
	PutApplType	MQLONG	o*6	登録アプリケーションタイプ
	PutApplName	MQCHAR28	o*6	登録アプリケーション名
	PutDate	MQCHAR8	o*6	登録日付
	PutTime	MQCHAR8	o*6	登録時刻
	ApplOriginData	MQCHAR4	o*6	登録元データ
	GroupId*3	MQBYTE24	i/o	グループ識別子
	MsgSeqNumber*3	MQLONG	i/o	メッセージシーケンス番号
Offset*3	MQLONG	i/o	オフセット	
MsgFlags*3	MQLONG	i/o	フラグ	
OriginalLength*3	MQLONG	o	登録元メッセージ長	
PutMsgOpts		MQPMO	—	メッセージ登録オプション

引数		データタイプ	向き	内容
フィールド	StrucId	MQCHAR4	i	構造体識別子
	Version	MQLONG	i	構造体バージョン番号
	Options	MQLONG	i	登録オプション
	Timeout	MQLONG	—	予備
	Context	MQHOBJ	i	コンテキスト
	KnownDestCount	MQLONG	o	ローカルキューの数
	UnknownDestCount	MQLONG	o	リモートキューの数
	InvalidDestCount	MQLONG	o	送信に失敗したキューの数
	ResolvedQName	MQCHAR48	o	受信キュー名
	ResolvedQMgrName	MQCHAR48	o	受信キューマネージャ名
	RecsPresent <sup>*3</sup>	MQLONG	i	登録メッセージレコードまたは応答レコードの数
	PutMsgRecFields <sup>*3</sup>	MQLONG	i	MQPMR 構造体の存在
	PutMsgRecOffset <sup>*3</sup>	MQLONG	i	最初の登録メッセージレコードまでのオフセット
	ResponseRecOffset <sup>*3</sup>	MQLONG	i	最初の応答レコードまでのオフセット
PutMsgRecPtr <sup>*3</sup>	MQPTR	i	登録メッセージレコードのアドレス	
ResponseRecPtr <sup>*3</sup>	MQPTR	i	応答レコードのアドレス	
BufferLength		MQLONG	i	バッファ長
Buffer		MQBYTE× BufferLength	i	バッファ
CompCode		MQLONG	o	完了コード
Reason		MQLONG	o	理由コード

(凡例)

i：入力用

i/o：入出力用

o：出力用

—：該当しません。

注※1

MQPUT 命令で指定してください。

注※2

MQPUT1 命令で指定してください。

注※3

Version フィールドが、MQ\*\_VERSION\_2 の場合だけ存在するフィールドです。

注※4

Version フィールドが、MQ\*\_VERSION\_3 の場合だけ存在するフィールドです。

注※5

MQPMO\_SET\_IDENTITY\_CONTEXT または MQPMO\_SET\_ALL\_CONTEXT を指定すると、入出力用のフィールドとなります。

注※6

MQPMO\_SET\_ALL\_CONTEXT を指定すると、入出力用のフィールドとなります。

MQPUT 命令および MQPUT1 命令の引数に指定できるオプションには、報告オプション (MsgDesc 引数の Report フィールド) と登録オプション (PutMsgOpts 引数の Options フィールド) の 2 種類があります。

それぞれのフィールドで指定できる報告オプションと登録オプションの一覧を次の各表に示します。

表 A-11 報告オプション一覧 (MQPUT 命令および MQPUT1 命令の MsgDesc 引数の Report フィールド)

オプション	指定方法	オープン時に必要なオプション
MQRO_EXCEPTION	必要時にどれか一つだけ指定	MQOO_OUTPUT* <sup>1</sup>
MQRO_EXCEPTION_WITH_DATA		
MQRO_EXCEPTION_WITH_FULL_DATA		
MQRO_EXPIRATION	必要時にどれか一つだけ指定	
MQRO_EXPIRATION_WITH_DATA		
MQRO_EXPIRATION_WITH_FULL_DATA		
MQRO_COA	必要時にどれか一つだけ指定	
MQRO_COA_WITH_DATA		
MQRO_COA_WITH_FULL_DATA		
MQRO_COD	必要時にどれか一つだけ指定	
MQRO_COD_WITH_DATA		
MQRO_COD_WITH_FULL_DATA		
MQRO_PAN	<ul style="list-style-type: none"> <li>• 必要時にどれか一つだけ指定</li> <li>• 省略可</li> </ul>	
MQRO_NAN		
MQRO_NEW_MSG_ID (省略時解釈値)		
MQRO_PASS_MSG_ID		
MQRO_COPY_MSG_ID_TO_CORRELL_ID (省略時解釈値)		

オプション	指定方法	オープン時に必要なオプション
MQRO_PASS_CORREL_ID	<ul style="list-style-type: none"> <li>• 必要時にどれか一つだけ指定</li> <li>• 省略可</li> </ul>	MQOO_OUTPUT※1
MQRO_DEAD_LETTER_Q	<ul style="list-style-type: none"> <li>• 必要時にどちらか一つだけ指定</li> <li>• 省略可</li> </ul>	
MQRO_DISCARD_MSG※2		
MQRO_NONE	報告を要求しない場合に指定	

注※1

MQPUT1 命令の場合、キューをオープンしておく必要がないため、この欄は該当しません。

注※2

TP1/Message Queue では、指定しても動作に影響がありません。

表 A-12 登録オプション一覧 (MQPUT 命令および MQPUT1 命令の PutMsgOpts 引数の Options フィールド)

オプション	指定方法	オープン時に必要なオプション	
MQPMO_SYNCPOINT	<ul style="list-style-type: none"> <li>• どちらか一つだけ指定</li> <li>• 省略可※1</li> </ul>	MQOO_OUTPUT※2	
MQPMO_NO_SYNCPOINT			
MQPMO_NEW_MSG_ID	必要時に指定		
MQPMO_NEW_CORREL_ID			
MQPMO_LOGICAL_ORDER			
MQPMO_NO_CONTEXT	<ul style="list-style-type: none"> <li>• どれか一つだけ指定</li> <li>• 省略可</li> </ul>		
MQPMO_DEFAULT_CONTEXT (省略時解釈値)			
MQPMO_PASS_IDENTITY_CONTEXT※3			<ul style="list-style-type: none"> <li>• MQOO_OUTPUT※2</li> <li>• MQOO_PASS_**2</li> </ul>
MQPMO_PASS_ALL_CONTEXT※3			<ul style="list-style-type: none"> <li>• MQOO_OUTPUT※2</li> <li>• MQOO_PASS_ALL_CONTEXT※2</li> </ul>
MQPMO_SET_IDENTITY_CONTEXT			<ul style="list-style-type: none"> <li>• MQOO_OUTPUT※2</li> <li>• MQOO_SET_**2</li> </ul>
MQPMO_SET_ALL_CONTEXT	<ul style="list-style-type: none"> <li>• MQOO_OUTPUT※2</li> <li>• MQOO_SET_ALL_CONTEXT※2</li> </ul>		
MQPMO_ALTERNATE_USER_AUTHORITY	必要時に指定 (MQPUT1 命令の場合)	なし	
MQPMO_FAIL_IF_QUIESCING※4	必要時に指定	MQOO_OUTPUT※2	

オプション	指定方法	オープン時に必要なオプション
MQPMO_NONE	ほかの登録オプションを指定しない場合に指定	MQOO_OUTPUT*2

注※1

省略時解釈値は環境に依存します。TP1/Message Queue では、トランザクションの範囲内から呼び出したかどうかで決まります。ほかの IBM MQ の製品では、省略時解釈値が異なる場合があります。

注※2

MQPUT1 命令の場合、キューをオープンしておく必要がないため、この欄は該当しません。

注※3

MQOO\_INPUT\_\*および MQOO\_SAVE\_ALL\_CONTEXT でオープンした、キューのハンドル（すでにメッセージを取り出したもの）が必要です。

注※4

TP1/Message Queue では、指定しても動作に影響がありません。

## 付録 A.8 MQSET 命令の引数一覧

MQSET 命令で指定できる引数の一覧を次の表に示します。

表 A-13 MQSET 命令の引数一覧

引数	データタイプ	向き	内容
Hconn	MQHCONN	i	コネクションハンドル
Hobj	MQHOBJ	i	オブジェクトハンドル
SelectorCount	MQLONG	i	セレクタ数
Selectors	MQLONG× SelectorCount	i	セレクタ
IntAttrCount	MQLONG	i	整数型属性数
IntAttrs	MQLONG× IntAttrCount	i	整数型属性値
CharAttrLength	MQLONG	i	文字型属性長
CharAttrs	MQCHAR× CharAttrLength	i	文字型属性値
CompCode	MQLONG	o	完了コード
Reason	MQLONG	o	理由コード

(凡例)

i : 入力用

o : 出力用

## 付録 B エラーコード

---

キューマネージャは各命令に対し、完了コードおよび理由コードを返し、命令の実行結果を報告します。

一つの命令から複数のエラーが発生した場合、返されるコードは実行環境によって異なります。したがって、返された完了コードまたは理由コード以外のエラーが発生している場合があるため、注意が必要です。各コードの定数値については、「付録 F TP1/Message Queue の定数値」を参照してください。

### 付録 B.1 完了コード

呼び出し元アプリケーションは、完了コード (CompCode) で命令が正常終了、部分終了、または失敗したかどうか即座に確認できます。次に、完了コードを示します。

#### MQCC\_OK

正常終了

命令は完全に完了し、出力引数はすべて設定されました。Reason 引数の値は MQRC\_NONE になります。

#### MQCC\_WARNING

警告 (部分終了)

命令は部分的に完了しました。CompCode 引数および Reason 引数に加えて幾つかの引数が設定されている場合があります。Reason 引数で部分終了の詳細情報を確認できます。

#### MQCC\_FAILED

命令失敗

命令は完了しませんでした。通常、キューマネージャの状態は変更されません。ただし、何らかの理由によって状態が変更される場合があります。その場合は、理由コードを参照してください。CompCode 引数および Reason 引数の出力引数は設定されました。上記以外の引数は変更されませんが、変更される場合は明確に報告されます。命令失敗の理由には次のようなものがあります。

- アプリケーションプログラムの障害
- アプリケーションプログラムへの外的要因 (アプリケーション権限が取り消されているなど)

Reason 引数で命令失敗の詳細情報を確認できます。

### 付録 B.2 理由コード

理由コード (Reason) では、完了コード (CompCode) より詳細な情報で、命令の実行結果を確認できます。報告する事象がない場合は、MQRC\_NONE が返されます。命令が正常終了した場合は、MQCC\_OK および MQRC\_NONE が返されます。完了コードが MQCC\_WARNING または MQCC\_FAILED の場合、キューマネージャは理由コードを返します。

また、理由コードは次に示すフィールドにも表示されます。

- MQDLH 構造体（デッドレターキューのメッセージに対する）の Reason フィールド
- MQMD 構造体（メッセージ記述子）の Feedback フィールド

次に、理由コードを示します。なお、TP1/Message Queue では返されない理由コードも含まれています。

値 (10 進数)	値 (16 進数)	説明
0	X'00000000'	<b>MQRC_NONE</b> 命令は正常終了しました。完了コード (CompCode) は、MQCC_OK になります。 対処 ありません。
900	X'00000384'	<b>MQRC_APPL_FIRST</b> データ変換出口によって返されるアプリケーション定義の理由コードの最低値です。データ変換出口は MQRC_APPL_FIRST から MQRC_APPL_LAST の範囲の理由コードを返し、この出口が検出した特定条件を示します。 対処 データ変換出口の作成者が定義しているとおり処置します。
999	X'000003E7'	<b>MQRC_APPL_LAST</b> データ変換出口によって返されるアプリケーション定義の理由コードの最高値です。データ変換出口は MQRC_APPL_FIRST から MQRC_APPL_LAST の範囲の理由コードを返し、この出口が検出した特定条件を示します。 対処 データ変換出口の作成者が定義しているとおり処置します。
2001	X'000007D1'	<b>MQRC_ALIAS_BASE_Q_TYPE_ERROR</b> あて先を別名キューとして、MQOPEN 命令または MQPUT1 命令が呼び出されました。しかし、別名キュー定義のベースキューの名称が、ローカルキューではない、またはリモートキューのローカル定義ではないキューと解釈されています。 対処 キューの定義を訂正してください。
2002	X'000007D2'	<b>MQRC_ALREADY_CONNECTED</b> MQCONN 命令が呼び出されましたが、アプリケーションはすでにキューマネージャに接続されています。 対処 ありません。返された Hconn 引数の値は、直前の MQCONN 命令で返された値と同じです。  MQCONN 命令でこの理由コードが返されても、キューマネージャから切り離すために MQDISC 命令を続けて呼び出す必要はありません。アプリケーションがすでに接続された状態でコールされた場合にこの理由コードが返されるとき、対応する MQDISC 命令を呼び出さないでください。MQDISC 命令を呼び出すと、元の MQCONN 命令を呼び出したアプリケーションも切り離されてしまいます。
2003	X'000007D3'	<b>MQRC_BACKED_OUT</b>

値 (10 進数)	値 (16 進数)	説明
2003	X'000007D3'	現在のトランザクションに致命的エラーが発生したか、またはロールバックされました。 対処 キューマネージャに対する前の命令のリターン情報を確認してください。例えば、前の MQPUT 命令が失敗していた可能性があります。
2004	X'000007D4'	<b>MQRC_BUFFER_ERROR</b> Buffer 引数または Buffer 引数のポインタが不正です。または Buffer 引数のポインタが、MQGET 命令で読み出し専用領域、または BufferLength 引数で指定された値より短い領域を指定しています。不正な Buffer 引数のポインタを検出できないことがあります。検出されないときの動作は保証しません。 対処 Buffer 引数を訂正してください。
2005	X'000007D5'	<b>MQRC_BUFFER_LENGTH_ERROR</b> BufferLength 引数または BufferLength 引数のポインタが不正です。不正な BufferLength 引数のポインタを検出できないことがあります。検出されないときの動作は保証しません。また、AuthorityBuffer パラメタが小さ過ぎて、サービスコンポーネントの呼び出し側に返されるデータを格納できない場合にも、MQZ_ENUMERATE_AUTHORITY_DATA インストール可能サービスコンポーネントによって、この理由コードが返されることがあります。 対処 0 以上の値を指定してください。
2006	X'000007D6'	<b>MQRC_CHAR_ATTR_LENGTH_ERROR</b> MQINQ 命令または MQSET 命令で、CharAttrLength 引数に負の値が指定されています。または、MQSET 命令で選択したすべての属性を保持できる長さではありません。この理由コードは、CharAttrLength 引数のポインタが不正な場合にも返されます。不正な CharAttrLength 引数のポインタを検出できないことがあります。検出されないときの動作は保証しません。 対処 選択した属性値がすべて保持される長さを指定してください。
2007	X'000007D7'	<b>MQRC_CHAR_ATTRS_ERROR</b> CharAttrs 引数または CharAttrs 引数のポインタが不正です。または、CharAttrs 引数のポインタが MQINQ 命令で、次に示す領域を指定しています。 <ul style="list-style-type: none"> <li>読み出し専用領域</li> <li>CharAttrLength 引数で指定された値より短い領域</li> </ul> 不正な CharAttrs 引数のポインタを検出できないことがあります。検出されないときの動作は保証しません。 対処 CharAttrs 引数を訂正してください。
2008	X'000007D8'	<b>MQRC_CHAR_ATTRS_TOO_SHORT</b> MQINQ 命令の CharAttrLength 引数の値が足りないため、Selectors 引数で指定された MQCA_*セクタの文字型属性をすべて格納できません。MQINQ 命令は、格納できるだけの文字属性を CharAttrs 引数の文字列に格納して完了します。格納された文字属性列だ

値 (10 進数)	値 (16 進数)	説明
2008	X'000007D8'	<p>けが返されます。属性の全体を格納できる領域がない場合、その属性とあとに続く文字属性は除かれます。属性が格納されない文字列の最後の領域は変化しません。</p> <p>対処</p> <p>必要な情報を設定できる値を指定してください。</p> <p>該当する MQRO_COD_* オプションの指定を解除してください。</p>
2009	X'000007D9'	<p><b>MQRC_CONNECTION_BROKEN</b></p> <p>MQC ゲートウェイサーバとの接続が消失しました。</p> <p>MQGMO_WAIT オプションを指定した MQGET 命令の場合、待ち合わせは取り消されています。接続ハンドルとオブジェクトハンドルは、すべて無効です。</p> <p>また、MQC クライアントでローカルトランザクションおよび XA 連携使用時は、エラーを検知した MQC ゲートウェイサーバが該当するトランザクションに関係したメッセージを自動的にロールバックさせます。</p> <p>対処</p> <p>MQC クライアント側 UAP で MQCONN 命令を呼び出して、接続を再確立してください。</p> <p>ただし、XA 連携を使用している場合は、トランザクション内でコミットされていない変更をロールバックしてください。なお、MQCONN 命令とロールバックの実行は順序が前後しても問題ありません。</p>
2010	X'000007DA'	<p><b>MQRC_DATA_LENGTH_ERROR</b></p> <p>DataLength 引数または DataLength 引数のポインタが不正です。または、DataLength 引数のポインタが読み出し専用領域を指定しています。不正な DataLength 引数のポインタを検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>DataLength 引数を訂正してください。</p>
2011	X'000007DB'	<p><b>MQRC_DYNAMIC_Q_NAME_ERROR</b></p> <p>MQOPEN 命令で、ObjDesc パラメタの ObjectName フィールドにモデルキューが指定されました。しかし、DynamicQName フィールドが次に示す理由のため不正です。</p> <ul style="list-style-type: none"> <li>• キュー名称として不正な文字があります。</li> <li>• アスタリスクが 34 文字目以降（ヌル文字より前）にあります。</li> <li>• アスタリスクがあり、その後ろにヌル文字でも空白でもない文字が続いています。</li> </ul> <p>対処</p> <p>正しい名称を指定してください。</p>
2012	X'000007DC'	<p><b>MQRC_ENVIRONMENT_ERROR</b></p> <p>命令が現在の環境に対して不正です。</p> <p>対処</p> <p>アプリケーションから不正な命令を取り除いてください。</p>
2013	X'000007DD'	<p><b>MQRC_EXPIRY_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージ記述子 (MQMD) の Expiry フィールドに指定された値が不正です。</p>

値 (10 進数)	値 (16 進数)	説明
2013	X'000007DD'	<p>対処</p> <p>0 以上の値または MQEI_UNLIMITED を指定してください。</p>
2014	X'000007DE'	<p><b>MQRC_FEEDBACK_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージ記述子 (MQMD) の Feedback フィールドに指定された値が不正です。指定された値は、MQFB_NONE ではありません。また、システムとアプリケーションのフィードバックコードに対して定義された範囲から外れています。</p> <p>対処</p> <p>次のどれかの値を指定してください。</p> <ul style="list-style-type: none"> <li>・MQFB_NONE</li> <li>・MQFB_SYSTEM_FIRST から MQFB_SYSTEM_LAST までの値</li> <li>・MQFB_APPL_FIRST から MQFB_APPL_LAST までの値</li> </ul>
2016	X'000007E0'	<p><b>MQRC_GET_INHIBITED</b></p> <p>キューまたはこのキューの別名キューの実体に対し、MQGET 命令が禁止されています。これは、次のどれかの方法でキューのオブジェクト属性が「キューの取り出し禁止」に設定されたまま解除されていない場合に発生します。</p> <ul style="list-style-type: none"> <li>・MQSET 命令によるオブジェクトの属性の設定</li> <li>・mqaset コマンドによるオブジェクト属性の設定</li> <li>・mqamkque コマンドによるオブジェクト作成で指定するモデルキューの定義 (mqaqueatl 定義コマンド) による属性の設定</li> </ul> <p>対処</p> <p>取り出し要求を、短い時間禁止するようにシステムが設計されている場合、しばらくしたあと、キューからの取り出しを再実行してください。</p> <p>キュー取り出し禁止をしない運用、またはキュー取り出し禁止を長い時間解除しない運用を前提としてシステムが設計されている場合は、アプリケーションを終了してください。</p>
2017	X'000007E1'	<p><b>MQRC_HANDLE_NOT_AVAILABLE</b></p> <p>MQOPEN 命令または MQPUT1 命令が呼び出されましたが、現在実行中の MQCONN 命令に対応できるオープンハンドルの最大数にすでに達しています。</p> <p>対処</p> <p>アプリケーションが、対応する MQCLOSE 命令なしに MQOPEN 命令を呼び出しているか確認してください。呼び出していなければ、使用するハンドルの数を減らしてください。一つの MQCONN 命令で同時にオープンできるハンドルの最大数は MaxHandles キューマネージャ属性で定義できます。</p>
2018	X'000007E2'	<p><b>MQRC_HCONN_ERROR</b></p> <p>コネクションハンドル (Hconn) が不正です。この理由コードは、ポインタ引数が不正、または MQCONN 命令で読み出し専用領域を指しているときにも返されます。不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>キューマネージャに対して MQCONN 命令が正常終了し、MQDISC 命令がまだ実行されていないことを確認してください。ハンドルが有効範囲の中で使用されているか確認</p>

値 (10 進数)	値 (16 進数)	説明
2018	X'000007E2'	<p>してください。ハンドルの有効範囲については、「2. メッセージキューイング機能の命令」の「MQCONN 命令 - キューマネージャへの接続」を参照してください。</p> <p>このエラーは原因を究明できない場合があります。不正なハンドルを受け渡した結果は定義されていません。</p>
2019	X'000007E3'	<p><b>MQRC_HOBJ_ERROR</b></p> <p>オブジェクトハンドル (Hobj) が不正です。この理由コードは、ポインタ引数が不正、または MQOPEN 命令で読み出し専用領域を指しているときにも返されます。不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>このオブジェクトに対して、MQOPEN 命令が正常終了し、MQCLOSE 命令がまだ実行されていないことを確認してください。MQGET 命令と MQPUT 命令に対して、ハンドルがキューオブジェクトを表していることも確認してください。ハンドルが有効範囲の中で使用されているか確認してください。ハンドルの有効範囲については、「2. メッセージキューイング機能の命令」の「MQOPEN 命令 - オブジェクトのオープン」を参照してください。</p>
2020	X'000007E4'	<p><b>MQRC_INHIBIT_VALUE_ERROR</b></p> <p>MQSET 命令の、MQIA_INHIBIT_GET または MQIA_INHIBIT_PUT 属性で指定された値が不正です。</p> <p>対処</p> <p>正しい値を指定してください。</p>
2021	X'000007E5'	<p><b>MQRC_INT_ATTR_COUNT_ERROR</b></p> <p>MQINQ 命令または MQSET 命令の IntAttrCount 引数が負です。または、MQSET 命令の IntAttrCount 引数に、Selectors 引数に指定された整数属性 (MQIA_*) より小さい値が指定されています。ポインタ引数の不正でもこのエラーコードが返されます。不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>選択した整数属性より大きな値を指定してください。</p>
2022	X'000007E6'	<p><b>MQRC_INT_ATTR_COUNT_TOO_SMALL</b></p> <p>MQINQ 命令の IntAttrCount 引数に、Selectors 引数で指定された整数属性セレクタ (MQIA_*) より小さい値が指定されています。MQINQ 命令は、格納できるだけの整数属性を IntAttrs 配列に格納し、MQCC_WARNING で完了します。</p> <p>対処</p> <p>値の一部だけを使用しなければ、大きな値を指定してください。</p>
2023	X'000007E7'	<p><b>MQRC_INT_ATTRS_ARRAY_ERROR</b></p> <p>MQINQ 命令または MQSET 命令の IntAttrs 引数または IntAttrs 引数のポインタが不正です。MQINQ 命令の場合、IntAttrs 引数のポインタが読み出し専用領域、または IntAttrCount 引数で指定された値より短い領域を指している場合もあります。不正な IntAttrs 引数のポインタを検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>引数を訂正してください。</p>

値 (10 進数)	値 (16 進数)	説明
2024	X'000007E8'	<p><b>MQRC_SYNCPOINT_LIMIT_REACHED</b></p> <p>トランザクションでコミットされていないメッセージ数が操作できる最大数を超えました。MQGET 命令、MQPUT 命令、または MQPUT1 命令が失敗しました。トランザクションのコミットされていないメッセージ数は、トランザクションの開始から次に示すメッセージ数の合計です。</p> <ul style="list-style-type: none"> <li>• MQPMO_SYNCPOINT を指定して登録したメッセージ数</li> <li>• MQGMO_SYNCPOINT を指定して取り出したメッセージ数</li> </ul> <p>対処</p> <p>アプリケーションがループしていないか確認してください。ループしていなければ、操作するメッセージ数を減らしてください。</p>
2025	X'000007E9'	<p><b>MQRC_MAX_CONNS_LIMIT_REACHED</b></p> <p>コンカレントコネクション数が最大値を超えたため、MQCONN 命令が受け付けられません。</p> <p>対処</p> <p>インストールパラメタ値のサイズを大きくする、またはコンカレントコネクション数を削減してください。</p>
2026	X'000007EA'	<p><b>MQRC_MD_ERROR</b></p> <p>MQMD 構造体が次のどれかの理由で不正です。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが MQMD_STRUC_ID ではありません。</li> <li>• Version フィールドに無効な値またはサポートされていない値が指定されています。</li> <li>• ポインタ引数が不正です (不正なポインタ引数を検出できないことがあります。検出できないときの動作は保証しません)。</li> <li>• 命令が成功しても、変更された構造体をキューマネージャがアプリケーションストレージにコピーできません。例えば、ポインタが読み取り専用ストレージを指している場合に発生します。</li> </ul> <p>対処</p> <p>メッセージ記述子の定義を訂正してください。また、入力フィールドが正しく設定されているか確認してください。</p>
2027	X'000007EB'	<p><b>MQRC_MISSING_REPLY_TO_Q</b></p> <p>MQPUT 命令および MQPUT1 命令で、メッセージ識別子 (MQMD 構造体) 内の ReplyToQ フィールドは空白です。しかし、次に示すうちの一つまたは両方の要求があります。</p> <ul style="list-style-type: none"> <li>• 応答が要求されました。このとき、メッセージ識別子の MsgType フィールドに MQMT_REQUEST が指定されています。</li> <li>• メッセージ識別子の Report フィールドに報告メッセージが要求されました。</li> </ul> <p>対処</p> <p>応答または報告メッセージを送信するキューの名称を指定してください。</p>
2029	X'000007ED'	<p><b>MQRC_MSG_TYPE_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージ記述子 (MQMD) の MsgType フィールドに指定した値が不正です。</p>

値 (10 進数)	値 (16 進数)	説明
2029	X'000007ED'	対処 正しい値を指定してください。
2030	X'000007EE'	<b>MQRC_MSG_TOO_BIG_FOR_Q</b> MQPUT 命令または MQPUT1 命令に、MaxMsgLength 属性で定義された最大値より大きいメッセージを登録しようとしていました。MaxMsgLength 属性については、「3. オブジェクトの属性」の「キューの属性」を参照してください。この理由コードは、報告メッセージのメッセージ記述子中の Feedback フィールドにも返されることがあります。これは、メッセージチャネルエージェントがメッセージをリモートキューに登録したときに、エラーが発生したことを示します。 対処 BufferLength 引数の指定を確認してください。指定が正しければ、メッセージを分割する、または MaxMsgLength 属性の値を大きくしてください。
2031	X'000007EF'	<b>MQRC_MSG_TOO_BIG_FOR_Q_MGR</b> リモートキューに登録されたメッセージが、リモートキューマネージャに割り当てられた最大メッセージ長を超えています。このエラーはあて先キューマネージャ、または仲介するキューマネージャで発生している場合もあります。この理由コードは、元のメッセージの送信者が報告メッセージを要求した場合に、報告メッセージのメッセージ記述子の Feedback フィールドに返されます。Feedback フィールドの詳細については、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」で Feedback フィールドの説明を参照してください。この理由コードは、ローカルキューマネージャの値より小さい MaxMsgLength キューマネージャ属性を持つリモートキューマネージャへメッセージを送る場合にだけ返されます。 また、メッセージが渡されるチャネルで、キューマネージャにサポートされた値より小さい値を最大メッセージ長に設定している場合に、メッセージ長がこの値より大きいときにも返されることがあります。 対処 BufferLength 引数の指定を確認してください。正しければ、そのメッセージを小さい単位に分割してください。また、チャネルの定義を確認してください。
2033	X'000007F1'	<b>MQRC_NO_MSG_AVAILABLE</b> MQGET 命令が呼び出されましたが、メッセージ記述子で指定された選択基準を満たすメッセージがキューにありません。このエラーは次に示すどちらかが原因です。 <ul style="list-style-type: none"> <li>• MQGMO_WAIT が指定されていません。</li> <li>• MQGMO 構造体の WaitInterval フィールドで指定された時間が経過しています。</li> </ul> この理由コードは、カーソルがキューの終端に到着したときの検索指定の MQGET 命令でも返されます。 対処 検索指定の MQGET 命令を発行した場合は、対処する必要はありません。 キューにメッセージが登録されたかどうかをチェックするようなシステム設計の場合、メッセージが登録されるまでアプリケーションを継続してください。 キューにメッセージが登録されたことが明かな場合に MQGET 命令を発行するシステム設計のとき、次のことを確認してください。 <ul style="list-style-type: none"> <li>• キューにメッセージが正常に登録されたかどうか。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2033	X'000007F1'	<ul style="list-style-type: none"> <li>MQPUT または MQPUT1 命令に使用されているトランザクションがあれば、正常にコミットしたかどうか。</li> <li>キューにメッセージが正しく登録されていた場合、MsgDesc 引数で MQMD 構造体の MsgId フィールドおよび CorrelId フィールドの値が正しく設定されているかどうか。時間を置いて再度 MQGET 命令を呼び出すか、Wait 時間を長くすることを検討してください。</li> </ul>
2034	X'000007F2'	<p><b>MQRC_NO_MSG_UNDER_CURSOR</b></p> <p>次に示すオプションのどちらかを指定して、MQGET 命令が呼び出されました。</p> <ul style="list-style-type: none"> <li>MQGMO_MSG_UNDER_CURSOR</li> <li>MQGMO_BROWSE_MSG_UNDER_CURSOR</li> </ul> <p>しかし、検索カーソルが取り出せるメッセージを指していません。原因を次に示します。</p> <ul style="list-style-type: none"> <li>検索カーソルが論理的に先頭のメッセージの前を指しています。オープンしてから一度もメッセージを検索していません。</li> <li>検索カーソルで示すメッセージの検索処理が終了後、ほかのアプリケーションによって取り出そうとしたメッセージがキューから削除、または排他状態にされました。</li> <li>検索カーソルで示すメッセージの保持時間が終了しています。</li> </ul> <p>対処</p> <p>メッセージの検索後、複数のサーバがメッセージを取り合うように設計されているアプリケーションの場合は、予期される理由コードです。MQGET 命令を再発行してほかのメッセージを取り出してください。</p> <p>それ以外の場合は、アプリケーションのロジックが正しいかどうかを確認してください。</p> <p>なお、メッセージの検索後、メッセージがキューから削除、または排他状態にされないようにするには、一つ前の検索 MQGET 命令に MQGMO_LOCK オプションを指定するようにアプリケーションを設計してください。</p>
2035	X'000007F3'	<p><b>MQRC_NOT_AUTHORIZED</b></p> <p>実行しようとした操作に対して、実行権限がありません。</p> <ul style="list-style-type: none"> <li>MQCONN 命令で、キューマネージャに接続する権限がありません。</li> <li>MQOPEN 命令または MQPUT1 命令で、指定されたオプションでオブジェクトをオープンする権限がありません。</li> <li>MQCLOSE 命令で、オブジェクトの永続的動的キューを削除する権限がなく指定される Hobj 引数は、キューを生成した MQOPEN 命令で返されるハンドルではありません。</li> </ul> <p>この理由コードは、報告メッセージ記述子中の Feedback フィールドにも返される場合があります。これは、メッセージチャネルエージェントがリモートキューにメッセージを登録したときに、エラーが発生したことを示します。</p> <p>対処</p> <p>キューマネージャまたはオブジェクトが正しく指定されているか、またはこれらに適切な権限があるか確認してください。</p>
2036	X'000007F4'	<p><b>MQRC_NOT_OPEN_FOR_BROWSE</b></p> <p>次に示すオプションのうちどれかを指定して、MQGET 命令が発行されました。</p> <ul style="list-style-type: none"> <li>MQGMO_BROWSE_FIRST</li> <li>MQGMO_BROWSE_NEXT</li> <li>MQGMO_BROWSE_MSG_UNDER_CURSOR</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2036	X'000007F4'	<ul style="list-style-type: none"> <li>• MQGMO_MSG_UNDER_CURSOR</li> </ul> <p>しかし、キューは検索用にオープンされていません。</p> <p>対処</p> <p>キューをオープンするときは、MQOO_BROWSE を指定してください。</p>
2037	X'000007F5'	<p><b>MQRC_NOT_OPEN_FOR_INPUT</b></p> <p>キューからメッセージを取り出すために MQGET 命令が呼び出されましたが、キューは取り出し用にオープンされていません。</p> <p>対処</p> <p>キューをオープンするときは、次に示すオプションのうち一つを指定してください。</p> <ul style="list-style-type: none"> <li>• MQOO_INPUT_SHARED</li> <li>• MQOO_INPUT_EXCLUSIVE</li> <li>• MQOO_INPUT_AS_Q_DEF</li> </ul>
2038	X'000007F6'	<p><b>MQRC_NOT_OPEN_FOR_INQUIRE</b></p> <p>オブジェクト属性を照会するために MQINQ 命令が呼び出されましたが、オブジェクトは照会用にオープンされていません。</p> <p>対処</p> <p>オブジェクトをオープンするときは、MQOO_INQUIRE を指定してください。</p>
2039	X'000007F7'	<p><b>MQRC_NOT_OPEN_FOR_OUTPUT</b></p> <p>キューにメッセージを登録するために MQPUT 命令が呼び出されましたが、キューは出力用にオープンされていません。</p> <p>対処</p> <p>キューをオープンするときは、MQOO_OUTPUT を指定してください。</p>
2040	X'000007F8'	<p><b>MQRC_NOT_OPEN_FOR_SET</b></p> <p>オブジェクトの属性を設定するために MQSET 命令が呼び出されましたが、オブジェクトは属性設定用にオープンされていません。</p> <p>対処</p> <p>オブジェクトをオープンするときは、MQOO_SET を指定してください。</p>
2041	X'000007F9'	<p><b>MQRC_OBJECT_CHANGED</b></p> <p>この命令で使用される Hobj ハンドルが、MQOPEN 命令で返されたあと、このオブジェクトに影響するオブジェクト定義が変更されました。</p> <p>MQPUT 命令または MQPUT1 命令で、オブジェクトハンドルが PutMsgOpts 引数の Context フィールドに指定されている場合、この理由コードは返されません。</p> <p>対処</p> <p>MQCLOSE 命令を呼び出してオブジェクトを終了したあと、オブジェクトを再度オープンしてください。なお、オブジェクト定義で変更された内容を確認したい場合は、オブジェクトをオープンしたあとに、MQINQ 命令を呼び出して、オブジェクトの新しい属性値を取得できます。</p>
2042	X'000007FA'	<p><b>MQRC_OBJECT_IN_USE</b></p> <p>MQOPEN 命令が呼び出されましたが、オブジェクトが Options 引数で指定されたものと矛盾するオプションです。また、それを呼び出したアプリケーションまたは別のアプリケー</p>

値 (10 進数)	値 (16 進数)	説明
2042	X'000007FA'	<p>ションによってすでにオープンされています。オブジェクトがすでに排他アクセスでオープンされていて、要求が共用入力の場合にこのエラーが発生します。また、オブジェクトがどんなタイプであっても、すでに入力用にオープンされていて、要求が排他アクセスのときにも発生します。</p> <p>対処</p> <p>対象のキューを排他アクセスで使用するシステムでは、先行のアプリケーションが MQCLOSE 命令を発行したあとで、このアプリケーションが MQOPEN 命令を再発行することで成功します。アプリケーションが待機、再試行、または別の処理を実行するかどうかをシステム設計で設定してください。</p> <p>対象のキューを排他アクセスで使わないシステムでは、引数不正が考えられるので、アプリケーションを見直してください。</p>
2043	X'000007FB'	<p><b>MQRC_OBJECT_TYPE_ERROR</b></p> <p>オブジェクトタイプが不正です。</p> <p>MQOPEN 命令または MQPUT1 命令で、MQOD 構造体の ObjectType フィールドに指定された値が不正です。MQPUT1 命令の場合、オブジェクトタイプは MQOT_Q です。</p> <p>対処</p> <p>正しいオブジェクトタイプを指定してください。</p>
2044	X'000007FC'	<p><b>MQRC_OD_ERROR</b></p> <p>MQOPEN 命令または MQPUT1 命令で、MQOD 構造体が次のどれかの理由で不正です。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが MQOD_STRUC_ID ではありません。</li> <li>• Version フィールドに無効な値、またはサポートされていない値が指定されています。</li> <li>• ポインタ引数が不正です。不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</li> <li>• 命令は成功しましたが、変更された構造体がアプリケーションの記憶領域にコピーされません。このエラーはポインタ引数が読み出し専用領域を指定している場合にも発生することがあります。</li> </ul> <p>対処</p> <p>オブジェクト記述子の定義を訂正してください。入力フィールドが正しく設定されているか確認してください。</p>
2045	X'000007FD'	<p><b>MQRC_OPTION_NOT_VALID_FOR_TYPE</b></p> <p>MQOPEN 命令または MQCLOSE 命令で、オープンまたはクローズされるオブジェクトやキューのタイプに対し、不正なオプションが指定されています。</p> <p>MQOPEN 命令の場合、次に示すオプションの両方が指定されています。</p> <ul style="list-style-type: none"> <li>• MQOT_PROCESS オブジェクトに対する MQOO_OUTPUT など、オブジェクトタイプに指定できないオプション</li> <li>• ローカル定義を持たないリモートキューに対する MQOO_INQUIRE など、キュータイプとしてサポートされていないオプション</li> </ul> <p>MQOPEN 命令の場合、キューマネージャの別名を指定するために、オブジェクト記述子の ObjectQMgrName にリモートキューのローカル定義名が指定されています。また、定義の RemoteQMgrName 属性に指定されたキューがローカルキューマネージャの名称で、次に示すオプションが含まれる場合も、この理由コードが返されます。</p> <p><b>MQOO_INPUT_AS_Q_DEF</b></p> <p><b>MQOO_INPUT_SHARED</b></p>

値 (10 進数)	値 (16 進数)	説明
2045	X'000007FD'	<p><b>MQOO_INPUT_EXCLUSIVE</b>  <b>MQOO_BROWSE</b>  <b>MQOO_INQUIRE</b>  <b>MQOO_SET</b></p> <p>MQOO_INQUIRE を指定した MQOPEN 命令の場合、次のどちらかの理由でこの理由コードが返されます。</p> <ul style="list-style-type: none"> <li>• オブジェクト記述子の ObjectQMgrName フィールドに、空白、ローカルキューマネージャ名、またはローカルに存在するキューの定義名以外の名称が指定されました。</li> <li>• オブジェクト記述子の ObjectQMgrName フィールドに、キューマネージャの別名定義が指定された場合に、定義内の XmitQName 属性が空白で、あて先キューマネージャと同じ名前 で定義されているキューが存在しません。</li> </ul> <p>MQCLOSE 命令の場合、次に示すどちらかのオプションが指定されましたが、キューが動的キューではありません。</p> <p><b>MQCO_DELETE</b>  <b>MQCO_DELETE_PURGE</b></p> <p>対処</p> <p>正しいオプションを指定してください。MQOPEN 命令のオプションについては、「2. メッセージキューイング機能の命令」の「<b>MQOPEN 命令 - オブジェクトのオープン</b>」を参照してください。MQCLOSE 命令のオプションについては「2. メッセージキューイング機能の命令」の「<b>MQCLOSE 命令 - オブジェクトのクローズ</b>」を参照してください。MQCLOSE 命令の場合、オプションを訂正する、またはキューを作成するために使用したモデルキューの定義タイプを変更してください。</p>
2046	X'000007FE'	<p><b>MQRC_OPTIONS_ERROR</b></p> <p>Options 引数またはフィールドのオプションが不正です。または、不正な組み合わせで指定されています。</p> <ul style="list-style-type: none"> <li>• MQOPEN 命令、MQCLOSE 命令、および MQXCNVC 命令の Options 引数は、各命令で指定する引数です。この理由コードはポインタ引数が不正な場合にも返されます。不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</li> <li>• 次に示す命令では、Options 引数は関連するオプション構造体 (MQGMO 構造体または MQPMO 構造体) のフィールドです。 <ul style="list-style-type: none"> <li>– MQGET 命令</li> <li>– MQPUT 命令</li> <li>– MQPUT1 命令</li> </ul> </li> </ul> <p>対処</p> <p>正しいオプションを指定してください。不正な組み合わせのオプションは、Options パラメタまたはフィールドで確認できます。個々のオプションを一緒に加算して複数オプションが設定されている場合、同じオプションが複数回加算されていないか確認してください。</p>
2047	X'000007FF'	<p><b>MQRC_PERSISTENCE_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージ記述子 (MQMD 構造体) の Persistence フィールドに指定された値が不正です。</p>

値 (10 進数)	値 (16 進数)	説明
2047	X'000007FF'	<p>対処</p> <p>次に示すオプションのうちどれかを指定してください。</p> <ul style="list-style-type: none"> <li>• MQPER_PERSISTENT</li> <li>• MQPER_NOT_PERSISTENT</li> <li>• MQPER_PERSISTENCE_AS_Q_DEF</li> </ul>
2048	X'00000800'	<p><b>MQRC_PERSISTENT_NOT_ALLOWED</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージ記述子 (MQMD) の Persistence フィールドに MQPER_PERSISTENT が指定されていますが、登録されるメッセージのキューが一時的動的キューであり、永続メッセージがサポートされていません。永続メッセージは一時的動的キューに登録できません。</p> <p>この理由コードは、報告メッセージのメッセージ記述子の Feedback フィールドにも返されることがあります。これは、メッセージチャネルエージェントがメッセージをリモートキューに登録したときにエラーが発生したことを示します。</p> <p>対処</p> <p>一時的動的キューにメッセージを登録する場合、MQPER_NOT_PERSISTENT を指定してください。永続性メッセージの場合、一時動的キューではなく永続動的キューまたは定義済みキューを使用してください。共有キューの場合、キューが回復可能であるか確認してください。</p> <p>アプリケーションサーバでは、元の要求メッセージ (MQMT_REQUEST) と同じ永続性の応答メッセージ (MQMT_REPLY) を送信することが推奨されています。要求メッセージが永続性の場合、メッセージ記述子 (MQMD) の ReplyToQ フィールドに指定される応答キューは一時的動的キューにできません。この場合は、応答キューとして永続動的キューまたは定義済みキューを使用してください。</p>
2049	X'00000801'	<p><b>MQRC_PRIORITY_EXCEEDS_MAXIMUM</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージ記述子 (MQMD) の Priority フィールドの値がローカルキューマネージャでサポートされた最大メッセージ優先度を超えています。メッセージはキューマネージャに受け付けられましたが、キューマネージャの最大メッセージ優先度でキューに登録されました。メッセージ記述子の Priority フィールドはメッセージを登録したアプリケーションで指定された値のままです。最大メッセージ優先度については、「3. オブジェクトの属性」の「キューマネージャの属性」で MaxPriority の説明を参照してください。</p> <p>対処</p> <p>メッセージを登録したアプリケーションがこの理由コードに対する処理を必要としない場合、対処する必要はありません。</p>
2050	X'00000802'	<p><b>MQRC_PRIORITY_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージ記述子 (MQMD) の Priority フィールドが不正です。キューマネージャによってサポートされている最大優先度は、キューマネージャの MaxPriority 属性の指定値です。</p> <p>対処</p> <p>0 から MaxPriority 属性の範囲で値を指定してください。または MQPRI_PRIORITY_AS_Q_DEF を指定してください。</p>
2051	X'00000803'	<p><b>MQRC_PUT_INHIBITED</b></p>

値 (10 進数)	値 (16 進数)	説明
2051	X'00000803'	<p>MQPUT 命令または MQPUT1 命令が、現在キューまたはこのキューの別名キューの実体に対し禁止されています。</p> <p>この理由コードは報告メッセージのメッセージ記述子の Feedback フィールドにも返されることがあります。これは、メッセージチャネルエージェントがメッセージをリモートキューに登録したときに、エラーが発生したことを示します。</p> <p>対処</p> <p>システム設計で短い期間の登録要求禁止が許されている場合は、登録を再実行してください。</p> <p>キュー取り出し禁止をしない運用、またはキュー取り出し禁止を長い時間解除しない運用を前提としてシステムが設計されている場合は、アプリケーションを終了してください。</p>
2052	X'00000804'	<p><b>MQRC_Q_DELETED</b></p> <p>命令で指定された Hobj キューハンドルは、キューがオープンされてから削除された動的キューを参照しています。</p> <p>対処</p> <p>MQCLOSE 命令を呼び出して、システムにハンドルと対応するリソースを返してください。この場合の MQCLOSE 命令は成功します。エラーが発生したアプリケーション設計を確認してください。</p>
2053	X'00000805'	<p><b>MQRC_Q_FULL</b></p> <p>次のどれかの理由で失敗しました。</p> <ul style="list-style-type: none"> <li>MQPUT 命令または MQPUT1 命令で、キューに格納できるメッセージの最大数を超えました。</li> <li>報告メッセージのメッセージ記述子の Feedback フィールドに、理由コードが返される場合は、メッセージチャネルエージェントがメッセージをリモートキューに登録したときに、エラーが発生したことを示します。</li> </ul> <p>キューに格納できるメッセージの最大数については、「3. オブジェクトの属性」の「キューの属性」で MaxQDepth の説明を参照してください。</p> <p>対処</p> <p>しばらくしたあと、登録を再実行してください。</p> <p>このエラーが頻繁に発生する場合、対象のキューを処理するアプリケーションを増やすことで改善できる可能性があります。</p> <p>また、メッセージ登録で待ち時間を許容できない場合、システム設計でキューのメッセージ最大数を増やしてください。</p>
2055	X'00000807'	<p><b>MQRC_Q_NOT_EMPTY</b></p> <p>永続動的キューに MQCLOSE 命令が発行されましたが、キューが空でないか使用中であるために、命令は失敗しました。次に示すどちらかに該当します。</p> <ul style="list-style-type: none"> <li>MQCO_DELETE が指定されましたが、キューにまだメッセージがあります。</li> <li>MQCO_DELETE または MQCO_DELETE_PURGE が指定されましたが、キューに対してコミットされていない取り出しまたは登録命令があります。</li> </ul> <p>対処</p> <p>キューにメッセージがある原因を調べてください。キューにメッセージが一つ以上あっても、キューの CurrentQDepth 属性は 0 の場合があります。これは、トランザクションで取り出されたメッセージが、コミットされていない場合に発生します。破棄しても</p>

値 (10 進数)	値 (16 進数)	説明
2055	X'00000807'	よいメッセージであれば、MQCO_DELETE_PURGE を指定して MQCLOSE 命令を呼び出したあと、MQCLOSE 命令を再実行してください。
2056	X'00000808'	<p><b>MQRC_Q_SPACE_NOT_AVAILABLE</b></p> <p>MQPUT 命令または MQPUT1 命令が呼び出されましたが、キューとして使用できる空き領域がディスクまたはほかの記憶装置にありません。</p> <p>この理由コードは、報告メッセージのメッセージ記述子の Feedback フィールドにも返されることがあります。これは、メッセージチャネルエージェントがメッセージをリモートキューに登録したときにエラーが発生したことを示します。</p> <p>対処</p> <p>システム設計で許容できる（メッセージ数とメッセージサイズの組み合わせ）以上のメッセージが登録されています。</p> <p>ほかのアプリケーションがメッセージを取り出したあと、MQRC_Q_SPACE_NOT_AVAILABLE を返却された MQI 命令を再発行することで成功する可能性があります。</p> <p>このエラーが頻繁に発生する場合、キュー用のディスク容量を大きくしてください。または、MQA サービス定義の 1 メッセージの最大レコード数 (mqa_max_msg_recnum) を大きくしてください。</p>
2057	X'00000809'	<p><b>MQRC_Q_TYPE_ERROR</b></p> <p>次に原因を示します。</p> <ul style="list-style-type: none"> <li>MQOPEN 命令で、オブジェクト記述子の ObjectQMgrName フィールドに、キューマネージャ別名を指定するためのリモートキューのローカル定義名が指定されています。また、RemoteQMgrName フィールドがローカルキューマネージャの名称になっています。しかし、ObjectName フィールドはローカルキューマネージャのモデルキューの名称を指定しています。リモートキューおよびローカルキューについては、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。</li> <li>MQPUT1 命令で、オブジェクト記述子 (MQOD) にモデルキューの名称を指定しています。</li> </ul> <p>対処</p> <p>キュータイプを訂正してください。</p>
2058	X'0000080A'	<p><b>MQRC_Q_MGR_NAME_ERROR</b></p> <p>MQCONN 命令で Name 引数に指定された値が不正か、または未知の値です。ただし、不正な Name 引数を検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <ul style="list-style-type: none"> <li>影響がなければ、Name 引数に空白を指定してください。または、使用中の名称が正しいか確認してください。</li> <li>Name 引数ポインタが正しいか確認してください。</li> </ul>
2059	X'0000080B'	<p><b>MQRC_Q_MGR_NOT_AVAILABLE</b></p> <p>MQCONN 命令の Name 引数で指定されたキューマネージャは、この時点では接続に使用できません。</p> <p>対処</p> <p>dc_rpc_open が実行されているかどうかを確認してください。</p>

値 (10 進数)	値 (16 進数)	説明
2061	X'0000080D'	<p><b>MQRC_REPORT_OPTIONS_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令が呼び出されましたが、メッセージ記述子 (MQMD) の Report フィールドに、ローカルキューマネージャで認識されないオプションがあります。このオプションは、メッセージのあて先によって異なります。</p> <p>対処</p> <p>対処方法を次に示します。</p> <ol style="list-style-type: none"> <li>1. メッセージ記述子の Report フィールドが、メッセージ記述子の宣言時に初期化されているかどうかを確認してください。または、MQPUT 命令または MQPUT1 命令の前に値を割り当てられているかどうかを確認してください。報告オプションが必要でない場合は、MQRO_NONE を指定してください。</li> <li>2. 指定された報告オプションが、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」の報告オプションに記載されているか確認してください。記載されていない報告オプションは削除してください。</li> <li>3. 個々の報告オプションと一緒に加算して複数の報告オプションを設定している場合、同じ報告オプションが複数回加算されていないか確認してください。</li> <li>4. 矛盾する報告オプションが指定されていないか確認してください。例えば、Report フィールドに、MQRO_EXCEPTION および MQRO_EXCEPTION_WITH_DATA の両方が加算されているとエラーになります。どちらか一つだけが指定できます。</li> </ol>
2062	X'0000080E'	<p><b>MQRC_SECOND_MARK_NOT_ALLOWED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2063	X'0000080F'	<p><b>MQRC_SECURITY_ERROR</b></p> <p>次に示すどれかの命令が発行されましたが、セキュリティエラーが起こったため失敗しました。</p> <ul style="list-style-type: none"> <li>• MQCLOSE 命令</li> <li>• MQCONN 命令</li> <li>• MQOPEN 命令</li> <li>• MQPUT1 命令</li> </ul> <p>対処</p> <p>セキュリティマネージャが示すエラーをシステム管理者に連絡してください。</p>
2065	X'00000811'	<p><b>MQRC_SELECTOR_COUNT_ERROR</b></p> <p>SelectorCount 引数に不正な値が指定されています。ポインタ引数が不正な場合にもこの理由コードが返されます。不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>0~256 の範囲で指定してください。</p>
2066	X'00000812'	<p><b>MQRC_SELECTOR_LIMIT_EXCEEDED</b></p> <p>MQINQ 命令または MQSET 命令で、SelectorCount 引数に最大値 (256) より大きい値が指定されています。</p> <p>対処</p> <p>命令で指定されたセクタ数を減らしてください。0~256 の範囲で指定してください。</p>

値 (10 進数)	値 (16 進数)	説明
2067	X'00000813'	<p><b>MQRC_SELECTOR_ERROR</b></p> <p>MQINQ 命令または MQSET 命令が発行されましたが、次のどれかの理由で、Selectors 配列に含まれるセレクタが無効でした。</p> <ul style="list-style-type: none"> <li>• セレクタがサポートされていない、または範囲外です。</li> <li>• セレクタが属性を照会または設定しようとしているオブジェクトのタイプに適用されません。</li> <li>• セレクタは、設定できない属性のものであります。</li> </ul> <p>対処</p> <p>セレクタに指定した値が、Hobj 引数で示すオブジェクトタイプに合っているかどうかを確認してください。MQSET 命令の場合、セレクタが設定できる整数属性になっているかどうかを確認してください。</p>
2068	X'00000814'	<p><b>MQRC_SELECTOR_NOT_FOR_TYPE</b></p> <p>MQINQ 命令で、Selectors 配列のセレクタが属性の参照されているキュータイプに適用できません。命令は MQCC_WARNING で完了し、適用できないセレクタの属性値に次に示すものが設定されます。</p> <ul style="list-style-type: none"> <li>• 整数属性の場合、IntAttrs に対応する要素が、MQIAV_NOT_APPLICABLE に設定されます。</li> <li>• 文字属性の場合、CharAttrs 文字列の適切な部分がすべてアスタリスクで設定されます。</li> </ul> <p>対処</p> <p>指定されているセレクタが、意図したセレクタであるかどうかを確認してください。</p>
2069	X'00000815'	<p><b>MQRC_SIGNAL_OUTSTANDING</b></p> <p>MQGET 命令が、MQGMO_SET_SIGNAL または MQGMO_WAIT を指定して呼び出されました。しかし、すでにオブジェクトハンドルに対して未解決のシグナルがあります。</p> <p>対処</p> <p>アプリケーションロジックを確認してください。同じキューに対して未解決のシグナルがある場合、シグナルまたはウェイトの設定が必要なときは、異なるオブジェクトハンドルを使用してください。</p>
2070	X'00000816'	<p><b>MQRC_SIGNAL_REQUEST_ACCEPTED</b></p> <p>GetMsgOpts 引数に MQGMO_SET_SIGNAL を指定して MQGET 命令が呼び出されましたが、利用できるメッセージがありません。命令はすぐに返されます。アプリケーションはシグナルが渡されるまで待つことができます。</p> <p>対処</p> <p>シグナルを待ってください。シグナルが到着したとき、シグナルを確認してメッセージが利用できることを確認してください。利用できれば、MQGET 命令を再度呼び出してください。</p>
2071	X'00000817'	<p><b>MQRC_STORAGE_NOT_AVAILABLE</b></p> <p>キューマネージャに利用できる記憶領域が少ないため、命令が完了されません。</p> <p>対処</p> <p>別のアプリケーションが終了すると、再発行で命令が成功する場合があります。このエラーが意図せず発生する場合、システムのメモリ設計が正しくないおそれがあります。次のことを確認してください。</p>

値 (10 進数)	値 (16 進数)	説明
2071	X'00000817'	<ul style="list-style-type: none"> <li>・実行中のアプリケーションが正しく動作しているかどうかを確認してください。正しく動作していれば、利用できる記憶領域を増やしてください。</li> <li>・OpenTPI のロックサービス定義に指定された MQA サーバの最大同時排他要求数を確認してください。</li> <li>・キューファイルグループの入出力バッファの値を見直してください。</li> </ul> <p>問題を解決したあと、アプリケーションを再実行してください。</p>
2072	X'00000818'	<p><b>MQRC_SYNCPOINT_NOT_AVAILABLE</b></p> <p>次に示す命令に各オプションが指定されていますが、ローカルキューマネージャがトランザクションをサポートしていません。</p> <ul style="list-style-type: none"> <li>・MQGMO_SYNCPOINT を指定した MQGET 命令</li> <li>・MQPMO_SYNCPOINT を指定した MQPUT 命令または MQPUT1 命令</li> </ul> <p>原因を次に示します。</p> <ul style="list-style-type: none"> <li>・trn_rm_open_close_scope = process の場合、dc_rpc_open()を発行していない。</li> <li>・trn_rm_open_close_scope = transaction の場合、トランザクション開始状態でない。</li> <li>・mqaquemgr (キューマネージャ定義) の-s オプションに sync を設定していない。</li> <li>・トランザクション制御オブジェクトがアプリケーションにリンクされていない。</li> <li>・64 ビット環境で 32 ビット製品を使用している場合、トランザクション制御オブジェクトを 32 ビットモードで作成していない。</li> </ul> <p>対処</p> <p>原因に対処するか、MQGMO_SYNCPOINT または MQPMO_SYNCPOINT の指定を削除してください。</p>
2075	X'0000081B'	<p><b>MQRC_TRIGGER_CONTROL_ERROR</b></p> <p>MQSET 命令の、MQIA_TRIGGER_CONTROL セレクタに指定された値が不正です。</p> <p>対処</p> <p>MQIA_TRIGGER_CONTROL セレクタの値を訂正してください。 MQIA_TRIGGER_CONTROL セレクタについては、「3. オブジェクトの属性」の「キューの属性」の TriggerControl の説明を参照してください。</p>
2076	X'0000081C'	<p><b>MQRC_TRIGGER_DEPTH_ERROR</b></p> <p>MQSET 命令の、MQIA_TRIGGER_DEPTH セレクタに指定された値が不正です。</p> <p>対処</p> <p>0 より大きい値を指定してください。</p>
2077	X'0000081D'	<p><b>MQRC_TRIGGER_MSG_PRIORITY_ERR</b></p> <p>MQSET 命令の、MQIA_TRIGGER_MSG_PRIORITY セレクタに指定された値が不正です。</p> <p>対処</p> <p>0 から MaxPriority キューマネージャ属性の範囲で値を指定してください。</p>
2078	X'0000081E'	<p><b>MQRC_TRIGGER_TYPE_ERROR</b></p> <p>MQSET 命令の、MQIA_TRIGGER_TYPE セレクタに指定された値が不正です。</p> <p>対処</p> <p>MQIA_TRIGGER_TYPE セレクタの値を訂正してください。</p>

値 (10 進数)	値 (16 進数)	説明
2079	X'0000081F'	<p><b>MQRC_TRUNCATED_MSG_ACCEPTED</b></p> <p>MQGET 命令で、メッセージ長が大き過ぎたため、用意されたバッファに格納されませんでした。MQGMO_ACCEPT_TRUNCATED_MSG が指定されたため命令は完了しました。メッセージは、トランザクションを考慮してキューから削除されます。または、これが検索操作の場合、検索カーソルがこのメッセージに進められます。DataLength 引数が切り捨て前のメッセージの長さに設定され、Buffer 引数に入るだけのメッセージが格納されます。MQMD 構造体には必要な項目が設定されます。</p> <p>対処</p> <p>アプリケーションで設定された処理のため、対処する必要はありません。</p>
2080	X'00000820'	<p><b>MQRC_TRUNCATED_MSG_FAILED</b></p> <p>MQGET 命令で、メッセージ長が大き過ぎたため、用意されたバッファに格納されませんでした。MQGMO_ACCEPT_TRUNCATED_MSG が指定されなかったため、メッセージはキューから削除されません。これが検索操作の場合、検索カーソルはコール前にあった位置にあります。しかし、MQGMO_BROWSE_FIRST が指定された場合、検索カーソルはキューの最も高い優先度のメッセージの前に論理的に位置づけられます。DataLength フィールドが切り捨て前のメッセージの長さに設定され、Buffer 引数に入るだけのメッセージが格納されます。MQMD 構造体には必要な項目が設定されます。</p> <p>対処</p> <p>アプリケーションがメッセージの全体を必要とする場合、アプリケーションで DataLength と同じ大きさのバッファを用意してから MQGET 命令を再発行してください。</p> <p>アプリケーションがメッセージデータの一部だけで処理できる場合、MQGMO_ACCEPT_TRUNCATED_MSG を指定した MQGET 命令を発行し、BufferLength に指定したサイズだけ取り出すように、アプリケーションの処理を見直してください。</p>
2082	X'00000822'	<p><b>MQRC_UNKNOWN_ALIAS_BASE_Q</b></p> <p>MQOPEN 命令または MQPUT1 命令が別名キューを対象として呼び出されました。しかし、別名キュー属性の BaseQName がキューの名称として認識されません。</p> <p>対処</p> <p>キュー定義を訂正してください。</p>
2085	X'00000825'	<p><b>MQRC_UNKNOWN_OBJECT_NAME</b></p> <p>MQOPEN または MQPUT1 命令が発行されましたが、オブジェクト記述子 (MQOD) の ObjectName および ObjectQMgrName フィールドに指定されたオブジェクトが見つかりません。次のどちらかに該当します。</p> <ul style="list-style-type: none"> <li>ObjectQMgrName フィールドが次に示すどれかです。 <ul style="list-style-type: none"> <li>空白</li> <li>ローカルキューマネージャの名前</li> <li>RemoteQMgrName フィールドに指定されたりリモートキューのローカル定義 (キューマネージャの別名) の名前がローカルキューマネージャの名前です。</li> </ul> </li> </ul> <p>しかし、指定された ObjectName および ObjectType のオブジェクトが、ローカルキューマネージャにありません。</p>

値 (10 進数)	値 (16 進数)	説明
2085	X'00000825'	<ul style="list-style-type: none"> <li>オープンしようとしているオブジェクトがリモートキューマネージャにあるクラスタキューだが、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への定義済み経路がありません。</li> </ul> <p>対処</p> <p>正しいオブジェクト名を指定してください。また、必要に応じて名称の右側が空白で埋まっているか確認してください。正しければ、そのキュー定義を確認してください。クラスタ機能を使用している場合、mqrsup、mqrspp が起動しているかどうか確認してください。</p>
2086	X'00000826'	<p><b>MQRC_UNKNOWN_OBJECT_Q_MGR</b></p> <p>MQOPEN 命令または MQPUT1 命令で、オブジェクト記述子 (MQOD) の ObjectQMgrName フィールドが不正です。キューマネージャの名称を含むオブジェクトの命名規則については、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。この理由コードは、オブジェクト記述子の ObjectType フィールドに MQOT_Q_MGR が指定され、ObjectQMgrName フィールドが空白ではなく、また、指定された名称がローカルキューマネージャ名でない場合にも返されます。</p> <p>対処</p> <p>正しいキューマネージャ名を指定してください。ローカルキューマネージャを指定する場合は、すべて空白の名前またはヌル文字で始まる名前を使用できます。必要であれば、名称の右側が空白で埋まっているまたは、名称がヌル文字で終わっていることを確認してください。クラスタ機能を使用している場合、mqrsup、mqrspp が起動しているかどうか確認してください。</p>
2087	X'00000827'	<p><b>MQRC_UNKNOWN_REMOTE_Q_MGR</b></p> <p>MQOPEN 命令または MQPUT1 命令のキュー名の決定で、エラーが発生しました。原因を次に示します。</p> <ul style="list-style-type: none"> <li>ObjectQMgrName が空白またはローカルキューマネージャの名称です。また、ObjectName が、空白の XmitQName を持つリモートキューローカル定義の名称です。しかし、RemoteQMgrName の名称で定義された転送キューはなく、DefXmitQName キューマネージャ属性が空白です。</li> <li>ObjectQMgrName は、空白の XmitQName を持つキューマネージャ別名定義の名称です。キューマネージャ別名定義は、リモートキューのローカル定義として保持されます。しかし、RemoteQMgrName の名称で定義された転送キューはなく、DefXmitQName キューマネージャ属性が空白です。</li> <li>指定された ObjectQMgrName が次に示す名称に当てはまりません。 <ul style="list-style-type: none"> <li>空白</li> <li>ローカルキューマネージャの名称</li> <li>ローカルキューの名称</li> <li>キューマネージャ別名定義の名称 (RemoteQName が空白のリモートキューローカル定義)</li> </ul> <p>ただし、DefXmitQName キューマネージャ属性は空白です。</p> </li> <li>ObjectQMgrName が空白またはローカルキューマネージャの名称です。また、ObjectName がリモートキューのローカル定義またはローカル定義の別名の名称です。ローカル定義または別名は、RemoteQMgrName が空白またはローカルキューマネージャの名称です。このエラーは XmitQName が空白でなくても発生する場合がありますので注意が必要です。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2087	X'00000827'	<ul style="list-style-type: none"> <li>ObjectQMGrName がリモートキューのローカル定義の名称です。この場合、ローカル定義はキューマネージャ別定義の<b>はず</b>ですが、定義中の RemoteQName が空白ではありません。</li> <li>ObjectQMGrName がモデルキューの名称です。</li> </ul> <p>対処</p> <p>ObjectQMGrName および ObjectName に指定された値を確認してください。正しければ、キュー定義を確認してください。</p>
2090	X'0000082A'	<p><b>MQRC_WAIT_INTERVAL_ERROR</b></p> <p>MQGET 命令で、GetMsgOpts 引数の WaitInterval フィールドに指定された値が不正です。</p> <p>対処</p> <p>0 以上の値を指定してください。無制限に待ち合わせる場合は、MQWI_UNLIMITED を指定してください。</p>
2091	X'0000082B'	<p><b>MQRC_XMIT_Q_TYPE_ERROR</b></p> <p>MQOPEN 命令または MQPUT1 命令で、メッセージがリモートキューマネージャに送信されることになっています。オブジェクト記述子の ObjectName または ObjectQMGrName フィールドにリモートキューのローカル定義名が指定されていますが、定義の XmitQName 属性が次のどちらかに当てはまります。</p> <ul style="list-style-type: none"> <li>XmitQName が空白でなくても、ローカルキューでないキューを指定します。</li> <li>XmitQName が空白であっても、RemoteQMGrName にローカルキューでないキューを指定します。</li> </ul> <p>対処</p> <p>ObjectName および ObjectQMGrName に指定された値を確認してください。正しい場合は、キュー定義を確認してください。名称解決については、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>
2092	X'0000082C'	<p><b>MQRC_XMIT_Q_USAGE_ERROR</b></p> <p>MQOPEN 命令または MQPUT1 命令で、メッセージがリモートキューマネージャに送信されますが、次に示す事象のどちらかが発生しました。</p> <ul style="list-style-type: none"> <li>ObjectQMGrName はローカルキューの名称を指定しますが、キューは MQUS_TRANSMISSION の Usage 属性を持っていません。</li> <li>オブジェクト記述子の ObjectName または ObjectQMGrName フィールドにリモートキューのローカル定義名が指定されています。しかし、定義の XmitQName 属性に次のどちらかが当てはまります。 <ul style="list-style-type: none"> <li>XmitQName が空白でなくても、MQUS_TRANSMISSION の Usage 属性を持たないキューが指定されています。</li> <li>XmitQName が空白であっても、RemoteQMGrName に MQUS_TRANSMISSION の Usage 属性を持たないキューが指定されています。</li> </ul> </li> </ul> <p>対処</p> <p>ObjectName および ObjectQMGrName に指定された値を確認してください。正しい場合は、キュー定義を確認してください。名称解決については、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>
2093	X'0000082D'	<p><b>MQRC_NOT_OPEN_FOR_PASS_ALL</b></p>

値 (10 進数)	値 (16 進数)	説明
2093	X'0000082D'	PutMsgOpts 引数に MQPMO_PASS_ALL_CONTEXT を指定して MQPUT 命令が呼び出されました。しかし、キューは MQOO_PASS_ALL_CONTEXT でオープンされていません。 対処 キューをオープンするときは、MQOO_PASS_ALL_CONTEXT、または、これに相当するオプションを指定してください。
2094	X'0000082E'	<b>MQRC_NOT_OPEN_FOR_PASS_IDENT</b> PutMsgOpts 引数に MQPMO_PASS_IDENTITY_CONTEXT を指定して、MQPUT 命令が呼び出されました。しかし、キューは MQOO_PASS_IDENTITY_CONTEXT でオープンされていません。 対処 キューをオープンするときは、MQOO_PASS_IDENTITY_CONTEXT、または、これに相当するオプションを指定してください。
2095	X'0000082F'	<b>MQRC_NOT_OPEN_FOR_SET_ALL</b> PutMsgOpts 引数に MQPMO_SET_ALL_CONTEXT を指定して MQPUT 命令が呼び出されました。しかし、キューは MQOO_SET_ALL_CONTEXT でオープンされていません。 対処 キューをオープンするときは、MQOO_SET_ALL_CONTEXT を指定してください。
2096	X'00000830'	<b>MQRC_NOT_OPEN_FOR_SET_IDENT</b> PutMsgOpts 引数に MQPMO_SET_IDENTITY_CONTEXT を指定して MQPUT 命令が呼び出されました。しかし、キューは MQOO_SET_IDENTITY_CONTEXT でオープンされていません。 対処 キューをオープンするときは、MQOO_SET_IDENTITY_CONTEXT、または、これに相当するオプションを指定してください。
2097	X'00000831'	<b>MQRC_CONTEXT_HANDLE_ERROR</b> MQPUT 命令または MQPUT1 命令で、次に示すオプションが指定されました。 <ul style="list-style-type: none"> <li>• MQPMO_PASS_IDENTITY_CONTEXT</li> <li>• MQPMO_PASS_ALL_CONTEXT</li> </ul> しかし、PutMsgOpts 引数の Context フィールドに指定されたハンドルが不正なキューハンドルです。または、正しいキューハンドルであっても、MQOO_SAVE_ALL_CONTEXT でオープンしたキューではありません。 対処 参照するキューをオープンするときには、MQOO_SAVE_ALL_CONTEXT を指定してください。
2098	X'00000832'	<b>MQRC_CONTEXT_NOT_AVAILABLE</b> MQPUT 命令または MQPUT1 命令で、次に示すオプションが指定されました。 <ul style="list-style-type: none"> <li>• MQPMO_PASS_IDENTITY_CONTEXT</li> <li>• MQPMO_PASS_ALL_CONTEXT</li> </ul> しかし、PutMsgOpts 引数の Context フィールド領域に指定されたハンドルに、対応するコンテキストがありません。このエラーは次に示す場合に発生します。

値 (10 進数)	値 (16 進数)	説明
2098	X'00000832'	<ul style="list-style-type: none"> <li>参照されたキューハンドルで取り出しに成功したメッセージがありません。</li> <li>最後に成功した MQGET 命令がメッセージの検索です。</li> </ul> <p>最後に取り出されたメッセージに対応するコンテキストがない場合は発生しません。</p> <p>対処</p> <p>参照されたキューハンドルで呼び出された、検索でない取り出しが成功しているか確認してください。</p>
2099	X'00000833'	<p><b>MQRC_SIGNAL1_ERROR</b></p> <p>GetMsgOpts 引数に MQGMO_SET_SIGNAL を指定して MQGET 命令が呼び出されましたが、Signal1 フィールドが不正です。</p> <p>対処</p> <p>Signal1 フィールドの設定を訂正してください。</p>
2100	X'00000834	<p><b>MQRC_OBJECT_ALREADY_EXISTS</b></p> <p>動的キューの作成で MQOPEN 命令が呼び出されましたが、作成しようとした動的キューと同じ名称のキューがすでに存在しています。</p> <p>対処</p> <p>動的キューの名称を確認してください。正しく設定されている場合、異なる名称にするか、必要でなければ存在しているキューを削除してください。または、キューマネージャが名称を作成できるようにしてください。キューマネージャがその名称の一部または全部を作成する場合、MQOPEN 命令を再度呼び出してください。</p>
2101	X'00000835'	<p><b>MQRC_OBJECT_DAMAGED</b></p> <p>命令でアクセスされたオブジェクトが破損して使用できなくなっています。原因を次に示します。</p> <ul style="list-style-type: none"> <li>主記憶装置でオブジェクトの定義が矛盾しています。</li> <li>主記憶装置の定義とディスクの定義が異なります。</li> <li>ディスク上の定義が読み出せません。</li> </ul> <p>オブジェクトを削除できます。ただし、対応するユーザ空間を削除できない場合もあります。</p> <p>対処</p> <p>キューマネージャを停止して再開始する、またはバックアップ装置からキューマネージャデータを回復してください。</p>
2102	X'00000836'	<p><b>MQRC_RESOURCE_PROBLEM</b></p> <p>命令が正常終了するために必要なシステム資源がありません。TP1/Message Queue では、オンライン中に使用できるキューの最大数を物理ファイルごとに規定しています。MQOPEN 命令で動的キュー生成時に使用できるキューの最大数を超えました。</p> <p>対処</p> <p>動的キューを作成するキューファイルがオンラインで使用できるかどうかを確認してください。キューファイルがオンラインで使用できる場合、未使用キューの個数を確認してください。</p>
2103	X'00000837'	<p><b>MQRC_ANOTHER_Q_MGR_CONNECTED</b></p> <p>MQCONN 命令が呼び出されましたが、スレッドまたはプロセスがすでに異なるキューマネージャに接続しています。スレッドまたはプロセスは、一度に一つのキューマネージャとだけしか接続できません。</p>

値 (10 進数)	値 (16 進数)	説明
2103	X'00000837'	<p>対処</p> <p>MQDISC 命令を呼び出し、すでに接続しているキューマネージャからスレッドまたはプロセスを切り離してください。その後、MQCONN 命令を再度呼び出して、新しいキューマネージャに接続してください。</p> <p>現在のキューマネージャから切り離すと、現在オープンしているキューはすべてクローズされます。そのため、MQDISC 命令を呼び出す前に、コミットしていないトランザクションをコミットまたはロールバックしてください。</p>
2104	X'00000838'	<p><b>MQRC_UNKNOWN_REPORT_OPTION</b></p> <p>MQPUT 命令または MQPUT1 命令が呼び出されましたが、メッセージ記述子 (MQMD) の Report フィールドに、ローカルキューマネージャで認識されないオプションがあります。このオプションが受け付けられます。ローカルキューマネージャで識別されないオプションは、メッセージのあて先によって異なります。</p> <p>対処</p> <p>この理由コードが予期される場合、訂正する必要はありません。この理由コードが予期されない場合は、次に示す対処を実行してください。</p> <ol style="list-style-type: none"> <li>1. メッセージ記述子の宣言時に、メッセージ記述子の Report フィールドが初期化されているかどうかを確認してください。または、MQPUT 命令または MQPUT1 命令を呼び出す前に、フィールドに値が割り当てられているかどうかを確認してください。</li> <li>2. 指定された報告オプションが、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」で報告オプションに記載されているかどうかを確認してください。記載されていない報告オプションは削除してください。</li> <li>3. 個々の報告オプションを一緒に加算して複数オプションが設定されている場合、同じ報告オプションが複数回加算されていないか確認してください。</li> <li>4. 矛盾している報告オプションが指定されていないことを確認してください。例えば、MQRO_EXCEPTION および MQRO_EXCEPTION_WITH_DATA の両方を Report フィールドに加算するとエラーになります。これらのうち一つだけが指定できます。</li> </ol>
2105	X'00000839'	<p><b>MQRC_STORAGE_CLASS_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2106	X'0000083A'	<p><b>MQRC_COD_NOT_VALID_FOR_XCF_Q</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2107	X'0000083B'	<p><b>MQRC_XWAIT_CANCELED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2108	X'0000083C'	<p><b>MQRC_XWAIT_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2109	X'0000083D'	<p><b>MQRC_SUPPRESSED_BY_EXIT</b></p> <p>MQCONN 命令または MQDISC 命令以外の呼び出しで、API 間出口が呼び出しを取り消しました。</p>

値 (10 進数)	値 (16 進数)	説明
2109	X'0000083D'	<p>対処</p> <p>出口が強制実行する MQI 呼び出しの規則に従ってください。規則を調べるには、出口の作成者に問い合わせてください。</p>
2110	X'0000083E'	<p><b>MQRC_FORMAT_ERROR</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令が発行されましたが、メッセージ形式のエラーがあるためメッセージ変換が正常に実行されません。このエラーには次に示すものがあります。</p> <ul style="list-style-type: none"> <li>• メッセージのフォーマット名が、MQFMT_NONE です。</li> </ul> <p>メッセージは変換されないで MQGET 命令を呼び出すアプリケーションに返されます。MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値はオブジェクト記述子の各フィールドに設定されます。また、命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>メッセージの登録時に指定されたフォーマット名を確認してください。</p>
2111	X'0000083F'	<p><b>MQRC_SOURCE_CCSID_ERROR</b></p> <p>メッセージデータの変換元のコード化文字セット識別子が不正です。またはサポートされていません。この理由コードは、GetMsgOpts 引数に MQGMO_CONVERT が指定された MQGET 命令で返されることがあります。エラーとなるコード化文字セット識別子は、取り出されたメッセージの CodedCharSetId フィールドです。メッセージデータは変換されないで返されます。MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値はオブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>メッセージの登録時に指定されたコード化文字セット識別子を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。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、「付録 G コード変換対象の文字セット識別子一覧」を参照してください。</p>
2112	X'00000840'	<p><b>MQRC_SOURCE_INTEGER_ENC_ERROR</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令で取り出されたメッセージの Encoding フィールドに、認識されない 2 進整数のマシンコード形式を指定しています。</p> <p>メッセージデータは変換されないで返されます。また、MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値はオブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>メッセージの登録時に指定された、2 進整数のマシンコード形式を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。</p>
2113	X'00000841'	<p><b>MQRC_SOURCE_DECIMAL_ENC_ERROR</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令で、取り出されたメッセージの Encoding フィールドが、認識されない 10 パック形式整数のマシンコード形式を指定しています。</p>

値 (10 進数)	値 (16 進数)	説明
2113	X'00000841'	<p>メッセージデータは変換されずに返されます。また、MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値は、オブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>メッセージの登録時に指定された、10 進パック形式整数のマシンコード形式を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。</p>
2114	X'00000842'	<p><b>MQRC_SOURCE_FLOAT_ENC_ERROR</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令で、取り出したメッセージの Encoding フィールドに、認識されない浮動小数点のマシンコード形式を指定しています。</p> <p>メッセージデータは変換されずに返されます。また、MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値はオブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>メッセージの登録時に指定された、浮動小数点のマシンコード形式を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。</p>
2115	X'00000843'	<p><b>MQRC_TARGET_CCSID_ERROR</b></p> <p>メッセージデータの変換後のコード化文字セット識別子が不正です。またはサポートされていません。この理由コードは、GetMsgOpts 引数に MQGMO_CONVERT が指定された MQGET 命令で返されることがあります。エラーとなるコード化文字セット識別子は、MsgDesc 引数の CodedCharSetId フィールドです。メッセージデータは変換されずに返されます。MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値は、オブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>MQGET 命令の MsgDesc 引数の CodedCharSetId フィールドに指定されたコード化文字セット識別子を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。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、「付録 G コード変換対象の文字セット識別子一覧」を参照してください。</p>
2116	X'00000844'	<p><b>MQRC_TARGET_INTEGER_ENC_ERROR</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令で、MsgDesc 引数の Encoding フィールドに、認識されない 2 進整数のマシンコード形式を指定しています。</p> <p>メッセージデータは変換されずに返されます。また、MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値はオブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>指定された 2 進整数のマシンコード形式を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。</p>
2117	X'00000845'	<p><b>MQRC_TARGET_DECIMAL_ENC_ERROR</b></p>

値 (10 進数)	値 (16 進数)	説明
2117	X'00000845'	<p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令で、MsgDesc 引数の Encoding フィールドに、認識されない 10 進パック形式整数のマシンコード形式を指定しています。</p> <p>メッセージデータは変換されないで返されます。また、MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値はオブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>指定された 10 進パック形式整数のマシンコード形式を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。</p>
2118	X'00000846'	<p><b>MQRC_TARGET_FLOAT_ENC_ERROR</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令で、MsgDesc 引数の Encoding フィールドに、認識されない浮動小数点のマシンコード形式を指定しています。</p> <p>メッセージデータは変換されないで返されます。また、MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドの値はオブジェクト記述子の各フィールドに設定されます。命令は MQCC_WARNING で完了します。</p> <p>対処</p> <p>指定された浮動小数点のマシンコード形式を確認してください。正しい場合は、キューマネージャ変換がサポートされているか確認してください。サポートされていない場合、変換はアプリケーションで実行してください。</p>
2119	X'00000847'	<p><b>MQRC_NOT_CONVERTED</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT を指定した MQGET 命令で、メッセージデータの変換中にエラーが発生しました。メッセージデータは変換されないで返却され、MsgDesc 引数の CodedCharSetId フィールドおよび Encoding フィールドには、未変換のまま返却されるメッセージの値が設定されます。このとき、命令は MQCC_WARNING で完了します。また、データ変換未サポートの値をパラメタに指定した場合にも、この理由コードが返却されることがあります。</p> <p>対処</p> <p>メッセージの登録時に指定した Format、CodedCharSetId、および Encoding の引数で、メッセージデータが正しく記述されているか確認してください。また、これらの値と MQGET 命令の MsgDesc 引数に指定された CodedCharSetId および Encoding フィールドが、キューマネージャ変換でサポートされているか確認してください。指定した変換がサポートされていない場合、アプリケーションで実行してください。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、「付録 G コード変換対象の文字セット識別子一覧」を参照してください。</p>
2120	X'00000848'	<p><b>MQRC_CONVERTED_MSG_TOO_BIG</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT オプションを指定した MQGET 命令で、メッセージデータがデータ変換の間に拡大されました。そのため、アプリケーションが用意したバッファサイズを超えました。しかし、メッセージデータは変換前に切り捨てをしないで、アプリケーションに格納できました。そのため、メッセージはキューからすでに取り出されています。データが消失されないために、MQGET 命令の CompCode 引数に MQCC_WARNING を設定し、メッセージを変換しないで返します。</p>

値 (10 進数)	値 (16 進数)	説明
2120	X'00000848'	<p>対処</p> <p>MQGET 命令を呼び出すアプリケーションの Buffer 引数に、さらに大きいバッファを用意してください。</p>
2121	X'00000849'	<p><b>MQRC_NO_EXTERNAL_PARTICIPANTS</b></p> <p>キューマネージャが調整したトランザクションを開始するための MQBEGIN 命令が呼び出されましたが、参加するリソースマネージャがキューマネージャに登録されていません。その結果、MQ リソースへの変更だけがキューマネージャによってトランザクション内で調整されます。</p> <p>対処</p> <p>アプリケーションがトランザクションに参加するために MQ リソース以外のリソースを必要としない場合、この理由コードは無視できるか、MQBEGIN 命令の呼び出しが削除されることがあります。それ以外の場合には、プログラマに問い合わせ、必要なリソースマネージャがキューマネージャに登録されなかった原因を調査してください。キューマネージャの構成ファイルにエラーがある可能性があります。</p>
2122	X'0000084A'	<p><b>MQRC_PARTICIPANT_NOT_AVAILABLE</b></p> <p>キューマネージャが調整したトランザクションを開始するための MQBEGIN 命令が呼び出されましたが、キューマネージャに登録済みの、参加するリソースマネージャが利用できません。その結果、リソースへの変更は、キューマネージャによってトランザクション内で調整されません。</p> <p>対処</p> <p>アプリケーションがトランザクションに参加するために MQ リソース以外のリソースを必要としない場合、この理由コードは無視できます。それ以外の場合には、プログラマに問い合わせ、必要なリソースマネージャが利用できない原因を調査してください。リソースマネージャは一時的に停止していたことがあるか、キューマネージャの構成ファイルにエラーがある可能性があります。</p>
2123	X'0000084B'	<p><b>MQRC_OUTCOME_MIXED</b></p> <p>キューマネージャが、ほかのリソースマネージャを扱うトランザクションに対してトランザクションコーディネータとして動作していますが、次のどちらかが発生しました。</p> <ul style="list-style-type: none"> <li>トランザクションをコミットするために MQCMIT 命令または MQDISC 命令が呼び出されましたが、参加している一つ以上のリソースマネージャがトランザクションをコミットしないでロールバックしました。その結果、トランザクションの結果が混在しています。</li> <li>トランザクションをロールバックするために MQBACK 命令が呼び出されましたが、参加している一つ以上のリソースマネージャがそのトランザクションをすでにコミットしています。</li> </ul> <p>対処</p> <p>キューマネージャのエラーログで結果の混在に関連するメッセージを調べてください。そのメッセージによって影響があったリソースマネージャを特定できます。影響があったリソースマネージャにある手順を使用してリソースを再同期させてください。</p> <p>この理由コードによってアプリケーションがそれ以降のトランザクションの開始を妨げられることはありません。</p>
2124	X'0000084C'	<p><b>MQRC_OUTCOME_PENDING</b></p>

値 (10 進数)	値 (16 進数)	説明
2124	X'0000084C'	<p>ほかのリソースマネージャを扱うトランザクションに対して、キューマネージャがトランザクションコーディネータとして動作していて、トランザクションをコミットするために MQCMIT 命令または MQDISC 命令が呼び出されましたが、参加している一つ以上のリソースマネージャでトランザクションが正常にコミットされたかどうかを確認されていません。コミット操作はこのあとのどこかの時点で完了しますが、結果が混在する可能性が残っています。</p> <p>対処</p> <p>通常のエラー報告機能を使用して、結果が混在しているかどうかを判断してください。混在している場合には、適切な処置をとってリソースを再同期させてください。</p> <p>この理由コードによってアプリケーションがそれ以降のトランザクションの開始を妨げられることはありません。</p>
2125	X'0000084D'	<p><b>MQRC_BRIDGE_STARTED</b> IMS ブリッジが開始しました。</p> <p>対処</p> <p>ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2126	X'0000084E'	<p><b>MQRC_BRIDGE_STOPPED</b> IMS ブリッジが停止しました。</p> <p>対処</p> <p>ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2127	X'0000084F'	<p><b>MQRC_ADAPTER_STORAGE_SHORTAGE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2128	X'00000850'	<p><b>MQRC_UOW_IN_PROGRESS</b> キューマネージャが調整したトランザクションを開始するために MQBEGIN 命令が呼び出されましたが、指定されたコネクションハンドルにすでにトランザクションがあります。このトランザクションは、前の MQBEGIN 命令の呼び出しによって開始されたグローバルトランザクションであるか、キューマネージャまたは関連するリソースマネージャのローカルなトランザクションの可能性があり、1 コネクションハンドルに同時に二つ以上のトランザクションはできません。</p> <p>対処</p> <p>アプリケーションロジックを検討して、既存のトランザクションがある原因を調査してください。MQBEGIN 命令をアプリケーションの適切な場所に移動してください。</p>
2129	X'00000851'	<p><b>MQRC_ADAPTER_CONN_LOAD_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2130	X'00000852'	<p><b>MQRC_ADAPTER_SERV_LOAD_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2131	X'00000853'	<p><b>MQRC_ADAPTER_DEFS_ERROR</b></p>

値 (10 進数)	値 (16 進数)	説明
2131	X'00000853'	この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2132	X'00000854'	<b>MQRC_ADAPTER_DEFS_LOAD_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2133	X'00000855'	<b>MQRC_ADAPTER_CONV_LOAD_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2134	X'00000856'	<b>MQRC_BO_ERROR</b> MQBEGIN 命令の呼び出しで、BeginOptions 引数の MQBO 構造体が次のどれかの理由で無効です。 <ul style="list-style-type: none"> <li>• StrucId ニーモニックアイキャッチャが MQBO_STRUC_ID ではありません。</li> <li>• Version フィールドが MQBO_VERSION_1 ではありません。</li> <li>• 引数ポインタが無効です。ただし、無効な引数ポインタを必ず検出できるとは限りません。検出できなかった場合には、予期しない結果が発生しています。</li> <li>• キューマネージャは、呼び出しが正常に終了しても、変更された構造体をアプリケーション記憶域へコピーできません。これは、例えば、ポインタが読み取り専用記憶域を指している場合などに起こります。</li> </ul> <b>対処</b> MQBO 構造体の定義を訂正します。必要な入力フィールドが正しく設定されていることを確認してください。
2135	X'00000857'	<b>MQRC_DH_ERROR</b> MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQDH 構造体があります。次のエラーなどがあります。 <ul style="list-style-type: none"> <li>• StrucId フィールドが MQDH_STRUC_ID ではありません。</li> <li>• Version フィールドが MQDH_VERSION_1 ではありません。</li> <li>• StrucLength フィールドに指定されている値が小さ過ぎて、構造体、および MQOR と MQPMR レコードの配列を格納できません。</li> <li>• CodedCharSetId フィールドに、0 または無効な負の値が指定されています。</li> <li>• 命令の BufferLength 引数の値が小さ過ぎて、構造体を格納できません（構造体がメッセージの末尾を超えて拡張されています）。</li> </ul> <b>対処</b> MQDH 構造体のフィールドが正しく設定されているか確認してください。また、アプリケーションで CodedCharSetId フィールドに有効な値が設定されているか確認してください。MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR および MQCCSI_UNDEFINED は、このフィールドに無効です。
2136	X'00000858'	<b>MQRC_MULTIPLE_REASONS</b> 配布リストをオープンしたり、配布リストにメッセージを登録したりするために MQOPEN 命令、MQPUT 命令、または MQPUT1 命令が発行されました。その命令の結果が、リスト内のすべてのあて先について同じではありませんでした。要因として、次の場合が考えられます。 <ul style="list-style-type: none"> <li>• 一部のあて先については命令が成功したが、そのほかのあて先については失敗した場合</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2136	X'00000858'	<p>この場合、完了コードは MQCC_WARNING となります。</p> <ul style="list-style-type: none"> <li>すべてのあて先について命令が失敗したが、失敗の理由はそれぞれ異なる場合</li> </ul> <p>この場合、完了コードは MQCC_FAILED です。</p> <p>対処</p> <p>MQRR 構造体の応答レコードを調べて、命令が失敗したあて先と失敗の理由を判定してください。また、アプリケーションが命令に関するエラーを判定するために十分な応答レコードを提供しているかどうかを確認してください。MQPUT1 命令の場合、MQPMO 構造体でなく、MQOD 構造体を使用して応答レコードを指定しなければなりません。</p>
2137	X'00000859'	<p><b>MQRC_OPEN_FAILED</b></p> <p>メッセージを配布リストに登録するための MQPUT 命令が発行されましたが、そのあて先が MQOPEN 命令によって正常にオープンできなかったために、この理由コードが適用されるあて先にメッセージを送信できませんでした。この理由コードは、MQRR 構造体の応答レコードの Reason フィールドだけに返されます。</p> <p>対処</p> <p>MQOPEN 命令に指定された MQRR 構造体の応答コードを調べて、キューのオープンに失敗した理由を判定してください。また、アプリケーションがその命令に関するエラーを判定するために十分な応答レコードを提供しているかどうかを確認してください。</p>
2138	X'0000085A'	<p><b>MQRC_ADAPTER_DISC_LOAD_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2139	X'0000085B'	<p><b>MQRC_CNO_ERROR</b></p> <p>MQCONN 命令の呼び出しで、ConnectOpts 引数の MQCNO 構造体が次のどれかの理由で無効です。</p> <ul style="list-style-type: none"> <li>StrucId ニーモニックアイキャッチャが MQCNO_STRUC_ID ではありません。</li> <li>Version フィールドに無効な値、またはサポートされていない値が指定されています。</li> <li>引数ポインタが無効です。無効な引数ポインタを必ず検出できるとは限りません。検出できなかった場合には、予期しない結果が発生しています。</li> <li>キューマネージャは呼び出しが正常に終了しても、変更された構造体をアプリケーション記憶域へコピーできません。これは、例えば、ポインタが読み取り専用記憶域を指している場合などに起こります。</li> </ul> <p>対処</p> <p>MQCNO 構造体の定義を訂正します。必要な入力フィールドが正しく設定されていることを確認してください。</p>
2140	X'0000085C'	<p><b>MQRC_CICS_WAIT_FAILED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2141	X'0000085D'	<p><b>MQRC_DLH_ERROR</b></p> <p>MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQDLH 構造体があります。次のエラーなどがあります。</p> <ul style="list-style-type: none"> <li>StrucId フィールドが MQDLH_STRUC_ID ではありません。</li> <li>Version フィールドが MQDLH_VERSION_1 ではありません。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2141	X'0000085D'	<ul style="list-style-type: none"> <li>• CodedCharSetId フィールドに、0 または無効な負の値が指定されています。</li> <li>• 命令の BufferLength 引数の値が小さすぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul> <p>対処</p> <p>MQDLH 構造体のフィールドが適切に設定されているか確認してください。また、アプリケーションで CodedCharSetId フィールドに有効な値が設定されているか確認してください。MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR および MQCCSI_UNDEFINED は、このフィールドに無効です。</p>
2142	X'0000085E'	<p><b>MQRC_HEADER_ERROR</b></p> <p>MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQ ヘッダ構造体があります。次のエラーなどがあります。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが無効です。</li> <li>• Version フィールドが無効です。</li> <li>• StrucLength フィールドに指定されている値が小さすぎます。</li> <li>• CodedCharSetId フィールドに、0 または無効な負の値が指定されています。</li> <li>• 命令の BufferLength 引数の値が小さすぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul> <p>対処</p> <p>MQ ヘッダ構造体の定義を修正してください。また、フィールドが正しく設定されていることを確認してください。</p>
2143	X'0000085F'	<p><b>MQRC_SOURCE_LENGTH_ERROR</b></p> <p>MQXCNVC 命令の SourceLength 引数に、負の長さが指定されています。この理由コードは SourceLength 引数のポインタが不正な場合にも返されます。不正な SourceLength 引数のポインタを検出できないことがあります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>0 以上の長さを指定してください。</p>
2144	X'00000860'	<p><b>MQRC_TARGET_LENGTH_ERROR</b></p> <p>MQXCNVC 命令の TargetLength 引数に、負の長さが指定されています。この理由コードは TargetLength 引数のポインタが不正な場合にも返されることがあります。不正な TargetLength 引数のポインタを検出できないこと場合があります。検出されないときの動作は保証しません。</p> <p>対処</p> <p>0 以上の長さを指定してください。</p>
2145	X'00000861'	<p><b>MQRC_SOURCE_BUFFER_ERROR</b></p> <p>MQXCNVC 命令の SourceBuffer 引数のポインタが不正です。または、SourceLength 引数で指定された値より短い領域を指しています。不正なポインタを検出できない場合があります。検出されない場合の動作は保証しません。</p> <p>対処</p> <p>SourceBuffer 引数の指定を訂正してください。</p>
2146	X'00000862'	<p><b>MQRC_TARGET_BUFFER_ERROR</b></p>

値 (10 進数)	値 (16 進数)	説明
2146	X'00000862'	MQXCNVC 命令の TargetBuffer 引数のポインタが不正です。または、ポインタが読み出し専用領域、または TargetLength 引数で指定された値より短い領域を指しています。不正なポインタを検出できない場合があります。検出されない場合の動作は保証しません。 対処 引数を訂正してください。
2148	X'00000864'	<b>MQRC_IIH_ERROR</b> MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQIIH 構造体があります。次のエラーなどがあります。 <ul style="list-style-type: none"> <li>• StrucId フィールドが MQIIH_STRUC_ID ではありません。</li> <li>• Version フィールドが MQIIH_VERSION_1 ではありません。</li> <li>• StrucLength フィールドが MQIIH_LENGTH_1 ではありません。</li> <li>• 命令の BufferLength 引数の値が小さすぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul> 対処 MQIIH 構造体の定義を訂正します。フィールドが正しく設定されていることを確認してください。
2149	X'00000865'	<b>MQRC_PCF_ERROR</b> PCF データを含むメッセージを登録するための MQPUT 命令または MQPUT1 命令が発行されましたが、メッセージの長さが、メッセージ内の PCF 構造体の長さの合計に一致していません。この理由コードは、次の名称のメッセージで返される場合があります。 <ul style="list-style-type: none"> <li>• MQFMT_ADMIN</li> <li>• MQFMT_EVENT</li> <li>• MQFMT_PCF</li> </ul> 対処 MQPUT 命令または MQPUT1 命令に指定されたメッセージの長さがメッセージデータ内に含まれている PCF 構造体の長さの合計に等しいことを確認してください。
2150	X'00000866'	<b>MQRC_DBCS_ERROR</b> 2 バイト文字セット (DBCS) 列に変換するときにエラーが発生しました。これは次の場合に発生します。 MQXCNVC 命令で、SourceCCSID パラメタに 2 バイト文字セット (DBCS) のコード化文字セット識別子が指定されました。しかし、SourceBuffer 引数に有効な DBCS 列がありません。原因は次に示すどちらかです。 <ul style="list-style-type: none"> <li>• 文字列に不正な DBCS 文字があります。</li> <li>• 文字列が SBCS と DBCS 文字列の混合で、シフトアウトとシフトインの制御文字が正しく組み合わせられていません。この場合、完了コードは MQCC_FAILED です。</li> </ul> 対処 正しい文字列を指定してください。
2152	X'00000868'	<b>MQRC_OBJECT_NAME_ERROR</b> MQOD 構造体の RecsPresent フィールドが 0 より大きい状態で、配布リストをオープンするための MQOPEN 命令または MQPUT1 命令が発行されたとき、ObjectName フィールドが空白でもヌル文字列でもありません。

値 (10 進数)	値 (16 進数)	説明
2152	X'00000868'	<p>対処</p> <p>配布リストをオープンするには、ObjectName フィールドを空白またはヌル文字列に設定します。配布リストをオープンしない場合は、RecsPresent フィールドを 0 に設定してください。</p>
2153	X'00000869'	<p><b>MQRC_OBJECT_Q_MGR_NAME_ERROR</b></p> <p>MQOD 構造体の RecsPresent フィールドが 0 より大きい状態で、配布リストをオープンするための MQOPEN 命令または MQPUT1 命令が発行されたとき、ObjectQMgrName フィールドが空白でもヌル文字列でもありません。</p> <p>対処</p> <p>配布リストをオープンする場合は、ObjectQMgrName フィールドを空白、またはヌル文字列に設定してください。配布リストをオープンしない場合は、RecsPresent フィールドを 0 に設定してください。</p>
2154	X'0000086A'	<p><b>MQRC_RECS_PRESENT_ERROR</b></p> <p>MQOPEN 命令または MQPUT1 命令が、次のどちらかの理由で失敗しました。</p> <ul style="list-style-type: none"> <li>MQOD 構造体の RecsPresent フィールドが 0 より小さくなっています。</li> <li>MQOD 構造体の ObjectType フィールドが MQOT_Q でなく、RecsPresent フィールドが 0 ではありません。オープン対象のオブジェクトがキューでない場合は、RecsPresent フィールドは 0 でなければなりません。</li> </ul> <p>対処</p> <p>配布リストをオープンする場合は、ObjectType フィールドに MQOT_Q を設定してください。また、RecsPresent フィールドを、リスト内のあて先の数に設定してください。配布リストをオープンしない場合は、RecsPresent フィールドを 0 に設定してください。</p>
2155	X'0000086B'	<p><b>MQRC_OBJECT_RECORDS_ERROR</b></p> <p>MQOD 構造体の RecsPresent フィールドが 0 より大きい状態で、配布リストをオープンするための MQOPEN 命令または MQPUT1 命令が発行されたとき、MQOR 構造体レコードが正しく指定されていません。次の理由が考えられます。</p> <ul style="list-style-type: none"> <li>ObjectRecOffset フィールドが 0 で、ObjectRecPtr フィールドがヌルポインタまたは 0 です。</li> <li>ObjectRecOffset フィールドが 0 以外の数値で、ObjectRecPtr フィールドがヌルポインタでも 0 でもありません。</li> <li>ObjectRecPtr フィールドが有効なポインタではありません。</li> <li>ObjectRecPtr フィールドまたは ObjectRecOffset フィールドが、アクセス不能な記憶領域を指しています。</li> </ul> <p>対処</p> <p>ObjectRecOffset フィールドと ObjectRecPtr フィールドのどちらか一方が 0 で、もう一方が 0 以外の数値になっていることを確認してください。また、使用するフィールドがアクセス可能な記憶領域を指していることも確認してください。</p>
2156	X'0000086C'	<p><b>MQRC_RESPONSE_RECORDS_ERROR</b></p> <p>配布リストをオープンするための、つまり、MQOD 構造体の RecsPresent フィールドが 0 より大きい MQOPEN 命令または MQPUT1 命令が発行されましたが、MQRR 構造体の応答レコードが正しく指定されていません。次のどれかの場合が該当します。</p>

値 (10 進数)	値 (16 進数)	説明
2156	X'0000086C'	<ul style="list-style-type: none"> <li>ResponseRecOffset フィールドが 0 でなく、ResponseRecPtr フィールドがヌルポインタでも 0 でもない場合</li> <li>ResponseRecPtr フィールドが有効なポインタでない場合</li> <li>ResponseRecPtr フィールドまたは ResponseRecOffset フィールドが、アクセス不能な記憶領域を指している場合</li> </ul> <p>対処</p> <p>ResponseRecOffset フィールドおよび ResponseRecPtr フィールドのうち、少なくともどちらかが 0 であることを確認してください。また、使用するフィールドがアクセス可能な記憶領域を指していることを確認してください。</p>
2157	X'0000086D'	<p><b>MQRC_ASID_MISMATCH</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2158	X'0000086E'	<p><b>MQRC_PMO_RECORD_FLAGS_ERROR</b></p> <p>メッセージを登録するための MQPUT 命令または MQPUT1 命令が発行されましたが、MQPMO 構造体の PutMsgRecFields フィールドが、次のどれかの理由で無効になっています。</p> <ul style="list-style-type: none"> <li>フィールドに無効なフラグが含まれています。</li> <li>メッセージが配布リストに登録されている最中で、登録メッセージレコードが提供された、つまり、RecsPresent フィールドが 0 より大きく、PutMsgRecOffset フィールド、または PutMsgRecPtr フィールドのどちらかが 0 以外であるが、PutMsgRecFields フィールドの値が MQPMRF_NONE になっています。</li> <li>MQPMRF_ACCOUNTING_TOKEN が指定されているが、MQPMO_SET_IDENTITY_CONTEXT も、MQPMO_SET_ALL_CONTEXT も指定されていません。</li> </ul> <p>対処</p> <p>登録メッセージレコード内にどんなフィールドがあるのか示すための適切な MQPMRF_* フラグを使用して、PutMsgRecFields フィールドが設定されていることを確認してください。</p> <p>MQPMRF_ACCOUNTING_TOKEN が指定されている場合は、同時に MQPMO_SET_IDENTITY_CONTEXT または MQPMO_SET_ALL_CONTEXT のどちらかが指定されていることを確認してください。</p> <p>MQPMRF_ACCOUNTING_TOKEN が指定されていない場合は、PutMsgRecOffset フィールドおよび PutMsgRecPtr フィールドの両方を 0 に設定してください。</p>
2159	X'0000086F'	<p><b>MQRC_PUT_MSG_RECORDS_ERROR</b></p> <p>メッセージを配布リストに登録するための MQPUT 命令または MQPUT1 命令が発行されましたが、MQPMR 構造体のメッセージレコードが正しく指定されていません。次のどれかの場合が該当します。</p> <ul style="list-style-type: none"> <li>PutMsgRecOffset フィールドが 0 以外の値であり、PutMsgRecPtr フィールドがヌルポインタでも 0 でもない場合</li> <li>PutMsgRecPtr フィールドが有効なポインタではない場合</li> <li>PutMsgRecPtr フィールドまたは PutMsgRecOffset フィールドが、アクセス不能な記憶領域を指している場合</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2159	X'0000086F'	<p>対処</p> <p>PutMsgRecOffset フィールドおよび PutMsgRecPtr フィールドのうち、少なくともどちら一つが 0 であることを確認してください。また、使用するフィールドがアクセス可能な記憶域を指していることを確認してください。</p>
2160	X'00000870'	<p><b>MQRC_CONN_ID_IN_USE</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2161	X'00000871'	<p><b>MQRC_Q_MGR QUIESCING</b></p> <p>MQI 命令が発行されましたが、キューマネージャが休止している（終了の準備をしている）ため、命令が失敗しました。</p> <p>キューマネージャが停止しているとき、MQOPEN, MQPUT, MQPUT1 および MQGET 命令は引き続き正常に実行できます。ただし、アプリケーションは、命令に次の適切なオプションを指定することによって、命令が失敗するよう指定できます。</p> <ul style="list-style-type: none"> <li>• MQOPEN 命令の MQOO_FAIL_IF QUIESCING</li> <li>• MQPUT または MQPUT1 命令の MQPMO_FAIL_IF QUIESCING</li> <li>• MQGET 命令の MQGMO_FAIL_IF QUIESCING</li> </ul> <p>対処</p> <p>アプリケーションの後処理をして停止します。失敗した命令で、MQOO_FAIL_IF QUIESCING, MQPMO_FAIL_IF QUIESCING または MQGMO_FAIL_IF QUIESCING オプションを指定した場合は、これらのオプションを削除して、命令を再発行します。これらのオプションを省略することによって、アプリケーションは現在のトランザクションを完了しコミットするために動作し続けます。しかし、別のトランザクションは開始できません。</p>
2162	X'00000872'	<p><b>MQRC_Q_MGR STOPPING</b></p> <p>MQI 命令が発行されましたが、アプリケーションまたは OpenTP1 が終了中のため、命令は失敗しました。</p> <p>アプリケーションまたは OpenTP1 の終了処理中に、命令が呼び出されました。MQGMO_WAIT を指定した MQGET 命令の場合、待ち合わせは取り消されます。メッセージキューイングする MQI 命令は呼び出せません。</p> <p>対処</p> <p>アプリケーションの後処理をして停止します。アプリケーションが、外部コーディネータによるトランザクションの途中である場合は、アプリケーションで命令を発行してトランザクションをロールバックします。</p>
2163	X'00000873'	<p><b>MQRC_DUPLICATE_RECOV_COORD</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2173	X'0000087D'	<p><b>MQRC_PMO_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令の MQPMO 構造体が次のどれかの理由で不正です。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが MQPMO_STRUC_ID ではありません。</li> <li>• Version フィールドに無効な値、またはサポートされていない値が指定されています。</li> <li>• ポインタ引数が不正です。不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2173	X'0000087D'	<ul style="list-style-type: none"> <li>命令は成功しましたが、変更された構造体がアプリケーションの記憶領域にコピーされません。これは、ポインタ引数が読み出し専用領域を指している場合にも発生することがあります。</li> </ul> <p>対処</p> <p>MQPMO 構造体の定義を訂正してください。要求された入力フィールドが正しく設定されているか確認してください。</p>
2182	X'00000886'	<p><b>MQRC_API_EXIT_NOT_FOUND</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2183	X'00000887'	<p><b>MQRC_API_EXIT_LOAD_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2184	X'00000888'	<p><b>MQRC_REMOTE_Q_NAME_ERROR</b></p> <p>MQOPEN 命令または MQPUT1 命令で、次に示すどちらかのエラーが発生しました。</p> <ul style="list-style-type: none"> <li>リモートキューのローカル定義または定義の別名が指定されましたが、リモートキュー定義の RemoteQName 属性が空白です。なお、このエラーは定義の XmitQName が空白でない場合にも発生します。</li> <li>オブジェクト記述子の ObjectQMgrName フィールドが空白、およびローカルキューマネージャ名ではありません。また、ObjectName フィールドが空白です。</li> </ul> <p>対処</p> <p>リモートキューのローカル定義を変更して、正しいリモートキュー名を指定してください。または、オブジェクト記述子の ObjectName フィールドに空白でないオブジェクト名を指定してください。</p>
2185	X'00000889'	<p><b>MQRC_INCONSISTENT_PERSISTENCE</b></p> <p>MQPUT 命令が発行され、グループのメッセージまたは論理メッセージのセグメントが登録されました。しかし、MQMD 構造体の Persistence フィールドの指定値またはデフォルト値が、キューハンドルに対してキューマネージャが保持する、現在のグループとセグメント情報と矛盾します。グループ内の全メッセージおよび論理メッセージの全セグメントの永続性は同じ値を持つ必要があります。つまり、すべて永続またはすべて非永続である必要があります。</p> <p>現在の命令が MQPMO_LOGICAL_ORDER を指定する場合、命令は失敗します。現在の命令は MQPMO_LOGICAL_ORDER を指定しないが、直前の MQPUT 命令がキューハンドルに指定した場合は、命令は完了コード MQCC_WARNING で成功します。</p> <p>対処</p> <p>グループ内のすべてのメッセージまたは論理メッセージのすべてのセグメントで、永続性について同じ値が使用されるよう、アプリケーションを変更してください。</p>
2186	X'0000088A'	<p><b>MQRC_GMO_ERROR</b></p> <p>MQGET 命令の、MQGMO 構造体が次のどれかの理由で不正です。</p> <ul style="list-style-type: none"> <li>StrucId フィールドが MQGMO_STRUC_ID ではありません。</li> <li>Version フィールドに無効な値またはサポートされていない値が指定されています。</li> <li>ポインタ引数が不正です。ただし、不正なポインタ引数を検出できないことがあります。検出されないときの動作は保証しません。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2186	X'0000088A'	<ul style="list-style-type: none"> <li>キューマネージャが変更した構造体をアプリケーション領域にコピーできません。命令が成功してもエラーになります。ポインタ引数が読み出し専用領域を指しているときも、このエラーが発生することがあります。</li> </ul> <p>対処</p> <p>MQGMO 構造体の定義を訂正してください。必要な入力フィールドが正しく設定されているか確認してください。</p>
2187	X'0000088B'	<p><b>MQRC_CICS_BRIDGE_RESTRICTION</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2188	X'0000088C'	<p><b>MQRC_STOPPED_BY_CLUSTER_EXIT</b></p> <p>オープンまたはメッセージ登録のためにクラスタキューに MQOPEN, MQPUT または MQPUT1 命令が発行されましたが、クラスタワークロード出口が命令を拒否しました。</p> <p>対処</p> <p>クラスタワークロード出口が正しく記述されているか確認してください。クラスタワークロード出口が命令を拒否した理由を調査し、問題を対策してください。</p>
2189	X'0000088D'	<p><b>MQRC_CLUSTER_RESOLUTION_ERROR</b></p> <p>MQOPEN 命令, MQPUT 命令, または MQPUT1 命令が、クラスタキューへのオープンまたはメッセージ登録のために発行されましたが、リポジトリマネージャからの応答がないため、キュー定義を正しく解決できません。</p> <p>リポジトリマネージャから応答がない原因として、TP1/MQ とフルリポジトリマネージャ間のチャネル障害やリポジトリマネージャの処理遅延が考えられます。</p> <p>対処</p> <p>MQRC_CLUSTER_RESOLUTION_ERROR を返却した MQI 命令を再発行してください。再発行で解決しない場合は、次に示す対処をしてください。</p> <ul style="list-style-type: none"> <li>リポジトリマネージャが動作していることをチェックしてください。</li> <li>キューおよびチャネル定義が正しいことをチェックしてください。</li> <li>クラスタ内に登録できるキューおよび使用できるチャネルの組み合わせがあることをチェックしてください (キューやチャネルが削除された場合に、このエラーが発生するおそれがあります)。</li> </ul> <p>問題を解決したあと、アプリケーションを再実行してください。</p>
2190	X'0000088E'	<p><b>MQRC_CONVERTED_STRING_TOO_BIG</b></p> <p>GetMsgOpts 引数に MQGMO_CONVERT オプションが指定されている MQGET 命令で、メッセージ内の固定長フィールドの文字列がデータ変換時に拡張しフィールド長を超えました。この場合、キューマネージャは、末尾の空白文字および最初のヌル文字に続く文字を廃棄して、文字列をフィールド長に合わせます。しかし、この場合は廃棄するだけの文字がありませんでした。</p> <p>この理由コードは、フォーマット名が MQFMT_IMS_VAR_STRING のメッセージの場合も発生することがあります。この場合は、IMS 可変文字列が拡張し、その長さが IMS 可変文字列の構造体内で 2 バイトバイナリ長フィールドのサイズを超えてしまいます (キューマネージャは IMS 可変文字列内の末尾の空白を廃棄しません)。</p> <p>MQGET 命令の CompCode 引数が MQCC_WARNING に設定され、メッセージは変換されずに返されます。メッセージが幾つかの部分から構成されていて、各部分が独自の文字セットとマシンコードで記述されている場合 (例えば、フォーマット名が</p>

値 (10 進数)	値 (16 進数)	説明
2190	X'0000088E'	MQFMT_DEAD_LETTER_HEADER のメッセージ), 変換される部分と変換されない部分が発生することがあります。しかし、文字セットとマシンコード形式のフィールドで返される値は、関連するメッセージデータを常に正確に表します。 この理由コードは、末尾の空白文字を廃棄することによって文字列をフィールドに合わせられる場合は発生しません。 対処 メッセージ内のフィールドに正しい値が含まれていること、メッセージの送信側と受信側が指定した文字セット識別子が正しいことを確認してください。正しい場合はメッセージのデータレイアウトを変更し、変換時に文字列が拡張しても十分な余裕があるようにフィールド長を増やします。
2191	X'0000088F'	<b>MQRC_TMC_ERROR</b> MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQTMC2 構造体があります。次のエラーなどがあります。 <ul style="list-style-type: none"> <li>• StrucId フィールドが MQTMC_STRUC_ID ではありません。</li> <li>• Version フィールドが MQTMC_VERSION_2 ではありません。</li> <li>• 命令の BufferLength 引数の値が小さすぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul> 対処 MQTMC 構造体または MQTMC2 構造体の定義を訂正します。各フィールドを正しく設定してください。
2192	X'00000890'	<b>MQRC_PAGESET_FULL</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2192	X'00000890'	<b>MQRC_STORAGE_MEDIUM_FULL</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2193	X'00000891'	<b>MQRC_PAGESET_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2194	X'00000892'	<b>MQRC_NAME_NOT_VALID_FOR_TYPE</b> MQOPEN 命令を呼び出して、キューマネージャ定義をオープンしました。しかし、ObjDesc 引数の ObjectName フィールドに空白が設定されていません。 対処 ObjectName フィールドに空白を設定してください。
2195	X'00000893'	<b>MQRC_UNEXPECTED_ERROR</b> 予期しないエラーが発生したため、命令が拒否されました。 また、次に示すすべての項目を指定して、キューマネージャがメッセージのセグメント分割をした場合も、この理由コードが返されます。 <ul style="list-style-type: none"> <li>• MQMD 構造体の Version フィールドに MQMD_VERSION_2 を指定</li> <li>• MQMD 構造体の MsgFlags フィールドに MQMF_SEGMENTATION_ALLOWED を指定</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2195	X'00000893'	<ul style="list-style-type: none"> <li>MQMD 構造体の Format フィールドに MQFMT_MD_EXTENSION を指定</li> <li>ユーザデータに MQMDE 構造体を指定</li> </ul> <p>対処</p> <p>システム管理者に連絡してください。</p> <p>また、キューマネージャがメッセージをセグメント分割する場合の対処を次に示します。</p> <ul style="list-style-type: none"> <li>MQMD 構造体の Version フィールドに MQMD_VERSION_1 を指定</li> <li>MQMD 構造体の MsgFlags フィールドに MQMF_NONE を指定</li> <li>ユーザデータの MQMDE 構造体を削除</li> </ul>
2196	X'00000894'	<p><b>MQRC_UNKNOW_N_XMIT_Q</b></p> <p>MQOPEN 命令または MQPUT1 命令で、メッセージがリモートキューマネージャに送信されることになっています。オブジェクト記述子の ObjectName フィールドまたは ObjectQMgrName フィールドに、リモートキューのローカル定義名が指定されています。なお、ObjectQMgrName の場合は、キューマネージャ別名が使われています。しかし、定義の XmitQName 属性が空白ではなく、ローカルに定義されるキューの名称でもありません。</p> <p>対処</p> <p>ObjectName と ObjectQMgrName に指定された値を確認してください。正しい場合、キュー定義を確認してください。名称解決については、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>
2197	X'00000895'	<p><b>MQRC_UNKNOW_N_DEF_XMIT_Q</b></p> <p>MQOPEN 命令または MQPUT1 命令があて先としてリモートキューを指定して呼び出されました。リモートキューのローカル定義が指定された、または、キューマネージャ別名の実名が解釈されていても、ローカル定義の XmitQName 属性が空白です。あて先キューマネージャと同じ名称で定義されたキューがありません。そのため、キューマネージャは省略時の転送キューを使用しようとしてしました。しかし、DefXmitQName キューマネージャ属性で定義された名称が、ローカルに定義されたキューの名称ではありません。</p> <p>対処</p> <p>キュー定義またはキューマネージャ属性を訂正してください。転送キューについては、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>
2198	X'00000896'	<p><b>MQRC_DEF_XMIT_Q_TYPE_ERROR</b></p> <p>あて先としてリモートキューを指定して、MQOPEN 命令または MQPUT1 命令が呼び出されました。リモートキューのローカル定義が指定された、またはキューマネージャ別名の実名が解釈されましたが、ローカル定義の XmitQName 属性が空白です。</p> <p>あて先キューマネージャと同じ名称で定義された転送キューがないため、ローカルキューマネージャは省略時の転送キューを使用しようとしてしました。しかし、DefXmitQName キューマネージャ属性で定義されたキューは、ローカルキューではないため使用できません。</p> <p>転送キューについては、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> <p>対処</p> <p>次に示す対処のうちどれかを実行してください。</p> <ul style="list-style-type: none"> <li>リモートキューのローカル定義の XmitQName 属性の値として、ローカル転送キューを指定してください。</li> <li>リモートキューマネージャと同じ名称でローカル転送キューを定義してください。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2198	X'00000896'	<ul style="list-style-type: none"> <li>DefXmitQName キューマネージャ属性の値としてローカル転送キューを指定してください。</li> </ul>
2199	X'00000897'	<p><b>MQRC_DEF_XMIT_Q_USAGE_ERROR</b></p> <p>あて先としてリモートキューを指定して、MQOPEN 命令または MQPUT1 命令が呼び出されました。リモートキューのローカル定義が指定された、またはキューマネージャ別名の実名が解釈されましたが、ローカル定義の XmitQName 属性が空白です。あて先キューマネージャと同じ名称で定義された転送キューがないため、ローカルキューマネージャは省略時の転送キューを使用しようとした。しかし、DefXmitQName キューマネージャ属性で定義されたキューが、値 MQUS_TRANSMISSION の Usage 属性を持っていないため、使用できません。</p> <p>転送キューについては、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> <p>対処</p> <p>次に示す対処のうちどれかを実行してください。</p> <ul style="list-style-type: none"> <li>リモートキューのローカル定義の XmitQName 属性の値として、ローカル転送キューを指定してください。</li> <li>リモートキューマネージャと同じ名称でローカル転送キューを定義してください。</li> <li>DefXmitQName キューマネージャ属性の値として異なるローカル転送キューを指定してください。</li> <li>DefXmitQName キューの Usage 属性を MQUS_TRANSMISSION に変更してください。</li> </ul>
2201	X'00000899'	<p><b>MQRC_NAME_IN_USE</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2202	X'0000089A'	<p><b>MQRC_CONNECTION_QUIESCING</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2203	X'0000089B'	<p><b>MQRC_CONNECTION_STOPPING</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2204	X'0000089C'	<p><b>MQRC_ADAPTER_NOT_AVAILABLE</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2206	X'0000089E'	<p><b>MQRC_MSG_ID_ERROR</b></p> <p>選択基準としてメッセージ識別子を使用してメッセージを取り出す MQGET 命令が発行されましたが、このキューではメッセージ識別子による選択がサポートされないため、失敗しました。</p> <p>対処</p> <p>メッセージ識別子による選択を実行しないようにアプリケーションを変更します。つまり、MsgId フィールドを MQMI_NONE に設定し、MQGMO 構造体で MQMO_MATCH_MSG_ID を指定しません。</p>

値 (10 進数)	値 (16 進数)	説明
2207	X'0000089F'	<p><b>MQRC_CORREL_ID_ERROR</b></p> <p>選択基準として相関識別子を使用してメッセージを取り出す MQGET 命令が発行されましたが、このキューでは相関識別子による選択がサポートされないため、失敗しました。</p> <p>対処</p> <p>相関識別子による選択を実行しないようにアプリケーションを変更します。つまり、CorrelId フィールドを MQCI_NONE に設定し、MQGMO 構造体で MQMO_MATCH_CORREL_ID を指定しません。</p>
2208	X'000008A0'	<p><b>MQRC_FILE_SYSTEM_ERROR</b></p> <p>キューで操作をしようとしたときに、ファイルシステムから予期しないリターンコードを受け取りました。</p> <p>この理由コードは VSE/ESA でだけ出力されます。</p> <p>対処</p> <p>アクセスしようとしたキューについてのファイルシステム定義をチェックしてください。VSAM ファイルの場合は、制御間隔がキューに許可されているメッセージの最大長の処理時間以上であるかどうかをチェックしてください。</p>
2209	X'000008A1'	<p><b>MQRC_NO_MSG_LOCKED</b></p> <p>MQGMO_UNLOCK オプションを指定して MQGET 命令が呼び出されましたが、排他状態にされたメッセージはありません。</p> <p>対処</p> <p>同一ハンドルを使用して、MQGMO_LOCK を指定した以前の MQGET 命令で、メッセージが排他状態にされていないか確認してください。また、呼び出された命令のうち、メッセージの排他状態を解除したものがないか確認してください。</p>
2216	X'000008A8'	<p><b>MQRC_FILE_NOT_AUDITED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2217	X'000008A9'	<p><b>MQRC_CONNECTION_NOT_AUTHORIZED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2218	X'000008AA'	<p><b>MQRC_MSG_TOO_BIG_FOR_CHANNEL</b></p> <p>メッセージがリモートキューに書き込まれましたが、その長さが、チャンネルによって認められている最大メッセージ長を超えています。この理由コードは、報告メッセージのメッセージ記述子内の Feedback フィールドに返されます。</p> <p>対処</p> <p>チャンネル定義を調べてください。チャンネルが受信できる最大メッセージ長を大きくするか、そのメッセージを複数の小さいメッセージに分割してください。</p>
2219	X'000008AB'	<p><b>MQRC_CALL_IN_PROGRESS</b></p> <p>コネクションに対して命令がすでに処理されている間に、アプリケーションがほかの命令を呼び出しました。一度に処理できる命令は、アプリケーションコネクション一つにつき 1 命令です。</p>

値 (10 進数)	値 (16 進数)	説明
2219	X'000008AB'	<p>アプリケーションがマルチスレッドを使用する場合、または MQI 命令処理の一部として出口が呼ばれる場合、同時並行の命令が発生します。例えば、MQGET 命令の処理で呼ばれるデータ変換出口は、MQI 命令を発行することがあります。</p> <p>対処</p> <p>命令の実行中は、そのほかの命令を発行しないでください。データ変換 UOC の中から命令を発行しないでください。</p>
2220	X'000008AC'	<p><b>MQRC_RMH_ERROR</b></p> <p>MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQRMH 構造体があります。次のエラーなどがあります。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが MQRMH_STRUC_ID ではありません。</li> <li>• Version フィールドが MQRMH_VERSION_1 ではありません。</li> <li>• StrucLength フィールドに指定されている値が小さ過ぎて、構造体、および構造体の末尾にある可変長データを格納できません。</li> <li>• CodedCharSetId フィールドに、0 または無効な負の値が指定されています。</li> <li>• 命令の BufferLength 引数の値が小さ過ぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul> <p>対処</p> <p>MQRMH 構造体の定義を修正してください。また、フィールドが正しく設定されていることを確認してください。</p>
2222	X'000008AE'	<p><b>MQRC_Q_MGR_ACTIVE</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2223	X'000008AF'	<p><b>MQRC_Q_MGR_NOT_ACTIVE</b></p> <p>この理由コードは、キューマネージャが停止または静止を要求されたときに返されます。</p> <p>対処</p> <p>ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2224	X'000008B0'	<p><b>MQRC_Q_DEPTH_HIGH</b></p> <p>MQPUT 命令または MQPUT1 命令が実行されたために、キューのサイズが QDepthHighLimit 属性で指定されている最大値を超えて増加しました。</p> <p>対処</p> <p>ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2225	X'000008B1'	<p><b>MQRC_Q_DEPTH_LOW</b></p> <p>MQGET 命令が実行されたために、キューのサイズが QDepthLowLimit 属性で指定されている最小値より減少しました。</p> <p>対処</p> <p>ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2226	X'000008B2'	<p><b>MQRC_Q_SERVICE_INTERVAL_HIGH</b></p>

値 (10 進数)	値 (16 進数)	説明
2226	X'000008B2'	<p>QServiceInterval 属性に指定された限界よりも長い間隔内で正常な読み取りまたは登録を検出ませんでした。</p> <p>対処 ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2227	X'000008B3'	<p><b>MQRC_Q_SERVICE_INTERVAL_OK</b></p> <p>QServiceInterval 属性に指定された限界以下の間隔内で正常な読み取りを検出しました。</p> <p>対処 ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2232	X'000008B8'	<p><b>MQRC_UNIT_OF_WORK_NOT_STARTED</b></p> <p>トランザクション内でメッセージの取り出しまたは登録を実行するために MQGET, MQPUT または MQPUT1 命令が発行されましたが、TM/MP トランザクションが開始していませんでした。MQGET 命令に MQGMO_NO_SYNCPOINT が指定されていない場合、および MQPUT または MQPUT1 命令に MQPMO_NO_SYNCPOINT が指定されていない場合 (デフォルト) は、命令にはトランザクションが必要です。</p> <p>対処 TM/MP トランザクションが使用可能であることを確認するか、MQGMO_NO_SYNCPOINT オプション指定の MQGET 命令、または MQPMO_NO_SYNCPOINT オプション指定の MQPUT または MQPUT1 命令を発行してください。これによってトランザクションを自動的に開始します。</p>
2233	X'000008B9'	<p><b>MQRC_CHANNEL_AUTO_DEF_OK</b></p> <p>この理由コードが返されるのは、チャンネルの自動定義が正常に終了した場合です。チャンネルはメッセージチャンネルエージェント (MCA) によって定義されています。</p> <p>対処 ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。</p>
2234	X'000008BA'	<p><b>MQRC_CHANNEL_AUTO_DEF_ERROR</b></p> <p>この理由コードが返されるのは、チャンネルの自動定義が失敗した場合です。失敗の理由としては、次のように考えられます。</p> <ul style="list-style-type: none"> <li>定義プロセスの実行中にエラーが発生した</li> <li>チャンネルの自動定義出口によって定義が禁止された</li> </ul> <p>イベントメッセージに、失敗の理由を示す追加情報が返されます。</p> <p>対処 イベントメッセージ中に返された追加情報で、障害の理由を確認してください。</p>
2235	X'000008BB'	<p><b>MQRC_CFH_ERROR</b></p> <p>MQPUT または MQPUT1 命令で、メッセージデータ内の PCF ヘッダである MQCFH 構造体が無効です。</p> <p>対処 MQCFH 構造体の定義を修正してください。フィールドの値が正しく設定されているか確認してください。</p>

値 (10 進数)	値 (16 進数)	説明
2236	X'000008BC'	<p><b>MQRC_CFIL_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージデータ中の PCF 整数リストの引数である MQCFIL 構造体が無効です。</p> <p>対処</p> <p>MQCFIL 構造体の定義を修正してください。また、フィールドが正しく設定されていることを確認してください。</p>
2237	X'000008BD'	<p><b>MQRC_CFIN_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージデータ中の PCF 整数の引数となる MQCFIN 構造体が無効です。</p> <p>対処</p> <p>MQCFIN 構造体の定義を修正してください。また、フィールドが正しく設定されていることを確認してください。</p>
2238	X'000008BE'	<p><b>MQRC_CFSL_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージデータ中の PCF スtringリスト引数である MQCFSL 構造体が無効です。</p> <p>対処</p> <p>MQCFSL 構造体の定義を修正してください。また、フィールドが正しく設定されていることを確認してください。</p>
2239	X'000008BF'	<p><b>MQRC_CFST_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令で、メッセージデータ中の PCF スtringの引数である MQCFST 構造体が無効です。</p> <p>対処</p> <p>MQCFST 構造体の定義を修正してください。また、フィールドが正しく設定されていることを確認してください。</p>
2241	X'000008C1'	<p><b>MQRC_INCOMPLETE_GROUP</b></p> <p>不完全なメッセージグループのあるキューハンドルを使用して、キューを操作しようとしてきました。この理由コードが返されるのは、次の場合です。</p> <ul style="list-style-type: none"> <li>MQPUT 命令で、アプリケーションがグループ内にはないメッセージを登録するために MQPMO_LOGICAL_ORDER を指定する場合 この場合、完了コードは MQCC_FAILED です。</li> <li>MQPUT 命令で、アプリケーションが MQPMO_LOGICAL_ORDER を指定しない が、キューハンドルに直前の MQPUT 命令で MQPMO_LOGICAL_ORDER を指定した場合 この場合、完了コードは MQCC_WARNING です。</li> <li>MQGET 命令で、アプリケーションが MQGMO_LOGICAL_ORDER を指定しない が、キューハンドルに直前の MQGET 命令で MQGMO_LOGICAL_ORDER を指定した場合 この場合、完了コードは MQCC_WARNING です。</li> <li>MQCLOSE 命令で、アプリケーションが、不完全なメッセージグループがあるキューをクローズする場合 この場合、完了コードは MQCC_WARNING です。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2241	X'000008C1'	<p>不完全なメッセージグループと同時に不完全な論理メッセージも存在する場合、理由コード MQRC_INCOMPLETE_MSG が MQRC_INCOMPLETE_GROUP の前に返されます。</p> <p>対処</p> <p>この理由コードを意図していた場合は、対処する必要はありません。意図したものでない場合は、そのグループ内の最後のメッセージに対する MQPUT 命令に MQMF_LAST_MSG_IN_GROUP が指定されていることを確認してください。</p>
2242	X'000008C2'	<p><b>MQRC_INCOMPLETE_MSG</b></p> <p>不完全な論理メッセージのあるキューハンドルを使用して、キューを操作しようとした。この理由コードが返されるのは、次の場合です。</p> <ul style="list-style-type: none"> <li>MQPUT 命令で、アプリケーションがセグメントでないメッセージを登録しようとして、MQPMO_LOGICAL_ORDER を指定した場合 この場合、命令は失敗します。</li> <li>MQPUT 命令で、アプリケーションが MQPMO_LOGICAL_ORDER の指定なしで順番と異なるセグメントのメッセージを登録しようとして、キューハンドル用の直前の MQPUT 命令に MQPMO_LOGICAL_ORDER が指定されていた場合 この場合、命令は成功し、完了コード MQCC_WARNING が返されます。</li> <li>MQGET 命令で、アプリケーションが MQGMO_LOGICAL_ORDER の指定なしで順番と異なるセグメントのメッセージを取り出そうとして、キューハンドル用の直前の MQGET 命令に MQGMO_LOGICAL_ORDER が指定されていた場合 この場合、命令は成功し、完了コード MQCC_WARNING が返されます。</li> <li>MQCLOSE 命令で、アプリケーションが不完全な論理メッセージのあるキューをクローズしようとした場合 この場合、命令は成功し、完了コード MQCC_WARNING が返されます。</li> <li>MQPUT 命令で、セグメントの MQMF_LAST_MSG_IN_GROUP フラグの設定が、直前のメッセージセグメントと異なる場合</li> </ul> <p>対処</p> <p>この理由コードを意図していた場合は、対処する必要はありません。意図したものでない場合は、最後のセグメントに対する MQPUT 命令に MQMF_LAST_SEGMENT が指定されていることを確認してください。</p>
2243	X'000008C3'	<p><b>MQRC_INCONSISTENT_CCSDS</b></p> <p>MQGMO_COMPLETE_MSG オプションを指定した MQGET 命令が発行されましたが、検索対象のメッセージは MQMD 構造体の CodedCharSetId フィールド中の、値の異なる複数のセグメントで構成されています。</p> <p>この理由コードが返されるのは、セグメントがネットワークを経由して異なるパスを通り、それらのパスの一部で MCA センダ側の変換が使用できる場合です。この場合命令は成功し、完了コード MQCC_WARNING が返されますが、同一の文字セット ID を持つセグメントの初期値と等しい数のセグメントだけが戻ります。</p> <p>対処</p> <p>MQGET 命令から MQGMO_COMPLETE_MSG オプションを解除し、残りのメッセージセグメントを一つずつ取り出してください。</p>
2244	X'000008C4'	<p><b>MQRC_INCONSISTENT_ENCODINGS</b></p>

値 (10 進数)	値 (16 進数)	説明
2244	X'000008C4'	<p>MQGMO_COMPLETE_MSG オプションを指定した MQGET 命令が発行されましたが、取り出し対象のメッセージは、MQMD 構造体の Encoding フィールド中の、値の異なる複数のセグメントで構成されています。</p> <p>この理由コードが返されるのは、セグメントがネットワークを経由して異なるパスを通り、それらのパスの一部で MCA センダ側の変換が使用できる場合です。このとき命令は成功し、完了コード MQCC_WARNING が返されますが、同一のマシンコード形式を持つセグメントの初期値と等しい数のセグメントだけが返されます。</p> <p>対処</p> <p>MQGET 命令から MQGMO_COMPLETE_MSG オプションを解除し、残りのメッセージセグメントを一つずつ取り出してください。</p>
2245	X'000008C5'	<p><b>MQRC_INCONSISTENT_UOW</b></p> <p>次のどれかが該当します。</p> <ol style="list-style-type: none"> <li>グループ内のメッセージまたは論理メッセージのセグメントを登録するために、MQPUT 命令が発行されました。しかし、MQPMO_SYNCPOINT オプションの指定値または初期値と、キューハンドル用にキューマネージャが保存している現在のグループおよびセグメントの情報が矛盾しています。</li> </ol> <p>現在の命令に MQPMO_LOGICAL_ORDER が指定されている場合は、その命令は失敗します。MQPMO_LOGICAL_ORDER が現在の命令には指定されないで、キューハンドルに対する直前の MQPUT 命令に指定されていた場合は、命令は成功して完了コード MQCC_WARNING が返されます。</p> <ol style="list-style-type: none"> <li>グループ内のメッセージまたは論理メッセージのセグメントをキューから取り出すために MQGET 命令が発行されました。しかし、MQGMO_SYNCPOINT オプションの指定値またはデフォルト値と、キューハンドル用にキューマネージャが保存している現在のグループおよびセグメントの情報が矛盾しています。</li> </ol> <p>現在の命令に MQGMO_LOGICAL_ORDER が指定されている場合は、その命令は失敗します。MQGMO_LOGICAL_ORDER が現在の命令には指定されないで、キューハンドルに対する直前の MQGET 命令に指定されていた場合は、命令は成功して完了コード MQCC_WARNING が返されます。</p> <ol style="list-style-type: none"> <li>セグメント方式 (mqtalcccha 定義コマンドの-g オプションに"buftype=seg"を指定) の受信側チャンネル (mqtalcccha 定義コマンドの-y オプションに"type=receiver", "type=requester", または"type=clusrchr"を指定) が、MQMDE 構造体の MsgFlags フィールドに MQMF_SEGMENTATION_ALLOWED が設定されているメッセージを受信しました。しかし、受信メッセージ長が、登録しようとしたキューの最大メッセージ長よりも大きいです。</li> </ol> <p>対処</p> <p>要因 1., 2. に対しては、アプリケーションを修正して、グループ内のすべてのメッセージ、または論理メッセージのすべてのセグメントについて、同じトランザクションを指定してください。</p> <p>要因 3. に対しては、メッセージ方式 (mqtalcccha 定義コマンドの-g オプションに"buftype=msg"を指定) に変更するか、キューの最大メッセージ長を大きくしてください。</p>
2246	X'000008C6'	<p><b>MQRC_INVALID_MSG_UNDER_CURSOR</b></p> <p>MQGMO_MSG_UNDER_CURSOR オプションまたは MQGMO_BROWSE_MSG_UNDER_CURSOR オプションのどちらかとともに、MQGMO_COMPLETE_MSG オプションを指定した MQGET 命令が発行されましたが、</p>

値 (10 進数)	値 (16 進数)	説明
2246	X'000008C6'	<p>カーソルの下にあるメッセージには、0 より大きい Offset フィールドを指定した MQMD 構造体があります。</p> <p>MQGMO_COMPLETE_MSG オプションが指定されているので、このメッセージは検索用として無効です。</p> <p>対処</p> <p>ブラウザカーソルの位置を変更し、MQMD 構造体の Offset フィールドが 0 になっているメッセージにカーソルを置いてください。または、MQGMO_COMPLETE_MSG オプションを解除してください。</p>
2247	X'000008C7'	<p><b>MQRC_MATCH_OPTIONS_ERROR</b></p> <p>MQGET 命令が発行されましたが、GetMsgOpts 引数の MatchOptions フィールドの値が次のどちらかの理由で無効です。</p> <ul style="list-style-type: none"> <li>未定義のオプションが指定されています。</li> <li>次のすべてが当てはまります。 <ul style="list-style-type: none"> <li>MQGMO_LOGICAL_ORDER が指定されています。</li> <li>キューハンドル用の現在のメッセージグループまたは論理メッセージがあります。</li> <li>次のオプションがどちらも指定されていません。 <ul style="list-style-type: none"> <li>MQGMO_BROWSE_MSG_UNDER_CURSOR</li> <li>MQGMO_MSG_UNDER_CURSOR</li> </ul> </li> <li>一つ以上の MQMO_* オプションが指定されています。</li> <li>指定されている MQMO_* オプションに対応する MsgDesc 引数のフィールドの値が、次に返されるメッセージ用の MQMD 構造体の対応するフィールドの値と異なっています。</li> </ul> </li> </ul> <p>対処</p> <p>フィールドに有効なオプションだけが指定されていることを確認してください。</p>
2248	X'000008C8'	<p><b>MQRC_MDE_ERROR</b></p> <p>MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQMDE 構造体が含まれています。エラーとして、次のことが考えられます。</p> <ul style="list-style-type: none"> <li>StrucId フィールドの値が、MQMDE_STRUC_ID ではありません。</li> <li>Version フィールドが MQMDE_VERSION_2 ではありません。</li> <li>StrucLength フィールドの値が MQMDE_LENGTH_2 ではありません。</li> <li>CodedCharSetId フィールドに、0 または無効な負の値が指定されました。</li> <li>命令の BufferLength 引数の値が小さすぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul> <p>対処</p> <p>メッセージ記述子の拡張の定義を修正してください。入力されなければならないフィールドが正しく設定されているか確認してください。</p>
2249	X'000008C9'	<p><b>MQRC_MSG_FLAGS_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令が発行されましたが、メッセージ記述子 MQMD 構造体の MsgFlags フィールドに、ローカルキューマネージャが認識できないメッセージフラグが一つ以上あります。この理由コードが返される原因になったメッセージフラグは、メッセージのあて先によって異なります。</p>

値 (10 進数)	値 (16 進数)	説明
2249	X'000008C9'	<p>この理由コードは、報告メッセージの MQMD 構造体の Feedback フィールド、またはデッドレターキューへ登録したメッセージの MQDLH 構造体の Reason フィールドに返されることもあります。どちらの場合も、メッセージの送信元が指定したメッセージフラグの中に、あて先キューマネージャがサポートしていないものが一つ以上あることを示しています。</p> <p>対処</p> <p>次に示す内容を確認してください。</p> <ul style="list-style-type: none"> <li>・メッセージ記述子の MsgFlags フィールドが、メッセージ記述子の宣言時の値を使用して初期設定されているかどうか、または MQPUT か MQPUT1 命令より前に MsgFlags フィールドに値が割り当てられているかどうか。メッセージフラグが不要な場合は、MQMF_NONE を指定します。</li> <li>・指定されたメッセージフラグが有効であるかどうか。フラグについては、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」を参照してください。</li> <li>・個々のメッセージフラグを追加することによって複数のメッセージフラグを設定している場合は、同じメッセージフラグを 2 回追加していないか確認してください。</li> </ul>
2250	X'000008CA'	<p><b>MQRC_MSG_SEQ_NUMBER_ERROR</b></p> <p>MQGET 命令、MQPUT 命令、または MQPUT1 命令が発行されましたが、MQMD 構造体または MQMDE 構造体の MsgSeqNumber フィールドに、1 より小さい値か 999999999 より大きい値が指定されています。MQPUT 命令で、実行結果として MsgSeqNumber フィールドが 999999999 より大きくなった場合も、このエラーが発生することがあります。</p> <p>対処</p> <p>1~999999999 の範囲内の値を指定してください。999999999 個を超えるメッセージを含むメッセージグループを作成しないでください。</p>
2251	X'000008CB'	<p><b>MQRC_OFFSET_ERROR</b></p> <p>MQPUT 命令または MQPUT1 命令が発行されましたが、MQMD 構造体または MQMDE 構造体の Offset フィールドに、0 より小さいか、999999999 より大きい値が指定されています。</p> <p>MQPUT 命令で、実行結果として Offset フィールドが 999999999 より大きくなった場合にも、このエラーが発生することがあります。</p> <p>対処</p> <p>0~999999999 の範囲内の値を指定してください。オフセットが 999999999 を超えるようなメッセージセグメントを作成しないでください。</p>
2252	X'000008CC'	<p><b>MQRC_ORIGINAL_LENGTH_ERROR</b></p> <p>セグメントである報告メッセージを登録するための MQPUT 命令または MQPUT1 命令が発行されましたが、MQMD 構造体または MQMDE 構造体の OriginalLength フィールドが、次のどれかに該当しています。</p> <ul style="list-style-type: none"> <li>・メッセージ内のデータの長さより小さい</li> <li>・1 より小さい (最後のセグメント以外のセグメントの場合)</li> <li>・0 より小さい (最後のセグメントの場合)</li> </ul> <p>対処</p> <p>0 より大きい値を指定してください。0 は、最後のセグメントにだけ有効です。</p>

値 (10 進数)	値 (16 進数)	説明
2253	X'000008CD'	<p><b>MQRC_SEGMENT_LENGTH_ZERO</b></p> <p>論理メッセージの開始位置、または中間のセグメントを登録するための MQPUT 命令または MQPUT1 命令が発行されましたが、そのセグメントの存在する可能性のある MQ ヘッダをすべて除いたアプリケーションメッセージデータ長が 0 です。開始位置、または中間のセグメントの長さは、少なくとも 1 でなければなりません。</p> <p>対処</p> <p>アプリケーションを調査して、セグメントが 1 以上の長さで登録されているか確認してください。ただし、論理メッセージの最後のセグメントだけは、セグメント長が 0 でもかまいません。</p>
2255	X'000008CF'	<p><b>MQRC_UOW_NOT_AVAILABLE</b></p> <p>トランザクションの外部で、メッセージの取り出しや登録をするための MQGET 命令、MQPUT 命令、または MQPUT1 命令が発行されましたが、その命令に指定されているオプションでは、キューマネージャがトランザクション内部で命令を処理する必要があります。しかし、ユーザ定義のトランザクションがすでに存在しているので、キューマネージャは、命令の実行中に使用する一時的なトランザクションを作成できませんでした。</p> <p>この理由コードは次の場合に返されます。</p> <ul style="list-style-type: none"> <li>• MQGET 命令で、MQGMO 構造体に MQGMO_COMPLETE_MSG オプションが指定されていて、取り出す論理メッセージが永続メッセージであり、かつ、二つ以上のセグメントで構成されている場合</li> <li>• MQPUT 命令または MQPUT1 命令で、MQMD 構造体に MQMF_SEGMENTATION_ALLOWED フラグを指定することによって、メッセージのセグメント化が必要な場合</li> </ul> <p>対処</p> <p>MQGET 命令、MQPUT 命令、または MQPUT1 命令を、ユーザ定義のトランザクションの内部で発行してください。MQPUT 命令または MQPUT1 命令の場合は、キューマネージャによるセグメント化を防ぐために、メッセージサイズを小さくすることもできます。</p>
2256	X'000008D0'	<p><b>MQRC_WRONG_GMO_VERSION</b></p> <p>バージョン番号が MQGMO_VERSION_2 以降の MQGMO 構造体を必要とするオプションを指定した MQGET 命令で、提供された MQGMO 構造体は、条件を満たしていません。</p> <p>対処</p> <p>バージョン 2 の MQGMO 構造体を渡すようにプログラムを変更してください。プログラムを調査して、MQGMO 構造体の Version フィールドが MQGMO_VERSION_2 に設定されていることを確認してください。または、バージョン 2 の MQGMO 構造体を必要とするオプションを解除してください。</p>
2257	X'000008D1'	<p><b>MQRC_WRONG_MD_VERSION</b></p> <p>バージョン番号が MQMD_VERSION_2 以降の MQMD 構造体を必要とするオプションを指定した MQGET 命令、MQPUT 命令、または MQPUT1 命令で、提供された MQMD 構造体は、条件を満たしていません。</p> <p>対処</p> <p>バージョン 2 の MQMD 構造体を渡すように、アプリケーションを変更してください。アプリケーションを調査して、MQMD 構造体の Version フィールドが MQMD_VERSION_2 に設定されていることを確認してください。または、バージョン 2 の MQMD 構造体を必要とするオプションを解除してください。</p>

値 (10 進数)	値 (16 進数)	説明
2258	X'000008D2'	<p><b>MQRC_GROUP_ID_ERROR</b></p> <p>グループ内のメッセージ、メッセージセグメントおよびセグメント化が許可されている配布リストメッセージを書き込むための MQPUT 命令または MQPUT1 命令が発行されましたが、指定したオプションと値の組み合わせが無効です。次に示す理由が考えられます。</p> <ul style="list-style-type: none"> <li>MQPMO 構造体の Options フィールドに、MQPMO_LOGICAL_ORDER が指定されていません。</li> <li>MQPMO 構造体から提供されるレコードの数が少な過ぎます。または MQPMR 構造体のレコードに GroupId フィールドがありません。</li> <li>MQMD 構造体または MQMDE 構造体の MsgFlags フィールドに、次の一つ以上のフラグが指定されています。 <ul style="list-style-type: none"> <li>MQMF_SEGMENTATION_ALLOWED</li> <li>MQMF_*_MSG_IN_GROUP</li> <li>MQMF_*_SEGMENT</li> </ul> </li> <li>MQMD 構造体または MQMDE 構造体の GroupId フィールドが、MQGI_NONE ではありません。</li> </ul> <p>上記の組み合わせのオプションと値を指定すると、同じグループ ID があて先リストにあるすべてのあて先に使用されることとなります。これは、キューマネージャによって許可されない組み合わせです。</p> <p>対処</p> <p>MQMD 構造体または MQMDE 構造体の GroupId フィールドに、MQGI_NONE を指定してください。ただし、MQPUT 命令の場合は、MQPMO 構造体の Options フィールドに MQPMO_LOGICAL_ORDER を指定してください。</p>
2259	X'000008D3'	<p><b>MQRC_INCONSISTENT_BROWSE</b></p> <p>MQGMO_BROWSE_NEXT オプションを指定した MQGET 命令が発行されましたが、その MQGET 命令用の MQGMO_LOGICAL_ORDER オプションの指定と、キューハンドル用の直前の MQGET 命令の MQGMO_LOGICAL_ORDER オプションの指定が異なります。両方の命令に MQGMO_LOGICAL_ORDER を指定するか、どちらの命令にも MQGMO_LOGICAL_ORDER を指定しないかの、どちらかにしなければなりません。</p> <p>対処</p> <p>MQGMO_LOGICAL_ORDER オプションを指定するか、解除するか、適切な方の処置をとってください。MQGMO_LOGICAL_ORDER オプションの代わりに、論理順序と物理順序を切り替えるためには、MQGET 命令に MQGMO_BROWSE_FIRST オプションを指定して、キューの先頭からメッセージを取り出します。このとき、必要に応じて MQGMO_LOGICAL_ORDER を指定または省略します。</p>
2260	X'000008D4'	<p><b>MQRC_XQH_ERROR</b></p> <p>MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQXQH 構造体があります。次のエラーなどがあります。</p> <ul style="list-style-type: none"> <li>StrucId フィールドが MQXQH_STRUC_ID ではありません。</li> <li>Version フィールドが MQXQH_VERSION_1 ではありません。</li> <li>命令の BufferLength 引数の値が小さ過ぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2260	X'000008D4'	<p>対処</p> <p>MQXQH 構造体の定義を修正してください。また、フィールドが正しく設定されていることを確認してください。</p>
2261	X'000008D5'	<p><b>MQRC_SRC_ENV_ERROR</b></p> <p>この理由コードは、参照メッセージを処理するチャンネル出口が、参照メッセージヘッダ (MQRMH 構造体) の送信元環境データでエラーを検出したときに出力されます。次のどれかの状態になっています。</p> <ul style="list-style-type: none"> <li>• SrcEnvLength が 0 よりも小さくなっています。</li> <li>• SrcEnvLength がゼロより大きいですが、送信元環境データがありません。</li> <li>• SrcEnvLength がゼロより大きいですが、SrcEnvOffset が負の値またはゼロです。または MQRMH 構造体の固定部分の長さより小さくなっています。</li> <li>• SrcEnvLength がゼロより大きいですが、SrcEnvOffset に SrcEnvLength を加えた値が、StrucLength より大きくなっています。</li> </ul> <p>出口はこの理由コードを MQCXP 構造体の Feedback フィールドに返します。例外報告が要求されていれば、報告に関連する MQMD 構造体の Feedback フィールドにコピーされます。</p> <p>対処</p> <p>送信元環境データを正しく指定してください。</p>
2262	X'000008D6'	<p><b>MQRC_SRC_NAME_ERROR</b></p> <p>この理由コードは、参照メッセージを処理するチャンネル出口が、参照メッセージヘッダ (MQRMH 構造体) の送信元オブジェクト名データでエラーを検出したときに出力されます。次のどれかの状態になっています。</p> <ul style="list-style-type: none"> <li>• SrcNameLength が 0 よりも小さくなっています。</li> <li>• SrcNameLength がゼロより大きいですが、送信元名データがありません。</li> <li>• SrcNameLength がゼロより大きいですが、SrcNameOffset が負の値またはゼロです。または MQRMH 構造体の固定部分の長さより小さくなっています。</li> <li>• SrcNameLength がゼロより大きいですが、SrcNameOffset に SrcNameLength を加えた値が、StrucLength より大きくなっています。</li> </ul> <p>出口はこの理由コードを MQCXP 構造体の Feedback フィールドに返します。例外報告が要求されていれば、報告に関連する MQMD 構造体の Feedback フィールドにコピーされます。</p> <p>対処</p> <p>送信元オブジェクト名データを正しく指定してください。</p>
2263	X'000008D7'	<p><b>MQRC_DEST_ENV_ERROR</b></p> <p>この理由コードは、参照メッセージを処理するチャンネル出口が、参照メッセージヘッダ (MQRMH 構造体) のあて先環境データでエラーを検出したときに出力されます。次のどれかの状態になっています。</p> <ul style="list-style-type: none"> <li>• DestEnvLength が 0 よりも小さくなっています。</li> <li>• DestEnvLength がゼロより大きいですが、あて先環境データがありません。</li> <li>• DestEnvLength がゼロより大きいですが、DestEnvOffset が負の値またはゼロです。または MQRMH 構造体の固定部分の長さより小さくなっています。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2263	X'000008D7'	<ul style="list-style-type: none"> <li>• DestEnvLength がゼロより大きいですが、DestEnvOffset に DestEnvLength を加えた値が、StrucLength より大きくなっています。</li> </ul> <p>出口はこの理由コードを MQCXP 構造体の Feedback フィールドに返します。例外報告が要求されていれば、報告に関連する MQMD 構造体の Feedback フィールドにコピーされます。</p> <p>対処 あて先環境データを正しく指定してください。</p>
2264	X'000008D8'	<p><b>MQRC_DEST_NAME_ERROR</b></p> <p>この理由コードは、参照メッセージを処理するチャネル出口が、参照メッセージヘッダ (MQRMH 構造体) のあて先オブジェクト名データでエラーを検出したときに出力されます。次のどれかの状態になっています。</p> <ul style="list-style-type: none"> <li>• DestNameLength が 0 よりも小さくなっています。</li> <li>• DestNameLength がゼロより大きいですが、あて先名データがありません。</li> <li>• DestNameLength がゼロより大きいですが、DestNameOffset が負の値か、または MQRMH の固定部分の長さより小さくなっています。</li> <li>• DestNameLength がゼロより大きいですが、DestNameOffset に DestNameLength を加えた値が、StrucLength より大きくなっています。</li> </ul> <p>出口はこの理由コードを MQCXP 構造体の Feedback フィールドに返します。例外報告が要求されていれば、報告に関連する MQMD 構造体の Feedback フィールドにコピーされます。</p> <p>対処 あて先オブジェクト名データを正しく指定してください。</p>
2265	X'000008D9'	<p><b>MQRC_TM_ERROR</b></p> <p>MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQTM 構造体があります。次のエラーなどがあります。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが MQTM_STRUC_ID ではありません。</li> <li>• Version フィールドが MQTM_VERSION_1 ではありません。</li> <li>• 命令の BufferLength 引数の値が小さすぎて、構造体を格納できません (構造体がメッセージの末尾を超えて拡張されています)。</li> </ul> <p>対処 MQTMC 構造体の定義を訂正します。各フィールドを正しく設定してください。</p>
2266	X'000008DA'	<p><b>MQRC_CLUSTER_EXIT_ERROR</b></p> <p>オープンまたはメッセージ登録のためにクラスタキューに MQOPEN, MQPUT または MQPUT1 命令が発行されましたが、キューマネージャの ClusterWorkloadExit 属性で定義したクラスタワークロード出口が失敗したか、または時間内に応答しませんでした。このキューハンドルに対する以降の MQOPEN, MQPUT および MQPUT1 命令は、ClusterWorkloadExit 属性が空白として処理されます。</p> <p>対処 クラスタワークロード出口が正しく記述されているか確認してください。</p>
2267	X'000008DB'	<p><b>MQRC_CLUSTER_EXIT_LOAD_ERROR</b></p>

値 (10 進数)	値 (16 進数)	説明
2267	X'000008DB'	<p>キューマネージャに接続するために MQCONN または MQCONNX 命令が発行されましたが、キューマネージャの ClusterWorkloadExit 属性で定義されたクラスタワークロード出口をロードできなかったため、命令は失敗しました。</p> <p>対処</p> <p>クラスタワークロード出口が正しい場所にインストールされているか確認してください。</p>
2268	X'000008DC'	<p><b>MQRC_CLUSTER_PUT_INHIBITED</b></p> <p>MQOPEN 命令が、MQOO_OUTPUT および MQOO_BIND_ON_OPEN オプションを指定して、クラスタキューに発行されましたが、次のすべての条件を満たしたため、失敗しました。</p> <ul style="list-style-type: none"> <li>すべてのクラスタキューが、現在は登録禁止です (つまり、すべてのキューで、InhibitPut 属性に MQQA_PUT_INHIBITED が設定されています)。</li> <li>キューのローカルインスタンスがありません。ローカルインスタンスがある場合は、ローカルインスタンスが登録禁止であっても MQOPEN 命令は成功します。</li> <li>クラスタワークロード出口がキューにない、またはあってもキューインスタンスを選択していません。クラスタワークロード出口がキューインスタンスを選択している場合は、そのインスタンスが登録禁止であっても MQOPEN 命令は成功します。</li> </ul> <p>MQOO_BIND_NOT_FIXED オプションが MQOPEN 命令に指定される場合、クラスタ内のすべてのキューが登録禁止であっても成功することがあります。しかし、以降の MQPUT 命令は、発行時にすべてのキューが登録禁止のままであると失敗することがあります。</p> <p>対処</p> <p>システム設計で短い期間の登録要求禁止が許されている場合は、MQOPEN 命令を再発行してください。</p> <p>キュー取り出し禁止をしない運用、またはキュー取り出し禁止を長い時間解除しない運用を前提としてシステムが設計されている場合は、アプリケーションを終了してください。</p>
2269	X'000008DD'	<p><b>MQRC_CLUSTER_RESOURCE_ERROR</b></p> <p>クラスタキューに MQOPEN、MQPUT または MQPUT1 命令が発行されましたが、クラスタに必要なリソースを使用するときにエラーが発生しました。</p> <p>対処</p> <p>次に示す対処を行ってください。</p> <ul style="list-style-type: none"> <li>SYSTEM.CLUSTER.*キューが登録禁止または満杯になっていないことを確認します。</li> <li>フルリポジトリキューマネージャが使用可能であることを確認します。</li> </ul>
2270	X'000008DE'	<p><b>MQRC_NO_DESTINATIONS_AVAILABLE</b></p> <p>MQPUT 命令または MQPUT1 命令がクラスタキューにメッセージを登録するために発行されましたが、発行時にクラスタにはキューがなくなっていたため、メッセージが送信できません。</p> <p>この状況は、キューをオープンする MQOPEN 命令で MQOO_BIND_NOT_FIXED が指定された延長で MQPUT 命令でメッセージを登録するとき、または MQPUT1 命令でメッセージを登録するときに発生します。</p>

値 (10 進数)	値 (16 進数)	説明
2270	X'000008DE'	<p>対処</p> <p>MQRC_NO_DESTINATIONS_AVAILABLE が返却された MQI 命令が MQPUT 命令の場合は、MQOPEN 命令から再発行してください。MQPUT1 命令の場合は、MQPUT1 命令を再発行してください。</p> <p>再発行で解決しない場合は、すべてのキューがクラスタから削除された理由を特定するために、キューの定義とキューの状態をチェックしてください。</p> <p>問題を解決したあと、アプリケーションを再実行してください。</p>
2271	X'000008DF'	<p><b>MQRC_CONN_TAG_IN_USE</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2272	X'000008E0'	<p><b>MQRC_PARTIALLY_CONVERTED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2273	X'000008E1'	<p><b>MQRC_CONNECTION_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2274	X'000008E2'	<p><b>MQRC_OPTION_ENVIRONMENT_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2277	X'000008E5'	<p><b>MQRC_CD_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2278	X'000008E6'	<p><b>MQRC_CLIENT_CONN_ERROR</b></p> <p>キューマネージャに接続するために MQCONNX 命令を発行しましたが、MQCD チャネル定義構造体が正しく指定されていません。次に示すどれかが該当します。</p> <ul style="list-style-type: none"> <li>• ClientConnOffset がゼロではなく、ClientConnPtr がゼロまたはヌルポインタのどちらでもありません。</li> <li>• ClientConnPtr が無効なポインタです。</li> <li>• ClientConnPtr または ClientConnOffset が、アクセス不可能なストレージを指しています。</li> </ul> <p>対処</p> <p>ClientConnOffset および ClientConnPtr の少なくとも一方がゼロであることを確認してください。使用しているフィールドがアクセス可能なストレージを指していることを確認します。</p>
2279	X'000008E7'	<p><b>MQRC_CHANNEL_STOPPED_BY_USER</b></p> <p>チャンネルがオペレータによって停止されていた場合にこの状況が検出されます。理由修飾子が停止理由を示します。</p> <p>対処</p> <p>ありません。この理由コードは対応するイベントメッセージの識別に使用されるだけです。</p>

値 (10 進数)	値 (16 進数)	説明
2280	X'000008E8'	<b>MQRC_HCONFIG_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2281	X'000008E9'	<b>MQRC_FUNCTION_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2282	X'000008EA'	<b>MQRC_CHANNEL_STARTED</b> これは、次のどちらかの場合です。 <ul style="list-style-type: none"> <li>オペレータが Start Channel (チャンネル開始) コマンドを発行した場合</li> <li>チャンネルのインスタンスの確立に成功した場合</li> </ul> この理由コードは、初期データの転送が完了していて、必要な場合には、再度同期化してメッセージのネゴシエーションを続行できるときに返されます。 対処 ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。
2283	X'000008EB'	<b>MQRC_CHANNEL_STOPPED</b> この理由コードは、チャンネルが停止したときに返されます。理由修飾子は停止の理由を表します。 対処 ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。
2284	X'000008EC'	<b>MQRC_CHANNEL_CONV_ERROR</b> この理由コードが返されるのは、チャンネルがデータ変換を実行できないで、転送キューからメッセージを読み取る MQGET 命令がデータ変換エラーの原因となった場合です。変換理由コードで障害の理由を確認してください。 対処 ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。
2285	X'000008ED'	<b>MQRC_SERVICE_NOT_AVAILABLE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2286	X'000008EE'	<b>MQRC_INITIALIZATION_FAILED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2287	X'000008EF'	<b>MQRC_TERMINATION_FAILED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2288	X'000008F0'	<b>MQRC_UNKNOWN_Q_NAME</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。

値 (10 進数)	値 (16 進数)	説明
2289	X'000008F1'	<b>MQRC_SERVICE_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2290	X'000008F2'	<b>MQRC_Q_ALREADY_EXISTS</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2291	X'000008F3'	<b>MQRC_USER_ID_NOT_AVAILABLE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2292	X'000008F4'	<b>MQRC_UNKNOWN_ENTITY</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2293	X'000008F5'	<b>MQRC_UNKNOWN_AUTH_ENTITY</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2294	X'000008F6'	<b>MQRC_UNKNOWN_REF_OBJECT</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2295	X'000008F7'	<b>MQRC_CHANNEL_ACTIVATED</b> この理由コードが返されるのは、Channel Not Activated イベントの生成対象になっていたチャンネルが活動状態になるのを待っている場合、別のチャンネルが活動状態のスロットを解放したために活動状態となったときです。Channel Not Activated イベントは、活動スロットが解放されるのを待たないで活動できるチャンネルに対しては生成されません。  対処 ありません。この理由コードは、対応するイベントメッセージを識別するためにだけ使用されます。
2296	X'000008F8'	<b>MQRC_CHANNEL_NOT_ACTIVATED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2297	X'000008F9'	<b>MQRC_UOW_CANCELED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2298	X'000008FA'	<b>MQRC_FUNCTION_NOT_SUPPORTED</b> 要求された機能は、現在の環境では利用できません。  対処 アプリケーションから命令を削除してください。
2299	X'000008FB'	<b>MQRC_SELECTOR_TYPE_ERROR</b> Selector 引数のデータ型が間違っています。この引数は long 型でなければいけません。

値 (10 進数)	値 (16 進数)	説明
2299	X'000008FB'	対処 Selector 引数を long として宣言してください。
2300	X'000008FC'	<b>MQRC_COMMAND_TYPE_ERROR</b> mqExecute 命令が発行されましたが、管理バッグ内の MQIASY_TYPE データ項目の値が MQCFT_COMMAND ではありません。 対処 管理バッグ内の MQIASY_TYPE データ項目の値が MQCFT_COMMAND であることを確認してください。
2301	X'000008FD'	<b>MQRC_MULTIPLE_INSTANCE_ERROR</b> Selector 引数でシステムセクタ (MQIASY_*値の一つ) を指定していますが、ItemIndex 引数が MQIND_NONE ではありません。各システムセクタのインスタンスは一つだけバッグ内に存在できます。 対処 ItemIndex 引数の値に MQIND_NONE を指定してください。
2302	X'000008FE'	<b>MQRC_SYSTEM_ITEM_NOT_ALTERABLE</b> バッグ内のシステムデータ項目 (MQIASY_*セクタの一つのあるデータ項目) の値を変更する命令が発行されましたが、データ項目をアプリケーションで変更できないため、命令は失敗しました。 対処 ユーザ定義データ項目のセクタを指定するか、命令を削除してください。
2303	X'000008FF'	<b>MQRC_BAG_CONVERSION_ERROR</b> mqBufferToBag または mqGetBag 命令が発行されましたが、バッファまたはメッセージ内のデータをバッグに変換できませんでした。変換するデータが無効な PCF の場合に発生します。 対処 バッファまたはメッセージを作成したアプリケーションのロジックを確認して、バッファまたはメッセージに有効な PCF が含まれることを確認してください。 メッセージに無効な PCF がある場合、そのメッセージは mqGetBag 命令で取り出せません。 ・MQGMO_BROWSE_*オプションの一つが指定されると、メッセージはキューに残り、MQGET 命令で取り出せます。 ・その他の場合は、メッセージがキューからすでに削除され廃棄されています。メッセージがトランザクション内で取り出された場合、トランザクションをロールバックして MQGET 命令でメッセージを取り出せます。
2304	X'00000900'	<b>MQRC_SELECTOR_OUT_OF_RANGE</b> Selector 引数の値が、命令の有効範囲外です。バッグが MQCBO_CHECK_SELECTORS オプションで作成された場合は、次に示す条件を満たしてください。 ・mqAddInteger 命令のとき、値は MQIA_FIRST から MQIA_LAST までの範囲です。 ・mqAddString 命令のとき、値は MQCA_FIRST から MQCA_LAST までの範囲です。 バッグが MQCBO_CHECK_SELECTORS オプションで作成されなかった場合、値はゼロ以上でなければいけません。

値 (10 進数)	値 (16 進数)	説明
2304	X'00000900'	対処 有効な値を指定してください。
2305	X'00000901'	<b>MQRC_SELECTOR_NOT_UNIQUE</b> ItemIndex 引数の値が MQIND_NONE ですが、バッグには Selector 引数で指定されたセクタ値を持つデータ項目が複数あります。MQIND_NONE の場合、バッグには指定セクタのオカレンスを一つだけ含められます。 この理由コードは、セクタを一つしか許可しない必要引数に二つ以上のセクタのオカレンスが管理バッグにあるときにも、mqExecute 命令で発生します。 対処 バッグを作成したアプリケーションのロジックを確認してください。ロジックが正しい場合は、ItemIndex にゼロ以上の値を指定し、バッグ内の全セクタのオカレンスを処理するためのアプリケーションロジックを追加します。 発行する管理コマンドの記述を見直して、バッグ内にすべての必要引数が正しく定義されていることを確認します。
2306	X'00000902'	<b>MQRC_INDEX_NOT_PRESENT</b> 指定された索引がありません。 <ul style="list-style-type: none"> <li>• バッグの場合には、Selector 引数で指定したセクタ値を持つデータ項目が一つ以上バッグにあるが、どのデータ項目も索引値が ItemIndex 引数で指定されていないことを示します。Selector および ItemIndex 引数で指定したデータ項目がバッグ内に必要です。</li> <li>• ネームリストの場合には、索引引数値が大き過ぎて、有効な値の範囲外であることを示します。</li> </ul> 対処 バッグまたはネームリストにあるデータ項目の索引を指定してください。mqCountItems 命令を使用してバッグにあるセクタを持つデータ項目の数を確認したり、nameCount メソッドを使用してネームリスト内の名前数を確認したりします。
2307	X'00000903'	<b>MQRC_STRING_ERROR</b> String 引数が無効です。引数ポインタが無効または読み取り専用ストレージを指しています。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。 対処 引数を訂正してください。
2308	X'00000904'	<b>MQRC_ENCODING_NOT_SUPPORTED</b> メッセージ記述子 MQMD 構造体の Encoding フィールドにサポートされない値があります。 <ul style="list-style-type: none"> <li>• mqPutBag 命令の場合、エラーに示されているフィールドは命令の MsgDesc 引数内にあります。</li> <li>• mqGetBag 命令の場合、エラーに示されているフィールドは次に示す場所にあります。  -MQGMO_CONVERT オプションが指定された場合は、命令の MsgDesc 引数です。  -MQGMO_CONVERT が指定されなかった場合は、取り出すメッセージのメッセージ記述子です。</li> </ul> 対処 値は MQENC_NATIVE でなくてはなりません。

値 (10 進数)	値 (16 進数)	説明
2308	X'00000904'	<p>メッセージ内の Encoding フィールドの値が無効な場合、メッセージは mqGetBag 命令で取り出せません。</p> <ul style="list-style-type: none"> <li>・MQGMO_BROWSE_*オプションの一つが指定されると、メッセージはキューに残り、MQGET 命令で取り出せます。</li> <li>・その他の場合は、メッセージがキューからすでに削除され廃棄されています。メッセージがトランザクション内で取り出された場合、トランザクションをロールバックし MQGET 命令でメッセージを取り出せます。</li> </ul>
2309	X'00000905'	<p><b>MQRC_SELECTOR_NOT_PRESENT</b> Selector 引数に、バッグにないセレクトアが指定されています。</p> <p>対処 バッグにあるセレクトアを指定してください。</p>
2310	X'00000906'	<p><b>MQRC_OUT_SELECTOR_ERROR</b> OutSelector 引数が無効です。引数ポインタが無効かまたは読み取り専用ストレージを指しています。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。</p> <p>対処 引数を修正してください。</p>
2311	X'00000907'	<p><b>MQRC_STRING_TRUNCATED</b> 命令によって返された文字列が長過ぎてバッファに入りません。文字列はバッファに入るように切り捨てられました。</p> <p>対処 文字列の全体が必要な場合は、バッファを大きくしてください。mqInquireString 命令では StringLength 引数を設定し、切り捨てないで文字列全体が入るバッファサイズを指定します。</p>
2312	X'00000908'	<p><b>MQRC_SELECTOR_WRONG_TYPE</b> 指定のセレクトアを持つデータ項目がバッグ内にありますが、そのデータ項目のデータ型が、使用する命令のデータ型と矛盾します。例えば、データ項目が整数データ型で、使用する命令は文字データ型の mqSetString という場合があります。</p> <p>mqBagToBuffer, mqExecute および mqPutBag 命令でも、MQIACF_INQUIRY データ項目をバッグに追加するために mqAddString または mqSetString が使用された場合にこの理由コードが発生します。</p> <p>対処 mqSetInteger および mqSetString 命令の場合、MQIND_ALL を ItemIndex 引数に指定して、必要なデータ型の新規オカレンスを作成する前に指定セレクトアの既存の全オカレンスをバッグから削除してください。</p> <p>mqInquireBag, mqInquireInteger および mqInquireString 命令の場合、mqInquireItemInfo 命令を使用して指定セレクトアを持つデータ項目のデータ型を決定し、次に適切な命令を使用してデータ項目の値を決定します。</p> <p>mqBagToBuffer, mqExecute および mqPutBag 命令の場合は、mqAddInteger または mqSetInteger 命令を使用して MQIACF_INQUIRY データ項目を必ずバッグに追加します。</p>
2313	X'00000909'	<p><b>MQRC_INCONSISTENT_ITEM_TYPE</b></p>

値 (10 進数)	値 (16 進数)	説明
2313	X'00000909'	<p>指定セクタの異なるオカレンスをバッグに追加するために mqAddInteger または mqAddString 命令が発行されましたが、オカレンスのデータ型が最初のオカレンスのデータ型と異なりました。</p> <p>mqBufferToBag および mqGetBag 命令で、バッファまたはメッセージ内の PCF に複数のオカレンスがあるセクタが含まれていて、それらのオカレンスのデータ型が不一致な場合にも、この理由コードが発生することがあります。</p> <p>対処</p> <p>mqAddInteger および mqAddString 命令の場合は、バッグ内のセクタで最初のオカレンスのデータ型に適した命令を使用してください。</p> <p>mqBufferToBag および mqGetBag 命令の場合は、バッファを作成したか、またはメッセージを送信したアプリケーションのロジックを確認して、セクタの複数オカレンスのデータ型が同じであるか確認します。セクタのデータ型が混在しているメッセージは、mqGetBag 命令では取り出せません。</p> <ul style="list-style-type: none"> <li>・MQGMO_BROWSE_*オプションの一つが指定されると、メッセージはキューに残り、MQGET 命令で取り出せます。</li> <li>・その他の場合は、メッセージがキューからすでに削除され廃棄されています。メッセージがトランザクション内で取り出された場合、トランザクションをロールバックしメッセージを MQGET 命令で取り出せます。</li> </ul>
2314	X'0000090A'	<p><b>MQRC_INDEX_ERROR</b></p> <p>命令またはメソッドの索引引数に無効な値があります。値はゼロ以上でなければいけません。バッグ命令の場合は特定の MQIND_*値も指定できます。</p> <ul style="list-style-type: none"> <li>・mqDeleteItem, mqSetInteger および mqSetString 命令の場合は、MQIND_ALL と MQIND_NONE が有効です。</li> <li>・mqInquireBag, mqInquireInteger, mqInquireString および mqInquireItemInfo 命令の場合は、MQIND_NONE が有効です。</li> </ul> <p>対処</p> <p>有効な値を指定してください。</p>
2315	X'0000090B'	<p><b>MQRC_SYSTEM_BAG_NOT_ALTERABLE</b></p> <p>バッグヘデータ項目を追加したり、バッグ内の既存データ項目値を変更したり、メッセージを取り出してバッグへ挿入したりする目的で命令が発行されました。しかし、前回の mqExecute 命令の結果としてシステムが作成したバッグであるため、命令は失敗しました。アプリケーションからシステムバッグを変更できません。</p> <p>対処</p> <p>アプリケーションによって作成されたバッグのハンドルを指定するか、命令を削除してください。</p>
2316	X'0000090C'	<p><b>MQRC_ITEM_COUNT_ERROR</b></p> <p>mqTruncateBag 命令が発行されましたが、ItemCount 引数に指定された値が無効です。値がゼロより小さいか、バッグ内のユーザ定義のデータ項目数より大きな値です。</p> <p>mqCountItems 命令でも、引数ポインタが無効か引数ポインタが読み取り専用ストレージを指している、この理由コードが発生します。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。</p>

値 (10 進数)	値 (16 進数)	説明
2316	X'0000090C'	<p>対処</p> <p>有効な値を指定してください。mqCountItems 命令を使用して、バッグ内のユーザ定義のデータ項目数を確認します。</p>
2317	X'0000090D'	<p><b>MQRC_FORMAT_NOT_SUPPORTED</b></p> <p>メッセージ記述子 MQMD 構造体の Format フィールドにサポートされない値があります。</p> <ul style="list-style-type: none"> <li>mqPutBag 命令の場合、エラーに示されているフィールドは命令の MsgDesc 引数内にあります。</li> <li>mqGetBag 命令の場合、エラーに示されているフィールドは、取り出すメッセージのメッセージ記述子にあります。</li> </ul> <p>対処</p> <p>値は次に示すどれかです。</p> <ul style="list-style-type: none"> <li>MQFMT_ADMIN</li> <li>MQFMT_EVENT</li> <li>MQFMT_PCF</li> </ul> <p>メッセージ内の Format フィールドの値が上記のどれでもない場合、メッセージは mqGetBag 命令で取り出せません。</p> <ul style="list-style-type: none"> <li>MQGMO_BROWSE_* オプションの一つが指定されると、メッセージはキューに残り、MQGET 命令で取り出せます。</li> <li>その他の場合は、メッセージがキューからすでに削除され、廃棄されています。メッセージがトランザクション内で取り出された場合、トランザクションをロールバックし MQGET 命令でメッセージを取り出せます。</li> </ul>
2318	X'0000090E'	<p><b>MQRC_SELECTOR_NOT_SUPPORTED</b></p> <p>Selector 引数にシステムセクタである値（負の値）が指定されていますが、そのシステムセクタは、命令でサポートされている値ではありません。</p> <p>対処</p> <p>サポートされているセクタ値を指定してください。</p>
2319	X'0000090F'	<p><b>MQRC_ITEM_VALUE_ERROR</b></p> <p>mqInquireBag または mqInquireInteger 命令が発行されましたが、ItemValue 引数が無効です。引数ポインタが無効か、または読み取り専用ストレージを指しています。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。</p> <p>対処</p> <p>引数を訂正してください。</p>
2320	X'00000910'	<p><b>MQRC_HBAG_ERROR</b></p> <p>引数としてバッグハンドルが指定された命令が発行されましたが、ハンドルが無効です。出力引数の場合、引数ポインタが無効か引数ポインタが読み取り専用ストレージを指しているときも、この理由コードが発生します。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。</p> <p>対処</p> <p>引数を訂正してください。</p>
2321	X'00000911'	<p><b>MQRC_PARAMETER_MISSING</b></p>

値 (10 進数)	値 (16 進数)	説明
2321	X'00000911'	<p>管理メッセージには、管理バッグ内には引数が必要です。この理由コードは、MQCBO_ADMIN_BAG または MQCBO_REORDER_AS_REQUIRED オプションによって作成されたバッグだけに発生します。</p> <p>対処 発行する管理コマンドの記述を見直して、バッグ内にすべての必要な引数があることを確認してください。</p>
2322	X'00000912'	<p><b>MQRC_CMD_SERVER_NOT_AVAILABLE</b> 管理コマンドを処理するコマンドサーバが使用できません。</p> <p>対処 コマンドサーバを開始してください。</p>
2323	X'00000913'	<p><b>MQRC_STRING_LENGTH_ERROR</b> StringLength 引数が無効です。引数ポインタが無効か、または読み取り専用ストレージを指しています。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。</p> <p>対処 引数を訂正してください。</p>
2324	X'00000914'	<p><b>MQRC_INQUIRY_COMMAND_ERROR</b> 事前に mqAddInquiry によって属性セクタがバッグに追加されましたが、mqBagToBuffer、mqExecute または mqPutBag 命令に使用するコマンドコードが認識されません。その結果、正しい PCF メッセージは生成されません。</p> <p>対処 mqAddInquiry 命令を削除し、代わりに適切な MQIACF_*_ATTRS または MQIACH_*_ATTRS セクタを指定した mqAddInteger 命令を使用してください。</p>
2325	X'00000915'	<p><b>MQRC_NESTED_BAG_NOT_SUPPORTED</b> 命令への入力であるバッグにネストバッグがあります。ネストバッグは、命令からの出力であるバッグの場合だけサポートされます。</p> <p>対処 異なるバッグを命令への入力として使用してください。</p>
2326	X'00000916'	<p><b>MQRC_BAG_WRONG_TYPE</b> 命令に対して不正な型のバッグのハンドルが、Bag 引数に指定されています。バッグは管理バッグでなくてはなりません。つまり、MQCBO_ADMIN_BAG オプションを指定した mqCreateBag 命令で作成するバッグの必要があります。</p> <p>対処 バッグを作成するときに MQCBO_ADMIN_BAG オプションを指定してください。</p>
2327	X'00000917'	<p><b>MQRC_ITEM_TYPE_ERROR</b> mqInquireItemInfo 命令が発行されましたが、ItemType 引数が無効です。引数ポインタが無効かまたは読み取り専用ストレージを指しています。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。</p> <p>対処 引数を訂正してください。</p>

値 (10 進数)	値 (16 進数)	説明
2328	X'00000918'	<p><b>MQRC_SYSTEM_BAG_NOT_DELETABLE</b></p> <p>バッグを削除するために mqDeleteBag 命令が発行されました。しかし、前回の mqExecute 命令の結果システムが作成したバッグだったため、命令は失敗しました。アプリケーションでシステムバッグを削除できません。</p> <p>対処</p> <p>アプリケーションによって作成されたバッグのハンドルを指定するか、命令を削除してください。</p>
2329	X'00000919'	<p><b>MQRC_SYSTEM_ITEM_NOT_DELETABLE</b></p> <p>バッグからシステムデータ項目 (MQIASY_*セクタの一つを持つデータ項目) の値を削除するための命令が発行されました。しかし、そのデータ項目はアプリケーションで削除できないため命令は失敗しました。</p> <p>対処</p> <p>ユーザ定義データ項目のセクタを指定するか、命令を削除してください。</p>
2330	X'0000091A'	<p><b>MQRC_CODED_CHAR_SET_ID_ERROR</b></p> <p>CodedCharSetId 引数が無効です。引数ポインタが無効かまたは読み取り専用ストレージを指しています。無効な引数ポインタを必ずしも検出できません。検出されなかった場合は予期しない結果が発生します。</p> <p>対処</p> <p>引数を訂正してください。</p>
2331	X'0000091B'	<p><b>MQRC_MSG_TOKEN_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2332	X'0000091C'	<p><b>MQRC_MISSING_WIH</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2333	X'0000091D'	<p><b>MQRC_WIH_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2334	X'0000091E'	<p><b>MQRC_RFH_ERROR</b></p> <p>MQPUT または MQPUT1 命令が発行されましたが、メッセージデータに無効な MQRFH または MQRFH2 構造体があります。次に示すエラーの可能性ががあります。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが MQRFH_STRUC_ID ではありません。</li> <li>• Version フィールドが MQRFH_VERSION_1 (MQRFH 構造体) または MQRFH_VERSION_2 (MQRFH2 構造体) ではありません。</li> <li>• StrucLength フィールドで指定する値が小さすぎるため、構造体および構造体の末尾にある可変長データを入れることができません。</li> <li>• CodedCharSetId フィールドに、0 または無効な負の値が指定されています。</li> <li>• 命令の BufferLength 引数の値が小さすぎるため、構造体を入れることができません。構造体がメッセージの末尾を超えて拡張されています。</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2334	X'0000091E'	<p>対処</p> <p>メッセージを生成したアプリケーションを変更して、有効な MQRFH 構造体がメッセージデータに配置されるようにしてください。</p>
2335	X'0000091F'	<p><b>MQRC_RFH_STRING_ERROR</b></p> <p>MQRFH 構造体の NameValueString フィールドの内容が無効です。NameValueString は、次に示す規則に正確に従ってください。</p> <ul style="list-style-type: none"> <li>文字列は、一つ以上の空白で区切られたゼロ以上の名前と値の対で構成されなければなりません。この場合の空白は有意ではありません。</li> <li>名前または値に有意のブランクが含まれている場合は、その名前または値を二重引用符で囲んでください。</li> <li>名前または値それ自体に一つ以上の二重引用符が含まれている場合は、その名前または値を二重引用符で囲み、かつ組み込まれている二重引用符は二重にしてください。</li> <li>名前または値には、区切り文字の役割を持つヌル以外のどの文字でも含めることができます。ヌルおよびそれに続く文字は、NameValueString に定義された長さまで無視されます。</li> </ul> <p>NameValueString を次に示します。</p> <p>Famous_Words "The program displayed ""Hello World"""</p> <p>対処</p> <p>メッセージを生成したアプリケーションを変更して、このアプリケーションが上記の規則に正確に従ったデータを NameValueString フィールドに配置するようにします。StrucLength フィールドが正しい値に設定されていることを確認してください。</p>
2336	X'00000920'	<p><b>MQRC_RFH_COMMAND_ERROR</b></p> <p>メッセージに MQRFH 構造体がありますが、NameValueString フィールドに含まれているコマンド名は無効です。</p> <p>対処</p> <p>メッセージを生成したアプリケーションを変更して、このアプリケーションが有効なコマンド名を NameValueString フィールドに配置するようにしてください。</p>
2337	X'00000921'	<p><b>MQRC_RFH_PARM_ERROR</b></p> <p>メッセージに MQRFH 構造体がありますが、NameValueString フィールドにある引数名は指定されたコマンドには無効です。</p> <p>対処</p> <p>メッセージを生成したアプリケーションを変更して、このアプリケーションが NameValueString フィールドに指定したコマンドに有効な引数だけを配置するようにしてください。</p>
2338	X'00000922'	<p><b>MQRC_RFH_DUPLICATE_PARM</b></p> <p>メッセージに MQRFH 構造体があります。しかし、指定したコマンドに引数の出現は 1 回だけが有効であるのに、NameValueString フィールドに 2 回以上出現しています。</p> <p>対処</p> <p>メッセージを生成したアプリケーションを変更して、NameValueString フィールドに引数が 1 回だけ出現するようにしてください。</p>
2339	X'00000923'	<p><b>MQRC_RFH_PARM_MISSING</b></p>

値 (10 進数)	値 (16 進数)	説明
2339	X'00000923'	メッセージに MQRFH 構造体があります。しかし、NameValueString フィールドに指定するコマンドには、必要な引数が不足しています。  対処 メッセージを生成したアプリケーションを変更して、指定したコマンドに必要なすべての引数を NameValueString フィールドに配置してください。
2340	X'00000924'	<b>MQRC_CHAR_CONVERSION_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2341	X'00000925'	<b>MQRC_UCS2_CONVERSION_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2342	X'00000926'	<b>MQRC_DB2_NOT_AVAILABLE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2343	X'00000927'	<b>MQRC_OBJECT_NOT_UNIQUE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2344	X'00000928'	<b>MQRC_CONN_TAG_NOT_RELEASED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2345	X'00000929'	<b>MQRC_CF_NOT_AVAILABLE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2346	X'0000092A'	<b>MQRC_CF_STRUC_IN_USE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2347	X'0000092B'	<b>MQRC_CF_STRUC_LIST_HDR_IN_USE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2348	X'0000092C'	<b>MQRC_CF_STRUC_AUTH_FAILED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2349	X'0000092D'	<b>MQRC_CF_STRUC_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2350	X'0000092E'	<b>MQRC_CONN_TAG_NOT_USABLE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2351	X'0000092F'	<b>MQRC_GLOBAL_UOW_CONFLICT</b>

値 (10 進数)	値 (16 進数)	説明
2351	X'0000092F'	<p>グローバルトランザクションの内部で、ほかのグローバルトランザクションに参加中の接続ハンドルを使用しようとしてしました。異なる DTC トランザクションに関するオブジェクト間で、アプリケーションが接続ハンドルを渡すとこのエラーが発生することがあります。トランザクションの完了は非同期であるため、アプリケーションが最初のオブジェクトを完了してトランザクションをコミットしたあとでこのエラーが発生することがあります。このエラーは、非トランザクション MQI 命令の場合には発生しません。</p> <p>この理由コードは Windows 環境でだけ発生します。</p> <p>対処</p> <p>オブジェクトのクラスに定義されている MTS Transaction Support 属性が正しく設定されているか確認してください。必要な場合には、異なるトランザクション内で実行中のオブジェクトによって接続ハンドルが使用されないようにアプリケーションを訂正してください。</p>
2352	X'00000930'	<p><b>MQRC_LOCAL_UOW_CONFLICT</b></p> <p>グローバルトランザクションの内部で、キューマネージャが調整するローカルトランザクションに参加中の接続ハンドルを使用しようとしてしました。アプリケーションがオブジェクト間で接続ハンドルを渡すとき、一方のオブジェクトが DTC トランザクションに関係していて他方が関係していない場合に、このエラーが発生することがあります。</p> <p>このエラーは、非トランザクション MQI 命令の場合には発生しません。</p> <p>この理由コードは Windows の環境でだけ返されます。</p> <p>対処</p> <p>オブジェクトのクラスに定義されている MTS Transaction Support 属性が正しく設定されているか確認してください。必要な場合には、接続ハンドルが異なるトランザクション内で実行中のオブジェクトによって使用されないようにアプリケーションを訂正してください。</p>
2353	X'00000931'	<p><b>MQRC_HANDLE_IN_USE_FOR_UOW</b></p> <p>トランザクション外で、グローバルトランザクションに参加中の接続ハンドルを使用しようとしてしました。</p> <p>アプリケーションがオブジェクト間で接続ハンドルを渡すとき、一方のオブジェクトが DTC トランザクションに関係していて他方が関係していない場合に、このエラーが発生することがあります。トランザクションの完了が非同期であるため、アプリケーションが最初のオブジェクトを完了してそのトランザクションをコミットしたあとでこのエラーが発生することがあります。</p> <p>トランザクションとともに作成され、トランザクションに関連づけられた単一オブジェクトが実行中にその関連を失った場合にも、このエラーが発生することがあります。DTC が MTS とは独立してそのトランザクションを終了するとき、関連は失われます。この終了は、トランザクションがタイムアウトになったか、DTC が停止されたためである可能性があります。</p> <p>このエラーは、非トランザクション MQI 命令の場合には発生しません。</p> <p>この理由コードは Windows の環境でだけ返されます。</p> <p>対処</p> <p>オブジェクトのクラスに定義されている MTS Transaction Support 属性が正しく設定されているか確認してください。必要な場合には、異なるトランザクション内で実行中のオブジェクトが同じ接続ハンドルの使用を試行しないようにアプリケーションを訂正してください。</p>

値 (10 進数)	値 (16 進数)	説明
2354	X'00000932'	<p><b>MQRC_UOW_ENLISTMENT_ERROR</b></p> <p>この理由コードはいろいろな理由で発生することがあります。最もよくある理由は、DTC トランザクションによって作成されたオブジェクトが、DTC トランザクションがタイムアウトするまでにトランザクションの MQI 命令を発行しないことです。トランザクションの MQI 命令が発行されたあとで DTC トランザクションがタイムアウトになる場合、MQI 命令の失敗によって理由コード MQRC_HANDLE_IN_USE_FOR_UOW が返されます。MQRC_UOW_ENLISTMENT_ERROR が返されるほかの理由は、不正なインストールです。Windows NT Service Pack は Windows NT Option Pack のあとにインストールしてください。</p> <p>この理由コードは Windows 環境でだけ返されます。</p> <p>対処</p> <p>DTC の Transaction timeout 値を確認してください。必要な場合には、インストール順序を確認してください。</p>
2355	X'00000933'	<p><b>MQRC_UOW_MIX_NOT_SUPPORTED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2356	X'00000934'	<p><b>MQRC_WXP_ERROR</b></p> <p>チェーンから次のレコードのアドレスを取得するために、クラスタワークロード出口から MQXCLWLN 命令が発行されました。次に示すどれかの理由で、ワークロード出口指数構造体 ExitParms が有効ではありません。</p> <ul style="list-style-type: none"> <li>• 引数ポインタが無効です。無効な引数およびポインタを必ず検出できるわけではありません。検出されなかった場合は予期しない結果が発生します。</li> <li>• StrucId フィールドが MQWXP_STRUC_ID ではありません。</li> <li>• Version フィールドが MQWXP_VERSION_2 ではありません。</li> </ul> <p>対処</p> <p>ExitParms に指定された引数が、出口が呼び出されたときに出口に渡された MQWXP 構造体であることを確認してください。</p>
2357	X'00000935'	<p><b>MQRC_CURRENT_RECORD_ERROR</b></p> <p>チェーンの次のレコードアドレスを取得するために、クラスタワークロード出口から MQXCLWLN 命令が発行されましたが、CurrentRecord 引数によって指定されたアドレスが有効なレコードのアドレスではありません。CurrentRecord はクラスタキャッシュ内にあるあて先レコード (MQWDR)、キューレコード (MQWQR)、またはクラスタレコード (MQWCR) でなければなりません。</p> <p>対処</p> <p>クラスタワークロード出口がクラスタキャッシュにある有効なレコードのアドレスを渡していることを確認してください。</p>
2358	X'00000936'	<p><b>MQRC_NEXT_OFFSET_ERROR</b></p> <p>チェーンの次のレコードアドレスを取得するために、クラスタワークロード出口から MQXCLWLN 命令が発行されましたが、NextOffset 引数によって指定されたオフセットが有効ではありません。NextOffset は以下のフィールドのどれかの値でなければなりません。</p> <ul style="list-style-type: none"> <li>• MQWDR の ChannelDefOffset フィールド</li> <li>• MQWDR の ClusterRecOffset フィールド</li> </ul>

値 (10 進数)	値 (16 進数)	説明
2358	X'00000936'	<ul style="list-style-type: none"> <li>MQWQR の ClusterRecOffset フィールド</li> <li>MQWCR の ClusterRecOffset フィールド</li> </ul> <p>対処</p> <p>NextOffset 引数に指定されている値が、上記のフィールドのどれかの値であることを確認してください。</p>
2359	X'00000937'	<p><b>MQRC_NO_RECORD_AVAILABLE</b></p> <p>チェーンの次のレコードアドレスを取得するために、クラスタワークロード出口から MQXCLWLN 命令が発行されましたが、現行レコードがチェーン内の最後のレコードです。</p> <p>対処</p> <p>ありません。</p>
2360	X'00000938'	<p><b>MQRC_OBJECT_LEVEL_INCOMPATIBLE</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2361	X'00000939'	<p><b>MQRC_NEXT_RECORD_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2362	X'0000093A'	<p><b>MQRC_BACKOUT_THRESHOLD_REACHED</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2363	X'0000093B'	<p><b>MQRC_MSG_NOT_MATCHED</b></p> <p>この理由コードは、MQDLH 構造体の Reason フィールドまたは報告メッセージの MQMD 構造体の Feedback フィールドにだけ発生します。</p> <p>ポイントツーポイントメッセージングを実行している間に、JMS は、キューをモニタする ConnectionConsumer のセレクトに合わないメッセージを検知しました。性能を維持するため、そのメッセージは MQMD 構造体の Report フィールドの後処理オプションに従って処理されました。</p> <p>Report フィールドで MQRO_EXCEPTION_*オプションのどれかを指定する場合、この理由コードが報告メッセージの Feedback フィールドに出力されます。Report フィールドで MQRO_DEAD_LETTER_Q を指定するか、または後処理オプションがデフォルトのままの場合、この理由コードが MQDLH 構造体の Reason フィールドに出力されます。</p> <p>対処</p> <p>キューをモニタする ConnectionConsumer が十分なセレクトを提供していることを確認してください。または、メッセージを保持するように QueueConnectionFactory を設定します。</p>
2364	X'0000093C'	<p><b>MQRC_JMS_FORMAT_ERROR</b></p> <p>この理由コードは、パースできないというメッセージを JMS が検出した場合に発生します。JMS ConnectionConsumer によってこのようなメッセージが検出された場合、そのメッセージは MQMD 構造体の Report フィールドの後処理オプションに従って処理されます。</p> <p>Report フィールドで MQRO_EXCEPTION_*オプションのどれかを指定する場合、この理由コードが報告メッセージの Feedback フィールドに出力されます。Report フィールド</p>

値 (10 進数)	値 (16 進数)	説明
2364	X'0000093C'	で MQRO_DEAD_LETTER_Q を指定するか、または後処理オプションがデフォルトのままの場合、この理由コードが MQDLH 構造体の Reason フィールドに出力されます。 対処 メッセージの発行元を調査してください。
2365	X'0000093D'	<b>MQRC_SEGMENTS_NOT_SUPPORTED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2366	X'0000093E'	<b>MQRC_WRONG_CF_LEVEL</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
2367	X'0000093F'	<b>MQRC_CONFIG_CREATE_OBJECT</b> この状態は、オブジェクトが作成される場合に検出されます。 対処 ありません。この理由コードは、対応するイベントメッセージの識別にだけ使用されます。
2368	X'00000940'	<b>MQRC_CONFIG_CHANGE_OBJECT</b> この状態は、オブジェクトが変更される場合に検出されます。 対処 ありません。この理由コードは、対応するイベントメッセージの識別にだけ使用されます。
2369	X'00000941'	<b>MQRC_CONFIG_DELETE_OBJECT</b> この状態は、オブジェクトが削除される場合に検出されます。 対処 ありません。この理由コードは、対応するイベントメッセージの識別にだけ使用されます。
2370	X'00000942'	<b>MQRC_CONFIG_REFRESH_OBJECT</b> この状態は、オブジェクトが更新される場合に検出されます。 対処 ありません。この理由コードは、対応するイベントメッセージの識別にだけ使用されます。
2371	X'00000943'	<b>MQRC_CHANNEL_SSL_ERROR</b> この状態は、SSL キー交換または認証の失敗のために接続が確立できない場合に検出されます。 対処 ありません。この理由コードは、対応するイベントメッセージの識別にだけ使用されます。
2373	X'00000945'	<b>MQRC_CF_STRUC_FAILED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。

値 (10 進数)	値 (16 進数)	説明
2374	X'00000946'	<p><b>MQRC_API_EXIT_ERROR</b></p> <p>API 出口機能が無効な応答コードを返したか、またはほかの理由で失敗しました。</p> <p>対処</p> <p>出口のロジックを調べ、出口が MQAXP 構造体の ExitResponse フィールドまたは ExitResponse2 フィールドに有効な値を返すことを確認してください。FFST レコードを調べて問題の詳細情報があるか確認してください。</p>
2375	X'00000947'	<p><b>MQRC_API_EXIT_INIT_ERROR</b></p> <p>API 出口機能のための実行環境の初期化中に、キューマネージャでエラーが発生しました。問題についての詳細情報を取得するために、FFST レコードを調査してください。</p>
2376	X'00000948'	<p><b>MQRC_API_EXIT_TERM_ERROR</b></p> <p>API 出口機能のための実行環境の終了中に、キューマネージャでエラーが発生しました。問題についての詳細情報を取得するために、FFST レコードを調査してください。</p>
2377	X'00000949'	<p><b>MQRC_EXIT_REASON_ERROR</b></p> <p>API 出口機能によって MQXEP 命令が発行されましたが、ExitReason 引数に指定された値が無効か、または指定された機能識別子 Function でサポートされていません。</p> <p>対処</p> <p>Function に指定された値に対して有効な値を ExitReason に指定するよう、出口機能を変更してください。</p>
2378	X'0000094A'	<p><b>MQRC_RESERVED_VALUE_ERROR</b></p> <p>API 出口機能によって MQXEP 命令が発行されましたが、Reserved 引数に指定された値が無効です。値はヌルポインタにしてください。</p> <p>対処</p> <p>Reserved 引数の値としてヌルポインタを指定するように出口を変更してください。</p>
2379	X'0000094B'	<p><b>MQRC_NO_DATA_AVAILABLE</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2380	X'0000094C'	<p><b>MQRC_SCO_ERROR</b></p> <p>MQCONNX 命令で、次に示すどちらかの理由によって MQSCO 構造体が無効です。</p> <ul style="list-style-type: none"> <li>• StrucId フィールドが MQSCO_STRUC_ID です。</li> <li>• Version フィールドが MQSCO_VERSION_1 ではありません。</li> </ul> <p>対処</p> <p>MQSCO 構造体の定義を訂正してください。</p>
2381	X'0000094D'	<p><b>MQRC_KEY_REPOSITORY_ERROR</b></p> <p>MQCONN または MQCONNX 命令で、キーリポジトリの位置が指定されていないか、無効か、または位置を指定してキーリポジトリにアクセスするとエラーが発生します。キーリポジトリの位置は、次に示すどちらかによって指定されます。</p> <ul style="list-style-type: none"> <li>• MQSSLKEYR 環境変数 (MQCONN または MQCONNX 命令の場合)</li> <li>• MQSCO 構造体の KeyRepository フィールドの値 (MQCONNX 命令の場合)</li> </ul> <p>MQCONNX 命令で MQSSLKEYR と KeyRepository の両方が指定されている場合、KeyRepository が使用されます。</p>

値 (10 進数)	値 (16 進数)	説明
2381	X'0000094D'	対処 キーリポジトリに有効な位置を指定してください。
2382	X'0000094E'	<b>MQRC_CRYPTO_HARDWARE_ERROR</b> MQCONN または MQCONNX 命令で、暗号ハードウェアの構成文字列が無効か、または構成文字列を使用して暗号ハードウェアを構成するとエラーが発生します。構成文字列は、次に示すどちらかによって指定されます。 <ul style="list-style-type: none"> <li>• MQSSLCRYP 環境変数 (MQCONN または MQCONNX 命令の場合)</li> <li>• MQSCO 構造体の CryptoHardware フィールドの値 (MQCONNX 命令の場合)</li> </ul> MQCONNX 命令で MQSSLCRYP と CryptoHardware の両方が指定されている場合、CryptoHardware が使用されます。 対処 暗号ハードウェアに有効な構成文字列を指定してください。
2383	X'0000094F'	<b>MQRC_AUTH_INFO_REC_COUNT_ERROR</b> MQCONNX 命令で、MQSCO 構造体の AuthInfoRecCount フィールドに、0 より小さい値が指定されています。 対処 AuthInfoRecCount フィールドにゼロ以上の値を指定してください。
2384	X'00000950'	<b>MQRC_AUTH_INFO_REC_ERROR</b> MQCONNX 命令で、MQSCO 構造体に MQAIR レコードのアドレスが正確に指定されていません。次に示すどれかが該当します。 <ul style="list-style-type: none"> <li>• AuthInfoRecCount はゼロより大きいですが、AuthInfoRecOffset がゼロであり、AuthInfoRecPtr がヌルポインタです。</li> <li>• AuthInfoRecOffset がゼロではなく、AuthInfoRecPtr がヌルポインタではありません。</li> <li>• AuthInfoRecPtr が無効なポインタです。</li> <li>• AuthInfoRecOffset または AuthInfoRecPtr が、アクセス不可能なストレージを指しています。</li> </ul> 対処 AuthInfoRecOffset または AuthInfoRecPtr の一方がゼロで、もう一方がゼロ以外の数値にしてください。使用しているフィールドがアクセス可能なストレージを指していることを確認します。
2385	X'00000951'	<b>MQRC_AIR_ERROR</b> MQCONNX 命令で MQAIR レコードが無効です。その理由は次に示すどちらかです。 <ul style="list-style-type: none"> <li>• StrucId フィールドが MQAIR_STRUC_ID ではありません。</li> <li>• Version フィールドが MQAIR_VERSION_1 ではありません。</li> </ul> 対処 MQAIR レコードの定義を訂正してください。
2386	X'00000952'	<b>MQRC_AUTH_INFO_TYPE_ERROR</b> MQCONNX 命令で、MQAIR レコードの AuthInfoType フィールドに無効な値が指定されています。

値 (10 進数)	値 (16 進数)	説明
2386	X'00000952'	対処 AuthInfoType に MQAIT_CRL_LDAP を指定してください。
2387	X'00000953'	<b>MQRC_AUTH_INFO_CONN_NAME_ERROR</b> MQCONNX 命令で、MQAIR レコードの AuthInfoConnName フィールドに無効な値が指定されています。 対処 有効な接続名を指定してください。
2388	X'00000954'	<b>MQRC_LDAP_USER_NAME_ERROR</b> MQCONNX 命令で、MQAIR レコードの LDAP ユーザ名が正しく指定されていません。次に示すどれかが該当します。 <ul style="list-style-type: none"> <li>LDAPUserNameLength がゼロより大きい、LDAPUserNameOffset がゼロであり、LDAPUserNamePtr がヌルポインタです。</li> <li>LDAPUserNameOffset がゼロではなく、LDAPUserNamePtr がヌルポインタではありません。</li> <li>LDAPUserNamePtr が無効なポインタです。</li> <li>LDAPUserNameOffset または LDAPUserNamePtr が、アクセスできないストレージを指しています。</li> </ul> 対処 LDAPUserNameOffset または LDAPUserNamePtr の一方がゼロで、もう一方がゼロ以外の数値にしてください。使用しているフィールドがアクセスできるストレージを指していることを確認します。
2389	X'00000955'	<b>MQRC_LDAP_USER_NAME_LENGTH_ERR</b> MQCONNX 命令で、MQAIR レコードの LDAPUserNameLength フィールドにゼロより小さい値が指定されています。 対処 LDAPUserNameLength に、ゼロ以上の値を指定してください。
2390	X'00000956'	<b>MQRC_LDAP_PASSWORD_ERROR</b> MQCONNX 命令で、MQAIR レコードの LDAPPASSWORD フィールドに値が指定されていますが、このフィールドには値を指定できません。 対処 ブランクまたはヌルの値を指定してください。
2391	X'00000957'	<b>MQRC_SSL_ALREADY_INITIALIZED</b> SSL 構成オプションを指定して MQCONN または MQCONNX 命令が発行されましたが、SSL 環境は初期化済みです。キューマネージャとの接続は正常に完了しましたが、命令に指定された SSL 構成オプションは無視されました。代わりに既存の SSL 環境が使用されました。 対処 MQCONN または MQCONNX 命令に定義した SSL 構成オプションでアプリケーションを実行する場合、MQDISC 命令を使用してキューマネージャとの接続を切断して、アプリケーションを終了してください。または、SSL 環境が初期設定されていないときにアプリケーションをあとで実行してください。

値 (10 進数)	値 (16 進数)	説明
2392	X'00000958'	<p><b>MQRC_SSL_CONFIG_ERROR</b></p> <p>MQCONNX 命令で、MQCNO 構造体は MQSCO 構造体を正しく指定していません。次に示すどれかが該当します。</p> <ul style="list-style-type: none"> <li>• SSLConfigOffset がゼロではなく、SSLConfigPtr がヌルポインタではありません。</li> <li>• SSLConfigPtr が無効なポインタです。</li> <li>• SSLConfigOffset または SSLConfigPtr が、アクセスできないストレージを指しています。</li> </ul> <p>対処</p> <p>SSLConfigOffset または SSLConfigPtr の一方がゼロで、もう一方がゼロ以外の数値にしてください。使用しているフィールドがアクセスできるストレージを指していることを確認します。</p>
2393	X'00000959'	<p><b>MQRC_SSL_INITIALIZATION_ERROR</b></p> <p>SSL 構成オプションを指定して MQCONN または MQCONNX 命令が発行されましたが、SSL 環境の初期化中にエラーが発生しました。</p> <p>対処</p> <p>SSL のインストールが正しいことを確認してください。</p>
2394	X'0000095A'	<p><b>MQRC_Q_INDEX_TYPE_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
2396	X'0000095C'	<p><b>MQRC_SSL_NOT_ALLOWED</b></p> <p>SSL 暗号化を指定したキューマネージャへの接続が要求されました。しかし、要求された接続モード (バインディング接続など) では SSL がサポートされていません。</p> <p>この理由コードは、Java アプリケーションの場合だけ発生します。</p> <p>対処</p> <p>クライアント接続モードを要求するか、または SSL 暗号化を使用不可にするよう、アプリケーションを訂正してください。</p>
2397	X'0000095D'	<p><b>MQRC_JSSE_ERROR</b></p> <p>JSSE がエラー (SSL 暗号化を使用したキューマネージャとの接続中など) を報告しました。この理由コードを含む MQException オブジェクトは、JSSE がスローする例外を参照します。これは、MQException.getCause() メソッドを使用して取得できます。JMS では、MQException はスローされた JMSEException にリンクされます。</p> <p>この理由コードは、Java アプリケーションの場合だけ発生します。</p> <p>対処</p> <p>原因となった例外を調査して、JSSE エラーを確認してください。</p>
2398	X'0000095E'	<p><b>MQRC_SSL_PEER_NAME_MISMATCH</b></p> <p>アプリケーションは SSL 暗号化を使用してキューマネージャに接続しようとしたますが、キューマネージャによって示された識別名は、指定されたパターンと一致しません。</p> <p>この理由コードは、Java アプリケーションの場合だけ発生します。</p> <p>対処</p> <p>キューマネージャの識別で使用する証明書を調査してください。また、アプリケーションによって指定されている sslPeerName プロパティの値も調査してください。</p>

値 (10 進数)	値 (16 進数)	説明
2399	X'0000095F'	<p><b>MQRC_SSL_PEER_NAME_ERROR</b></p> <p>アプリケーションに指定されているピア名の形式が不正です。</p> <p>この理由コードは、Java アプリケーションの場合だけ発生します。</p> <p>対処</p> <p>アプリケーションによって指定されている sslPeerName プロパティの値を調査してください。</p>
2400	X'00000960'	<p><b>MQRC_UNSUPPORTED_CIPHER_SUITE</b></p> <p>SSL 暗号化を指定したキューマネージャへの接続が要求されました。しかし、アプリケーションに指定されている CipherSuite はサポートされていないことが JSSE によって報告されました。</p> <p>この理由コードは、Java アプリケーションの場合だけ発生します。</p> <p>対処</p> <p>アプリケーションに指定されている CipherSuite を調査してください。JSSECipherSuite の名前は、キューマネージャで使用される同等の CipherSpec とは異なることに注意してください。</p> <p>また、JSSE が正しくインストールされているか確認してください。</p>
2401	X'00000961'	<p><b>MQRC_SSL_CERTIFICATE_REVOKED</b></p> <p>SSL 暗号化を指定したキューマネージャへの接続が要求されました。しかし、キューマネージャによって示された証明書は、指定された CertStore の一つによって取り消されていました。</p> <p>この理由コードは、Java アプリケーションの場合だけ発生します。</p> <p>対処</p> <p>キューマネージャの識別に使用する証明書を調査してください。</p>
2402	X'00000962'	<p><b>MQRC_SSL_CERT_STORE_ERROR</b></p> <p>SSL 暗号化を指定したキューマネージャへの接続が要求されました。しかし、アプリケーションが提供する CertStore オブジェクトの中から、キューマネージャによって示される証明書を検索できませんでした。この理由コードのある MQException オブジェクトは、最初の CertStore の検索時に検出される Exception を参照します。これは MQException.getCause() メソッドを使用して取得できます。JMS では、MQException はスローされる JMSEException にリンクされます。</p> <p>この理由コードは、Java アプリケーションの場合だけ発生します。</p> <p>対処</p> <p>原因となった例外を調査して、エラーを確認してください。アプリケーションが提供する CertStore オブジェクトを調査してください。原因となった例外が java.lang.NoSuchElementException の場合、アプリケーションで CertStore オブジェクトの空のコレクションを指定していないか確認してください。</p>
6100	X'000017D4'	<p><b>MQRC_REOPEN_EXCL_INPUT_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>
6101	X'000017D5'	<p><b>MQRC_REOPEN_INQUIRE_ERROR</b></p> <p>この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。</p>

値 (10 進数)	値 (16 進数)	説明
6102	X'000017D6'	<b>MQRC_REOPEN_SAVED_CONTEXT_ERR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6103	X'000017D7'	<b>MQRC_REOPEN_TEMPORARY_Q_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6104	X'000017D8'	<b>MQRC_ATTRIBUTE_LOCKED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6105	X'000017D9'	<b>MQRC_CURSOR_NOT_VALID</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6106	X'000017DA'	<b>MQRC_ENCODING_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6107	X'000017DB'	<b>MQRC_STRUC_ID_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6108	X'000017DC'	<b>MQRC_NULL_POINTER</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6109	X'000017DD'	<b>MQRC_NO_CONNECTION_REFERENCE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6110	X'000017DE'	<b>MQRC_NO_BUFFER</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6111	X'000017DF'	<b>MQRC_BINARY_DATA_LENGTH_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6112	X'000017E0'	<b>MQRC_BUFFER_NOT_AUTOMATIC</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6113	X'000017E1'	<b>MQRC_INSUFFICIENT_BUFFER</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6114	X'000017E2'	<b>MQRC_INSUFFICIENT_DATA</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。

値 (10 進数)	値 (16 進数)	説明
6115	X'000017E3'	<b>MQRC_DATA_TRUNCATED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6116	X'000017E4'	<b>MQRC_ZERO_LENGTH</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6117	X'000017E5'	<b>MQRC_NEGATIVE_LENGTH</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6118	X'000017E6'	<b>MQRC_NEGATIVE_OFFSET</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6119	X'000017E7'	<b>MQRC_INCONSISTENT_FORMAT</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6120	X'000017E8'	<b>MQRC_INCONSISTENT_OBJECT_STATE</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6121	X'000017E9'	<b>MQRC_CONTEXT_OBJECT_NOT_VALID</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6122	X'000017EA'	<b>MQRC_CONTEXT_OPEN_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6123	X'000017EB'	<b>MQRC_STRUC_LENGTH_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6124	X'000017EC'	<b>MQRC_NOT_CONNECTED</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6125	X'000017ED'	<b>MQRC_NOT_OPEN</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6126	X'000017EE'	<b>MQRC_DISTRIBUTION_LIST_EMPTY</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6127	X'000017EF'	<b>MQRC_INCONSISTENT_OPEN_OPTIONS</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。

値 (10 進数)	値 (16 進数)	説明
6128	X'000017F0'	<b>MQRC_WRONG_VERSION</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。
6129	X'000017F1'	<b>MQRC_REFERENCE_ERROR</b> この値は TP1/Message Queue では返りません。詳細については、IBM MQ のマニュアルを参照してください。

## 付録 C マシンコード形式

---

メッセージデータの Encoding フィールドの構造について説明します。メッセージの構成およびメッセージデータの形式については、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。

Encoding フィールドは、次に示す四つのサブフィールドで構成される 32 ビット整数です。

- 2 進整数に必要なマシンコード形式
- 10 進パック形式整数に必要なマシンコード形式
- 浮動小数点数に必要なマシンコード形式
- 予備ビット

サブフィールドは、ビットマスクで識別されます。ビットマスクは、各サブフィールドに対応する位置のビットが 1 で、そのほかのビットは 0 です。最上位のビットは 0、最下位のビットは 31 のようにナンバリングされ、各マスクは次のように定義されています。

### MQENC\_INTEGER\_MASK

2 進整数マシンコード形式用のマスク

Encoding フィールド中のビット 28~31 を使用します。

### MQENC\_DECIMAL\_MASK

10 進パック形式整数のマシンコード形式用のマスク

Encoding フィールド中のビット 24~27 を使用します。

### MQENC\_FLOAT\_MASK

浮動小数点マシンコード形式用のマスク

Encoding フィールド中のビット 20~23 を使用します。

### MQENC\_RESERVED\_MASK

予約ビット用のマスク

Encoding フィールド中のビット 0~19 を使用します。

## 付録 C.1 2 進整数のマシンコード形式

2 進整数のマシンコード形式に有効なオプションを次に示します。

### MQENC\_INTEGER\_UNDEFINED

未定義整数のマシンコード形式

2 進整数は、未定義のマシンコード形式で表現されます。

### MQENC\_INTEGER\_NORMAL

通常の整数のマシンコード形式

2 進整数は、次に示す形式で表現されます。

- ある数の中にある最下位のバイトは、最も高位なアドレスを持ち、最上位のバイトは、最も低位なアドレスを持ちます。
- 各バイトの最下位のビットは、次の高位アドレスを持つバイトと隣接しています。
- 各バイトの最上位ビットは、次の低位アドレスを持つバイトと隣接しています。

#### MQENC\_INTEGER\_REVERSED

整数マシンコード形式の反転

2 進整数は、MQENC\_INTEGER\_NORMAL と同じ方法で表現されますが、バイトは逆の順序に配列されます。各バイト中のビットは、MQENC\_INTEGER\_NORMAL と同じ方法で配列されます。

## 付録 C.2 10 進パック形式整数のマシンコード形式

10 進パック形式整数のマシンコード形式に有効なオプションを次に示します。

#### MQENC\_DECIMAL\_UNDEFINED

未定義の 10 進パック形式のマシンコード形式

10 進パック形式整数は、未定義のマシンコード形式で表現されます。

#### MQENC\_DECIMAL\_NORMAL

通常の 10 進パック形式のマシンコード形式

10 進パック形式整数は、通常の方法で表現されます。

- 各 10 進数字は、X'0'~X'9'のうち、一つの 16 進数字を使って 10 進パック形式で表現されます。各 16 進数字は 4 ビットを使用します。10 進パック形式の各バイトは、二つの 10 進数字を表現します。
- 10 進パック形式の最下位のバイトは、最下位の 10 進数字を持つバイトです。その最下位のバイト中にある上位の 4 ビットに最下位の 10 進数字を持ち、下位 4 ビットに符号を持ちます。その符号は、X'C' (正符号)、X'D' (負符号)、X'F' (符号なし) のどれかです。
- 数字の最下位のバイトは高位アドレスを持ち、最上位のバイトは低位アドレスを持ちます。
- 各バイトの最下位のビットは、次の高位アドレスを持つバイトと隣接しています。バイトの最上位ビットは、次の低位アドレスを持つバイトと隣接しています。

#### MQENC\_DECIMAL\_REVERSED

10 進パック形式のマシンコード形式の反転

10 進パック形式整数は、MQENC\_DECIMAL\_NORMAL と同じ方法で表現されますが、バイト単位では反転した順序に配列されます。各バイト中のビットは、MQENC\_DECIMAL\_NORMAL と同じ方法で配置されます。

## 付録 C.3 浮動小数点のマシコード形式

浮動小数点のマシコード形式に有効なオプションを次に示します。

### MQENC\_FLOAT\_UNDEFINED

未定義の浮動小数点のマシコード形式

浮動小数点数は、未定義のマシコード形式で表現されます。

### MQENC\_FLOAT\_IEEE\_NORMAL

通常の IEEE フロートのマシコード形式

浮動小数点数は、標準の IEEE (The Institute of Electrical and Electronics Engineers : 米国電気電子学会) で使用している浮動小数点フォーマットで表現されます。バイト中の配列を次に示します。

- 仮数の最下位バイトは最も高位なアドレスを持ち、べき指数を含むバイトは最も低位なアドレスを持ちます。
- 各バイトの最下位のビットは、次の高位アドレスを持つバイトと隣接しています。
- 各バイトの最上位ビットは、次の低位アドレスを持つバイトと隣接しています。

### MQENC\_FLOAT\_IEEE\_REVERSED

IEEE フロートのマシコード形式の反転

浮動小数点数は、MQENC\_FLOAT\_IEEE\_NORMAL と同じ方法で表現されますが、バイト単位では、反転した順序に配列されます。各バイト中のビットは、MQENC\_FLOAT\_IEEE\_NORMAL と同じ方法で配置されます。

### MQENC\_FLOAT\_S390

System/390 アーキテクチャフロートのマシコード形式

浮動小数点数は、標準の System/390 浮動小数点フォーマットで表現されます。これは、System/370 でも使用されています。

## 付録 C.4 マシコード形式の作成と検証

Encoding フィールドの値を作成するには、要求されたマシコード形式を示す適切な定数値を、論理的 OR で結合します。論理的オペレーションをサポートしていない言語では、定数を加算できます。ただし、同じ定数は 2 回以上加算できません。

特定のマシコード形式について Encoding フィールドをテストするには、そのフィールドを論理的 AND で要求されるマシコード形式のマスクで結合してください。また、そのマシコード形式に有効な値と値が同じかどうか確認してください。マシコード形式に有効な値は、MQENC\_INTEGER\_\*, MQENC\_DECIMAL\_\*, または MQENC\_FLOAT\_\* です。

論理的オペレーションがサポートされていない言語では、次に示す方法で各マシコード形式化の値を確認できます。

1. 要求されたマシコード形式に対応する値を選択し、その値を"A"とします。

2 進整数の場合：1

10 進パック形式の場合：16

浮動小数点の場合：256

2. Encoding フィールドの値を"A"で割り、その結果を"B"とします。

3. "B"を 16 で割り、その結果を"C"とします。

4. "C"に 16 を掛けた値を"B"から引きます。その結果を"D"とします。

5. "D"に"A"を掛け、その結果を"E"とします。

6. "E"が要求されるマシンコード形式に有効な値と等しいか確認してください。

## 付録 C.5 マシンアーキテクチャのマシンコード形式の要約

マシンアーキテクチャのマシンコード形式の要約を次の表に示します。

表 C-1 マシンアーキテクチャのマシンコード形式の要約

マシンアーキテクチャ	2 進整数 マシンコード形式	10 進パック形式 マシンコード形式	浮動小数点 マシンコード形式
3500, 3050RX シリーズ	通常	通常	IEEE 通常
AS/400	通常	通常	IEEE 通常
EP8000	通常	通常	IEEE 通常
H9000, H9000V シリーズ	通常	通常	IEEE 通常
M シリーズ	通常	通常	IEEE 通常
PC/AT 互換機	反転	反転	IEEE 反転
PS/2	反転	反転	IEEE 反転
RS/6000	通常	通常	IEEE 通常
System/390	通常	通常	System/390

### 付録 D.1 報告オプション

MQPUT 命令および MQPUT1 命令で指定される、メッセージ記述子の Report フィールドの処理について説明します。Report フィールドの詳細については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」を参照してください。

Report フィールドは三つのサブフィールドに分けられた 32 ビット整数です。三つのサブフィールドが示す報告オプションを次に示します。

- ローカルキューマネージャが認識できない場合拒否される報告オプション
- ローカルキューマネージャが認識できなくても受け付けられる報告オプション
- ある条件が満たされたときに受け付けられる報告オプション

各サブフィールドはサブフィールドに対応する位置にビットが立ち、そのほかの位置にはビットが立っていないビットマスクで識別されます。サブフィールドの各ビットは隣接している必要はありません。ビットは、ビット 0 が最大重要ビットでありビット 31 が最小重要ビットです。次に示すマスクはサブフィールドを識別するために定義されています。

#### MQRO\_REJECT\_UNSUP\_MASK

未サポートの報告オプションの場合拒否されるオプションのマスク

ローカルキューマネージャにサポートされていないため、MQPUT 命令または MQPUT1 命令が、次に示すコードでエラーとなる報告オプションの Report フィールド中のビット位置を示します。

この場合に返される完了コードと理由コードを次に示します。

- 完了コード：MQCC\_FAILED
- 理由コード：MQRC\_REPORT\_OPTIONS\_ERROR

このサブフィールドはビット位置 3 と、ビット位置 11～13 を使用します。

#### MQRO\_ACCEPT\_UNSUP\_MASK

未サポートの報告オプションでも受け付けられるオプションのマスク

このマスクは報告フィールド内で、ローカルキューマネージャにサポートされていなくても、MQPUT 命令または MQPUT1 命令を受け付ける報告オプションのビット位置を示します。

この場合に返される完了コードと理由コードを次に示します。

- 完了コード：MQCC\_WARNING
- 理由コード：MQRC\_UNKNOWN\_REPORT\_OPTION

このサブフィールドは、ビット位置 0～2、ビット位置 4～10、ビット位置 24～31 を使用します。

#### MQRO\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK

ある条件のときだけ受け付けられる報告オプションのマスク

このマスクは報告フィールド内で、ローカルキューマネージャにサポートされていなくても、次に示す条件がすべて満たされたとき、MQPUT 命令または MQPUT1 命令を受け付ける報告オプションのビット位置を示します。

- メッセージが置かれる最初のキューが転送キューです。
- アプリケーションは直接転送キューに登録していません。

これらの条件が満たされた場合は、次に示す完了コードと理由コードが返されます。

- 完了コード：MQCC\_WARNING
- 理由コード：MQRC\_UNKNOWN\_REPORT\_OPTION

また、条件が満たされない場合は、次に示す完了コードと理由コードが返されます。

- 完了コード：MQCC\_FAILED
- 理由コード：MQRC\_REPORT\_OPTIONS\_ERROR

Report フィールドにキューマネージャが認識しないオプションを指定した場合、キューマネージャは、Report フィールドとサブフィールドのマスクをビットごとに AND 結合して、各サブフィールドを順番に確認します。AND 結合の結果が 0 でなかった場合、完了コードと理由コードは上記に示した値が返されます。

MQCC\_WARNING が返される場合、ほかの警告条件があるときはどんな理由コードが返却されるかは定義されていません。

ローカルキューマネージャで認識できない報告オプションの指定および受け付けができれば、リモートキューマネージャで認識され処置される報告オプションを持つメッセージを送信するときに便利です。

## 付録 D.2 メッセージフラグ

MsgFlags フィールドは三つのサブフィールドに分けられた 32 ビット整数です。三つのサブフィールドが示すメッセージフラグを次に示します。

- ローカルキューマネージャが認識できない場合、拒否されるメッセージフラグ
- ローカルキューマネージャが認識できなくても受け付けられるメッセージフラグ
- ある条件が満たされたときに受け付けられるメッセージフラグ

### 注意

MsgFlags フィールドのすべてのサブフィールドは、キューマネージャが使用するために予約されています。

各サブフィールドはサブフィールドに対応する位置にビットが立ち、そのほかの位置にはビットが立っていないビットマスクで識別されます。ビットは、ビット 0 が最大重要ビットでありビット 31 が最小重要ビットです。次に示すマスクはサブフィールドを識別するために定義されています。

## MQMF\_REJECT\_UNSUP\_MASK

未サポートのメッセージフラグの場合、拒否されるメッセージフラグのマスク

ローカルキューマネージャにサポートされていないため、MQPUT 命令または MQPUT1 命令が、次に示すコードでエラーとなるメッセージフラグフィールドの MsgFlags フィールドの中のビット位置を示します。

この場合に返される完了コードと理由コードを次に示します。

- 完了コード：MQCC\_FAILED
- 理由コード：MQRC\_MSG\_FLAGS\_ERROR

このサブフィールドは、ビット位置 20~31 を使用します。

なお、このサブフィールドには、次のメッセージフラグがあります。

- MQMF\_LAST\_MSG\_IN\_GROUP
- MQMF\_LAST\_SEGMENT
- MQMF\_MSG\_IN\_GROUP
- MQMF\_SEGMENT
- MQMF\_SEGMENTATION\_ALLOWED

## MQMF\_ACCEPT\_UNSUP\_MASK

未サポートのメッセージフラグでも受け付けられるメッセージフラグのマスク

このマスクは、メッセージフラグがローカルキューマネージャによってサポートされていなくても、MQPUT 命令または MQPUT1 命令で受け付ける MsgFlags フィールドのビット位置を示します。この場合、完了コードは MQCC\_OK です。

このサブフィールドは、ビット位置 0~11 を使用します。

## MQMF\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK

ある条件のときだけ受け付けられる、未サポートメッセージフラグのマスク

このマスクは、メッセージフラグがローカルキューマネージャによってサポートされていなくても、メッセージフラグを MQPUT 命令または MQPUT1 命令で受け付ける MsgFlags フィールドのビット位置を示します。ただし、次の条件を満たす場合だけです。

- メッセージのあて先が、リモートキューマネージャです。
- アプリケーションが直接ローカルの転送キューにメッセージを登録していません。つまり、MQOPEN 命令または MQPUT1 命令の呼び出しによって指定されたオブジェクト記述子の ObjectQMGrName フィールドおよび ObjectName フィールドによって識別されるキューが、ローカルの転送キューではありません。

これらの条件が二つとも満たされた場合は、次に示す完了コードが返されます。

- 完了コード：MQCC\_OK

また、条件が満たされない場合は、次に示す完了コードと理由コードが返されます。

- 完了コード：MQCC\_FAILED

- 理由コード：MQRC\_MSG\_FLAGS\_ERROR

このサブフィールドは、ビット位置 12~19 を使用します。

MsgFlags フィールドにキューマネージャが認識しないオプションを指定した場合、キューマネージャは MsgFlags フィールドとサブフィールドのマスクをビットごとに AND 結合して、各サブフィールドを順番に確認します。AND 結合の結果が 0 でなかった場合は、完了コードと理由コードは上記に示した値が返されます。

## 付録 E 各プログラミング言語での実現例

C 言語および COBOL 言語を使用する場合の、メッセージキューイング機能の呼び出し、引数、およびデータタイプの宣言の方法について説明します。

### 付録 E.1 C 言語での実現例

#### (1) C 言語での注意事項

##### (a) ヘッダファイル

ヘッダファイルは、メッセージキューイング機能で使用する定数などを定義し、C 言語で記述されたアプリケーションプログラムでメッセージキューイング機能を使用できるようにします。ヘッダファイルを次に示します。

- CMQC : MQI のプロトタイプ宣言、データタイプ、および定数を定義します。

なお、ヘッダファイルのファイル名を次に示すように、`#include` 指令で指定するとアプリケーションの移植性が良くなります。

```
#include "cmqc.h"
```

##### (b) 関数

入力専用、MQHCONN 型、MQHOBJ 型、または MQLONG 型の引数は値渡しです。そのほかの引数はアドレス渡しです。

関数が呼び出される時、アドレス渡しの引数を常に指定する必要はありません。特定の引数が必要でない場合、引数データにあるアドレスの関数の呼び出しでは、引数のアドレスとしてヌルを指定できます。引数のアドレスとしてヌルを指定できる引数については、「2. メッセージキューイング機能の命令」の各命令の説明を参照してください。

C 言語では、関数の戻り値はありません。すべて void 型で宣言されています。

##### (c) データタイプとして宣言されない引数

MQGET、MQPUT、および MQPUT1 関数には、データタイプとして宣言されない引数やバッファ引数があります。この引数はアプリケーションメッセージデータの受け渡しをするときに使用します。

データタイプとして宣言されない引数は、MQBYTE 型の配列として宣言します。この方法で引数を宣言するほかに、メッセージのデータ構造を示す構造体として宣言することもできます。実際の関数の引数は、void 型のポインタとして宣言されているため、どんなデータのアドレスも関数呼び出しの引数として指定できます。

## (d) データタイプ

すべてのデータタイプは、C 言語の記憶域クラス指定子 typedef で定義されます。また、各データタイプのポインタ型のデータタイプも定義されます。ポインタ型のデータタイプ名は、ポインタを意味する"P"を先頭に付けた基本データタイプ名または構造体データタイプ名になります。ポインタの属性は MQPOINTER 変数で定義します。次にポインタデータタイプの宣言の方法を示します。

```
#define MQPOINTER *          /* *を設定します          */
. . .
typedef MQLONG MQPOINTER PMQLONG; /* MQLONG型へのポインタ */
typedef MQMD MQPOINTER PMQMD; /* MQMD型へのポインタ */
```

## (e) ビット列の操作

ビット列は、MQBYTEn 型の一つとして宣言されます。この型のフィールドが複写、比較、または設定される場合は、C 言語の関数 (memcpy, memcmp, memset) を使用してください。次に使用例を示します。

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

MQMD MyMsgDesc;

memcpy(MyMsgDesc.MsgId, /* ヌルをMsgIdフィールドに設定します */
       MQMI_NONE,      /* 定数を使用して設定します          */
       sizeof(MyMsgDesc.MsgId));

memset(MyMsgDesc.CorrelId,
       /* ヌルをCorrelIdフィールドに設定します */
       0x00,          /* 定数を使用しないで設定します          */
       sizeof(MQBYTE24));
```

文字列を操作する関数 (strcpy, strcmp, strncpy, strncmp) は、MQBYTEn 型で宣言されたデータには使用できません。

## (f) 文字列の操作

キューマネージャがアプリケーションに文字データを返す場合、そのデータ長がフィールド長に足りないときは余りを空白で埋めます。また、キューマネージャはヌルで終わる文字列を返しません。そのため、文字列を操作する関数 (strncpy, strncmp, strncat) を使用してください。

ヌルで終わる文字列を操作する関数 (strcpy, strcmp, strcat) は使用しないでください。フィールド長を調べるときは、sizeof 演算子を使用してください。文字列の長さを決める strlen 関数は使用しないでください。

## (g) 構造体の初期値

ヘッダファイル (CMQC) は、MQ の構造体に初期値を設定するとき使用するマクロ変数を定義します。これらのマクロ変数は、MQXXX\_DEFAULT という形式の名称です。MQXXX は構造体の名称を示します。次にマクロ変数の使用例を示します。

```
MQMD MyMsgDesc = {MQMD_DEFAULT};
MQPMO MyPutOpts = {MQPMO_DEFAULT};
```

MQI が定義した値には、文字列のフィールドに設定できる値もあります。この値は次に示す二つの方法で定義されています。

- 文字列の長さと一致するマクロ変数として定義します。

定義されたフィールド長と同じ長さの文字列として定義します。なお、この長さにはヌルは含みません。MQMD 構造体の Format フィールドには、次のマクロ変数を定義します。

```
#define MQFMT_STRING "MQSTR△△△" (△△△は空白を示します)
```

このマクロを使用するときは、memcpy 関数および memcmp 関数で操作してください。

- 文字の配列として定義します。

このマクロ変数の名称の最後には、"\_ARRAY"が付きます。次にマクロ変数の定義例を示します。

```
#define MQFMT_STRING_ARRAY 'M','Q','S','T','R',' ',' ',' '
```

C++の場合、MQMD\_DEFAULT マクロ変数で初期値が設定されています。この初期値と異なる値を使用したい場合は、この形式のマクロを使用してください。

## (h) 動的構造体の初期値

必要な構造体の個数が決まっていない場合、calloc 関数または malloc 関数を使用して、動的にメモリを確保します。このようなフィールドを初期化する方法を次に示します。

1. 構造体を初期化するため、MQXXX\_DEFAULT マクロ変数を使用して構造体の変数を宣言します。この変数はほかの変数の初期値になります。

```
MQMD Model = {MQMD_DEFAULT}; /* 初期値の変数の定義 */
```

静的変数か動的変数かを定めるために、static または auto を指定できます。

2. calloc 関数または malloc 関数を使用して、構造体の動的変数用のメモリを確保します。

```
PMQMD Instance;
Instance = malloc(sizeof(MQMD));
/* 動的変数用のメモリの取得 */
```

3. memcpy 関数を使用して、初期値の変数を動的変数に複写します。

```
memcpy(Instance, &Model, sizeof(MQMD)); /* 動的変数の初期化 */
```

## (i) C++での使用

C++では次に示すステートメントが、ヘッダファイルに記述されています。

```
#ifdef_cplusplus
extern "C"{
#endif
.
.
.
#ifdef_cplusplus
}
#endif
```

## (j) リンケージ方法

HP-UX, Windows, およびLinuxの場合, リンケージのコマンドに引数 (-lmqa) を指定してください。

AIXの場合, リンケージのコマンドに引数 (-lmqa2) を指定してください。

一つのロードモジュールにC言語およびCOBOL言語を混在できません。

## (2) 関数

### (a) MQCLOSE 命令 - オブジェクトのクローズ

```
MQCLOSE(Hconn, &Hobj, Options, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQHOBJ Hobj; /* オブジェクトハンドル */
MQLONG Options; /* クローズオプション */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード
```

### (b) MQCONN 命令 - キューマネージャへの接続

```
MQCONN(Name, &Hconn, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQCHAR48 Name; /* キューマネージャ名 */
MQHCONN Hconn; /* コネクションハンドル */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード
```

### (c) MQDISC 命令 - キューマネージャからの切り離し

```
MQDISC(&Hconn, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード */
```

### (d) MQGET 命令 - メッセージの取り出し

```
MQGET(Hconn, Hobj, &MsgDesc, &GetMsgOpts, BufferLength, Buffer,
&DataLength, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQHOBJ Hobj; /* オブジェクトハンドル */
MQMD MsgDesc; /* メッセージ記述子 */
MQGMO GetMsgOpts; /* メッセージ取り出しオプション */
MQLONG BufferLength; /* バッファ長 */
MQBYTE Buffer[n]; /* バッファ */
MQLONG DataLength; /* データ長 */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード */
```

### (e) MQINQ 命令 - オブジェクトの属性の照会

```
MQINQ(Hconn, Hobj, SelectorCount, Selectors, IntAttrCount,
IntAttrs, CharAttrLength, CharAttrs, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQHOBJ Hobj; /* オブジェクトハンドル */
MQLONG SelectorCount; /* セレクタ数 */
MQLONG Selectors[n]; /* セレクタ */
MQLONG IntAttrCount; /* 整数型属性数 */
MQLONG IntAttrs[n]; /* 整数型属性値 */
MQLONG CharAttrLength; /* 文字型属性長 */
MQCHAR CharAttrs[n]; /* 文字型属性値 */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード */
```

### (f) MQOPEN 命令 - オブジェクトのオープン

```
MQOPEN(Hconn, &ObjDesc, Options, &Hobj, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQOD ObjDesc; /* オブジェクト記述子 */
MQLONG Options; /* オープンオプション */
MQHOBJ Hobj; /* オブジェクトハンドル */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード
```

## (g) MQPUT 命令 - メッセージの登録

```
MQPUT(Hconn, Hobj, &MsgDesc, &PutMsgOpts, BufferLength, Buffer,
&CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQHOBJ Hobj; /* オブジェクトハンドル */
MQMD MsgDesc; /* メッセージ記述子 */
MQPMO PutMsgOpts; /* メッセージ登録オプション */
MQLONG BufferLength; /* バッファ長 */
MQBYTE Buffer[n]; /* バッファ */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード
```

## (h) MQPUT1 命令 - 1 メッセージの登録

```
MQPUT1(Hconn, &ObjDesc, &MsgDesc, &PutMsgOpts, BufferLength,
Buffer, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQOD ObjDesc; /* オブジェクト記述子 */
MQMD MsgDesc; /* メッセージ記述子 */
MQPMO PutMsgOpts; /* メッセージ登録オプション */
MQLONG BufferLength; /* バッファ長 */
MQBYTE Buffer[n]; /* バッファ */
MQLONG CompCode; /* 完了コード */
MQLONG Reason; /* 理由コード
```

## (i) MQSET 命令 - オブジェクトの属性の設定

```
MQSET(Hconn, Hobj, SelectorCount, Selectors, IntAttrCount,
IntAttrs, CharAttrLength, CharAttrs, &CompCode, &Reason);
```

引数の宣言方法を次に示します。

```
MQHCONN Hconn; /* コネクションハンドル */
MQHOBJ Hobj; /* オブジェクトハンドル
```

MQLONG	SelectorCount;	/* セレクタ数	*/
MQLONG	Selectors[n];	/* セレクタ	*/
MQLONG	IntAttrCount;	/* 整数型属性数	*/
MQLONG	IntAttrs[n];	/* 整数型属性値	*/
MQLONG	CharAttrLength;	/* 文字型属性長	*/
MQCHAR	CharAttrs[n];	/* 文字型属性値	*/
MQLONG	CompCode;	/* 完了コード	*/
MQLONG	Reason;	/* 理由コード	*/

### (3) 基本データタイプ

基本データタイプの宣言を、表 E-1 に示します。

表 E-1 基本データタイプの宣言 (C 言語)

データタイプ	表記
MQBYTE 型	typedef unsigned char MQBYTE;
MQBYTE8 型	typedef MQBYTE MQBYTE8[8];
MQBYTE16 型	typedef MQBYTE MQBYTE16[16];
MQBYTE24 型	typedef MQBYTE MQBYTE24[24];
MQBYTE32 型	typedef MQBYTE MQBYTE32[32];
MQBYTE40 型	typedef MQBYTE MQBYTE40[40];
MQBYTE128 型	typedef MQBYTE MQBYTE128[128];
MQCHAR 型	typedef char MQCHAR;
MQCHAR4 型	typedef MQCHAR MQCHAR4[4];
MQCHAR8 型	typedef MQCHAR MQCHAR8[8];
MQCHAR12 型	typedef MQCHAR MQCHAR12[12];
MQCHAR20 型	typedef MQCHAR MQCHAR20[20];
MQCHAR28 型	typedef MQCHAR MQCHAR28[28];
MQCHAR32 型	typedef MQCHAR MQCHAR32[32];
MQCHAR48 型	typedef MQCHAR MQCHAR48[48];
MQCHAR64 型	typedef MQCHAR MQCHAR64[64];
MQCHAR128 型	typedef MQCHAR MQCHAR128[128];
MQCHAR256 型	typedef MQCHAR MQCHAR256[256];
MQHCONN 型	typedef MQLONG MQHCONN;
MQHOBJ 型	typedef MQLONG MQHOBJ;
MQLONG 型	LP64 データ型モデルの場合 typedef int MQLONG;

データタイプ	表記
MQLONG 型	その他の場合 typedef long MQLONG;
MQPTR 型	typedef void MQPOINTER MQPTR;
MQULONG 型	LP64 データ型モデルの場合 typedef unsigned int MQULONG; その他の場合 typedef unsigned long MQULONG;
PMQBYTE 型	typedef MQBYTE MQPOINTER PMQBYTE;
PMQBYTE8 型	typedef MQBYTE8[8] MQPOINTER PMQBYTE8[8];
PMQBYTE16 型	typedef MQBYTE16[16] MQPOINTER PMQBYTE16[16];
PMQBYTE24 型	typedef MQBYTE24[24] MQPOINTER PMQBYTE24[24];
PMQBYTE32 型	typedef MQBYTE32[32] MQPOINTER PMQBYTE32[32];
PMQBYTE40 型	typedef MQBYTE40[40] MQPOINTER PMQBYTE40[40];
PMQBYTE128 型	typedef MQBYTE128[128] MQPOINTER PMQBYTE128[128];
PMQCHAR 型	typedef MQCHAR MQPOINTER PMQCHAR;
PMQCHAR4 型	typedef MQCHAR4[4] MQPOINTER PMQCHAR4[4];
PMQCHAR8 型	typedef MQCHAR8[8] MQPOINTER PMQCHAR8[8];
PMQCHAR12 型	typedef MQCHAR12[12] MQPOINTER PMQCHAR12[12];
PMQCHAR20 型	typedef MQCHAR20[20] MQPOINTER PMQCHAR20[20];
PMQCHAR28 型	typedef MQCHAR28[28] MQPOINTER PMQCHAR28[28];
PMQCHAR32 型	typedef MQCHAR32[32] MQPOINTER PMQCHAR32[32];
PMQCHAR48 型	typedef MQCHAR48[48] MQPOINTER PMQCHAR48[48];
PMQCHAR64 型	typedef MQCHAR64[64] MQPOINTER PMQCHAR64[64];
PMQCHAR128 型	typedef MQCHAR128[128] MQPOINTER PMQCHAR128[128];
PMQCHAR256 型	typedef MQCHAR256[256] MQPOINTER PMQCHAR256[256];
PMQCHAR264 型	typedef MQCHAR264[264] MQPOINTER PMQCHAR264[264];
PMQDLH 型	typedef MQDLH MQPOINTER PMQDLH;
PMQGMO 型	typedef MQGMO MQPOINTER PMQGMO;
PMQHCONN 型	typedef MQHCONN MQPOINTER PMQHCONN;
PMQHOBJ 型	typedef MQHOBJ MQPOINTER PMQHOBJ;
PMQLONG 型	typedef MQLONG MQPOINTER PMQLONG;
PMQMD 型	typedef MQMD MQPOINTER PMQMD;

データタイプ	表記
PMQMD1 型	typedef MQMD1[1] MQPOINTER PMQMD1[1];
PMQMDE 型	typedef MQMDE MQPOINTER PMQMDE;
PMQOD 型	typedef MQOD MQPOINTER PMQOD;
PMQPMO 型	typedef MQPMO MQPOINTER PMQPMO;
PMQPTR 型	typedef MQPTR MQPOINTER PMQPTR;
PMQRMH 型	typedef MQRMH MQPOINTER PMQRMH;
PMQTM 型	typedef MQTM MQPOINTER PMQTM;
PMQULONG 型	typedef MQULONG MQPOINTER PMQULONG;
PMQVOID 型	typedef void MQPOINTER PMQVOID;
PMQXQH 型	typedef MQXQH MQPOINTER PMQXQH;
PPMQBYTE 型	typedef PMQBYTE MQPOINTER PPMQBYTE;
PPMQCHAR 型	typedef PMQCHAR MQPOINTER PPMQCHAR;
PPMQGMO 型	typedef PMQGMO MQPOINTER PPMQGMO;
PPMQHCONN 型	typedef PMQHCONN MQPOINTER PPMQHCONN;
PPMQHOBJ 型	typedef PMQHOBJ MQPOINTER PPMQHOBJ;
PPMQLONG 型	typedef PMLONG MQPOINTER PPMQLONG;
PPMQMD 型	typedef PMQMD MQPOINTER PPMQMD;
PPMQOD 型	typedef PMQOD MQPOINTER PPMQOD;
PPMQPMO 型	typedef PMQPMO MQPOINTER PPMQPMO;
PPMQULONG 型	typedef PMQULONG MQPOINTER PPMQULONG;
PPMQVOID 型	typedef PMQVOID MQPOINTER PPMQVOID;

## (4) 構造体データタイプ

### (a) MQCFH 構造体 – PCF ヘッダ

```
typedef struct tagMQCFH {
    MQLONG  Type;           /* 構造体タイプ          */
    MQLONG  StrucLength;   /* 構造体長              */
    MQLONG  Version;      /* 構造体バージョン番号  */
    MQLONG  Command;      /* 命令識別子            */
    MQLONG  MsgSeqNumber; /* メッセージシーケンス番号 */
    MQLONG  Control;      /* 制御オプション        */
    MQLONG  CompCode;     /* 完了コード            */
    MQLONG  Reason;       /* 理由コード            */
    MQLONG  ParameterCount; /* 引数構造体の数        */
} MQCFH;
```

## (b) MQCFIL 構造体 — PCF 整数リスト引数

```
typedef struct tagMQCFIL {
    MQLONG  Type;          /* 構造体タイプ */
    MQLONG  StrucLength;   /* 構造体長 */
    MQLONG  Parameter;    /* 引数識別子 */
    MQLONG  Count;        /* 引数値の数 */
    MQLONG  Values[1];    /* 引数の値 */
} MQCFIL;
```

## (c) MQCFIN 構造体 — PCF 整数引数

```
typedef struct tagMQCFIN {
    MQLONG  Type;          /* 構造体タイプ */
    MQLONG  StrucLength;   /* 構造体長 */
    MQLONG  Parameter;    /* 引数識別子 */
    MQLONG  Value;        /* 引数の値 */
} MQCFIN;
```

## (d) MQCFSL 構造体 — PCF 文字列リスト引数

```
typedef struct tagMQCFSL {
    MQLONG  Type;          /* 構造体タイプ */
    MQLONG  StrucLength;   /* 構造体長 */
    MQLONG  Parameter;    /* 引数識別子 */
    MQLONG  CodedCharSetId; /* 文字セット識別子 */
    MQLONG  Count;        /* 引数の値の数 */
    MQLONG  StringLength;  /* 文字列長 */
    MQCHAR  Strings;      /* 文字列の値 */
} MQCFSL;
```

## (e) MQCFST 構造体 — PCF スtring引数

```
typedef struct tagMQCFST {
    MQLONG  Type;          /* 構造体タイプ */
    MQLONG  StrucLength;   /* 構造体長 */
    MQLONG  Parameter;    /* 引数識別子 */
    MQLONG  CodedCharSetId; /* 文字セット識別子 */
    MQLONG  StringLength;  /* 文字列長 */
    MQCHAR  String[1];    /* 文字列の値 */
} MQCFST;
```

## (f) MQDHF 構造体 — 配布ヘッダ

```
typedef struct tagMQDHF {
    MQCHAR4  StrucId;      /* 構造体識別子 */
    MQLONG  Version;      /* 構造体バージョン番号 */
    MQLONG  StrucLength;   /* MQDHF構造体とそれに続く
    MQOR構造体とMQPMR構造体を足した長さ */
    MQLONG  Encoding;     /* MQOR構造体およびMQPMR構造体
    データのマシンコード形式 */
    MQLONG  CodedCharSetId; /* MQOR構造体およびMQPMR構造体
    に続く */
}
```

```

        データの文字セット識別子          */
MQCHAR8  Format;          /* MQOR構造体およびMQPMR構造体続く          */
        データのフォーマット名          */
MQLONG   Flags;          /* フラグ          */
MQLONG   PutMsgRecFields; /* MQPMR構造体フィールドの存在          */
MQLONG   RecsPresent;    /* MQOR構造体の数          */
MQLONG   ObjectRecOffset; /* 最初のMQOR構造体までのオフセット          */
MQLONG   PutMsgRecOffset; /* 最初のMQPMR構造体までのオフセット          */
} MQDH;

```

## (g) MQDLH 構造体 – デッドレターヘッダ

```

typedef struct tagMQDLH {
    MQCHAR4  StrucId;          /* 構造体識別子          */
    MQLONG   Version;         /* 構造体バージョン番号          */
    MQLONG   Reason;          /* メッセージ通知理由          */

    MQCHAR48 DestQName;       /* あて先キュー名          */
    MQCHAR48 DestQMGrName;    /* あて先キューマネージャ名          */
    MQLONG   Encoding;        /* MQDLH構造体続く          */
        データのマシンコード形式          */
    MQLONG   CodedCharSetId; /* MQDLH構造体続く          */
        データの文字セット識別子          */
    MQCHAR8  Format;          /* MQDLH構造体続く          */
        データのフォーマット名          */
    MQLONG   PutApplType;     /* 登録アプリケーションタイプ          */
    MQCHAR28 PutApplName;     /* 登録アプリケーション名          */
    MQCHAR8  PutDate;         /* 登録日付          */
    MQCHAR8  PutTime;        /* 登録時刻          */
} MQDLH;

```

## (h) MQGMO 構造体 – メッセージ取り出しオプション

```

typedef struct tagMQGMO {
    MQCHAR4  StrucId;          /* 構造体識別子          */
    MQLONG   Version;         /* 構造体バージョン番号          */
    MQLONG   Options;         /* 取り出しオプション          */
    MQLONG   WaitInterval;    /* 待ち合わせ最大時間          */
    MQLONG   Signal1;         /* シグナル          */
    MQLONG   Signal2;         /* 予備          */
    MQCHAR48 ResolvedQName;    /* 受信キュー名          */
    MQLONG   MatchOptions;     /* 一致オプション          */
    MQCHAR   GroupStatus;      /* メッセージグループフラグ          */
    MQCHAR   SegmentStatus;    /* 論理メッセージフラグ          */
    MQCHAR   Segmentation;     /* セグメント分割フラグ          */
    MQCHAR   Reserved1;        /* 予備          */
} MQGMO;

```

## (i) MQMD 構造体 – メッセージ記述子

```

typedef struct tagMQMD {
    MQCHAR4  StrucId;          /* 構造体識別子          */
    MQLONG   Version;         /* 構造体バージョン番号          */
    MQLONG   Report;          /* 報告メッセージオプション          */
} MQMD;

```

```

MQLONG  MsgType;          /* メッセージタイプ          */
MQLONG  Expiry;          /* メッセージ保持時間        */
MQLONG  Feedback;       /* 報告メッセージ返答コード */
MQLONG  Encoding;       /* メッセージデータの
                          マシンコード形式 */
MQLONG  CodedCharSetId; /* メッセージデータの
                          文字セット識別子 */
MQCHAR8  Format;         /* メッセージデータの
                          フォーマット名 */
MQLONG  Priority;       /* メッセージ優先度          */
MQLONG  Persistence;   /* メッセージ永続性          */
MQBYTE24 MsgId;        /* メッセージ識別子          */
MQBYTE24 CorrelId;     /* 相関識別子                */
MQLONG  BackoutCount;  /* ロールバック回数         */
MQCHAR48 ReplyToQ;     /* 応答キュー名              */
MQCHAR48 ReplyToQMgr; /* 応答キューマネージャ名   */
MQCHAR12 UserIdentifier; /* ユーザ識別子              */
MQBYTE32 AccountingToken; /* 課金トークン              */
MQCHAR32 ApplIdentityData; /* アプリケーション識別データ */
MQLONG  PutApplType;   /* 登録アプリケーションタイプ */
MQCHAR28 PutApplName; /* 登録アプリケーション名    */
MQCHAR8  PutDate;     /* 登録日付                  */
MQCHAR8  PutTime;     /* 登録時刻                  */
MQCHAR4  ApplOriginData; /* 登録元データ              */
MQBYTE24 GroupId;     /* グループ識別子            */
MQLONG  MsgSeqNumber; /* メッセージシーケンス番号  */
MQLONG  Offset;       /* オフセット                 */
MQLONG  MsgFlags;     /* メッセージフラグ          */
MQLONG  OriginalLength; /* 登録元メッセージ長        */
} MQMD;

```

## (j) MQMDE 構造体 — メッセージ記述子の拡張

```

typedef struct tagMQMDE {
MQCHAR4  StrucId;       /* 構造体識別子              */
MQLONG  Version;       /* 構造体バージョン番号      */
MQLONG  StrucLength    /* 構造体長                  */
MQLONG  Encoding;     /* MQMDE構造体に行くデータの
                          マシンコード形式 */
MQLONG  CodedCharSetId; /* MQMDE構造体に行くデータの
                          文字セット識別子 */
MQCHAR8  Format;       /* MQMDE構造体に行くデータの
                          フォーマット名 */
MQLONG  Flags;        /* フラグ                     */
MQBYTE24 GroupId;     /* グループ識別子            */
MQLONG  MsgSeqNumber; /* メッセージシーケンス番号  */
MQLONG  Offset;       /* オフセット                 */
MQLONG  MsgFlags;     /* メッセージフラグ          */
MQLONG  OriginalLength; /* 登録元メッセージ長        */
} MQMDE;

```

## (k) MQOD 構造体 — オブジェクト記述子

```

typedef struct tagMQOD {
MQCHAR4  StrucId;       /* 構造体識別子              */
MQLONG  Version;       /* 構造体バージョン番号      */

```

```

MQLONG  ObjectType;          /* オブジェクトタイプ          */
MQCHAR48 ObjectName;        /* オブジェクト名              */
MQCHAR48 ObjectQMgrName;    /* オブジェクトキューマネージャ名 */
MQCHAR48 DynamicQName;     /* 動的キュー名                */
MQCHAR12 AlternateUserId;   /* 代替ユーザ識別子           */
MQLONG  RecsPresent;        /* オブジェクトレコード数      */
MQLONG  KnownDestCount;     /* ローカルキューの数          */
MQLONG  UnknownDestCount;   /* リモートキューの数          */
MQLONG  InvalidDestCount;   /* オープンに失敗したキューの数 */
MQLONG  ObjectRecOffset;    /* 最初のオブジェクトレコードまでの
                               オフセット                    */
MQLONG  ResponseRecOffset; /* 最初の応答レコードまでのオフセット */
MQPTR   ObjectRecPtr;       /* 最初のオブジェクトレコードのアドレス */
MQPTR   ResponseRecPtr;     /* 最初の応答レコードのアドレス */
MQBYTE40 AlternateSecurityId; /* 代替セキュリティ識別子      */
MQCHAR48 ResolvedQName;     /* 解決したキュー名称          */
MQCHAR48 ResolvedQMgrName; /* 解決したキューマネージャ名称 */
} MQOD;

```

## (l) MQOR 構造体 - オブジェクトレコード

```

typedef struct tagMQOR {
    MQCHAR48 ObjectName;          /* オブジェクトキュー名          */
    MQCHAR48 ObjectQMgrName;     /* オブジェクトキューマネージャ名 */
} MQOR;

```

## (m) MQPMO 構造体 - メッセージ登録オプション

```

typedef struct tagMQPMO {
    MQCHAR4  StrucId;           /* 構造体識別子                  */
    MQLONG  Version;           /* 構造体バージョン番号          */
    MQLONG  Options;           /* 登録オプション                */
    MQLONG  Timeout;           /* 予備                          */
    MQHOBJ  Context;           /* コンテキスト                  */
    MQLONG  KnownDestCount;     /* ローカルキューの数            */
    MQLONG  UnknownDestCount;   /* リモートキューの数            */
    MQLONG  InvalidDestCount;   /* 送信に失敗したキューの数      */
    MQCHAR48 ResolvedQName;     /* 受信キュー名                  */
    MQCHAR48 ResolvedQMgrName; /* 受信キューマネージャ名        */
    MQLONG  RecsPresent;        /* 登録メッセージレコード数
                               または応答レコード数 */
    MQLONG  PutMsgRecFields;    /* MQPMR構造体フィールドの存在  */
    MQLONG  PutMsgRecOffset;    /* 最初の登録メッセージレコードまでの
                               オフセット                    */
    MQLONG  ResponseRecOffset; /* 最初の応答レコードまでの
                               オフセット                    */
    MQLONG  PutMsgRecPtr;       /* 最初の登録メッセージレコードの
                               アドレス                        */
    MQLONG  ResponseRecPtr;     /* 最初の応答レコードのアドレス */
} MQPMO;

```

## (n) MQPMR 構造体 - 登録メッセージレコード

```
typedef struct tagMQPMR {
    MQBYTE24  MsgId;           /* メッセージ識別子 */
    MQBYTE24  CorrelId;        /* 相関識別子 */
    MQBYTE24  GroupId;         /* グループ識別子 */
    MQLONG    Feedback;        /* 報告メッセージ返答コード */
    MQBYTE32  AccountingToken; /* 課金トークン */
} MQPMR;
```

## (o) MQRMH 構造体 - メッセージ参照ヘッダ

```
typedef struct tagMQRMH {
    MQCHAR4   StrucId;         /* 構造体識別子 */
    MQLONG    Version;         /* 構造体バージョン番号 */
    MQLONG    StrucLength;     /* 構造体長 */
    MQLONG    Encoding;        /* 長大データのマシンコード形式 */
    MQLONG    CodedCharSetId;  /* 長大データの文字セット識別子 */
    MQCHAR8   Format;          /* 長大データのフォーマット名 */
    MQLONG    Flags;           /* フラグ */
    MQCHAR8   ObjectType;      /* オブジェクトタイプ */
    MQBYTE24  ObjectInstanceId; /* オブジェクトインスタンス識別子 */
    MQLONG    SrcEnvLength     /* 送信元環境データ長 */
    MQLONG    SrcEnvOffset     /* 送信元環境データオフセット */
    MQLONG    SrcNameLength    /* 送信元オブジェクト名長 */
    MQLONG    SrcNameOffset    /* 送信元オブジェクトのオフセット */
    MQLONG    DestEnvLength    /* あて先環境データ長 */
    MQLONG    DestEnvOffset    /* あて先環境データオフセット */
    MQLONG    DestNameLength   /* あて先オブジェクト名長 */
    MQLONG    DestNameOffset   /* あて先オブジェクトのオフセット */
    MQLONG    DataLogicalLength /* 長大データ長 */
    MQLONG    DataLogicalOffset /* 長大データ低位オフセット */
    MQLONG    DataLogicalOffset2 /* 長大データ高位オフセット */
} MQRMH;
```

## (p) MQRR 構造体 - 応答レコード

```
typedef struct tagMQRR {
    MQLONG    CompCode; /* 完了コード */
    MQLONG    Reason;   /* 理由コード */
} MQRR;
```

## (q) MQTM 構造体 - トリガメッセージ

```
typedef struct tagMQTM {
    MQCHAR4   StrucId; /* 構造体識別子 */
    MQLONG    Version; /* 構造体バージョン番号 */
    MQCHAR48  QName;   /* イベント発生キュー名 */
    MQCHAR48  ProcessName; /* プロセス名 */
    MQCHAR64  TriggerData; /* トリガデータ */
    MQLONG    ApplType; /* アプリケーションタイプ */
    MQCHAR256 ApplId;   /* アプリケーション識別子 */
    MQCHAR128 EnvData;  /* 環境データ */
}
```

```
MQCHAR128 UserData; /* ユーザデータ */
} MQTM;
```

## (r) MQXQH 構造体 — 転送キューヘッダ

```
typedef struct tagMQXQH {
MQCHAR4 StrucId; /* 構造体識別子 */
MQLONG Version; /* 構造体バージョン番号 */
MQCHAR48 RemoteQName; /* リモートキュー名 */
MQCHAR48 RemoteQMgrName; /* リモートキューマネージャ名 */
MQMD1 MsgDesc; /* 登録元メッセージ記述子 */
} MQXQH;
```

## 付録 E.2 COBOL 言語での実現例

### (1) COBOL 言語での注意事項

#### (a) コピーファイル

コピーファイルは、メッセージキューイング機能を定義し、COBOL 言語で記述されたアプリケーションプログラムでメッセージキューイング機能を使用できるようにします。コピーファイルには定数を含む二つのファイルおよび各構造体に二つのファイルがあります。

各構造体には、二つの形式があります。初期値がある形式と初期値がない形式です。

- 初期値がある構造体は作業場所節で使用でき、"V"が最後に付く名称のコピーファイルに含まれています。
- 初期値がない構造体は連絡節で使用でき、"L"が最後に付く名称のコピーファイルに含まれています。

表 E-2 にコピーファイルの一覧を示します。

表 E-2 コピーファイルの一覧

ファイル名		内容
初期値がある形式	初期値がない形式	
CMQCFHV	CMQCFHL	PCF ヘッダの構造体
CMQCFILV	CMQCFILL	PCF 整数リスト引数の構造体
CMQCFINV	CMQCFINL	PCF 整数引数の構造体
CMQCFSLV	CMQCFSSL	PCF 文字列リスト引数の構造体
CMQCFSTV	CMQCFSTL	PCF スtring引数の構造体
CMQCFV	—	PCF の定数
CMQDHSV	CMQDHL	配布ヘッダの構造体

ファイル名		内容
初期値がある形式	初期値がない形式	
CMQDLHV	CMQDLHL	デッドレターヘッダの構造体
CMQGMOV	CMQGMOL	取り出し動作オプションの構造体
CMQMD1V	CMQMD1L	バージョン 1 のメッセージ記述子の構造体
CMQMDEV	CMQMDEL	メッセージ記述子拡張の構造体
CMQMDV	CMQMDL	メッセージ記述子の構造体
CMQODV	CMQODL	オブジェクト記述子の構造体
CMQORV	CMQORL	オブジェクトレコードの構造体
CMQPMOV	CMQPMOL	登録動作オプションの構造体
CMQRMHV	CMQRMHL	メッセージ参照ヘッダの構造体
CMQRRV	CMQRRL	応答レコードの構造体
CMQTMV	CMQTML	トリガメッセージの構造体
CMQXQHV	CMQXQHL	転送キューヘッダの構造体
CMQV	—	MQI の定数

(凡例)

—：該当しません。

## (b) データ構造

各構造の宣言をレベル番号 10 の項目で始めます。このようにすることで、データ項目をレベル番号 01 の宣言でコーディングしたあと、残りのデータ項目をコピー文で複写して、幾つかのインスタンスを宣言できます。インスタンスを参照するときは、IN を指定できます。

```
* Declare two instances of MQMD
01 MY_MQMD.
   COPY CMQMDV.
01 MY_OTHER_MQMD.
   COPY CMQMDV.
*
* Set MSGTYPE field in MY_OTHER_MQMD
MOVE MQMT_REQUEST TO MQMD_MSGTYPE IN MY_OTHER_MQMD.
```

4バイト単位でバウンダリ調整してください。レベル番号 01 でない項目に続く構造体にコピー文を使用している場合、レベル番号 01 の項目の始まりから、4バイトの倍数に位置するように構造体を宣言してください。

「1. データタイプ」の「[データタイプの概要](#)」で示す、構造体のフィールド名には接頭語がありません。COBOL 言語では、フィールド名は構造体の名称にハイフンが付けられた形式で表現されます。MSGTYPE という名称を COBOL 言語で表記すると、MQMD-MSGTYPE になります。

V が最後に付く名称の構造体では、初期値が設定されています。そのため、初期値と異なる値を使用したい場合はフィールドを初期化してください。

## (c) 定数

- このマニュアルでは、定数にアンダスコア ( \_ ) が使用されていますが、COBOL 言語では、アンダスコアの代わりにハイフン ( - ) を使用してください。
- 文字列値を含む定数では、文字列のデリミタとしてシングルコーテーションを使用しています。そのため、環境によってはコンパイラオプションを指定して、コンパイラが文字列のデリミタとしてシングルコーテーションを受け取るようにする必要があります。
- 定数は、コピーファイルにレベル番号 10 の項目として宣言されています。定数を使用するには、レベル番号 01 の項目として定義してから、定義の宣言でコピー文を使用します。

```
*構造体を大域データとして定義します
01 MY_MQ_CONSTANTS.
   COPY CMQV.
```

この方法で定数が定義された場合、定数がプログラム中の領域を占有します。プログラムが幾つかの部分に分かれた場合は、定数が多重に複写されます。定数が多重に複写されるとメモリが必要以上に使用されるため、次に示す方法を使用してください。

- グローバル句をレベル番号 01 で指定してください。

```
* 定数を大域データとして宣言します
01 MY_MQ_CONSTANTS GLOBAL.
   COPY CMQV.
```

このようにすると、定数が一つの領域で割り当てられ、それぞれのプログラムで参照できます。なお、グローバル句は一部の環境ではサポートされていません。

- プログラムから参照するデータだけを選んで複写してください。すべての定数を複写することはありません。

## (d) リンケージ方法

HP-UX, Windows, および Linux の場合、リンケージのコマンドに引数 (-lmqacb) を指定してください。

AIX の場合、リンケージのコマンドに引数 (-lmqacb2) を指定してください。

一つのロードモジュールに C 言語および COBOL 言語を混在できません。

## (2) 呼ぶプログラム

### (a) MQCLOSE 命令 - オブジェクトのクローズ

PROCEDURE DIVISION の指定

```
CALL 'MQCLOSE' USING HCONN, HOBJ, OPTIONS, COMPCODE, REASON.
```

## DATA DIVISION の指定

```
** コネクションハンドル
01 HCONN    PIC S9(9) BINARY.
** オブジェクトハンドル
01 HOBJ     PIC S9(9) BINARY.
** クローズオプション
01 OPTIONS  PIC S9(9) BINARY.
** 完了コード
01 COMPCODE PIC S9(9) BINARY.
** 理由コード
01 REASON   PIC S9(9) BINARY.
```

## (b) MQCONN 命令 - キューマネージャへの接続

### PROCEDURE DIVISION の指定

```
CALL 'MQCONN' USING NAME, HCONN, COMPCODE, REASON.
```

### DATA DIVISION の指定

```
** キューマネージャ名
01 NAME     PIC X(48).
** コネクションハンドル
01 HCONN    PIC S9(9) BINARY.
** 完了コード
01 COMPCODE PIC S9(9) BINARY.
** 理由コード
01 REASON   PIC S9(9) BINARY.
```

## (c) MQDISC 命令 - キューマネージャからの切り離し

### PROCEDURE DIVISION の指定

```
CALL 'MQDISC' USING HCONN, COMPCODE, REASON.
```

### DATA DIVISION の指定

```
** コネクションハンドル
01 HCONN    PIC S9(9) BINARY.
** 完了コード
01 COMPCODE PIC S9(9) BINARY.
** 理由コード
01 REASON   PIC S9(9) BINARY.
```

## (d) MQGET 命令 - メッセージの取り出し

### PROCEDURE DIVISION の指定

```
CALL 'MQGET' USING
HCONN, HOBJ, MSGDESC, GETMSGOPTS, BUFFERLENGTH,
BUFFER, DATALENGTH, COMPCODE, REASON.
```

## DATA DIVISION の指定

```
** コネクションハンドル
01 HCONN          PIC S9(9) BINARY.
** オブジェクトハンドル
01 HOBJ           PIC S9(9) BINARY.
** メッセージ記述子
01 MSGDESC.
   COPY CMQMDV.
** メッセージ取り出しオプション
01 GETMSGOPTS.
   COPY CMQGMV.
** バッファ長
01 BUFFERLENGTH PIC S9(9) BINARY.
** バッファ
01 BUFFER          PIC X(n).
** データ長
01 DATALENGTH    PIC S9(9) BINARY.
** 完了コード
01 COMPCODE        PIC S9(9) BINARY.
** 理由コード
01 REASON          PIC S9(9) BINARY.
```

## (e) MQINQ 命令 – オブジェクトの属性の照会

### PROCEDURE DIVISION の指定

```
CALL 'MQINQ' USING
HCONN, HOBJ, SELECTORCOUNT, SELECTORS, INTATTRCOUNT,
INTATTRS, CHARATTRLENGTH, CHARATTRS, COMPCODE,
REASON.
```

### DATA DIVISION の指定

```
** コネクションハンドル
01 HCONN          PIC S9(9) BINARY.
** オブジェクトハンドル
01 HOBJ           PIC S9(9) BINARY.
** セレクタ数
01 SELECTORCOUNT PIC S9(9) BINARY.
** セレクタ
01 SELECTORS.
   02 SELECTORS    PIC S9(9) BINARY OCCURS n TIMES.
** 整数型属性数
01 INTATTRCOUNT PIC S9(9) BINARY.
** 整数型属性値
01 INTATTRS.
   02 INTATTRS    PIC S9(9) BINARY OCCURS n TIMES.
** 文字型属性長
01 CHARATTRLENGTH PIC S9(9) BINARY.
** 文字型属性値
01 CHARATTRS      PIC X(n).
** 完了コード
01 COMPCODE        PIC S9(9) BINARY.
** 理由コード
01 REASON          PIC S9(9) BINARY.
```

## (f) MQOPEN 命令 - オブジェクトのオープン

PROCEDURE DIVISION の指定

```
CALL 'MQOPEN' USING  
HCONN, OBJDESC, OPTIONS, HOBJ, COMPCODE, REASON.
```

DATA DIVISION の指定

```
** コネクションハンドル  
01 HCONN PIC S9(9) BINARY.  
** オブジェクト記述子  
01 OBJDESC.  
COPY CMQODV.  
** オープンオプション  
01 OPTIONS PIC S9(9) BINARY.  
** オブジェクトハンドル  
01 HOBJ PIC S9(9) BINARY.  
** 完了コード  
01 COMPCODE PIC S9(9) BINARY.  
** 理由コード  
01 REASON PIC S9(9) BINARY.
```

## (g) MQPUT 命令 - メッセージの登録

PROCEDURE DIVISION の指定

```
CALL 'MQPUT' USING  
HCONN, HOBJ, MSGDESC, PUTMSGOPTS, BUFFERLENGTH,  
BUFFER, COMPCODE, REASON.
```

DATA DIVISION の指定

```
** コネクションハンドル  
01 HCONN PIC S9(9) BINARY.  
** オブジェクトハンドル  
01 HOBJ PIC S9(9) BINARY.  
** メッセージ記述子  
01 MSGDESC.  
COPY CMQMDV.  
** メッセージ登録オプション  
01 PUTMSGOPTS.  
COPY CMQPMOV.  
** バッファ長  
01 BUFFERLENGTH PIC S9(9) BINARY.  
** バッファ  
01 BUFFER PIC X(n).  
** 完了コード  
01 COMPCODE PIC S9(9) BINARY.  
** 理由コード  
01 REASON PIC S9(9) BINARY.
```

## (h) MQPUT1 命令 – 1 メッセージの登録

PROCEDURE DIVISION の指定

```
CALL 'MQPUT1' USING  
HCONN, OBJDESC, MSGDESC, PUTMSGOPTS, BUFFERLENGTH,  
BUFFER, COMPCODE, REASON.
```

DATA DIVISION の指定

```
** コネクションハンドル  
01 HCONN          PIC S9(9) BINARY.  
** オブジェクト記述子  
01 OBJDESC.  
   COPY CMQODV.  
** メッセージ記述子  
01 MSGDESC.  
   COPY CMQMDV.  
** メッセージ登録オプション  
01 PUTMSGOPTS.  
   COPY CMQPMOV.  
** バッファ長  
01 BUFFERLENGTH PIC S9(9) BINARY.  
** バッファ  
01 BUFFER        PIC X(n).  
** 完了コード  
01 COMPCODE      PIC S9(9) BINARY.  
** 理由コード  
01 REASON        PIC S9(9) BINARY.
```

## (i) MQSET 命令 – オブジェクトの属性の設定

PROCEDURE DIVISION の指定

```
CALL 'MQSET' USING  
HCONN, HOBJ, SELECTORCOUNT, SELECTORS, INTATTRCOUNT,  
INTATTRS, CHARATTRLENGTH, CHARATTRS, COMPCODE, REASON.
```

DATA DIVISION の指定

```
** コネクションハンドル  
01 HCONN          PIC S9(9) BINARY.  
** オブジェクトハンドル  
01 HOBJ          PIC S9(9) BINARY.  
** セレクタ数  
01 SELECTORCOUNT PIC S9(9) BINARY.  
** セレクタ  
01 SELECTORS.  
   02 SELECTORS  PIC S9(9) BINARY OCCURS n TIMES.  
** 整数型属性数  
01 INTATTRCOUNT PIC S9(9) BINARY.  
** 整数型属性値  
01 INTATTRS.  
   02 INTATTRS  PIC S9(9) BINARY OCCURS n TIMES.  
** 文字型属性長  
01 CHARATTRLENGTH PIC S9(9) BINARY.
```

```

** 文字型属性値
01 CHARATTRS      PIC X(n).
** 完了コード
01 COMPCODE       PIC S9(9) BINARY.
** 理由コード
01 REASON         PIC S9(9) BINARY.

```

### (3) 基本データタイプ

基本データタイプの宣言を次の表に示します。

表 E-3 基本データタイプの宣言 (COBOL 言語)

データタイプ	表記
MQBYTE 型	PIC X
MQBYTE8 型	PIC X(8)
MQBYTE16 型	PIC X(16)
MQBYTE24 型	PIC X(24)
MQBYTE32 型	PIC X(32)
MQBYTE40 型	PIC X(40)
MQCHAR 型	PIC X
MQCHAR4 型	PIC X(4)
MQCHAR8 型	PIC X(8)
MQCHAR12 型	PIC X(12)
MQCHAR20 型	PIC X(20)
MQCHAR28 型	PIC X(28)
MQCHAR32 型	PIC X(32)
MQCHAR48 型	PIC X(48)
MQCHAR64 型	PIC X(64)
MQCHAR128 型	PIC X(128)
MQCHAR256 型	PIC X(256)
MQHCONN 型	PIC S9(9) BINARY
MQHOBJ 型	PIC S9(9) BINARY
MQLONG 型	PIC S9(9) BINARY
MQPTR	POINTER

## (4) 構造体データタイプ

### (a) MQCFH 構造体 – PCF ヘッダ

```
** MQCFH構造体
 10 MQCFH.
** 構造体タイプ
 15 MQCFH-TYPE          PIC S9(9) BINARY.
** 構造体長
 15 MQCFH-STRUCLength PIC S9(9) BINARY.
** 構造体バージョン番号
 15 MQCFH-VERSION      PIC S9(9) BINARY.
** 命令識別子
 15 MQCFH-COMMAND      PIC S9(9) BINARY.
** メッセージシーケンス番号
 15 MQCFH-MSGSEQNUMBER PIC S9(9) BINARY.
** 制御オプション
 15 MQCFH-CONTROL      PIC S9(9) BINARY.
** 完了コード
 15 MQCFH-COMPCODE     PIC S9(9) BINARY.
** 理由コード
 15 MQCFH-REASON       PIC S9(9) BINARY.
** 引数構造体の数
 15 MQCFH-PARAMETERCOUNT PIC S9(9) BINARY.
```

### (b) MQCFIL 構造体 – PCF 整数リスト引数

```
** MQCFIL構造体
 10 MQCFIL.
** 構造体タイプ
 15 MQCFIL-TYPE          PIC S9(9) BINARY.
** 構造体長
 15 MQCFIL-STRUCLength PIC S9(9) BINARY.
** 引数識別子
 15 MQCFIL-PARAMETER     PIC S9(9) BINARY.
** 引数の値の数
 15 MQCFIL-COUNT        PIC S9(9) BINARY.
```

### (c) MQCFIN 構造体 – PCF 整数引数

```
** MQCFIN構造体
 10 MQCFIN.
** 構造体タイプ
 15 MQCFIN-TYPE          PIC S9(9) BINARY.
** 構造体長
 15 MQCFIN-STRUCLength PIC S9(9) BINARY.
** 引数識別子
 15 MQCFIN-PARAMETER     PIC S9(9) BINARY.
** 引数の値
 15 MQCFIN-VALUE        PIC S9(9) BINARY.
```

## (d) MQCFSL 構造体 – PCF 文字列リスト引数

```
** MQCFSL構造体
 10 MQCFSL.
** 構造体タイプ
 15 MQCFSL-TYPE          PIC S9(9) BINARY.
** 構造体長
 15 MQCFSL-STRUCLength  PIC S9(9) BINARY.
** 引数識別子
 15 MQCFSL-PARAMETER    PIC S9(9) BINARY.
** 文字セット識別子
 15 MQCFSL-CODEDCHARSETID PIC S9(9) BINARY.
** 引数の値の数
 15 MQCFSL-COUNT        PIC S9(9) BINARY.
** 文字列長
 15 MQCFSL-STRINGLENGTH PIC S9(9) BINARY.
```

## (e) MQCFST 構造体 – PCF スtring引数

```
** MQCFST構造体
 10 MQCFST.
** 構造体タイプ
 15 MQCFST-TYPE          PIC S9(9) BINARY.
** 構造体長
 15 MQCFST-STRUCLength  PIC S9(9) BINARY.
** 引数識別子
 15 MQCFST-PARAMETER    PIC S9(9) BINARY.
** 文字コード識別子
 15 MQCFST-CODEDCHARSETID PIC S9(9) BINARY.
** 文字列長
 15 MQCFST-STRINGLENGTH PIC S9(9) BINARY.
```

## (f) MQDHF 構造体 – 配布ヘッダ

```
** MQDHF構造体
 10 MQDHF.
** 構造体識別子
 15 MQDHF-STRUCID       PIC X(4).
** 構造体バージョン番号
 15 MQDHF-VERSION       PIC S9(9) BINARY.
** MQDHF構造体とそれに続くMQOR構造体とMQPMR構造体を足した長さ
 15 MQDHF-STRUCLength  PIC S9(9) BINARY.
** MQOR構造体およびMQPMR構造体に続くデータのマシンコード形式
 15 MQDHF-ENCODING      PIC S9(9) BINARY.
** MQOR構造体およびMQPMR構造体に続くデータの文字セット識別子
 15 MQDHF-CODEDCHARSETID PIC S9(9) BINARY.
** MQOR構造体およびMQPMR構造体に続くデータのフォーマット名
 15 MQDHF-FORMAT        PIC X(8).
** フラグ
 15 MQDHF-FLAGS         PIC S9(9) BINARY.
** MQPMR構造体フィールドの存在
 15 MQDHF-PUTMSGRECFIELDS PIC S9(9) BINARY.
** MQOR構造体の数
 15 MQDHF-RECSPRESENT   PIC S9(9) BINARY.
** 最初のMQOR構造体までのオフセット
```

```
15 MQDH-OBJECTRECOFFSET PIC S9(9) BINARY.  
** 最初のMQPMR構造体までのオフセット  
15 MQDH-PUTMSGRECOFFSET PIC S9(9) BINARY.
```

## (g) MQDLH 構造体 – デッドレターヘッダ

```
** MQDLH構造体  
10 MQDLH.  
** 構造体識別子  
15 MQDLH-STRUCID PIC X(4).  
** 構造体バージョン番号  
15 MQDLH-VERSION PIC S9(9) BINARY.  
** メッセージ通知理由  
15 MQDLH-REASON PIC S9(9) BINARY.  
** あて先キュー名  
15 MQDLH-DESTQNAME PIC X(48).  
** あて先キューマネージャ名  
15 MQDLH-DESTQMGRNAME PIC X(48).  
** MQDLH構造体に続くデータのマシンコード形式  
15 MQDLH-ENCODING PIC S9(9) BINARY.  
** MQDLH構造体に続くデータの文字セット識別子  
15 MQDLH-CODEDCHARSETID PIC S9(9) BINARY.  
** MQDLH構造体に続くデータのフォーマット名  
15 MQDLH-FORMAT PIC X(8).  
** 登録アプリケーションタイプ  
15 MQDLH-PUTAPPLTYPE PIC S9(9) BINARY.  
** 登録アプリケーション名  
15 MQDLH-PUTAPPLNAME PIC X(28).  
** 登録日付  
15 MQDLH-PUTDATE PIC X(8).  
** 登録時刻  
15 MQDLH-PUTTIME PIC X(8).
```

## (h) MQGMO 構造体 – メッセージ取り出しオプション

```
** MQGMO構造体  
10 MQGMO.  
** 構造体識別子  
15 MQGMO-STRUCID PIC X(4).  
** 構造体バージョン番号  
15 MQGMO-VERSION PIC S9(9) BINARY.  
** 取り出しオプション  
15 MQGMO-OPTIONS PIC S9(9) BINARY.  
** 待ち合わせ最大時間  
15 MQGMO-WAITINTERVAL PIC S9(9) BINARY.  
** シグナル  
15 MQGMO-SIGNAL1 PIC S9(9) BINARY.  
** 予備  
15 MQGMO-SIGNAL2 PIC S9(9) BINARY.  
** 受信キュー名  
15 MQGMO-RESOLVEDQNAME PIC X(48).  
** 一致オプション  
15 MQGMO-MATCHOPTIONS PIC S9(9) BINARY.  
** メッセージグループフラグ  
15 MQGMO-GROUPSTATUS PIC X.  
** 論理メッセージフラグ
```

```

15 MQGMO-SEGMENTSTATUS PIC X.
** セグメント分割フラグ
15 MQGMO-SEGMENTATION PIC X.
** 予備
15 MQGMO-RESERVED1 PIC X.

```

## (i) MQMD 構造体 — メッセージ記述子

```

** MQMD構造体
10 MQMD.
** 構造体識別子
15 MQMD-STRUCID PIC X(4).
** 構造体バージョン番号
15 MQMD-VERSION PIC S9(9) BINARY.
** 報告メッセージオプション
15 MQMD-REPORT PIC S9(9) BINARY.
** メッセージタイプ
15 MQMD-MSGTYPE PIC S9(9) BINARY.
** メッセージ保持時間
15 MQMD-EXPIRY PIC S9(9) BINARY.
** 報告メッセージ返答コード
15 MQMD-FEEDBACK PIC S9(9) BINARY.
** メッセージデータのマシンコード形式
15 MQMD-ENCODING PIC S9(9) BINARY.
** メッセージデータの文字セット識別子
15 MQMD-CODEDCHARSETID PIC S9(9) BINARY.
** メッセージデータのフォーマット名
15 MQMD-FORMAT PIC X(8).
** メッセージ優先度
15 MQMD-PRIORITY PIC S9(9) BINARY.
** メッセージ永続性
15 MQMD-PERSISTENCE PIC S9(9) BINARY.
** メッセージ識別子
15 MQMD-MSGID PIC X(24).
** 相関識別子
15 MQMD-CORRELID PIC X(24).
** ロールバック回数
15 MQMD-BACKOUTCOUNT PIC S9(9) BINARY.
** 応答キュー名
15 MQMD-REPLYTOQ PIC X(48).
** 応答キューマネージャ名
15 MQMD-REPLYTOQMGR PIC X(48).
** ユーザ識別子
15 MQMD-USERIDENTIFIER PIC X(12).
** 課金トークン
15 MQMD-ACCOUNTINGTOKEN PIC X(32).
** アプリケーション識別データ
15 MQMD-APPLIDENTITYDATA PIC X(32).
** 登録アプリケーションタイプ
15 MQMD-PUTAPPLTYPE PIC S9(9) BINARY.
** 登録アプリケーション名
15 MQMD-PUTAPPLNAME PIC X(28).
** 登録日付
15 MQMD-PUTDATE PIC X(8).
** 登録時刻
15 MQMD-PUTTIME PIC X(8).
** 登録元データ

```

```

15 MQMD-APPLORIGINDATA PIC X(4).
** グループ識別子
15 MQMD-GROUPID PIC X(24).
** メッセージシーケンス番号
15 MQMD-MSGSEQNUMBER PIC S9(9) BINARY.
** オフセット
15 MQMD-OFFSET PIC S9(9) BINARY.
** メッセージフラグ
15 MQMD-MSGFLAGS PIC S9(9) BINARY.
** 登録元メッセージ長
15 MQMD-ORIGINLENGTH PIC S9(9) BINARY.

```

## (j) MQMDE 構造体 - メッセージ記述子の拡張

```

** MQMDE構造体
10 MQMDE.
** 構造体識別子
15 MQMDE-STRUCID PIC X(4).
** 構造体バージョン番号
15 MQMDE-VERSION PIC S9(9) BINARY.
** 構造体長
15 MQMDE-STRUCLength PIC S9(9) BINARY.
** MQMDE構造体続くデータのマシンコード形式
15 MQMDE-ENCODING PIC S9(9) BINARY.
** MQMDE構造体続くデータの文字セット識別子
15 MQMDE-CODEDCHARSETID PIC S9(9) BINARY.
** MQMDE構造体続くデータのフォーマット名
15 MQMDE-FORMAT PIC X(8).
** フラグ
15 MQMDE-FLAGS PIC S9(9) BINARY.
** グループ識別子
15 MQMDE-GROUPID PIC X(24).
** メッセージシーケンス番号
15 MQMDE-MSGSEQNUMBER PIC S9(9) BINARY.
** オフセット
15 MQMDE-OFFSET PIC S9(9) BINARY.
** メッセージフラグ
15 MQMDE-MSGFLAGS PIC S9(9) BINARY.
** 登録元メッセージ長
15 MQMDE-ORIGINLENGTH PIC S9(9) BINARY.

```

## (k) MQOD 構造体 - オブジェクト記述子

```

** MQOD構造体
10 MQOD.
** 構造体識別子
15 MQOD-STRUCID PIC X(4).
** 構造体バージョン番号
15 MQOD-VERSION PIC S9(9) BINARY.
** オブジェクトタイプ
15 MQOD-OBJECTTYPE PIC S9(9) BINARY.
** オブジェクト名
15 MQOD-OBJECTNAME PIC X(48).
** オブジェクトキューマネージャ名
15 MQOD-OBJECTQMGRNAME PIC X(48).
** 動的キュー名

```

```

15 MQOD-DYNAMICQNAME      PIC X(48).
** 代替ユーザ識別子
15 MQOD-ALTERNATEUSERID   PIC X(12).
** オブジェクトレコード数
15 MQOD-RECSPRESENT       PIC S9(9) BINARY.
** ローカルキューの数
15 MQOD-KNOWNDDESTCOUNT  PIC S9(9) BINARY.
** リモートキューの数
15 MQOD-UNKNOWNDDESTCOUNT PIC S9(9) BINARY.
** オープンに失敗したキューの数
15 MQOD-INVALIDDESTCOUNT PIC S9(9) BINARY.
** 最初のオブジェクトレコードまでのオフセット
15 MQOD-OBJECTRECOFFSET   PIC S9(9) BINARY.
** 最初の応答レコードまでのオフセット
15 MQOD-RESPONSERECOFFSET PIC S9(9) BINARY.
** 最初のオブジェクトレコードのアドレス
15 MQOD-OBJECTRECPTTR     POINTER.
** 最初の応答レコードのアドレス
15 MQOD-RESPONSERECPTTR   POINTER.
** 代替セキュリティ識別子
15 MQOD-ALTERNATESECURITYID PIC X(40).
** 解決したキュー名称
15 MQOD-RESOLVEDQNAME     PIC X(48).
** 解決したキューマネージャ名称
15 MQOD-RESOLVEDQMGRNAME  PIC X(48).

```

## (l) MQOR 構造体 - オブジェクトレコード

```

** MQOR構造体
10 MQOR.
** オブジェクト名
15 MQOR-OBJECTNAME       PIC X(48).
** オブジェクトキューマネージャ名
15 MQOR-OBJECTQMGRNAME   PIC X(48).

```

## (m) MQPMO 構造体 - メッセージ登録オプション

```

** MQPMO構造体
10 MQPMO.
** 構造体識別子
15 MQPMO-STRUCID         PIC X(4).
** 構造体バージョン番号
15 MQPMO-VERSION        PIC S9(9) BINARY.
** 登録オプション
15 MQPMO-OPTIONS        PIC S9(9) BINARY.
** 予備
15 MQPMO-TIMEOUT        PIC S9(9) BINARY.
** コンテキスト
15 MQPMO-CONTEXT        PIC S9(9) BINARY.
** ローカルキューの数
15 MQPMO-KNOWNDDESTCOUNT PIC S9(9) BINARY.
** リモートキューの数
15 MQPMO-UNKNOWNDDESTCOUNT PIC S9(9) BINARY.
** 送信に失敗したキューの数
15 MQPMO-INVALIDDESTCOUNT PIC S9(9) BINARY.
** 受信キュー名

```

```

15 MQPMO-RESOLVEDQNAME      PIC X(48).
** 受信キューマネージャ名
15 MQPMO-RESOLVEDQMGRNAME   PIC X(48).
** 登録メッセージレコード数または応答レコード数
15 MQPMO-RECSPRESENT        PIC S9(9) BINARY.
** MQPMR構造体フィールドの存在
15 MQPMO-PUTMSGRECFIELDS    PIC S9(9) BINARY.
** 最初の登録メッセージレコードまでのオフセット
15 MQPMO-PUTMSGRECOFFSET    PIC S9(9) BINARY.
** 最初の応答レコードまでのオフセット
15 MQPMO-RESPONSERECOFFSET  PIC S9(9) BINARY.
** 最初の登録メッセージレコードのアドレス
15 MQPMO-PUTMSGRECPTTR      POINTER.
** 最初の応答レコードのアドレス
15 MQPMO-RESPONSERECPTR     POINTER.

```

## (n) MQPMR 構造体 - 登録メッセージレコード

```

** MQPMR構造体
10 MQPMR.
** メッセージ識別子
15 MQPMR-MSGID              PIC X(24).
** 相関識別子
15 MQPMR-CORRELID           PIC X(24).
** グループ識別子
15 MQPMR-GROUPID            PIC X(24).
** 報告メッセージ返答コード
15 MQPMR-FEEDBACK           PIC S(9) BINARY.
** 課金トークン
15 MQPMR-ACCOUNTINGTOKEN    PIC X(32).

```

## (o) MQRMH 構造体 - メッセージ参照ヘッダ

```

** MQRMH構造体
10 MQRMH.
** 構造体識別子
15 MQRMH-STRUCID            PIC X(4).
** 構造体バージョン番号
15 MQRMH-VERSION            PIC S9(9) BINARY.
** 構造体長
15 MQRMH-STRUCLength        PIC S9(9) BINARY.
** 長大データのマシンコード形式
15 MQRMH-ENCODING           PIC S9(9) BINARY.
** 長大データの文字セット識別子
15 MQRMH-CODEDCHARSETID     PIC S9(9) BINARY.
** 長大データのフォーマット名
15 MQRMH-FORMAT             PIC X(8).
** フラグ
15 MQRMH-FLAGS              PIC S9(9) BINARY.
** オブジェクトタイプ
15 MQRMH-OBJECTTYPE         PIC X(8).
** オブジェクトインスタンス識別子
15 MQRMH-OBJECTINSTANCEID   PIC X(24).
** 送信元環境データ長
15 MQRMH-SRCENVLENGTH        PIC S9(9) BINARY.
** 送信元環境データオフセット

```

```

15 MQRMH-SRCENVOFFSET      PIC S9(9) BINARY.
** 送信元オブジェクト名長
15 MQRMH-SRCNAMELENGTH    PIC S9(9) BINARY.
** 送信元オブジェクトのオフセット
15 MQRMH-SRCNAMEOFFSET    PIC S9(9) BINARY.
** あて先環境データ長
15 MQRMH-DESTENVLENGTH    PIC S9(9) BINARY.
** あて先環境データオフセット
15 MQRMH-DESTENVOFFSET    PIC S9(9) BINARY.
** あて先オブジェクト名長
15 MQRMH-DESTNAMELENGTH   PIC S9(9) BINARY.
** あて先オブジェクトのオフセット
15 MQRMH-DESTNAMEOFFSET   PIC S9(9) BINARY.
** 長大データ長
15 MQRMH-DATALOGICALENGTH PIC S9(9) BINARY.
** 長大データ低位オフセット
15 MQRMH-DATALOGICALOFFSET PIC S9(9) BINARY.
** 長大データ高位オフセット
15 MQRMH-DATALOGICALOFFSET2 PIC S9(9) BINARY.

```

## (p) MQRR 構造体 - 応答レコード

```

** MQRR構造体
10 MQRR.
** 完了コード
15 MQRR-COMPCODE          PIC S9(9) BINARY.
** 理由コード
15 MQRR-REASON            PIC S9(9) BINARY.

```

## (q) MQTM 構造体 - トリガメッセージ

```

** MQTM構造体
10 MQTM.
** 構造体識別子
15 MQTM-STRUCID          PIC X(4).
** 構造体バージョン番号
15 MQTM-VERSION          PIC S9(9) BINARY.
** イベント発生キュー名
15 MQTM-QNAME            PIC X(48).
** プロセス名
15 MQTM-PROCESSNAME     PIC X(48).
** トリガデータ
15 MQTM-TRIGGERDATA      PIC X(64).
** アプリケーションタイプ
15 MQTM-APPLTYPE         PIC S9(9) BINARY.
** アプリケーション識別子
15 MQTM-APPLID           PIC X(256).
** 環境データ
15 MQTM-ENVDATA          PIC X(128).
** ユーザデータ
15 MQTM-USERDATA         PIC X(128).

```

## (r) MQXQH 構造体 - キューヘッダ

**	MQXQH構造体	
10	MQXQH.	
**	構造体識別子	
15	MQXQH-STRUCID	PIC X(4).
**	構造体バージョン番号	
15	MQXQH-VERSION	PIC S9(9) BINARY.
**	リモートキュー名	
15	MQXQH-REMOTEQNAME	PIC X(48).
**	リモートキューマネージャ名	
15	MQXQH-REMOTEQMGRNAME	PIC X(48).
**	登録元メッセージ記述子	
15	MQXQH-MSGDESC.	
**	構造体識別子	
20	MQXQH-MSGDESC-STRUCID	PIC X(4).
**	構造体バージョン番号	
20	MQXQH-MSGDESC-VERSION	PIC S9(9) BINARY.
**	報告オプション	
20	MQXQH-MSGDESC-REPORT	PIC S9(9) BINARY.
**	メッセージタイプ	
20	MQXQH-MSGDESC-MSGTYPE	PIC S9(9) BINARY.
**	メッセージ保持時間	
20	MQXQH-MSGDESC-EXPIRY	PIC S9(9) BINARY.
**	報告メッセージ返答コードまたは理由コード	
20	MQXQH-MSGDESC-FEEDBACK	PIC S9(9) BINARY.
**	メッセージデータのマシンコード形式	
20	MQXQH-MSGDESC-ENCODING	PIC S9(9) BINARY.
**	メッセージデータの文字セット識別子	
20	MQXQH-MSGDESC-CODEDCHARSETID	PIC S9(9) BINARY.
**	メッセージデータのフォーマット名	
20	MQXQH-MSGDESC-FORMAT	PIC X(8).
**	メッセージ優先度	
20	MQXQH-MSGDESC-PRIORITY	PIC S9(9) BINARY.
**	メッセージ永続性	
20	MQXQH-MSGDESC-PERSISTENCE	PIC S9(9) BINARY.
**	メッセージ識別子	
20	MQXQH-MSGDESC-MSGID	PIC X(24).
**	関連識別子	
20	MQXQH-MSGDESC-CORRELID	PIC X(24).
**	ロールバック回数	
20	MQXQH-MSGDESC-BACKOUTCOUNT	PIC S9(9) BINARY.
**	応答キュー名	
20	MQXQH-MSGDESC-REPLYTOQ	PIC X(48).
**	応答キューマネージャ名	
20	MQXQH-MSGDESC-REPLYTOQMGR	PIC X(48).
**	ユーザ識別子	
20	MQXQH-MSGDESC-USERIDENTIFIER	PIC X(12).
**	課金トークン	
20	MQXQH-MSGDESC-ACCOUNTINGTOKEN	PIC X(32).
**	アプリケーション識別データ	
20	MQXQH-MSGDESC-APPLIDENTITYDATA	PIC X(32).
**	登録アプリケーションタイプ	
20	MQXQH-MSGDESC-PUTAPPLTYPE	PIC S9(9) BINARY.
**	登録アプリケーション名	
20	MQXQH-MSGDESC-PUTAPPLNAME	PIC X(28).
**	登録日付	

20	MQXQH-MSGDESC-PUTDATE	PIC X(8).
**	登録時刻	
20	MQXQH-MSGDESC-PUTTIME	PIC X(8).
**	登録元データ	
20	MQXQH-MSGDESC-APPLORIGINDATA	PIC X(4).

## 付録 F TP1/Message Queue の定数値

TP1/Message Queue で使用する定数について、定数名とその値を示します。定数名は、関連するフィールドまたはパラメタに従ってグループ化されます。一つのグループに属するすべての定数名は、先頭に同じ文字列を持ちます。

ここでは、定数名のグループをアルファベット順に示します。

定数値の表記形式を次に示します。

MQxxxx_* (定数名)	10進数	16進数	または	文字
-------------------	------	------	-----	----

xxxx：0 から 4 文字の任意の文字

\*：任意の文字列

10 進数：10 進数です。

16 進数：X'nnnnnnnn'の形式で表される 16 進数です。(n：16 進数字)

文字：文字または文字列です。文字は' 'で囲み、文字列は" "で囲みます。

### • MQ\_\* (文字列および、バイト列の長さ)

MQ\_\*は、文字列、およびバイト列の長さを表す定数です。次に定数とその値を示します。

MQ_ACCOUNTING_TOKEN_LENGTH	32	X'00000020'
MQ_APPL_IDENTITY_DATA_LENGTH	32	X'00000020'
MQ_APPL_ORIGIN_DATA_LENGTH	4	X'00000004'
MQ_CHANNEL_NAME_LENGTH	20	X'00000014'
MQ_CLUSTER_NAME_LENGTH	48	X'00000030'
MQ_CORREL_ID_LENGTH	24	X'00000018'
MQ_CREATION_DATE_LENGTH	12	X'0000000C'
MQ_CREATION_TIME_LENGTH	8	X'00000008'
MQ_FORMAT_LENGTH	8	X'00000008'
MQ_GROUP_ID_LENGTH	24	X'00000018'
MQ_MODE_NAME_LENGTH	8	X'00000008'
MQ_MSG_HEADER_LENGTH	4000	X'00000FA0'
MQ_MSG_ID_LENGTH	24	X'00000018'
MQ_PROCESS_APPL_ID_LENGTH	256	X'00000100'
MQ_PROCESS_DESC_LENGTH	64	X'00000040'
MQ_PROCESS_ENV_DATA_LENGTH	128	X'00000080'
MQ_PROCESS_NAME_LENGTH	48	X'00000030'
MQ_PROCESS_USER_DATA_LENGTH	128	X'00000080'
MQ_PUT_APPL_NAME_LENGTH	28	X'0000001C'
MQ_PUT_DATE_LENGTH	8	X'00000008'
MQ_PUT_TIME_LENGTH	8	X'00000008'
MQ_Q_DESC_LENGTH	64	X'00000040'
MQ_Q_MGR_DESC_LENGTH	64	X'00000040'
MQ_Q_MGR_NAME_LENGTH	48	X'00000030'
MQ_Q_NAME_LENGTH	48	X'00000030'



命令 「オブジェクトの属性の設定」で Selectors 引数を参照してください。次に定数とその値を示します。

MQCA_FIRST	2001	X' 000007D1'
MQCA_APPL_ID	2001	X' 000007D1'
MQCA_BASE_Q_NAME	2002	X' 000007D2'
MQCA_CREATION_DATE	2004	X' 000007D4'
MQCA_CREATION_TIME	2005	X' 000007D5'
MQCA_DEAD_LETTER_Q_NAME	2006	X' 000007D6'
MQCA_ENV_DATA	2007	X' 000007D7'
MQCA_INITIATION_Q_NAME	2008	X' 000007D8'
MQCA_PROCESS_DESC	2011	X' 000007DB'
MQCA_PROCESS_NAME	2012	X' 000007DC'
MQCA_Q_DESC	2013	X' 000007DD'
MQCA_Q_MGR_DESC	2014	X' 000007DE'
MQCA_Q_MGR_NAME	2015	X' 000007DF'
MQCA_Q_NAME	2016	X' 000007E0'
MQCA_REMOTE_Q_MGR_NAME	2017	X' 000007E1'
MQCA_REMOTE_Q_NAME	2018	X' 000007E2'
MQCA_NAMES	2020	X' 000007E4'
MQCA_USER_DATA	2021	X' 000007E5'
MQCA_TRIGGER_DATA	2023	X' 000007E7'
MQCA_XMIT_Q_NAME	2024	X' 000007E8'
MQCA_DEF_XMIT_Q_NAME	2025	X' 000007E9'
MQCA_LAST_USED	2026	X' 000007EA'
MQCA_CLUSTER_NAME	2029	X' 000007ED'
MQCA_LAST	4000	X' 00000FA0'

• MQCC\_\* (完了コード)

MQCC\_\*は、完了コードを表す定数です。完了コードについては、「付録 B.1 完了コード」を参照してください。次に定数とその値を示します。

MQCC_OK	0	X' 00000000'
MQCC_WARNING	1	X' 00000001'
MQCC_FAILED	2	X' 00000002'

• MQCCSI\_\* (文字セット識別子)

MQCCSI\_\*は、文字セット識別子を表す定数です。文字セット識別子については、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」で CodedCharSetId フィールドを参照してください。次に定数とその値を示します。

MQCCSI_INHERIT	-2	X' FFFFFFFE'
MQCCSI_EMBEDDED	-1	X' FFFFFFFF'
MQCCSI_DEFAULT	0	X' 00000000'
MQCCSI_Q_MGR	0	X' 00000000'
MQCCSI_UNDEFINED	0	X' 00000000'

• MQCFH\_\* (構造体識別子)

MQCFH\_\*は、構造体識別子を表す定数です。構造体識別子については、「1. データタイプ」の「MQCFH 構造体 - PCF ヘッダ」で StrucLength フィールド、Version フィールド、および Control フィールドを参照してください。次に定数とその値を示します。

MQCFH_STRUC_LENGTH	36	X' 00000024'
MQCFH_VERSION_1	1	X' 00000001'

MQCFC_LAST	1	X' 00000001'
MQCFC_NOT_LAST	0	X' 00000000'

- MQCFIL\_\* (構造体識別子)

MQCFIL\_\*は、構造体識別子を表す定数です。構造体識別子については、「1. データタイプ」の「MQCFIL 構造体 - PCF 整数リスト引数」で StrucLength フィールドを参照してください。次に定数とその値を示します。

MQCFIL_STRUC_LENGTH	16	X' 00000010'
---------------------	----	--------------

- MQCFIN\_\* (構造体識別子)

MQCFIN\_\*は、構造体識別子を表す定数です。構造体識別子については、「1. データタイプ」の「MQCFIN 構造体 - PCF 整数引数」で StrucLength フィールドを参照してください。次に定数とその値を示します。

MQCFIN_STRUC_LENGTH	16	X' 00000010'
---------------------	----	--------------

- MQCFST\_\* (構造体識別子)

MQCFST\_\*は、構造体識別子を表す定数です。構造体識別子については、「1. データタイプ」の「MQCFST 構造体 - PCF スtring引数」で StrucLength フィールドを参照してください。次に定数とその値を示します。

MQCFST_STRUC_LENGTH	20	X' 00000014'
---------------------	----	--------------

- MQCFT\_\* (構造体タイプ)

MQCFT\_\*は、構造体タイプを表す定数です。構造体タイプについては、「1. データタイプ」の「MQCFH 構造体 - PCF ヘッダ」, 「MQCFIN 構造体 - PCF 整数引数」, 「MQCFST 構造体 - PCF スtring引数」, および「MQCFIL 構造体 - PCF 整数リスト引数」で Type フィールドを参照してください。次に定数とその値を示します。

MQCFT_COMMAND	1	X' 00000001'
MQCFT_INTEGER	3	X' 00000003'
MQCFT_STRING	4	X' 00000004'
MQCFT_INTEGER_LIST	5	X' 00000005'

- MQCI\_\* (相関識別子)

MQCI\_\*は、相関識別子を表す定数です。相関識別子については、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」で CorrelId フィールドを参照してください。次に定数とその値を示します。

MQCI_NONE	"¥0"
MQCI_NONE_ARRAY	'¥0', '¥0'

- MQCO\_\* (クローズオプション)

MQCO\_\*は、クローズオプションを表す定数です。クローズオプションについては、「2. メッセージキューイング機能の命令」の「MQCLOSE 命令 - オブジェクトのクローズ」で Options 引数を参照してください。次に定数とその値を示します。

MQCO_NONE	0	X' 00000000'
MQCO_DELETE	1	X' 00000001'
MQCO_DELETE_PURGE	2	X' 00000002'

- MQDCC\_\* (変換文字列オプション)

MQDCC\_\*は、変換文字列オプションを表す定数です。次に定数とその値を示します。

MQDCC_NONE	0	X' 00000000'
------------	---	--------------

- MQDH\_\* (配布ヘッダ構造体識別子)

MQDH\_\*は、配布ヘッダ構造体識別子を表す定数です。配布ヘッダ構造体識別子については、「1. データタイプ」の「MQDH 構造体 - 配布ヘッダ」を参照してください。次に定数とその値を示します。

MQDH_STRUC_ID		"DH"
MQDH_VERSION_1	1	X' 00000001'
MQDH_CURRENT_VERSION	1	X' 00000001'

- MQDHF\_\* (配布ヘッダフラグ)

MQDHF\_\*は、配布ヘッダフラグを表す定数です。配布ヘッダフラグについては、「1. データタイプ」の「MQDH 構造体 - 配布ヘッダ」を参照してください。次に定数とその値を示します。

MQDHF_NONE	0	X' 00000000'
MQDHF_NEW_MSG_IDS	1	X' 00000001'

- MQDL\_\* (配布リストのサポート)

MQDL\_\*は、配布リストを表す定数です。配布リストのサポートについては、「1. データタイプ」の「MQDH 構造体 - 配布ヘッダ」を参照してください。次に定数とその値を示します。

MQDL_NOT_SUPPORTED	0	X' 00000000'
MQDL_SUPPORTED	1	X' 00000001'

- MQDLH\_STRUC\_\* (デッドレターヘッダの構造体識別子)

MQDLH\_STRUC\_\*は、デッドレターヘッダの構造体識別子を表す定数です。デッドレターヘッダの構造体識別子については、「1. データタイプ」の「MQDLH 構造体 - デッドレターヘッダ」で StrucId フィールドを参照してください。次に定数とその値を示します。

MQDLH_STRUC_ID		"DLH "
MQDLH_STRUC_ID_ARRAY		'D','L','H',' '

- MQDLH\_VERSION\_\* (デッドレターヘッダの構造体バージョン番号)

MQDLH\_VERSION\_\*は、デッドレターヘッダの構造体バージョン番号を表す定数です。デッドレターヘッダの構造体バージョン番号については、「1. データタイプ」の「MQDLH 構造体 - デッドレターヘッダ」で Version フィールドを参照してください。次に定数とその値を示します。

MQDLH_VERSION_1	1	X' 00000001'
MQDLH_CURRENT_VERSION	1	X' 00000001'

- MQDXP\_\* (データ変換出力引数の構造体識別子)

MQDXP\_\*は、データ変換出力引数の構造体識別子を表す定数です。次に定数とその値を示します。



MQENC_INTEGER_UNDEFINED	0	X'00000000'
MQENC_INTEGER_NORMAL	1	X'00000001'
MQENC_INTEGER_REVERSED	2	X'00000002'

• MQFB\_\* (報告メッセージ返答コード, メッセージ通知理由)

MQFB\_\*は、報告メッセージ返答コード、メッセージ通知理由を表す定数です。報告メッセージ返答コードについては、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で Feedback フィールドを参照してください。メッセージ通知理由については、「1. データタイプ」の「MQDLH 構造体 – デッドレターヘッダ」で Reason フィールドを参照してください。次に定数とその値を示します。

MQFB_NONE	0	X'00000000'
MQFB_SYSTEM_FIRST	1	X'00000001'
MQFB_QUIT	256	X'00000100'
MQFB_EXPIRATION	258	X'00000102'
MQFB_COA	259	X'00000103'
MQFB_COD	260	X'00000104'
MQFB_CHANNEL_COMPLETED	262	X'00000106'
MQFB_CHANNEL_FAIL_RETRY	263	X'00000107'
MQFB_CHANNEL_FAIL	264	X'00000108'
MQFB_APPL_CANNOT_BE_STARTED	265	X'00000109'
MQFB_TM_ERROR	266	X'0000010A'
MQFB_APPL_TYPE_ERROR	267	X'0000010B'
MQFB_XMIT_Q_MSG_ERROR	271	X'0000010F'
MQFB_PAN	275	X'00000113'
MQFB_NAN	276	X'00000114'
MQFB_STOPPED_BY_CHAD_EXIT	277	X'00000115'
MQFB_STOPPED_BY_PUBSUB_EXIT	279	X'00000117'
MQFB_NOT_A_REPOSITORY_MSG	280	X'00000118'
MQFB_BIND_OPEN_CLUSRCVR_DEL	281	X'00000119'
MQFB_DATA_LENGTH_ZERO	291	X'00000123'
MQFB_DATA_LENGTH_NEGATIVE	292	X'00000124'
MQFB_DATA_LENGTH_TOO_BIG	293	X'00000125'
MQFB_BUFFER_OVERFLOW	294	X'00000126'
MQFB_LENGTH_OFF_BY_ONE	295	X'00000127'
MQFB_IIH_ERROR	296	X'00000128'
MQFB_NOT_AUTHORIZED_FOR_IMS	298	X'0000012A'
MQFB_IMS_ERROR	300	X'0000012C'
MQFB_IMS_FIRST	301	X'0000012D'
MQFB_IMS_LAST	399	X'0000018F'
MQFB_CICS_INTERNAL_ERROR	401	X'00000191'
MQFB_CICS_NOT_AUTHORIZED	402	X'00000192'
MQFB_CICS_BRIDGE_FAILURE	403	X'00000193'
MQFB_CICS_CORREL_ID_ERROR	404	X'00000194'
MQFB_CICS_CCSID_ERROR	405	X'00000195'
MQFB_CICS_ENCODING_ERROR	406	X'00000196'
MQFB_CICS_CIH_ERROR	407	X'00000197'
MQFB_CICS_UOW_ERROR	408	X'00000198'
MQFB_CICS_COMMAREA_ERROR	409	X'00000199'
MQFB_CICS_APPL_NOT_STARTED	410	X'0000019A'
MQFB_CICS_APPL_ABENDED	411	X'0000019B'
MQFB_CICS_DLQ_ERROR	412	X'0000019C'
MQFB_CICS_UOW_BACKED_OUT	413	X'0000019D'
MQFB_SYSTEM_LAST	65535	X'0000FFFF'



MQGMO_ACCEPT_TRUNCATED_MSG	64	X' 00000040'
MQGMO_MSG_UNDER_CURSOR	256	X' 00000100'
MQGMO_LOCK	512	X' 00000200'
MQGMO_UNLOCK	1024	X' 00000400'
MQGMO_BROWSE_MSG_UNDER_CURSOR	2048	X' 00000800'
MQGMO_SYNCPOINT_IF_PERSISTENT	4096	X' 00001000'
MQGMO_FAIL_IF QUIESCING	8192	X' 00002000'
MQGMO_CONVERT	16384	X' 00004000'
MQGMO_LOGICAL_ORDER	32768	X' 00008000'
MQGMO_COMPLETE_MSG	65536	X' 00010000'
MQGMO_ALL_MSGS_AVAILABLE	131072	X' 00020000'
MQGMO_ALL_SEGMENTS_AVAILABLE	262144	X' 00040000'

- MQGMO\_STRUC\_\* (構造体識別子)

MQGMO\_STRUC\_\*は、構造体識別子を表す定数です。構造体識別子については、「1. データタイプ」の「MQGMO 構造体 - メッセージ取り出しオプション」で StrucId フィールドを参照してください。次に定数とその値を示します。

MQGMO_STRUC_ID	"GMO "
MQGMO_STRUC_ID_ARRAY	'G','M','O',' '

- MQGMO\_VERSION\_\* (構造体バージョン番号)

MQGMO\_VERSION\_\*は、構造体バージョン番号を表す定数です。構造体バージョン番号については、「1. データタイプ」の「MQGMO 構造体 - メッセージ取り出しオプション」で Version フィールドを参照してください。次に定数とその値を示します。

MQGMO_VERSION_1	1	X' 00000001'
MQGMO_VERSION_2	2	X' 00000002'
MQGMO_CURRENT_VERSION	2	X' 00000002'

- MQGS\_\* (グループフラグ)

MQGS\_\*は、グループフラグを表す定数です。グループフラグは取り出されたメッセージがグループに属するかどうかを示します。グループフラグについては、「1. データタイプ」の「MQGMO 構造体 - メッセージ取り出しオプション」で GroupStatus フィールドを参照してください。次に定数とその値を示します。

MQGS_NOT_IN_GROUP	' '
MQGS_MSG_IN_GROUP	'G'
MQGS_LAST_MSG_IN_GROUP	'L'

- MQHC\_\* (コネクションハンドル)

MQHC\_\*は、コネクションハンドルを表す定数です。コネクションハンドルについては、「1. データタイプ」の「MQDISC 命令 - キューマネージャからの切り離し」で Hconn 引数を参照してください。次に定数とその値を示します。

MQHC_UNUSABLE_HCONN	-1	X' FFFFFFFF'
---------------------	----	--------------

- MQHO\_\* (オブジェクトハンドル)

MQHO\_\*は、オブジェクトハンドルを表す定数です。オブジェクトハンドルについては、「2. メッセージキューイング機能の命令」の「MQCLOSE 命令 - オブジェクトのクローズ」で Hobj 引数を参照してください。次に定数とその値を示します。

MQHO_UNUSABLE_HOBJ	-1 X'FFFFFFFF'
--------------------	----------------

- MQIA\_\* (整数型属性のセレクタ)

MQIA\_\*は、整数型属性のセレクタを表す定数です。整数型属性のセレクタについては、「2. メッセージキューイング機能の命令」の「MQINQ 命令 - オブジェクトの属性の照会」および「MQSET 命令 - オブジェクトの属性の設定」で Selectors 引数を参照してください。次に定数とその値を示します。

MQIA_FIRST	1	X'00000001'
MQIA_APPL_TYPE	1	X'00000001'
MQIA_CODED_CHAR_SET_ID	2	X'00000002'
MQIA_CURRENT_Q_DEPTH	3	X'00000003'
MQIA_DEF_INPUT_OPEN_OPTION	4	X'00000004'
MQIA_DEF_PERSISTENCE	5	X'00000005'
MQIA_DEF_PRIORITY	6	X'00000006'
MQIA_DEFINITION_TYPE	7	X'00000007'
MQIA_HARDEN_GET_BACKOUT	8	X'00000008'
MQIA_INHIBIT_GET	9	X'00000009'
MQIA_INHIBIT_PUT	10	X'0000000A'
MQIA_MAX_HANDLES	11	X'0000000B'
MQIA_USAGE	12	X'0000000C'
MQIA_MAX_MSG_LENGTH	13	X'0000000D'
MQIA_MAX_PRIORITY	14	X'0000000E'
MQIA_MAX_Q_DEPTH	15	X'0000000F'
MQIA_MSG_DELIVERY_SEQUENCE	16	X'00000010'
MQIA_OPEN_INPUT_COUNT	17	X'00000011'
MQIA_OPEN_OUTPUT_COUNT	18	X'00000012'
MQIA_NAME_COUNT	19	X'00000013'
MQIA_Q_TYPE	20	X'00000014'
MQIA_RETENTION_INTERVAL	21	X'00000015'
MQIA_SHAREABILITY	23	X'00000017'
MQIA_TRIGGER_CONTROL	24	X'00000018'
MQIA_TRIGGER_INTERVAL	25	X'00000019'
MQIA_TRIGGER_MSG_PRIORITY	26	X'0000001A'
MQIA_TRIGGER_TYPE	28	X'0000001C'
MQIA_TRIGGER_DEPTH	29	X'0000001D'
MQIA_SYNCPOINT	30	X'0000001E'
MQIA_PLATFORM	32	X'00000020'
MQIA_MAX_UNCOMMITTED_MSGS	33	X'00000021'
MQIA_DIST_LISTS	34	X'00000022'
MQIA_SCOPE	45	X'0000002D'
MQIA_LAST_USED	56	X'00000038'
MQIA_DEF_BIND	61	X'0000003D'
MQIA_LAST	2000	X'000007D0'

- MQIAV\_\* (整数型属性の値)

MQIAV\_\*は、整数型属性の値を表す定数です。次に定数とその値を示します。

MQIAV_NOT_APPLICABLE	-1 X'FFFFFFFF'
----------------------	----------------

- MQMD\_STRUC\_\* (メッセージ記述子の構造体識別子)

MQMD\_STRUC\_\*は、メッセージ記述子の構造体識別子を表す定数です。メッセージ記述子の構造体識別子については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で StructId フィールドを参照してください。次に定数とその値を示します。

MQMD_STRUC_ID	"MD "
MQMD_STRUC_ID_ARRAY	'M','D',' ',' ',' '

- MQMD\_VERSION\_\* (構造体バージョン番号)

MQMD\_VERSION\_\*は、構造体バージョン番号を表す定数です。構造体バージョン番号については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で Version フィールドを参照してください。次に定数とその値を示します。

MQMD_VERSION_1	1 X'00000001'
MQMD_VERSION_2	2 X'00000002'
MQMD_CURRENT_VERSION	2 X'00000002'

- MQMDE\_\* (構造体長)

MQMDE\_\*は、構造体長を表す定数です。構造体長はメッセージ記述子の拡張の長さを示します。メッセージ記述子の拡張の長さについては、「1. データタイプ」の「MQMDE 構造体 – メッセージ記述子の拡張」で Version フィールドを参照してください。次に定数とその値を示します。

MQMDE_LENGTH_2	72 X'00000048'
----------------	----------------

- MQMDE\_\* (構造体識別子)

MQMDE\_\*は、メッセージ記述子の拡張の構造体識別子を表す定数です。メッセージ記述子の拡張の構造体識別子については、「1. データタイプ」の「MQMDE 構造体 – メッセージ記述子の拡張」で Version フィールドを参照してください。次に定数とその値を示します。

MQMDE_STRUC_ID	'MDE '
MQMDE_STRUC_ID_ARRAY	'M','D','E',' ',' '

- MQMDE\_\* (構造体バージョン番号)

MQMDE\_\*は、構造体バージョン番号を表す定数です。構造体バージョン番号はメッセージ記述子の拡張のバージョンを示します。メッセージ記述子の拡張のバージョンについては、「1. データタイプ」の「MQMDE 構造体 – メッセージ記述子の拡張」で Version フィールドを参照してください。次に定数とその値を示します。

MQMDE_VERSION_2	2 X'00000002'
MQMDE_CURRENT_VERSION	2 X'00000002'

- MQMDEF\_\* (メッセージ記述子拡張フラグ)

MQMDEF\_\*は、メッセージ記述子の拡張フラグを表す定数です。メッセージ記述子の拡張フラグについては、「1. データタイプ」の「MQMDE 構造体 – メッセージ記述子の拡張」で Flags フィールドを参照してください。次に定数とその値を示します。

MQMDEF_NONE	0 X'00000000'
-------------	---------------

- MQMDS\_\* (メッセージ配布順序)

MQMDS\_\*は、メッセージ配布順序を表す定数です。メッセージ配布順序については、「3. オブジェクトの属性」の「キューの属性」で MsgDeliverySequence 属性を参照してください。次に定数とその値を示します。

MQMDS_PRIORITY	0	X' 00000000'
MQMDS_FIFO	1	X' 00000001'

• MQMF\_\* (メッセージフラグ)

MQMF\_\*は、メッセージフラグを表す定数です。メッセージフラグについては、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で MsgFlags フィールドを参照してください。次に定数とその値を示します。

MQMF_SEGMENTATION_INHIBITED	0	X' 00000000'
MQMF_NONE	0	X' 00000000'
MQMF_SEGMENTATION_ALLOWED	1	X' 00000001'
MQMF_SEGMENT	2	X' 00000002'
MQMF_LAST_SEGMENT	4	X' 00000004'
MQMF_MSG_IN_GROUP	8	X' 00000008'
MQMF_LAST_MSG_IN_GROUP	16	X' 00000010'

• MQMI\_\* (メッセージ識別子)

MQMI\_\*は、メッセージ識別子を表す定数です。メッセージ識別子については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で MsgId フィールドを参照してください。次に定数とその値を示します。

MQMI_NONE	"¥0"
MQMI_NONE_ARRAY	'¥0', '¥0'

• MQMO\_\* (一致オプション)

MQMO\_\*は、一致オプションを表す定数です。一致オプションについては、「1. データタイプ」の「MQGMO 構造体 – メッセージ取り出しオプション」で MatchOptions フィールドを参照してください。次に定数とその値を示します。

MQMO_NONE	0	X' 00000000'
MQMO_MATCH_MSG_ID	1	X' 00000001'
MQMO_MATCH_CORREL_ID	2	X' 00000002'
MQMO_MATCH_GROUP_ID	4	X' 00000004'
MQMO_MATCH_MSG_SEQ_NUMBER	8	X' 00000008'
MQMO_MATCH_OFFSET	16	X' 00000010'

• MQMT\_\* (メッセージタイプ)

MQMT\_\*は、メッセージタイプを表す定数です。メッセージタイプについては、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で MsgType フィールドを参照してください。次に定数とその値を示します。

MQMT_SYSTEM_FIRST	1	X' 00000001'
MQMT_REQUEST	1	X' 00000001'
MQMT_REPLY	2	X' 00000002'
MQMT_REPORT	4	X' 00000004'
MQMT_DATAGRAM	8	X' 00000008'

MQMT_SYSTEM_LAST	65535	X' 0000FFFF'
MQMT_APPL_FIRST	65536	X' 00010000'
MQMT_APPL_LAST	999999999	X' 3B9AC9FF'

- MQOD\_\* (オブジェクト記述子長)

MQOD\_\*は、オブジェクト記述子の長さを表します。オブジェクト記述子長については、「1. データタイプ」の「MQOD 構造体 – オブジェクト記述子」を参照してください。次に定数とその値を示します。

MQOD_CURRENT_LENGTH	(環境に依存します)
---------------------	------------

- MQOD\_STRUC\_\* (構造体識別子)

MQOD\_STRUC\_\*は、構造体識別子を表す定数です。構造体識別子については、「1. データタイプ」の「MQOD 構造体 – オブジェクト記述子」で StrucId フィールドを参照してください。次に定数とその値を示します。

MQOD_STRUC_ID	"OD "
MQOD_STRUC_ID_ARRAY	'O','D',' ',' ',' '

- MQOD\_VERSION\_\* (構造体バージョン番号)

MQOD\_VERSION\_\*は、構造体バージョン番号を表す定数です。構造体バージョン番号については、「1. データタイプ」の「MQOD 構造体 – オブジェクト記述子」で Version フィールドを参照してください。次に定数とその値を示します。

MQOD_VERSION_1	1	X' 00000001'
MQOD_VERSION_2	2	X' 00000002'
MQOD_VERSION_3	3	X' 00000003'
MQOD_CURRENT_VERSION	3	X' 00000003'

- MQOII\_\* (オブジェクトインスタンス識別子)

MQOII\_\*は、オブジェクトインスタンス識別子を表す定数です。オブジェクトインスタンス識別子については、「1. データタイプ」の「MQORMH 構造体 – メッセージ参照ヘッダ」で ObjectInstanceId フィールドを参照してください。次に定数とその値を示します。

MQOII_NONE	"¥0"
MQOII_NONE_ARRAY	'¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0', '¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0', '¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0','¥0'

- MQOL\_\* (登録元データ長)

MQOL\_\*は、登録元のデータ長を表す定数です。登録元データ長については、「1. データタイプ」の「MQOD 構造体 – オブジェクト記述子」で OriginalLength フィールド参照してください。次に定数とその値を示します。

MQOL_UNDEFINED	-1	X' FFFFFFFF'
----------------	----	--------------

- MQOO\_\* (オープンオプション)

MQOO\_\*は、オープンオプションを表す定数です。オープンオプションについては、「2. メッセージキューイング機能の命令」の「MQOPEN 命令 – オブジェクトのオープン」で Options 引数を参照してください。次に定数とその値を示します。

MQ00_BIND_AS_Q_DEF	0	X'00000000'
MQ00_INPUT_AS_Q_DEF	1	X'00000001'
MQ00_INPUT_SHARED	2	X'00000002'
MQ00_INPUT_EXCLUSIVE	4	X'00000004'
MQ00_BROWSE	8	X'00000008'
MQ00_OUTPUT	16	X'00000010'
MQ00_INQUIRE	32	X'00000020'
MQ00_SET	64	X'00000040'
MQ00_SAVE_ALL_CONTEXT	128	X'00000080'
MQ00_PASS_IDENTITY_CONTEXT	256	X'00000100'
MQ00_PASS_ALL_CONTEXT	512	X'00000200'
MQ00_SET_IDENTITY_CONTEXT	1024	X'00000400'
MQ00_SET_ALL_CONTEXT	2048	X'00000800'
MQ00_ALTERNATE_USER_AUTHORITY	4096	X'00001000'
MQ00_FAIL_IF QUIESCING	8192	X'00002000'
MQ00_BIND_ON_OPEN	16384	X'00004000'
MQ00_BIND_NOT_FIXED	32768	X'00008000'

- MQOT\_\* (オブジェクトタイプ)

MQOT\_\*は、オブジェクトタイプを表す定数です。オブジェクトタイプについては、「1. データタイプ」の「MQOD 構造体 - オブジェクト記述子」で ObjectType フィールドを参照してください。次に定数とその値を示します。

MQOT_Q	1	X'00000001'
MQOT_PROCESS	3	X'00000003'
MQOT_Q_MGR	5	X'00000005'

- MQPER\_\* (メッセージ永続性)

MQPER\_\*は、メッセージ永続性を表す定数です。メッセージ永続性については、「1. データタイプ」の「MQMD 構造体 - メッセージ記述子」で Persistence フィールドを参照してください。次に定数とその値を示します。

MQPER_NOT_PERSISTENT	0	X'00000000'
MQPER_PERSISTENT	1	X'00000001'
MQPER_PERSISTENCE_AS_Q_DEF	2	X'00000002'

- MQPL\_\* (プラットフォーム)

MQPL\_\*は、プラットフォームを表す定数です。プラットフォームについては、「3. オブジェクトの属性」の「キューマネージャの属性」で Platform 属性を参照してください。次に定数とその値を示します。

MQPL_MVS	1	X'00000001'
MQPL_OS2	2	X'00000002'
MQPL_AIX	3	X'00000003'
MQPL_UNIX	3	X'00000003'
MQPL_OS400	4	X'00000004'
MQPL_WINDOWS	5	X'00000005'
MQPL_WINDOWS_NT	11	X'0000000B'
MQPL_OPEN_TP1	15	X'0000000F'
MQPL_XDM	16	X'00000010'
MQPL_TMS_4V	17	X'00000011'

- MQPMO\_\* (メッセージ登録オプション)

MQPMO\_\*は、メッセージ登録オプションを表す定数です。メッセージ登録オプションについては、「1. データタイプ」の「MQPMO 構造体 - メッセージ登録オプション」で Options フィールドを参照してください。次に定数とその値を示します。

MQPMO_NONE	0	X' 00000000'
MQPMO_SYNCPOINT	2	X' 00000002'
MQPMO_NO_SYNCPOINT	4	X' 00000004'
MQPMO_DEFAULT_CONTEXT	32	X' 00000020'
MQPMO_NEW_MSG_ID	64	X' 00000040'
MQPMO_NEW_CORREL_ID	128	X' 00000080'
MQPMO_PASS_IDENTITY_CONTEXT	256	X' 00000100'
MQPMO_PASS_ALL_CONTEXT	512	X' 00000200'
MQPMO_SET_IDENTITY_CONTEXT	1024	X' 00000400'
MQPMO_SET_ALL_CONTEXT	2048	X' 00000800'
MQPMO_ALTERNATE_USER_AUTHORITY	4096	X' 00001000'
MQPMO_FAIL_IF QUIESCING	8192	X' 00002000'
MQPMO_NO_CONTEXT	16384	X' 00004000'
MQPMO_LOGICAL_ORDER	32768	X' 00008000'

• MQPMO\_\* (メッセージ登録オプションの構造体長)

MQPMO\_\*は、メッセージ登録オプションの構造体長を表す定数です。次に定数とその値を示します。

MQPMO_CURRENT_LENGTH	環境に依存
----------------------	-------

• MQPMO\_STRUC\_\* (メッセージ登録オプションの構造体識別子)

MQPMO\_STRUC\_\*は、メッセージ登録オプションの構造体識別子を表す定数です。メッセージ登録オプションの構造体識別子については、「1. データタイプ」の「MQPMO 構造体 - メッセージ登録オプション」で StrucId フィールドを参照してください。次に定数とその値を示します。

MQPMO_STRUC_ID	"PMO "
MQPMO_STRUC_ID_ARRAY	'P','M','O',' '

• MQPMO\_VERSION\_\* (構造体バージョン番号)

MQPMO\_VERSION\_\*は、構造体バージョン番号を表す定数です。構造体バージョン番号については、「1. データタイプ」の「MQPMO 構造体 - メッセージ登録オプション」で Version フィールドを参照してください。次に定数とその値を示します。

MQPMO_VERSION_1	1	X' 00000001'
MQPMO_VERSION_2	2	X' 00000002'
MQPMO_CURRENT_VERSION	2	X' 00000002'

• MQPMRF\_\* (メッセージ登録レコードフィールドのフラグ)

MQPMRF\_\*は、メッセージ登録レコードフィールドのフラグを表す定数です。メッセージ登録レコードフィールドのフラグについては、「1. データタイプ」の「MQPDH 構造体 - 配布ヘッダ」で PutMsgRecFields フィールドを参照してください。次に定数とその値を示します。

MQPMRF_NONE	0	X' 00000000'
MQPMRF_MSG_ID	1	X' 00000001'
MQPMRF_CORREL_ID	2	X' 00000002'
MQPMRF_GROUP_ID	4	X' 00000004'
MQPMRF_FEEDBACK	8	X' 00000008'
MQPMRF_ACCOUNTING_TOKEN	16	X' 00000010'

- MQPRI\_\* (メッセージ優先度)

MQPRI\_\*は、メッセージ優先度を表す定数です。メッセージ優先度については、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で Priority フィールドを参照してください。次に定数とその値を示します。

MQPRI_PRIORITY_AS_Q_DEF	-1	X' FFFFFFFF'
-------------------------	----	--------------

- MQQA\_BACKOUT\_\* (ロールバック回数記録方法)

MQQA\_BACKOUT\_\*は、ロールバック回数記録方法を表す定数です。ロールバック回数記録方法については、「3. オブジェクトの属性」の「キューの属性」で HardenGetBackout 属性を参照してください。次に定数とその値を示します。

MQQA_BACKOUT_NOT_HARDENED	0	X' 00000000'
MQQA_BACKOUT_HARDENED	1	X' 00000001'

- MQQA\_GET\_\* (取り出し許可)

MQQA\_GET\_\*は、取り出し許可を表す定数です。取り出し許可については、「3. オブジェクトの属性」の「キューの属性」で InhibitGet 属性を参照してください。次に定数とその値を示します。

MQQA_GET_ALLOWED	0	X' 00000000'
MQQA_GET_INHIBITED	1	X' 00000001'

- MQQA\_PUT\_\* (登録許可)

MQQA\_PUT\_\*は、登録許可を表す定数です。登録許可については、「3. オブジェクトの属性」の「キューの属性」で InhibitPut 属性を参照してください。次に定数とその値を示します。

MQQA_PUT_ALLOWED	0	X' 00000000'
MQQA_PUT_INHIBITED	1	X' 00000001'

- MQQA\_\*\_SHAREABLE (共用性)

MQQA\_\*\_SHAREABLE は、共用性を表す定数です。共用性については、「3. オブジェクトの属性」の「キューの属性」で Shareability 属性を参照してください。次に定数とその値を示します。

MQQA_NOT_SHAREABLE	0	X' 00000000'
MQQA_SHAREABLE	1	X' 00000001'

- MQQDT\_\* (定義タイプ)

MQQDT\_\*は、定義タイプを表す定数です。定義タイプについては、「3. オブジェクトの属性」の「キューの属性」で DefinitionType 属性を参照してください。次に定数とその値を示します。

MQQDT_PREDEFINED	1	X' 00000001'
MQQDT_PERMANENT_DYNAMIC	2	X' 00000002'
MQQDT_TEMPORARY_DYNAMIC	3	X' 00000003'

- MQQT\_\* (キュータイプ)

MQQT\_\*は、キュータイプを表す定数です。キュータイプについては、「3. オブジェクトの属性」の「キューの属性」で QType 属性を参照してください。次に定数とその値を示します。

MQQT_LOCAL	1	X' 00000001'
MQQT_MODEL	2	X' 00000002'

MQQT_ALIAS	3	X' 00000003'
MQQT_REMOTE	6	X' 00000006'
MQQT_CLUSTER	7	X' 00000007'

• MQRC\_\* (理由コード)

MQRC\_\*は、理由コードを表す定数です。理由コードについては、「付録 B.2 理由コード」を参照してください。次に定数とその値を示します。

MQRC_NONE	0	X' 00000000'
MQRC_APPL_FIRST	900	X' 00000384'
MQRC_APPL_LAST	999	X' 000003E7'
MQRC_ALIAS_BASE_Q_TYPE_ERROR	2001	X' 000007D1'
MQRC_ALREADY_CONNECTED	2002	X' 000007D2'
MQRC_BACKED_OUT	2003	X' 000007D3'
MQRC_BUFFER_ERROR	2004	X' 000007D4'
MQRC_BUFFER_LENGTH_ERROR	2005	X' 000007D5'
MQRC_CHAR_ATTR_LENGTH_ERROR	2006	X' 000007D6'
MQRC_CHAR_ATTRS_ERROR	2007	X' 000007D7'
MQRC_CHAR_ATTRS_TOO_SHORT	2008	X' 000007D8'
MQRC_CONNECTION_BROKEN	2009	X' 000007D9'
MQRC_DATA_LENGTH_ERROR	2010	X' 000007DA'
MQRC_DYNAMIC_Q_NAME_ERROR	2011	X' 000007DB'
MQRC_ENVIRONMENT_ERROR	2012	X' 000007DC'
MQRC_EXPIRY_ERROR	2013	X' 000007DD'
MQRC_FEEDBACK_ERROR	2014	X' 000007DE'
MQRC_GET_INHIBITED	2016	X' 000007E0'
MQRC_HANDLE_NOT_AVAILABLE	2017	X' 000007E1'
MQRC_HCONN_ERROR	2018	X' 000007E2'
MQRC_HOBJ_ERROR	2019	X' 000007E3'
MQRC_INHIBIT_VALUE_ERROR	2020	X' 000007E4'
MQRC_INT_ATTR_COUNT_ERROR	2021	X' 000007E5'
MQRC_INT_ATTR_COUNT_TOO_SMALL	2022	X' 000007E6'
MQRC_INT_ATTRS_ARRAY_ERROR	2023	X' 000007E7'
MQRC_SYNCPOINT_LIMIT_REACHED	2024	X' 000007E8'
MQRC_MAX_CONNS_LIMIT_REACHED	2025	X' 000007E9'
MQRC_MD_ERROR	2026	X' 000007EA'
MQRC_MISSING_REPLY_TO_Q	2027	X' 000007EB'
MQRC_MSG_TYPE_ERROR	2029	X' 000007ED'
MQRC_MSG_TOO_BIG_FOR_Q	2030	X' 000007EE'
MQRC_MSG_TOO_BIG_FOR_Q_MGR	2031	X' 000007EF'
MQRC_NO_MSG_AVAILABLE	2033	X' 000007F1'
MQRC_NO_MSG_UNDER_CURSOR	2034	X' 000007F2'
MQRC_NOT_AUTHORIZED	2035	X' 000007F3'
MQRC_NOT_OPEN_FOR_BROWSE	2036	X' 000007F4'
MQRC_NOT_OPEN_FOR_INPUT	2037	X' 000007F5'
MQRC_NOT_OPEN_FOR_INQUIRE	2038	X' 000007F6'
MQRC_NOT_OPEN_FOR_OUTPUT	2039	X' 000007F7'
MQRC_NOT_OPEN_FOR_SET	2040	X' 000007F8'
MQRC_OBJECT_CHANGED	2041	X' 000007F9'
MQRC_OBJECT_IN_USE	2042	X' 000007FA'
MQRC_OBJECT_TYPE_ERROR	2043	X' 000007FB'
MQRC_OD_ERROR	2044	X' 000007FC'
MQRC_OPTION_NOT_VALID_FOR_TYPE	2045	X' 000007FD'
MQRC_OPTIONS_ERROR	2046	X' 000007FE'
MQRC_PERSISTENCE_ERROR	2047	X' 000007FF'
MQRC_PERSISTENT_NOT_ALLOWED	2048	X' 00000800'
MQRC_PRIORITY_EXCEEDS_MAXIMUM	2049	X' 00000801'

MQRC_PRIORITY_ERROR	2050	X' 00000802'
MQRC_PUT_INHIBITED	2051	X' 00000803'
MQRC_Q_DELETED	2052	X' 00000804'
MQRC_Q_FULL	2053	X' 00000805'
MQRC_Q_NOT_EMPTY	2055	X' 00000807'
MQRC_Q_SPACE_NOT_AVAILABLE	2056	X' 00000808'
MQRC_Q_TYPE_ERROR	2057	X' 00000809'
MQRC_Q_MGR_NAME_ERROR	2058	X' 0000080A'
MQRC_Q_MGR_NOT_AVAILABLE	2059	X' 0000080B'
MQRC_REPORT_OPTIONS_ERROR	2061	X' 0000080D'
MQRC_SECOND_MARK_NOT_ALLOWED	2062	X' 0000080E'
MQRC_SECURITY_ERROR	2063	X' 0000080F'
MQRC_SELECTOR_COUNT_ERROR	2065	X' 00000811'
MQRC_SELECTOR_LIMIT_EXCEEDED	2066	X' 00000812'
MQRC_SELECTOR_ERROR	2067	X' 00000813'
MQRC_SELECTOR_NOT_FOR_TYPE	2068	X' 00000814'
MQRC_SIGNAL_OUTSTANDING	2069	X' 00000815'
MQRC_SIGNAL_REQUEST_ACCEPTED	2070	X' 00000816'
MQRC_STORAGE_NOT_AVAILABLE	2071	X' 00000817'
MQRC_SYNCPOINT_NOT_AVAILABLE	2072	X' 00000818'
MQRC_TRIGGER_CONTROL_ERROR	2075	X' 0000081B'
MQRC_TRIGGER_DEPTH_ERROR	2076	X' 0000081C'
MQRC_TRIGGER_MSG_PRIORITY_ERR	2077	X' 0000081D'
MQRC_TRIGGER_TYPE_ERROR	2078	X' 0000081E'
MQRC_TRUNCATED_MSG_ACCEPTED	2079	X' 0000081F'
MQRC_TRUNCATED_MSG_FAILED	2080	X' 00000820'
MQRC_UNKNOWN_ALIAS_BASE_Q	2082	X' 00000822'
MQRC_UNKNOWN_OBJECT_NAME	2085	X' 00000825'
MQRC_UNKNOWN_OBJECT_Q_MGR	2086	X' 00000826'
MQRC_UNKNOWN_REMOTE_Q_MGR	2087	X' 00000827'
MQRC_WAIT_INTERVAL_ERROR	2090	X' 0000082A'
MQRC_XMIT_Q_TYPE_ERROR	2091	X' 0000082B'
MQRC_XMIT_Q_USAGE_ERROR	2092	X' 0000082C'
MQRC_NOT_OPEN_FOR_PASS_ALL	2093	X' 0000082D'
MQRC_NOT_OPEN_FOR_PASS_IDENT	2094	X' 0000082E'
MQRC_NOT_OPEN_FOR_SET_ALL	2095	X' 0000082F'
MQRC_NOT_OPEN_FOR_SET_IDENT	2096	X' 00000830'
MQRC_CONTEXT_HANDLE_ERROR	2097	X' 00000831'
MQRC_CONTEXT_NOT_AVAILABLE	2098	X' 00000832'
MQRC_SIGNAL1_ERROR	2099	X' 00000833'
MQRC_OBJECT_ALREADY_EXISTS	2100	X' 00000834'
MQRC_OBJECT_DAMAGED	2101	X' 00000835'
MQRC_RESOURCE_PROBLEM	2102	X' 00000836'
MQRC_ANOTHER_Q_MGR_CONNECTED	2103	X' 00000837'
MQRC_UNKNOWN_REPORT_OPTION	2104	X' 00000838'
MQRC_STORAGE_CLASS_ERROR	2105	X' 00000839'
MQRC_COD_NOT_VALID_FOR_XCF_Q	2106	X' 0000083A'
MQRC_XWAIT_CANCELED	2107	X' 0000083B'
MQRC_XWAIT_ERROR	2108	X' 0000083C'
MQRC_SUPPRESSED_BY_EXIT	2109	X' 0000083D'
MQRC_FORMAT_ERROR	2110	X' 0000083E'
MQRC_SOURCE_CCSID_ERROR	2111	X' 0000083F'
MQRC_SOURCE_INTEGER_ENC_ERROR	2112	X' 00000840'
MQRC_SOURCE_DECIMAL_ENC_ERROR	2113	X' 00000841'
MQRC_SOURCE_FLOAT_ENC_ERROR	2114	X' 00000842'
MQRC_TARGET_CCSID_ERROR	2115	X' 00000843'
MQRC_TARGET_INTEGER_ENC_ERROR	2116	X' 00000844'
MQRC_TARGET_DECIMAL_ENC_ERROR	2117	X' 00000845'

MQRC_TARGET_FLOAT_ENC_ERROR	2118	X' 00000846'
MQRC_NOT_CONVERTED	2119	X' 00000847'
MQRC_CONVERTED_MSG_TOO_BIG	2120	X' 00000848'
MQRC_NO_EXTERNAL_PARTICIPANTS	2121	X' 00000849'
MQRC_PARTICIPANT_NOT_AVAILABLE	2122	X' 0000084A'
MQRC_OUTCOME_MIXED	2123	X' 0000084B'
MQRC_OUTCOME_PENDING	2124	X' 0000084C'
MQRC_BRIDGE_STARTED	2125	X' 0000084D'
MQRC_BRIDGE_STOPPED	2126	X' 0000084E'
MQRC_ADAPTER_STORAGE_SHORTAGE	2127	X' 0000084F'
MQRC_UOW_IN_PROGRESS	2128	X' 00000850'
MQRC_ADAPTER_CONN_LOAD_ERROR	2129	X' 00000851'
MQRC_ADAPTER_SERV_LOAD_ERROR	2130	X' 00000852'
MQRC_ADAPTER_DEFS_ERROR	2131	X' 00000853'
MQRC_ADAPTER_DEFS_LOAD_ERROR	2132	X' 00000854'
MQRC_ADAPTER_CONV_LOAD_ERROR	2133	X' 00000855'
MQRC_BO_ERROR	2134	X' 00000856'
MQRC_DH_ERROR	2135	X' 00000857'
MQRC_MULTIPLE_REASONS	2136	X' 00000858'
MQRC_OPEN_FAILED	2137	X' 00000859'
MQRC_ADAPTER_DISC_LOAD_ERROR	2138	X' 0000085A'
MQRC_CNO_ERROR	2139	X' 0000085B'
MQRC_CICS_WAIT_FAILED	2140	X' 0000085C'
MQRC_DLH_ERROR	2141	X' 0000085D'
MQRC_HEADER_ERROR	2142	X' 0000085E'
MQRC_SOURCE_LENGTH_ERROR	2143	X' 0000085F'
MQRC_TARGET_LENGTH_ERROR	2144	X' 00000860'
MQRC_SOURCE_BUFFER_ERROR	2145	X' 00000861'
MQRC_TARGET_BUFFER_ERROR	2146	X' 00000862'
MQRC_IIH_ERROR	2148	X' 00000864'
MQRC_PCF_ERROR	2149	X' 00000865'
MQRC_DBCS_ERROR	2150	X' 00000866'
MQRC_OBJECT_NAME_ERROR	2152	X' 00000868'
MQRC_OBJECT_Q_MGR_NAME_ERROR	2153	X' 00000869'
MQRC_RECS_PRESENT_ERROR	2154	X' 0000086A'
MQRC_OBJECT_RECORDS_ERROR	2155	X' 0000086B'
MQRC_RESPONSE_RECORDS_ERROR	2156	X' 0000086C'
MQRC_ASID_MISMATCH	2157	X' 0000086D'
MQRC_PMO_RECORD_FLAGS_ERROR	2158	X' 0000086E'
MQRC_PUT_MSG_RECORDS_ERROR	2159	X' 0000086F'
MQRC_CONN_ID_IN_USE	2160	X' 00000870'
MQRC_Q_MGR QUIESCING	2161	X' 00000871'
MQRC_Q_MGR STOPPING	2162	X' 00000872'
MQRC_DUPLICATE_RECOV_COORD	2163	X' 00000873'
MQRC_PMO_ERROR	2173	X' 0000087D'
MQRC_API_EXIT_NOT_FOUND	2182	X' 00000886'
MQRC_API_EXIT_LOAD_ERROR	2183	X' 00000887'
MQRC_REMOTE_Q_NAME_ERROR	2184	X' 00000888'
MQRC_INCONSISTENT_PERSISTENCE	2185	X' 00000889'
MQRC_GMO_ERROR	2186	X' 0000088A'
MQRC_CICS_BRIDGE_RESTRICTION	2187	X' 0000088B'
MQRC_STOPPED_BY_CLUSTER_EXIT	2188	X' 0000088C'
MQRC_CLUSTER_RESOLUTION_ERROR	2189	X' 0000088D'
MQRC_CONVERTED_STRING_TOO_BIG	2190	X' 0000088E'
MQRC_TMC_ERROR	2191	X' 0000088F'
MQRC_PAGESET_FULL	2192	X' 00000890'
MQRC_STORAGE_MEDIUM_FULL	2192	X' 00000890'
MQRC_PAGESET_ERROR	2193	X' 00000891'

MQRC_NAME_NOT_VALID_FOR_TYPE	2194	X' 00000892'
MQRC_UNEXPECTED_ERROR	2195	X' 00000893'
MQRC_UNKNOWN_XMIT_Q	2196	X' 00000894'
MQRC_UNKNOWN_DEF_XMIT_Q	2197	X' 00000895'
MQRC_DEF_XMIT_Q_TYPE_ERROR	2198	X' 00000896'
MQRC_DEF_XMIT_Q_USAGE_ERROR	2199	X' 00000897'
MQRC_NAME_IN_USE	2201	X' 00000899'
MQRC_CONNECTION QUIESCING	2202	X' 0000089A'
MQRC_CONNECTION STOPPING	2203	X' 0000089B'
MQRC_ADAPTER_NOT_AVAILABLE	2204	X' 0000089C'
MQRC_MSG_ID_ERROR	2206	X' 0000089D'
MQRC_CORREL_ID_ERROR	2207	X' 0000089E'
MQRC_FILE_SYSTEM_ERROR	2208	X' 0000089F'
MQRC_NO_MSG_LOCKED	2209	X' 000008A1'
MQRC_FILE_NOT_AUDITED	2216	X' 000008A8'
MQRC_CONNECTION NOT AUTHORIZED	2217	X' 000008A9'
MQRC_MSG_TOO_BIG_FOR_CHANNEL	2218	X' 000008AA'
MQRC_CALL_IN_PROGRESS	2219	X' 000008AB'
MQRC_RMH_ERROR	2220	X' 000008AC'
MQRC_Q_MGR_ACTIVE	2222	X' 000008AE'
MQRC_Q_MGR_NOT_ACTIVE	2223	X' 000008AF'
MQRC_Q_DEPTH_HIGH	2224	X' 000008B0'
MQRC_Q_DEPTH_LOW	2225	X' 000008B1'
MQRC_Q_SERVICE_INTERVAL_HIGH	2226	X' 000008B2'
MQRC_Q_SERVICE_INTERVAL_OK	2227	X' 000008B3'
MQRC_UNIT_OF_WORK_NOT_STARTED	2232	X' 000008B8'
MQRC_CHANNEL_AUTO_DEF_OK	2233	X' 000008B9'
MQRC_CHANNEL_AUTO_DEF_ERROR	2234	X' 000008BA'
MQRC_CFH_ERROR	2235	X' 000008BB'
MQRC_CFIL_ERROR	2236	X' 000008BC'
MQRC_CFIN_ERROR	2237	X' 000008BD'
MQRC_CFSL_ERROR	2238	X' 000008BE'
MQRC_CFST_ERROR	2239	X' 000008BF'
MQRC_INCOMPLETE_GROUP	2241	X' 000008C1'
MQRC_INCOMPLETE_MSG	2242	X' 000008C2'
MQRC_INCONSISTENT_CCSDS	2243	X' 000008C3'
MQRC_INCONSISTENT_ENCODINGS	2244	X' 000008C4'
MQRC_INCONSISTENT_UOW	2245	X' 000008C5'
MQRC_INVALID_MSG_UNDER_CURSOR	2246	X' 000008C6'
MQRC_MATCH_OPTIONS_ERROR	2247	x' 000008C7'
MQRC_MDE_ERROR	2248	X' 000008C8'
MQRC_MSG_FLAGS_ERROR	2249	X' 000008C9'
MQRC_MSG_SEQ_NUMBER_ERROR	2250	X' 000008CA'
MQRC_OFFSET_ERROR	2251	X' 000008CB'
MQRC_ORIGINAL_LENGTH_ERROR	2252	X' 000008CC'
MQRC_SEGMENT_LENGTH_ZERO	2253	X' 000008CD'
MQRC_UOW_NOT_AVAILABLE	2255	X' 000008CF'
MQRC_WRONG_GMO_VERSION	2256	X' 000008D0'
MQRC_WRONG_MD_VERSION	2257	X' 000008D1'
MQRC_GROUP_ID_ERROR	2258	X' 000008D2'
MQRC_INCONSISTENT_BROWSE	2259	X' 000008D3'
MQRC_XQH_ERROR	2260	X' 000008D4'
MQRC_SRC_ENV_ERROR	2261	X' 000008D5'
MQRC_SRC_NAME_ERROR	2262	X' 000008D6'
MQRC_DEST_ENV_ERROR	2263	X' 000008D7'
MQRC_DEST_NAME_ERROR	2264	X' 000008D8'
MQRC_TM_ERROR	2265	X' 000008D9'
MQRC_CLUSTER_EXIT_ERROR	2266	X' 000008DA'

MQRC_CLUSTER_EXIT_LOAD_ERROR	2267	X' 000008DB'
MQRC_CLUSTER_PUT_INHIBITED	2268	X' 000008DC'
MQRC_CLUSTER_RESOURCE_ERROR	2269	X' 000008DD'
MQRC_NO_DESTINATIONS_AVAILABLE	2270	X' 000008DE'
MQRC_CONN_TAG_IN_USE	2271	X' 000008DF'
MQRC_PARTIALLY_CONVERTED	2272	X' 000008E0'
MQRC_CONNECTION_ERROR	2273	X' 000008E1'
MQRC_OPTION_ENVIRONMENT_ERROR	2274	X' 000008E2'
MQRC_CD_ERROR	2277	X' 000008E5'
MQRC_CLIENT_CONN_ERROR	2278	X' 000008E6'
MQRC_CHANNEL_STOPPED_BY_USER	2279	X' 000008E7'
MQRC_HCONFIG_ERROR	2280	X' 000008E8'
MQRC_FUNCTION_ERROR	2281	X' 000008E9'
MQRC_CHANNEL_STARTED	2282	X' 000008EA'
MQRC_CHANNEL_STOPPED	2283	X' 000008EB'
MQRC_CHANNEL_CONV_ERROR	2284	X' 000008EC'
MQRC_SERVICE_NOT_AVAILABLE	2285	X' 000008ED'
MQRC_INITIALIZATION_FAILED	2286	X' 000008EE'
MQRC_TERMINATION_FAILED	2287	X' 000008EF'
MQRC_UNKNOWN_Q_NAME	2288	X' 000008F0'
MQRC_SERVICE_ERROR	2289	X' 000008F1'
MQRC_Q_ALREADY_EXISTS	2290	X' 000008F2'
MQRC_USER_ID_NOT_AVAILABLE	2291	X' 000008F3'
MQRC_UNKNOWN_ENTITY	2292	X' 000008F4'
MQRC_UNKNOWN_AUTH_ENTITY	2293	X' 000008F5'
MQRC_UNKNOWN_REF_OBJECT	2294	X' 000008F6'
MQRC_CHANNEL_ACTIVATED	2295	X' 000008F7'
MQRC_CHANNEL_NOT_ACTIVATED	2296	X' 000008F8'
MQRC_UOW_CANCELED	2297	X' 000008F9'
MQRC_FUNCTION_NOT_SUPPORTED	2298	X' 000008FA'
MQRC_SELECTOR_TYPE_ERROR	2299	X' 000008FB'
MQRC_COMMAND_TYPE_ERROR	2300	X' 000008FC'
MQRC_MULTIPLE_INSTANCE_ERROR	2301	X' 000008FD'
MQRC_SYSTEM_ITEM_NOT_ALTERABLE	2302	X' 000008FE'
MQRC_BAG_CONVERSION_ERROR	2303	X' 000008FF'
MQRC_SELECTOR_OUT_OF_RANGE	2304	X' 00000900'
MQRC_SELECTOR_NOT_UNIQUE	2305	X' 00000901'
MQRC_INDEX_NOT_PRESENT	2306	X' 00000902'
MQRC_STRING_ERROR	2307	X' 00000903'
MQRC_ENCODING_NOT_SUPPORTED	2308	X' 00000904'
MQRC_SELECTOR_NOT_PRESENT	2309	X' 00000905'
MQRC_OUT_SELECTOR_ERROR	2310	X' 00000906'
MQRC_STRING_TRUNCATED	2311	X' 00000907'
MQRC_SELECTOR_WRONG_TYPE	2312	X' 00000908'
MQRC_INCONSISTENT_ITEM_TYPE	2313	X' 00000909'
MQRC_INDEX_ERROR	2314	X' 0000090A'
MQRC_SYSTEM_BAG_NOT_ALTERABLE	2315	X' 0000090B'
MQRC_ITEM_COUNT_ERROR	2316	X' 0000090C'
MQRC_FORMAT_NOT_SUPPORTED	2317	X' 0000090D'
MQRC_SELECTOR_NOT_SUPPORTED	2318	X' 0000090E'
MQRC_ITEM_VALUE_ERROR	2319	X' 0000090F'
MQRC_HBAG_ERROR	2320	X' 00000910'
MQRC_PARAMETER_MISSING	2321	X' 00000911'
MQRC_CMD_SERVER_NOT_AVAILABLE	2322	X' 00000912'
MQRC_STRING_LENGTH_ERROR	2323	X' 00000913'
MQRC_INQUIRY_COMMAND_ERROR	2324	X' 00000914'
MQRC_NESTED_BAG_NOT_SUPPORTED	2325	X' 00000915'
MQRC_BAG_WRONG_TYPE	2326	X' 00000916'

MQRC_ITEM_TYPE_ERROR	2327	X' 00000917'
MQRC_SYSTEM_BAG_NOT_DELETABLE	2328	X' 00000918'
MQRC_SYSTEM_ITEM_NOT_DELETABLE	2329	X' 00000919'
MQRC_CODED_CHAR_SET_ID_ERROR	2330	X' 0000091A'
MQRC_MSG_TOKEN_ERROR	2331	X' 0000091B'
MQRC_MISSING_WIH	2332	X' 0000091C'
MQRC_WIH_ERROR	2333	X' 0000091D'
MQRC_RFH_ERROR	2334	X' 0000091E'
MQRC_RFH_STRING_ERROR	2335	X' 0000091F'
MQRC_RFH_COMMAND_ERROR	2336	X' 00000920'
MQRC_RFH_PARM_ERROR	2337	X' 00000921'
MQRC_RFH_DUPLICATE_PARM	2338	X' 00000922'
MQRC_RFH_PARM_MISSING	2339	X' 00000923'
MQRC_CHAR_CONVERSION_ERROR	2340	X' 00000924'
MQRC_UCS2_CONVERSION_ERROR	2341	X' 00000925'
MQRC_DB2_NOT_AVAILABLE	2342	X' 00000926'
MQRC_OBJECT_NOT_UNIQUE	2343	X' 00000927'
MQRC_CONN_TAG_NOT_RELEASED	2344	X' 00000928'
MQRC_CF_NOT_AVAILABLE	2345	X' 00000929'
MQRC_CF_STRUC_IN_USE	2346	X' 0000092A'
MQRC_CF_STRUC_LIST_HDR_IN_USE	2347	X' 0000092B'
MQRC_CF_STRUC_AUTH_FAILED	2348	X' 0000092C'
MQRC_CF_STRUC_ERROR	2349	X' 0000092D'
MQRC_CONN_TAG_NOT_USABLE	2350	X' 0000092E'
MQRC_GLOBAL_UOW_CONFLICT	2351	X' 0000092F'
MQRC_LOCAL_UOW_CONFLICT	2352	X' 00000930'
MQRC_HANDLE_IN_USE_FOR_UOW	2353	X' 00000931'
MQRC_UOW_ENLISTMENT_ERROR	2354	X' 00000932'
MQRC_UOW_MIX_NOT_SUPPORTED	2355	X' 00000933'
MQRC_WXP_ERROR	2356	X' 00000934'
MQRC_CURRENT_RECORD_ERROR	2357	X' 00000935'
MQRC_NEXT_OFFSET_ERROR	2358	X' 00000936'
MQRC_NO_RECORD_AVAILABLE	2359	X' 00000937'
MQRC_OBJECT_LEVEL_INCOMPATIBLE	2360	X' 00000938'
MQRC_NEXT_RECORD_ERROR	2361	X' 00000939'
MQRC_BACKOUT_THRESHOLD_REACHED	2362	X' 0000093A'
MQRC_MSG_NOT_MATCHED	2363	X' 0000093B'
MQRC_JMS_FORMAT_ERROR	2364	X' 0000093C'
MQRC_SEGMENTS_NOT_SUPPORTED	2365	X' 0000093D'
MQRC_WRONG_CF_LEVEL	2366	X' 0000093E'
MQRC_CONFIG_CREATE_OBJECT	2367	X' 0000093F'
MQRC_CONFIG_CHANGE_OBJECT	2368	X' 00000940'
MQRC_CONFIG_DELETE_OBJECT	2369	X' 00000941'
MQRC_CONFIG_REFRESH_OBJECT	2370	X' 00000942'
MQRC_CHANNEL_SSL_ERROR	2371	X' 00000943'
MQRC_CF_STRUC_FAILED	2373	X' 00000945'
MQRC_API_EXIT_ERROR	2374	X' 00000946'
MQRC_API_EXIT_INIT_ERROR	2375	X' 00000947'
MQRC_API_EXIT_TERM_ERROR	2376	X' 00000948'
MQRC_EXIT_REASON_ERROR	2377	X' 00000949'
MQRC_RESERVED_VALUE_ERROR	2378	X' 0000094A'
MQRC_NO_DATA_AVAILABLE	2379	X' 0000094B'
MQRC_SCO_ERROR	2380	X' 0000094C'
MQRC_KEY_REPOSITORY_ERROR	2381	X' 0000094D'
MQRC_CRYPTO_HARDWARE_ERROR	2382	X' 0000094E'
MQRC_AUTH_INFO_REC_COUNT_ERROR	2383	X' 0000094F'
MQRC_AUTH_INFO_REC_ERROR	2384	X' 00000950'
MQRC_AIR_ERROR	2385	X' 00000951'

MQRC_AUTH_INFO_TYPE_ERROR	2386	X'00000952'
MQRC_AUTH_INFO_CONN_NAME_ERROR	2387	X'00000953'
MQRC_LDAP_USER_NAME_ERROR	2388	X'00000954'
MQRC_LDAP_USER_NAME_LENGTH_ERR	2389	X'00000955'
MQRC_LDAP_PASSWORD_ERROR	2390	X'00000956'
MQRC_SSL_ALREADY_INITIALIZED	2391	X'00000957'
MQRC_SSL_CONFIG_ERROR	2392	X'00000958'
MQRC_SSL_INITIALIZATION_ERROR	2393	X'00000959'
MQRC_Q_INDEX_TYPE_ERROR	2394	X'0000095A'
MQRC_SSL_NOT_ALLOWED	2396	X'0000095C'
MQRC_JSSÉ_ERROR	2397	X'0000095D'
MQRC_SSL_PEER_NAME_MISMATCH	2398	X'0000095E'
MQRC_SSL_PEER_NAME_ERROR	2399	X'0000095F'
MQRC_UNSUPPORTED_CIPHER_SUITE	2400	X'00000960'
MQRC_SSL_CERTIFICATE_REVOKED	2401	X'00000961'
MQRC_SSL_CERT_STORE_ERROR	2402	X'00000962'
MQRC_REOPEN_EXCL_INPUT_ERROR	6100	X'000017D4'
MQRC_REOPEN_INQUIRE_ERROR	6101	X'000017D5'
MQRC_REOPEN_SAVED_CONTEXT_ERR	6102	X'000017D6'
MQRC_REOPEN_TEMPORARY_Q_ERROR	6103	X'000017D7'
MQRC_ATTRIBUTE_LOCKED	6104	X'000017D8'
MQRC_CURSOR_NOT_VALID	6105	X'000017D9'
MQRC_ENCODING_ERROR	6106	X'000017DA'
MQRC_STRUC_ID_ERROR	6107	X'000017DB'
MQRC_NULL_POINTER	6108	X'000017DC'
MQRC_NO_CONNECTION_REFERENCE	6109	X'000017DD'
MQRC_NO_BUFFER	6110	X'000017DE'
MQRC_BINARY_DATA_LENGTH_ERROR	6111	X'000017DF'
MQRC_BUFFER_NOT_AUTOMATIC	6112	X'000017E0'
MQRC_INSUFFICIENT_BUFFER	6113	X'000017E1'
MQRC_INSUFFICIENT_DATA	6114	X'000017E2'
MQRC_DATA_TRUNCATED	6115	X'000017E3'
MQRC_ZERO_LENGTH	6116	X'000017E4'
MQRC_NEGATIVE_LENGTH	6117	X'000017E5'
MQRC_NEGATIVE_OFFSET	6118	X'000017E6'
MQRC_INCONSISTENT_FORMAT	6119	X'000017E7'
MQRC_INCONSISTENT_OBJECT_STATE	6120	X'000017E8'
MQRC_CONTEXT_OBJECT_NOT_VALID	6121	X'000017E9'
MQRC_CONTEXT_OPEN_ERROR	6122	X'000017EA'
MQRC_STRUC_LENGTH_ERROR	6123	X'000017EB'
MQRC_NOT_CONNECTED	6124	X'000017EC'
MQRC_NOT_OPEN	6125	X'000017ED'
MQRC_DISTRIBUTION_LIST_EMPTY	6126	X'000017EE'
MQRC_INCONSISTENT_OPEN_OPTIONS	6127	X'000017EF'
MQRC_WRONG_VERSION	6128	X'000017F0'
MQRC_REFERENCE_ERROR	6129	X'000017F1'

- **MQRMH\_STRUC\_\*** (参照メッセージヘッダ構造体識別子)

MQRMH\_STRUC\_\*は、参照メッセージヘッダ構造体識別子を表す定数です。参照メッセージヘッダ構造体識別子については、「1. データタイプ」の「[MQRMH 構造体](#) - [メッセージ参照ヘッダ](#)」で StrucId フィールドを参照してください。次に定数とその値を示します。

MQRMH_STRUC_ID	"RMH "
MQRMH_STRUC_ID_ARRAY	'R','M','H',' '

- **MQRMH\_VERSION\_\*** (構造体バージョン番号)

MQRMH\_VERSION\_\*は、構造体バージョン番号を表す定数です。構造体バージョン番号については、「1. データタイプ」の「MQRMH 構造体 – メッセージ参照ヘッダ」で StrucId フィールドを参照してください。次に定数とその値を示します。

MQRMH_VERSION_1	1	X' 00000001'
MQRMH_CURRENT_VERSION	1	X' 00000001'

- MQRMHF\_\* (参照メッセージヘッダフラグ)

MQRMHF\_\*は、参照メッセージヘッダフラグを表す定数です。参照メッセージヘッダフラグについては、「1. データタイプ」の「MQRMH 構造体 – メッセージ参照ヘッダ」で Flags フィールドを参照してください。次に定数とその値を示します。

MQRMHF_NOT_LAST	0	X' 00000000'
MQRMHF_LAST	1	X' 00000001'

- MQRO\_\* (報告オプション)

MQRO\_\*は、報告オプションを表す定数です。報告オプションについては、「1. データタイプ」の「MQMD 構造体 – メッセージ記述子」で Report フィールドを参照してください。次に定数とその値を示します。

MQRO_COPY_MSG_ID_TO_CORREL_ID	0	X' 00000000'
MQRO_NEW_MSG_ID	0	X' 00000000'
MQRO_DEAD_LETTER_Q	0	X' 00000000'
MQRO_NONE	0	X' 00000000'
MQRO_PAN	1	X' 00000001'
MQRO_NAN	2	X' 00000002'
MQRO_PASS_CORREL_ID	64	X' 00000040'
MQRO_PASS_MSG_ID	128	X' 00000080'
MQRO_COA	256	X' 00000100'
MQRO_COA_WITH_DATA	768	X' 00000300'
MQRO_COA_WITH_FULL_DATA	1792	X' 00000700'
MQRO_COD	2048	X' 00000800'
MQRO_COD_WITH_DATA	6144	X' 00001800'
MQRO_COD_WITH_FULL_DATA	14336	X' 00003800'
MQRO_EXPIRATION	2097152	X' 00200000'
MQRO_EXPIRATION_WITH_DATA	6291456	X' 00600000'
MQRO_EXPIRATION_WITH_FULL_DATA	14680064	X' 00E00000'
MQRO_EXCEPTION	16777216	X' 01000000'
MQRO_EXCEPTION_WITH_DATA	50331648	X' 03000000'
MQRO_EXCEPTION_WITH_FULL_DATA	117440512	X' 07000000'
MQRO_DISCARD_MSG	134217728	X' 08000000'

- MQRO\_\*\_MASK (報告オプションのマスク)

MQRO\_\*\_MASK は、報告オプションのマスクを表す定数です。次に定数とその値を示します。

MQRO_REJECT_UNSUP_MASK	270270464	X' 101C0000'
MQRO_ACCEPT_UNSUP_MASK	-270532353	X' EFE000FF'
MQRO_ACCEPT_UNSUP_IF_XMIT_MASK	261888	X' 0003FF00'

- MQSCO\_\* (キューの有効範囲)

MQSCO\_\*は、キューの有効範囲を表す定数です。キューの有効範囲については、「3. オブジェクトの属性」の「キューの属性」で Scope 属性を参照してください。次に定数とその値を示します。

MQSCO_Q_MGR	1	X'00000001'
MQSCO_CELL	2	X'00000002'

- MQSEG\_\* (セグメント分割)

MQSEG\_\*は、セグメント分割を表す定数です。セグメント分割については、「1. データタイプ」の「MQGMO 構造体 – メッセージ取り出しオプション」で Segmentation フィールドを参照してください。次に定数とその値を示します。

MQSEG_INHIBITED	,
MQSEG_ALLOWED	'A'

- MQSIDT\_\* (代替セキュリティ識別子)

MQSIDT\_\*は、代替セキュリティ識別子を表す定数です。代替セキュリティ識別子については、「1. データタイプ」の「MQOD 構造体 – オブジェクト記述子」で AlternateSecurityId フィールドを参照してください。次に定数とその値を示します。

MQSIDT_NONE	'¥x00'
MQSIDT_NT_SECURITY_ID	'¥x01'

- MQSP\_\* (トランザクション有効性)

MQSP\_\*は、トランザクション有効性を表す定数です。トランザクション有効性については、「3. オブジェクトの属性」の「キューマネージャの属性」で SyncPoint 属性を参照してください。次に定数とその値を示します。

MQSP_NOT_AVAILABLE	0	X'00000000'
MQSP_AVAILABLE	1	X'00000001'

- MQSS\_\* (セグメント状態)

MQSS\_\*は、セグメント状態を表す定数です。セグメント状態については、「1. データタイプ」の「MQGMO 構造体 – メッセージ取り出しオプション」で SegmentStatus フィールドを参照してください。次に定数とその値を示します。

MQSS_NOT_A_SEGMENT	,
MQSS_LAST_SEGMENT	'L'
MQSS_SEGMENT	'S'

- MQTC\_\* (トリガ制御)

MQTC\_\*は、トリガ制御を表す定数です。トリガ制御については、「3. オブジェクトの属性」の「キューの属性」で TriggerControl 属性を参照してください。次に定数とその値を示します。

MQTC_OFF	0	X'00000000'
MQTC_ON	1	X'00000001'

- MQTM\_STRUC\_\* (トリガメッセージの構造体識別子)

MQTM\_STRUC\_\*は、トリガメッセージの構造体識別子を表す定数です。トリガメッセージの構造体識別子については、「1. データタイプ」の「MQTM 構造体 – トリガメッセージ」で StrucId フィールドを参照してください。次に定数とその値を示します。

MQTM_STRUC_ID	"TM "
MQTM_STRUC_ID_ARRAY	'T','M',' ',' ',' '

- MQTM\_VERSION\_\* (トリガメッセージの構造体バージョン番号)

MQTM\_VERSION\_\*は、トリガメッセージの構造体バージョン番号を表す定数です。トリガメッセージの構造体バージョン番号については、「1. データタイプ」の「MQTM 構造体 - トリガメッセージ」で Version フィールドを参照してください。次に定数とその値を示します。

MQTM_VERSION_1	1 X'00000001'
MQTM_CURRENT_VERSION	1 X'00000001'

- MQTT\_\* (トリガタイプ)

MQTT\_\*は、トリガタイプを表す定数です。トリガタイプについては、「3. オブジェクトの属性」の「キューの属性」で TriggerType 属性を参照してください。次に定数とその値を示します。

MQTT_NONE	0 X'00000000'
MQTT_FIRST	1 X'00000001'
MQTT_EVERY	2 X'00000002'
MQTT_DEPTH	3 X'00000003'

- MQUS\_\* (使用種別)

MQUS\_\*は、使用種別を表す定数です。使用種別については、「3. オブジェクトの属性」の「キューの属性」で Usage 属性を参照してください。次に定数とその値を示します。

MQUS_NORMAL	0 X'00000000'
MQUS_TRANSMISSION	1 X'00000001'

- MQWI\_\* (待ち合わせ最大時間)

MQWI\_\*は、待ち合わせ最大時間を表す定数です。待ち合わせ最大時間については、「1. データタイプ」の「MQGMO 構造体 - メッセージ取り出しオプション」で WaitInterval フィールドを参照してください。次に定数とその値を示します。

MQWI_UNLIMITED	-1 X'FFFFFFFF'
----------------	----------------

- MQXDR\_\* (データ変換出力応答)

MQXDR\_\*は、データ変換出力応答を表す定数です。次に定数とその値を示します。

MQXDR_OK	0 X'00000000'
MQXDR_CONVERSION_FAILED	1 X'00000001'

- MQXQH\_\* (構造体識別子)

MQXQH\_\*は、構造体識別子を表す定数です。構造体識別子については、「1. データタイプ」の「MQXQH 構造体 - 転送キューヘッダ」を参照してください。次に定数とその値を示します。

MQXQH_STRUC_ID	"XQH "
MQXQH_STRUC_ID_ARRAY	'X','Q',' ','H',' ',' '

- MQXQH\_\* (構造体バージョン番号)

MQXQH\_\*は、構造体バージョン番号を表す定数です。構造体バージョン番号については、「1. データタイプ」の「MQXQH 構造体 - 転送キューヘッダ」を参照してください。次に定数とその値を示します。

MQXQH_VERSION_1	1	X'00000001'
MQXQH_CURRENT_VERSION	1	X'00000001'

## 付録 G コード変換対象の文字セット識別子一覧

ここでは、文字セット識別子と各 OS でサポートされている文字コードセットとの対応、および日立コード変換を使用した場合の文字コードの対応について説明します。

### 付録 G.1 文字セット識別子と各 OS でサポートされている文字コードセットとの対応

MQGET 命令の MQGMO\_CONVERT オプション、または mqtalcccha (チャンネル定義) の -d オプションで cnvccsid オペランドを指定してコード変換する場合、文字セット識別子は、ここで説明するコード変換対象の文字コードセットで対応づけられた文字コードセットとして変換されます。

HP-UX の場合、次に示す機能によって変換されます。

- 文字セット識別子 1279 以外：OS の機能 (iconv 関数)
- 文字セット識別子 1279：日立コード変換

Windows の場合、日立コード変換の機能によって変換されます。

Linux、または AIX の場合、次に示す機能によって変換されます。

- 文字セット識別子 1200 と 1279：日立コード変換
- 文字セット識別子 1200 と 1279 以外：OS の機能 (iconv 関数)

日立コード変換を使用した場合の変換については、「付録 G.2 日立コード変換を使用した場合の文字コードの対応」を参照してください。日立コード変換については、マニュアル「日立コード変換 ユーザーズガイド」を参照してください。

文字セット識別子と、HP-UX、Windows、Linux、および AIX でサポートされている文字コードセットとの対応を、次の表に示します。

表 G-1 コード変換対象の文字コードセット

文字セット識別子	HP-UX	Windows	Linux	AIX
37	cp037	—	IBM037	IBM-037
277	cp277	—	IBM277	IBM-277
437	cp437	—	IBM437	—
500	cp500	—	IBM500	IBM-500
813	iso88597	—	ISO88597	ISO8859-7
819	iso88591	—	ISO88591	ISO8859-1

文字セット識別子	HP-UX	Windows	Linux	AIX
850	sjis*1	sjis*1	IBM850	IBM-850
852	cp852	—	IBM852	IBM-852
866	cp866	—	IBM866	—
870	cp870	—	IBM870	IBM-870
880	cp880	—	IBM880	IBM-880
897	sjis*1	sjis*1	—	—
912	iso88592	—	ISO88592	ISO8859-2
915	iso88595	—	ISO88595	ISO8859-5
916	iso88598	—	ISO88598	ISO8859-8
920	iso88599	—	ISO88599	ISO8859-9
930	cp930	cp930	IBM930	IBM-930
932	sjis	sjis	SJIS	IBM-932
938	ccdc	—	—	—
939	cp939	cp939	IBM939	IBM-939
950	big5	—	BIG5	big5
954	ujis	ujis	UJIS	IBM-eucJP
964	eucTW	—	EUCTW	IBM-eucTW
970	eucKR	—	EUCKR	IBM-eucKR
1051	roman8	—	ROMAN8	—
1089	iso88596	—	ISO88596	ISO8859-6
1200	—	Unicode	Unicode*2	Unicode*3
1208	—	—	UTF-8	—
1279	KEIS*4	KEIS	KEIS*2	KEIS*3
1381	hp15CN	—	—	—
5026	cp930	cp930	IBM930	IBM-930
5035	cp939	cp939	IBM939	IBM-939
5050	ujis	ujis	UJIS	IBM-eucJP
33722	ujis	ujis	UJIS	IBM-eucJP

(凡例)

—：該当しません。

注※1

文字セット識別子 850 および 897 は 1 バイト文字コードであり、コード変換時には、シフト JIS コードとして扱います。

注※2

文字セット識別子 1200 および 1279 は、32 ビット版の Linux (P-8164-C311) では変換対象外です。

注※3

文字セット識別子 1200 および 1279 は、64 ビット版の AIX (P-1M64-D121) では変換対象外です。

注※4

文字セット識別子 1279 は、64 ビット版の HP-UX (P-1J64-C581) では変換対象外です。

## 付録 G.2 日立コード変換を使用した場合の文字コードの対応

ここでは、日立コード変換を使用した場合の文字コード変換の対応表を、OS 別に説明します。

### (1) HP-UX

HP-UX 使用時の、日立コード変換を使用した場合の文字コード変換の対応を次に示します。

表 G-2 日立コード変換を使用した場合の文字コード変換の対応 (HP-UX)

文字セット 識別子 (変換元)		文字セット識別子 (変換先)												
		850	897	930	932	939	954	120 0	120 8	127 9	502 6	503 5	505 0	337 22
		sjis	sjis	cp9 30	sjis	cp9 39	ujis			KEIS	cp9 30	cp9 39	ujis	ujis
850	sjis	-	-	/	-	/	/	× <sub>3</sub>	× <sub>3</sub>	△	/	/	/	/
897	sjis	-	-	/	-	/	/	× <sub>3</sub>	× <sub>3</sub>	△	/	/	/	/
930	cp9 30	/	/	-	/	/	/	× <sub>3</sub>	× <sub>3</sub>	× <sub>1</sub>	-	/	/	/
932	sjis	-	-	/	-	/	/	× <sub>3</sub>	× <sub>3</sub>	△	/	/	/	/
939	cp9 39	/	/	/	/	-	/	× <sub>3</sub>	× <sub>3</sub>	× <sub>1</sub>	/	-	/	/
954	ujis	/	/	/	/	/	-	× <sub>3</sub>	× <sub>3</sub>	△	/	/	-	-
1200		× <sub>2</sub>	-	× <sub>2</sub>										
1208		× <sub>2</sub>	-	× <sub>2</sub>										
127 9	KEIS	△	△	× <sub>1</sub>	△	× <sub>1</sub>	△	× <sub>3</sub>	× <sub>3</sub>	-	× <sub>1</sub>	× <sub>1</sub>	△	△
502 6	cp9 30	/	/	-	/	/	/	× <sub>3</sub>	× <sub>3</sub>	× <sub>1</sub>	-	/	/	/

文字セット 識別子 (変換元)		文字セット識別子 (変換先)												
		850	897	930	932	939	954	120 0	120 8	127 9	502 6	503 5	505 0	337 22
		sjis	sjis	cp9 30	sjis	cp9 39	ujis			KEIS	cp9 30	cp9 39	ujis	ujis
503 5	cp9 39	/	/	/	/	-	/	× <sub>3</sub>	× <sub>3</sub>	× <sub>1</sub>	/	-	/	/
505 0	ujis	/	/	/	/	/	-	× <sub>3</sub>	× <sub>3</sub>	△	/	/	-	-
337 22	ujis	/	/	/	/	/	-	× <sub>3</sub>	× <sub>3</sub>	△	/	/	-	-

(凡例)

- : 同一文字コードセットのため変換処理は実行されません。
- △ : 変換できます。ただし、64ビット版の TP1/Message Queue の場合は×<sub>2</sub>、または×<sub>3</sub>となります。
- / : OS の機能 (iconv 関数) による変換のため、この表の対象外です。
- ×<sub>1</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_NOT\_CONVERTED」を返却、またはチャネル送信時にエラーを検知します。
- ×<sub>2</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_SOURCE\_CCSID\_ERROR」を返却、またはチャネル送信時にエラーを検知します。
- ×<sub>3</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_TARGET\_CCSID\_ERROR」を返却、またはチャネル送信時にエラーを検知します。

## 注意事項

- 対応表に記載がない文字セット識別子を使った文字コード変換は、OS の機能 (iconv 関数) での変換となります。ただし、表 G-1 の文字コードセットに「-」が記載されているものは、OS の機能 (iconv 関数) での変換もできません (×<sub>2</sub>または×<sub>3</sub>)。
- iconv 関数で変換できるかどうかについては、OS の仕様を確認してください。

## (2) Windows

Windows 使用時の、日立コード変換を使用した場合の文字コード変換を次に示します。

表 G-3 日立コード変換を使用した場合の文字コード変換の対応 (Windows)

文字セット 識別子 (変換元)		文字セット識別子 (変換先)												
		850	897	930	932	939	954	120 0	120 8	127 9	502 6	503 5	505 0	337 22
		sjis	sjis	cp9 30	sjis	cp9 39	ujis	Unic ode	UTF- 8	KEIS	cp9 30	cp9 39	ujis	ujis
850	sjis	-	-	○	-	○	○	○	×	○	○	○	○	○

文字セット 識別子 (変換元)		文字セット識別子 (変換先)												
		850	897	930	932	939	954	120 0	120 8	127 9	502 6	503 5	505 0	337 22
		sjis	sjis	cp9 30	sjis	cp9 39	ujis	Unic ode	UTF- 8	KEIS	cp9 30	cp9 39	ujis	ujis
897	sjis	—	—	○	—	○	○	○	×	○	○	○	○	○
930	cp9 30	○	○	—	○	×	○	○	×	×	—	×	○	○
932	sjis	—	—	○	—	○	○	○	×	○	○	○	○	○
939	cp9 39	○	○	×	○	—	○	○	×	×	×	—	○	○
954	ujis	○	○	○	○	○	—	○	×	○	○	○	—	—
120 0	Unic ode	○	○	○	○	○	○	—	×	○	○	○	○	○
120 8	UTF- 8	×	×	×	×	×	×	×	—	×	×	×	×	×
127 9	KEIS	○	○	×	○	×	○	○	×	—	×	×	○	○
502 6	cp9 30	○	○	—	○	×	○	○	×	×	—	×	○	○
503 5	cp9 39	○	○	×	○	—	○	○	×	×	×	—	○	○
505 0	ujis	○	○	○	○	○	—	○	×	○	○	○	—	—
337 22	ujis	○	○	○	○	○	—	○	×	○	○	○	—	—

(凡例)

—：同一文字コードセットのため変換処理は実行されません。

○：変換できます。

×：変換できません。MQGET 命令が理由コード「MQRC\_NOT\_CONVERTED」を返却，またはチャネル送信時にエラーを検知します。

## ■ 注意事項

対応表に記載がない文字セット識別子を使った文字コード変換はできません (×)。

## (3) Linux

Linux 使用時の，日立コード変換を使用した場合の文字コード変換を次に示します。

表 G-4 日立コード変換を使用した場合の文字コード変換の対応 (Linux)

文字セット 識別子 (変換元)		文字セット識別子 (変換先)													
		850	897	930	932	939	954	120 0	120 8	127 9	502 6	503 5	505 0	337 22	
		IBM 850		IBM 930	SJIS	IBM 939	UJIS	Unic ode	UTF- 8	KEIS	IBM 930	IBM 939	UJIS	UJIS	
850	IBM 850	-	× <sub>3</sub>	/	/	/	/	× <sub>1</sub>	/	× <sub>1</sub>	/	/	/	/	
897		× <sub>2</sub>	-	× <sub>2</sub>											
930	IBM 930	/	× <sub>3</sub>	-	/	/	/	△	/	× <sub>1</sub>	-	/	/	/	
932	SJIS	/	× <sub>3</sub>	/	-	/	/	△	/	△	/	/	/	/	
939	IBM 939	/	× <sub>3</sub>	/	/	-	/	△	/	× <sub>1</sub>	/	-	/	/	
954	UJIS	/	× <sub>3</sub>	/	/	/	-	△	/	△	/	/	-	-	
120 0	Unic ode	× <sub>1</sub>	× <sub>3</sub>	△	△	△	△	-	△	△	△	△	△	△	
120 8	UTF- 8	/	× <sub>3</sub>	/	/	/	/	△	-	△	/	/	/	/	
127 9	KEIS	× <sub>1</sub>	× <sub>3</sub>	× <sub>1</sub>	△	× <sub>1</sub>	△	△	△	-	× <sub>1</sub>	× <sub>1</sub>	△	△	
502 6	IBM 930	/	× <sub>3</sub>	-	/	/	/	△	/	× <sub>1</sub>	-	/	/	/	
503 5	IBM 939	/	× <sub>3</sub>	/	/	-	/	△	/	× <sub>1</sub>	/	-	/	/	
505 0	UJIS	/	× <sub>3</sub>	/	/	/	-	△	/	△	/	/	-	-	
337 22	UJIS	/	× <sub>3</sub>	/	/	/	-	△	/	△	/	/	-	-	

(凡例)

- : 同一文字コードセットのため変換処理は実行されません。
- △ : 変換できます。ただし、32ビット版の TP1/Message Queue の場合は×<sub>2</sub>、または×<sub>3</sub>となります。
- / : OS の機能 (iconv 関数) による変換のため、この表の対象外です。
- ×<sub>1</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_NOT\_CONVERTED」を返却、またはチャネル送信時にエラーを検知します。
- ×<sub>2</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_SOURCE\_CCsid\_ERROR」を返却、またはチャネル送信時にエラーを検知します。
- ×<sub>3</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_TARGET\_CCsid\_ERROR」を返却、またはチャネル送信時にエラーを検知します。

## 注意事項

- 対応表に記載がない文字セット識別子を使った文字コード変換は、OS の機能 (iconv 関数) での変換となります。ただし、表 G-1 の文字コードセットに「-」が記載されているものは、OS の機能 (iconv 関数) での変換もできません (×<sub>2</sub> または ×<sub>3</sub>)。
- iconv 関数で変換できるかどうかについては、OS の仕様を確認してください。

## (4) AIX

AIX 使用時の、日立コード変換を使用した場合の文字コード変換を次に示します。

表 G-5 日立コード変換を使用した場合の文字コード変換の対応 (AIX)

文字セット 識別子 (変換元)		文字セット識別子 (変換先)													
		850	897	930	932	939	954	120 0	120 8	127 9	502 6	503 5	505 0	337 22	
		IBM- 850		IBM- 930	IBM- 932	IBM- 939	IBM- eucJ P	Unic ode		KEIS	IBM- 930	IBM- 939	IBM- eucJ P	IBM- eucJ P	
850	IBM- 850	-	× <sub>3</sub>	/	/	/	/	× <sub>1</sub>	× <sub>3</sub>	× <sub>1</sub>	/	/	/	/	
897		× <sub>2</sub>	-	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	
930	IBM- 930	/	× <sub>3</sub>	-	/	/	/	△	× <sub>3</sub>	× <sub>1</sub>	-	/	/	/	
932	IBM- 932	/	× <sub>3</sub>	/	-	/	/	△	× <sub>3</sub>	△	/	/	/	/	
939	IBM- 939	/	× <sub>3</sub>	/	/	-	/	△	× <sub>3</sub>	× <sub>1</sub>	/	-	/	/	
954	IBM- eucJ P	/	× <sub>3</sub>	/	/	/	-	△	× <sub>3</sub>	△	/	/	-	-	
120 0	Unic ode	× <sub>1</sub>	× <sub>3</sub>	△	△	△	△	-	× <sub>3</sub>	△	△	△	△	△	
1208		× <sub>2</sub>	× <sub>2</sub>	-	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>	× <sub>2</sub>						
127 9	KEIS	× <sub>1</sub>	× <sub>3</sub>	× <sub>1</sub>	△	× <sub>1</sub>	△	△	× <sub>3</sub>	-	× <sub>1</sub>	× <sub>1</sub>	△	△	
502 6	IBM- 930	/	× <sub>3</sub>	-	/	/	/	△	× <sub>3</sub>	× <sub>1</sub>	-	/	/	/	
503 5	IBM- 939	/	× <sub>3</sub>	/	/	-	/	△	× <sub>3</sub>	× <sub>1</sub>	/	-	/	/	

文字セット 識別子 (変換元)		文字セット識別子 (変換先)												
		850	897	930	932	939	954	120 0	120 8	127 9	502 6	503 5	505 0	337 22
		IBM- 850		IBM- 930	IBM- 932	IBM- 939	IBM- eucJ P	Unic ode		KEIS	IBM- 930	IBM- 939	IBM- eucJ P	IBM- eucJ P
505 0	IBM- eucJ P	/	× <sub>3</sub>	/	/	/	-	△	× <sub>3</sub>	△	/	/	-	-
337 22	IBM- eucJ P	/	× <sub>3</sub>	/	/	/	-	△	× <sub>3</sub>	△	/	/	-	-

(凡例)

- : 同一文字コードセットのため変換処理は実行されません。
- △ : 変換できます。ただし、64ビット版の TP1/Message Queue の場合は×<sub>2</sub>、または×<sub>3</sub>となります。
- / : OS の機能 (iconv 関数) による変換のため、この表の対象外です。
- ×<sub>1</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_NOT\_CONVERTED」を返却、またはチャネル送信時にエラーを検知します。
- ×<sub>2</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_SOURCE\_CCSID\_ERROR」を返却、またはチャネル送信時にエラーを検知します。
- ×<sub>3</sub> : 変換できません。MQGET 命令が理由コード「MQRC\_TARGET\_CCSID\_ERROR」を返却、またはチャネル送信時にエラーを検知します。

## 注意事項

- 対応表に記載がない文字セット識別子を使った文字コード変換は、OS の機能 (iconv 関数) での変換となります。ただし、表 G-1 の文字コードセットに「-」が記載されているものは、OS の機能 (iconv 関数) での変換もできません (×<sub>2</sub> または ×<sub>3</sub>)。
- iconv 関数で変換できるかどうかについては、OS の仕様を確認してください。

# 索引

## 数字

- 10 進パック形式整数のマシンコード形式 381
- 2 進整数のマシンコード形式 380

## C

- C++での使用 391
- COBOL 言語での実現例 402
- C 言語での実現例 388

## M

- MQACT\_NONE [MQMD 構造体 (AccountingToken)] 76
- MQAT\_AIX [MQMD 構造体 (PutApplType)] 104
- MQAT\_AIX [MQTM 構造体 (ApplType)] 179
- MQAT\_AIX [プロセス定義の属性 (ApplType)] 279
- MQAT\_BROKER [MQMD 構造体 (PutApplType)] 104
- MQAT\_CICS [MQMD 構造体 (PutApplType)] 104
- MQAT\_CICS [MQTM 構造体 (ApplType)] 179
- MQAT\_CICS [プロセス定義の属性 (ApplType)] 279
- MQAT\_DEFAULT [MQMD 構造体 (PutApplType)] 105
- MQAT\_DOS [MQMD 構造体 (PutApplType)] 104
- MQAT\_DOS [MQTM 構造体 (ApplType)] 179
- MQAT\_DOS [プロセス定義の属性 (ApplType)] 279
- MQAT\_DQM [MQMD 構造体 (PutApplType)] 104
- MQAT\_IMS [MQMD 構造体 (PutApplType)] 104
- MQAT\_IMS [MQTM 構造体 (ApplType)] 179
- MQAT\_IMS [プロセス定義の属性 (ApplType)] 279
- MQAT\_JAVA [MQMD 構造体 (PutApplType)] 104
- MQAT\_MVS [MQMD 構造体 (PutApplType)] 104
- MQAT\_MVS [MQTM 構造体 (ApplType)] 179
- MQAT\_MVS [プロセス定義の属性 (ApplType)] 279
- MQAT\_NO\_CONTEXT [MQMD 構造体 (PutApplType)] 105
- MQAT\_OPEN\_TP1 [MQMD 構造体 (PutApplType)] 104
- MQAT\_OPEN\_TP1 [MQTM 構造体 (ApplType)] 179
- MQAT\_OPEN\_TP1 [プロセス定義の属性 (ApplType)] 279
- MQAT\_OS2 [MQMD 構造体 (PutApplType)] 104
- MQAT\_OS2 [MQTM 構造体 (ApplType)] 179
- MQAT\_OS2 [プロセス定義の属性 (ApplType)] 279
- MQAT\_OS400 [MQMD 構造体 (PutApplType)] 104
- MQAT\_OS400 [MQTM 構造体 (ApplType)] 179
- MQAT\_OS400 [プロセス定義の属性 (ApplType)] 280
- MQAT\_QMGR [MQMD 構造体 (PutApplType)] 104
- MQAT\_TMS\_4V [MQMD 構造体 (PutApplType)] 105
- MQAT\_TMS\_4V [MQTM 構造体 (ApplType)] 179
- MQAT\_TMS\_4V [プロセス定義の属性 (ApplType)] 280
- MQAT\_UNIX [MQTM 構造体 (ApplType)] 179
- MQAT\_UNIX [プロセス定義の属性 (ApplType)] 280
- MQAT\_UNKNOWN [MQMD 構造体 (PutApplType)] 105

MQAT\_USER\_FIRST [MQMD 構造体 (PutApplType)] 105

MQAT\_USER\_FIRST [MQTM 構造体 (ApplType)] 179

MQAT\_USER\_FIRST [プロセス定義の属性 (ApplType)] 280

MQAT\_USER\_LAST [MQMD 構造体 (PutApplType)] 105

MQAT\_USER\_LAST [MQTM 構造体 (ApplType)] 180

MQAT\_USER\_LAST [プロセス定義の属性 (ApplType)] 280

MQAT\_WINDOWS\_NT [MQTM 構造体 (ApplType)] 179

MQAT\_WINDOWS\_NT [プロセス定義の属性 (ApplType)] 280

MQAT\_WINDOWS [MQMD 構造体 (PutApplType)] 105

MQAT\_WINDOWS [MQTM 構造体 (ApplType)] 179

MQAT\_WINDOWS [プロセス定義の属性 (ApplType)] 280

MQAT\_WLM [MQMD 構造体 (PutApplType)] 105

MQAT\_XDM [MQMD 構造体 (PutApplType)] 105

MQAT\_XDM [MQTM 構造体 (ApplType)] 179

MQAT\_XDM [プロセス定義の属性 (ApplType)] 280

MQAT\_ZOS [MQMD 構造体 (PutApplType)] 105

MQBND\_BIND\_NOT\_FIXED [キューの属性 (DefBind)] 261

MQBND\_BIND\_ON\_OPEN [キューの属性 (DefBind)] 261

MQCA\_APPL\_ID [MQINQ 命令 (Selectors)] 213

MQCA\_BASE\_Q\_NAME [MQINQ 命令 (Selectors)] 212

MQCA\_CLUSTER\_NAME [MQINQ 命令 (Selectors)] 212

MQCA\_CREATION\_DATE [MQINQ 命令 (Selectors)] 212

MQCA\_CREATION\_TIME [MQINQ 命令 (Selectors)] 212

MQCA\_DEAD\_LETTER\_Q\_NAME [MQINQ 命令 (Selectors)] 213

MQCA\_DEF\_XMIT\_Q\_NAME [MQINQ 命令 (Selectors)] 214

MQCA\_ENV\_DATA [MQINQ 命令 (Selectors)] 213

MQCA\_INITIATION\_Q\_NAME [MQINQ 命令 (Selectors)] 212

MQCA\_PROCESS\_DESC [MQINQ 命令 (Selectors)] 213

MQCA\_PROCESS\_NAME [MQINQ 命令 (Selectors) (キュー用)] 212

MQCA\_PROCESS\_NAME [MQINQ 命令 (Selectors) (プロセス定義用)] 213

MQCA\_Q\_DESC [MQINQ 命令 (Selectors)] 212

MQCA\_Q\_MGR\_DESC [MQINQ 命令 (Selectors)] 214

MQCA\_Q\_MGR\_NAME [MQINQ 命令 (Selectors)] 214

MQCA\_Q\_NAME [MQINQ 命令 (Selectors)] 212

MQCA\_REMOTE\_Q\_MGR\_NAME [MQINQ 命令 (Selectors)] 212

MQCA\_REMOTE\_Q\_NAME [MQINQ 命令 (Selectors)] 212

MQCA\_REPOSITORY\_NAME [MQINQ 命令 (Selectors)] 214

MQCA\_TRIGGER\_DATA [MQINQ 命令 (Selectors)] 212

MQCA\_TRIGGER\_DATA [MQSET 命令 (Selectors)] 251

MQCA\_USER\_DATA [MQINQ 命令 (Selectors)] 213

MQCA\_XMIT\_Q\_NAME [MQINQ 命令 (Selectors)] 212

MQCC\_FAILED 302

MQCC\_OK 302

MQCC\_WARNING 302

MQCCSI\_DEFAULT [MQCFSL 構造体 (CodedCharSetId)] 24

MQCCSI\_DEFAULT [MQCFST 構造体 (CodedCharSetId)] 27

MQCCSI\_DEFAULT [MQMD 構造体 (CodedCharSetId)] 78

MQCCSI\_EMBEDDED [MQMD 構造体 (CodedCharSetId)] 78

MQCCSI\_INHERIT [MQDH 構造体 (CodedCharSetId)] 32

MQCCSI\_INHERIT [MQDLH 構造体 (CodedCharSetId)] 39

MQCCSI\_INHERIT [MQMDE 構造体 (CodedCharSetId)] 127

MQCCSI\_INHERIT [MQMD 構造体 (CodedCharSetId)] 78

MQCCSI\_INHERIT [MQRMH 構造体 (CodedCharSetId)] 169

MQCCSI\_Q\_MGR [MQMD 構造体 (CodedCharSetId)] 78

MQCFC\_LAST [MQCFH 構造体 (Control)] 18

MQCFC\_NOT\_LAST [MQCFH 構造体 (Control)] 18

MQCFH\_CURRENT\_VERSION [MQCFH 構造体 (Version)] 19

MQCFH\_STRUC\_LENGTH [MQCFH 構造体 (StrucLength)] 18

MQCFH\_VERSION\_1 [MQCFH 構造体 (Version)] 19

MQCFH 構造体 17

MQCFIL\_STRUC\_LENGTH\_FIXED [MQCFIL 構造体 (StrucLength)] 21

MQCFIL 構造体 20

MQCFIN\_STRUC\_LENGTH [MQCFIN 構造体 (StrucLength)] 22

MQCFIN 構造体 22

MQCFSL\_STRUC\_LENGTH\_FIXED [MQCFSL 構造体 (StrucLength)] 26

MQCFSL 構造体 24

MQCFST\_STRUC\_LENGTH\_FIXED [MQCFST 構造体 (StrucLength)] 28

MQCFST 構造体 27

MQCFT\_INTEGER\_LIST [MQCFIL 構造体 (Type)] 21

MQCFT\_INTEGER [MQCFIN 構造体 (Type)] 23

MQCFT\_STRING\_LIST [MQCFSL 構造体 (Type)] 26

MQCFT\_STRING [MQCFST 構造体 (Type)] 29

MQCI\_NONE [MQMD 構造体 (CorrelId)] 80

MQCLOSE 命令 191

MQCLOSE 命令の引数一覧 288

MQCO\_DELETE\_PURGE [MQCLOSE 命令 (Options)] 192

MQCO\_DELETE [MQCLOSE 命令 (Options)] 192

MQCO\_NONE [MQCLOSE 命令 (Options)] 191

MQCONN 命令 196

MQCONN 命令の引数一覧 288

MQDH\_CURRENT\_VERSION [MQDH 構造体 (Version)] 35

MQDH\_STRUC\_ID [MQDH 構造体 (StrucId)] 34

MQDH\_VERSION\_1 [MQDH 構造体 (Version)] 35

MQDHF\_NEW\_MSG\_IDS [MQDH 構造体 (Flags)] 32

MQDHF\_NONE [MQDH 構造体 (Flags)] 33

MQDH 構造体 30

MQDH 構造体とそれに続く MQOR 構造体および MQPMR 構造体を足した長さ [MQDH 構造体 (StrucLength)] 34

MQDISC 命令 199

MQDISC 命令の引数一覧 289

MQDL\_NOT\_SUPPORTED [キューの属性 (DistLists)] 265

MQDL\_NOT\_SUPPORTED [キューマネージャの属性 (DistLists)] 283

MQDL\_SUPPORTED [キューの属性 (DistLists)] 264

MQDL\_SUPPORTED [キューマネージャの属性 (DistLists)] 283

MQDLH\_CURRENT\_VERSION [MQDLH 構造体 (Version)] 43

MQDLH\_STRUC\_ID [MQDLH 構造体 (StrucId)] 43

MQDLH\_VERSION\_1 [MQDLH 構造体 (Version)] 43

MQDLH 構造体 36

MQEI\_UNLIMITED [MQMD 構造体 (Expiry)] 83

MQENC\_DECIMAL\_MASK 380

MQENC\_DECIMAL\_NORMAL 381

MQENC\_DECIMAL\_REVERSED 381

MQENC\_DECIMAL\_UNDEFINED 381

MQENC\_FLOAT\_IEEE\_NORMAL 382

MQENC\_FLOAT\_IEEE\_REVERSED 382

MQENC\_FLOAT\_MASK 380

MQENC\_FLOAT\_S390 382

MQENC\_FLOAT\_UNDEFINED 382

MQENC\_INTEGER\_MASK 380

MQENC\_INTEGER\_NORMAL 380

MQENC\_INTEGER\_REVERSED 381

MQENC\_INTEGER\_UNDEFINED 380

MQENC\_NATIVE [MQMD 構造体 (Encoding)] 80

MQENC\_RESERVED\_MASK 380

MQFB\_APPL\_CANNOT\_BE\_STARTED [MQDLH 構造体 (Reason)] 42

MQFB\_APPL\_FIRST [MQMD 構造体 (Feedback)] 84

MQFB\_APPL\_LAST [MQMD 構造体 (Feedback)] 84

MQFB\_APPL\_TYPE\_ERROR [MQDLH 構造体 (Reason)] 42

MQFB\_BIND\_OPEN\_CLUSRCVR\_DEL [MQDLH 構造体 (Reason)] 42

MQFB\_BUFFER\_OVERFLOW [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_APPL\_ABENDED [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_APPL\_NOT\_STARTED [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_BRIDGE\_FAILURE [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_CCSID\_ERROR [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_CIH\_ERROR [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_COMMAREA\_ERROR [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_CORREL\_ID\_ERROR [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_DLQ\_ERROR [MQMD 構造体 (Feedback)] 85

MQFB\_CICS\_ENCODING\_ERROR [MQMD 構造体 (Feedback)] 86

MQFB\_CICS\_INTERNAL\_ERROR [MQMD 構造体 (Feedback)] 86

MQFB\_CICS\_NOT\_AUTHORIZED [MQMD 構造体 (Feedback)] 86

MQFB\_CICS\_UOW\_BACKED\_OUT [MQMD 構造体 (Feedback)] 86

MQFB\_CICS\_UOW\_ERROR [MQMD 構造体 (Feedback)] 86

MQFB\_COA [MQMD 構造体 (Feedback)] 84

MQFB\_COD [MQMD 構造体 (Feedback)] 84

MQFB\_DATA\_LENGTH\_NEGATIVE [MQMD 構造体 (Feedback)] 85

MQFB\_DATA\_LENGTH\_TOO\_BIG [MQMD 構造体 (Feedback)] 85

MQFB\_DATA\_LENGTH\_ZERO [MQMD 構造体 (Feedback)] 85

MQFB\_EXPIRATION [MQMD 構造体 (Feedback)] 84

MQFB\_IIH\_ERROR [MQMD 構造体 (Feedback)] 85

MQFB\_IMS\_ERROR [MQMD 構造体 (Feedback)] 85

MQFB\_IMS\_FIRST [MQMD 構造体 (Feedback)] 85

MQFB\_IMS\_LAST [MQMD 構造体 (Feedback)] 85

MQFB\_LENGTH\_OFF\_BY\_ONE [MQMD 構造体 (Feedback)] 85

MQFB\_NAN [MQMD 構造体 (Feedback)] 84

MQFB\_NONE [MQMD 構造体 (Feedback)] 84

MQFB\_NOT\_A\_REPOSITORY\_MSG [MQDLH 構造体 (Reason)] 42

MQFB\_NOT\_AUTHORIZED\_FOR\_IMS [MQMD 構造体 (Feedback)] 85

MQFB\_PAN [MQMD 構造体 (Feedback)] 84

MQFB\_QUIT [MQMD 構造体 (Feedback)] 84

MQFB\_STOPPED\_BY\_CHAD\_EXIT [MQDLH 構造体 (Reason)] 42

MQFB\_SYSTEM\_FIRST [MQMD 構造体 (Feedback)] 84

MQFB\_SYSTEM\_LAST [MQMD 構造体 (Feedback)] 84

MQFB\_TM\_ERROR [MQDLH 構造体 (Reason)] 42

MQFB\_XMIT\_Q\_MSG\_ERROR [MQDLH 構造体 (Reason)] 42

MQFMT\_ADMIN [MQMD 構造体 (Format)] 87

MQFMT\_DEAD\_LETTER\_HEADER [MQMD 構造体 (Format)] 87

MQFMT\_DIST\_HEADER [MQMD 構造体 (Format)] 88

MQFMT\_EVENT [MQMD 構造体 (Format)] 88

MQFMT\_MD\_EXTENSION [MQMD 構造体 (Format)] 88

MQFMT\_NONE [MQMD 構造体 (Format)] 87

MQFMT\_PCF [MQMD 構造体 (Format)] 88

MQFMT\_REF\_MSG\_HEADER [MQMD 構造体 (Format)] 88

MQFMT\_STRING [MQMD 構造体 (Format)] 89

MQFMT\_TRIGGER [MQMD 構造体 (Format)] 89

MQFMT\_XMIT\_Q\_HEADER [MQMD 構造体 (Format)] 89

MQGET 命令 201

MQGET 命令の引数一覧 289

MQGI\_NONE [MQMD 構造体 (GroupId)] 91

MQGMO\_ACCEPT\_TRUNCATED\_MSG [MQGMO 構造体 (Options)] 59

MQGMO\_ALL\_MSGS\_AVAILABLE [MQGMO 構造体 (Options)] 66

MQGMO\_ALL\_SEGMENTS\_AVAILABLE [MQGMO 構造体 (Options)] 67

MQGMO\_BROWSE\_FIRST [MQGMO 構造体 (Options)] 52

MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR [MQGMO 構造体 (Options)] 55

MQGMO\_BROWSE\_NEXT [MQGMO 構造体 (Options)] 53

MQGMO\_COMPLETE\_MSG [MQGMO 構造体 (Options)] 65

MQGMO\_CONVERT [MQGMO 構造体 (Options)] 59

MQGMO\_CURRENT\_VERSION [MQGMO 構造体 (Version)] 70

MQGMO\_FAIL\_IF QUIESCING [MQGMO 構造体 (Options)] 50

MQGMO\_LOCK [MQGMO 構造体 (Options)] 57

MQGMO\_LOGICAL\_ORDER [MQGMO 構造体 (Options)] 60

MQGMO\_MSG\_UNDER\_CURSOR [MQGMO 構造体 (Options)] 56

MQGMO\_NO\_SYNCPOINT [MQGMO 構造体 (Options)] 50

MQGMO\_NO\_WAIT [MQGMO 構造体 (Options)] 49

MQGMO\_NONE [MQGMO 構造体 (Options)] 68

MQGMO\_STRUC\_ID [MQGMO 構造体 (StrucId)] 70

MQGMO\_SYNCPOINT\_IF\_PERSISTENT [MQGMO 構造体 (Options)] 51

MQGMO\_SYNCPOINT [MQGMO 構造体 (Options)] 50

MQGMO\_UNLOCK [MQGMO 構造体 (Options)] 58

MQGMO\_VERSION\_1 [MQGMO 構造体 (Version)] 70

MQGMO\_VERSION\_2 [MQGMO 構造体 (Version)] 70

MQGMO\_WAIT [MQGMO 構造体 (Options)]  
48

MQGMO 構造体 44

MQGS\_LAST\_MSG\_IN\_GROUP [MQGMO 構造体 (GroupStatus)] 45

MQGS\_MSG\_IN\_GROUP [MQGMO 構造体 (GroupStatus)] 45

MQGS\_NOT\_IN\_GROUP [MQGMO 構造体 (GroupStatus)] 45

MQHC\_UNUSABLE\_HCONN [MQDISC 命令 (Hconn)] 199

MQHO\_UNUSABLE\_HOBJ [MQCLOSE 命令 (Hobj)] 191

MQIA\_APPL\_TYPE [MQINQ 命令 (Selectors)]  
213

MQIA\_CODED\_CHAR\_SET\_ID [MQINQ 命令 (Selectors)] 214

MQIA\_CURRENT\_Q\_DEPTH [MQINQ 命令 (Selectors)] 212

MQIA\_DEF\_BIND [MQINQ 命令 (Selectors)]  
212

MQIA\_DEF\_INPUT\_OPEN\_OPTION [MQINQ 命令 (Selectors)] 212

MQIA\_DEF\_PERSISTENCE [MQINQ 命令 (Selectors)] 212

MQIA\_DEF\_PRIORITY [MQINQ 命令 (Selectors)] 212

MQIA\_DEFINITION\_TYPE [MQINQ 命令 (Selectors)] 212

MQIA\_DIST\_LISTS [MQINQ 命令 (Selectors) (キューマネージャ用)] 214

MQIA\_DIST\_LISTS [MQINQ 命令 (Selectors) (キュー用)] 212

MQIA\_DIST\_LISTS [MQSET 命令 (Selectors)]  
251

MQIA\_HARDEN\_GET\_BACKOUT [MQINQ 命令 (Selectors)] 212

MQIA\_INHIBIT\_GET [MQINQ 命令 (Selectors)]  
212

MQIA\_INHIBIT\_GET [MQSET 命令 (Selectors)]  
251

MQIA\_INHIBIT\_PUT [MQINQ 命令 (Selectors)]  
213

MQIA\_INHIBIT\_PUT [MQSET 命令 (Selectors)]  
251

MQIA\_MAX\_HANDLES [MQINQ 命令 (Selectors)] 214

MQIA\_MAX\_MSG\_LENGTH [MQINQ 命令 (Selectors) (キューマネージャ用)] 214

MQIA\_MAX\_MSG\_LENGTH [MQINQ 命令 (Selectors) (キュー用)] 213

MQIA\_MAX\_PRIORITY [MQINQ 命令 (Selectors)] 214

MQIA\_MAX\_Q\_DEPTH [MQINQ 命令 (Selectors)] 213

MQIA\_MAX\_UNCOMMITTED\_MSGS [MQINQ 命令 (Selectors)] 214

MQIA\_MSG\_DELIVERY\_SEQUENCE [MQINQ 命令 (Selectors)] 213

MQIA\_OPEN\_INPUT\_COUNT [MQINQ 命令 (Selectors)] 213

MQIA\_OPEN\_OUTPUT\_COUNT [MQINQ 命令 (Selectors)] 213

MQIA\_PLATFORM [MQINQ 命令 (Selectors)]  
214

MQIA\_Q\_TYPE [MQINQ 命令 (Selectors)] 213

MQIA\_RETENTION\_INTERVAL [MQINQ 命令 (Selectors)] 213

MQIA\_SCOPE [MQINQ 命令 (Selectors)] 213

MQIA\_SHAREABILITY [MQINQ 命令 (Selectors)] 213

MQIA\_SYNCPOINT [MQINQ 命令 (Selectors)]  
214

MQIA\_TRIGGER\_CONTROL [MQINQ 命令 (Selectors)] 213

MQIA\_TRIGGER\_CONTROL [MQSET 命令 (Selectors)] 251

MQIA\_TRIGGER\_DEPTH [MQINQ 命令 (Selectors)] 213

MQIA\_TRIGGER\_DEPTH [MQSET 命令 (Selectors)] 251

MQIA\_TRIGGER\_INTERVAL [MQINQ 命令 (Selectors)] 214

MQIA\_TRIGGER\_MSG\_PRIORITY [MQINQ 命令 (Selectors)] 213

MQIA\_TRIGGER\_MSG\_PRIORITY [MQSET 命令 (Selectors)] 251

MQIA\_TRIGGER\_TYPE [MQINQ 命令 (Selectors)] 213

MQIA\_TRIGGER\_TYPE [MQSET 命令 (Selectors)] 251

MQIA\_USAGE [MQINQ 命令 (Selectors)] 213

MQINQ 命令 210

MQINQ 命令の引数一覧 293

MQMD\_CURRENT\_VERSION [MQMD 構造体 (Version)] 123

MQMD\_STRUC\_ID [MQMD 構造体 (StrucId)] 121

MQMD\_VERSION\_1 [MQMD 構造体 (Version)] 123

MQMD\_VERSION\_2 [MQMD 構造体 (Version)] 123

MQMDE\_CURRENT\_VERSION [MQMDE 構造体 (Version)] 129

MQMDE\_LENGTH\_2 [MQMDE 構造体 (StrucLength)] 129

MQMDE\_STRUC\_ID [MQMDE 構造体 (StrucId)] 129

MQMDE\_VERSION\_2 [MQMDE 構造体 (Version)] 129

MQMDEF\_NONE [MQMDE 構造体 (Flags)] 127

MQMDE 構造体 124

MQMDS\_FIFO [キューの属性 (MsgDeliverySequence)] 268

MQMDS\_PRIORITY [キューの属性 (MsgDeliverySequence)] 268

MQMD 構造体 72

MQMF\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK 386

MQMF\_ACCEPT\_UNSUP\_MASK 386

MQMF\_LAST\_MSG\_IN\_GROUP [MQMD 構造体 (MsgFlags)] 93

MQMF\_LAST\_SEGMENT [MQMD 構造体 (MsgFlags)] 94

MQMF\_MSG\_IN\_GROUP [MQMD 構造体 (MsgFlags)] 93

MQMF\_NONE [MQMD 構造体 (MsgFlags)] 95

MQMF\_REJECT\_UNSUP\_MASK 386

MQMF\_SEGMENTATION\_ALLOWED [MQMD 構造体 (MsgFlags)] 92

MQMF\_SEGMENTATION\_INHIBITED [MQMD 構造体 (MsgFlags)] 92

MQMF\_SEGMENT [MQMD 構造体 (MsgFlags)] 93

MQMI\_NONE [MQMD 構造体 (MsgId)] 97

MQMO\_MATCH\_CORREL\_ID [MQGMO 構造体 (MatchOptions)] 46

MQMO\_MATCH\_GROUP\_ID [MQGMO 構造体 (MatchOptions)] 47

MQMO\_MATCH\_MSG\_ID [MQGMO 構造体 (MatchOptions)] 46

MQMO\_MATCH\_MSG\_SEQ\_NUMBER [MQGMO 構造体 (MatchOptions)] 47

MQMO\_MATCH\_OFFSET [MQGMO 構造体 (MatchOptions)] 47

MQMO\_NONE [MQGMO 構造体 (MatchOptions)] 47

MQMT\_APPL\_FIRST [MQMD 構造体 (MsgType)] 99

MQMT\_APPL\_LAST [MQMD 構造体 (MsgType)] 99

MQMT\_DATAGRAM [MQMD 構造体 (MsgType)] 98

MQMT\_REPLY [MQMD 構造体 (MsgType)] 98

MQMT\_REPORT [MQMD 構造体 (MsgType)] 98

MQMT\_REQUEST [MQMD 構造体 (MsgType)] 98

MQMT\_SYSTEM\_FIRST [MQMD 構造体 (MsgType)] 98

MQMT\_SYSTEM\_LAST [MQMD 構造体 (MsgType)] 98

MQOD\_CURRENT\_VERSION [MQOD 構造体 (Version)] 140

MQOD\_STRUC\_ID [MQOD 構造体 (StrucId)] 140

MQOD\_VERSION\_1 [MQOD 構造体 (Version)]  
140

MQOD\_VERSION\_2 [MQOD 構造体 (Version)]  
140

MQOD\_VERSION\_3 [MQOD 構造体 (Version)]  
140

MQOD 構造体 130

MQOII\_NONE [MQRMH 構造体  
(ObjectInstanceId)] 172

MQOL\_UNDEFINED [MQMD 構造体  
(OriginalLength)] 100

MQOO\_ALTERNATE\_USER\_AUTHORITY  
[MQOPEN 命令 (Options)] 225

MQOO\_BIND\_AS\_Q\_DEF [MQOPEN 命令  
(Options)] 223

MQOO\_BIND\_NOT\_FIXED [MQOPEN 命令  
(Options)] 222

MQOO\_BIND\_ON\_OPEN [MQOPEN 命令  
(Options)] 222

MQOO\_BROWSE [MQOPEN 命令 (Options)]  
221

MQOO\_FAIL\_IF QUIESCING [MQOPEN 命令  
(Options)] 225

MQOO\_INPUT\_AS\_Q\_DEF [MQOPEN 命令  
(Options)] 220

MQOO\_INPUT\_EXCLUSIVE [MQOPEN 命令  
(Options)] 221

MQOO\_INPUT\_EXCLUSIVE [キューの属性  
(DefInputOpenOption)] 262

MQOO\_INPUT\_SHARED [MQOPEN 命令  
(Options)] 220

MQOO\_INPUT\_SHARED [キューの属性  
(DefInputOpenOption)] 262

MQOO\_INQUIRE [MQOPEN 命令 (Options)]  
222

MQOO\_OUTPUT [MQOPEN 命令 (Options)]  
222

MQOO\_PASS\_ALL\_CONTEXT [MQOPEN 命令  
(Options)] 224

MQOO\_PASS\_IDENTITY\_CONTEXT [MQOPEN  
命令 (Options)] 224

MQOO\_SAVE\_ALL\_CONTEXT [MQOPEN 命令  
(Options)] 223

MQOO\_SET\_ALL\_CONTEXT [MQOPEN 命令  
(Options)] 225

MQOO\_SET\_IDENTITY\_CONTEXT [MQOPEN  
命令 (Options)] 224

MQOO\_SET [MQOPEN 命令 (Options)] 222

MQOPEN 命令 219

MQOPEN 命令の引数一覧 294

MQOR 構造体 142

MQOR 構造体の数 [MQDH 構造体  
(RecsPresent)] 34

MQOT\_PROCESS [MQOD 構造体  
(ObjectType)] 137

MQOT\_Q\_MGR [MQOD 構造体 (ObjectType)]  
137

MQOT\_Q [MQOD 構造体 (ObjectType)] 137

MQPER\_NOT\_PERSISTENT [MQMD 構造体  
(Persistence)] 101

MQPER\_NOT\_PERSISTENT [キューの属性  
(DefPersistence)] 263

MQPER\_PERSISTENCE\_AS\_Q\_DEF [MQMD 構造  
体 (Persistence)] 101

MQPER\_PERSISTENT [MQMD 構造体  
(Persistence)] 101

MQPER\_PERSISTENT [キューの属性  
(DefPersistence)] 263

MQPL\_AIX [キューマネージャの属性 (Platform)]  
284

MQPL\_MVS [キューマネージャの属性 (Platform)]  
284

MQPL\_OPEN\_TP1 [キューマネージャの属性  
(Platform)] 284

MQPL\_OS2 [キューマネージャの属性 (Platform)]  
284

MQPL\_OS400 [キューマネージャの属性 (Platform)]  
284

MQPL\_TMS\_4V [キューマネージャの属性  
(Platform)] 285

MQPL\_UNIX [キューマネージャの属性 (Platform)]  
284

MQPL\_WINDOWS\_NT [キューマネージャの属性 (Platform)] 285

MQPL\_WINDOWS [キューマネージャの属性 (Platform)] 285

MQPL\_XDM [キューマネージャの属性 (Platform)] 285

MQPMO\_ALTERNATE\_USER\_AUTHORITY [MQPMO 構造体 (Options)] 156

MQPMO\_CURRENT\_VERSION [MQPMO 構造体 (Version)] 163

MQPMO\_DEFAULT\_CONTEXT [MQPMO 構造体 (Options)] 155

MQPMO\_FAIL\_IF\_QUIESCING [MQPMO 構造体 (Options)] 156

MQPMO\_LOGICAL\_ORDER [MQPMO 構造体 (Options)] 149

MQPMO\_NEW\_CORREL\_ID [MQPMO 構造体 (Options)] 148

MQPMO\_NEW\_MSG\_ID [MQPMO 構造体 (Options)] 148

MQPMO\_NO\_CONTEXT [MQPMO 構造体 (Options)] 155

MQPMO\_NO\_SYNCPOINT [MQPMO 構造体 (Options)] 147

MQPMO\_NONE [MQPMO 構造体 (Options)] 157

MQPMO\_PASS\_ALL\_CONTEXT [MQPMO 構造体 (Options)] 155

MQPMO\_PASS\_IDENTITY\_CONTEXT [MQPMO 構造体 (Options)] 155

MQPMO\_SET\_ALL\_CONTEXT [MQPMO 構造体 (Options)] 156

MQPMO\_SET\_IDENTITY\_CONTEXT [MQPMO 構造体 (Options)] 156

MQPMO\_STRUC\_ID [MQPMO 構造体 (StrucId)] 162

MQPMO\_SYNCPOINT [MQPMO 構造体 (Options)] 146

MQPMO\_VERSION\_1 [MQPMO 構造体 (Version)] 163

MQPMO\_VERSION\_2 [MQPMO 構造体 (Version)] 163

MQPMO 構造体 144

MQPMRF\_ACCOUNTING\_TOKEN [MQDH 構造体 (PutMsgRecFields)] 34

MQPMRF\_ACCOUNTING\_TOKEN [MQPMO 構造体 (PutMsgRecFields)] 157

MQPMRF\_CORREL\_ID [MQDH 構造体 (PutMsgRecFields)] 33

MQPMRF\_CORREL\_ID [MQPMO 構造体 (PutMsgRecFields)] 157

MQPMRF\_FEEDBACK [MQDH 構造体 (PutMsgRecFields)] 33

MQPMRF\_FEEDBACK [MQPMO 構造体 (PutMsgRecFields)] 157

MQPMRF\_GROUP\_ID [MQDH 構造体 (PutMsgRecFields)] 33

MQPMRF\_GROUP\_ID [MQPMO 構造体 (PutMsgRecFields)] 157

MQPMRF\_MSG\_ID [MQDH 構造体 (PutMsgRecFields)] 33

MQPMRF\_MSG\_ID [MQPMO 構造体 (PutMsgRecFields)] 157

MQPMRF\_NONE [MQDH 構造体 (PutMsgRecFields)] 34

MQPMRF\_NONE [MQPMO 構造体 (PutMsgRecFields)] 158

MQPMR 構造体 164

MQPMR 構造体フィールドの存在 [MQDH 構造体 (PutMsgRecFields)] 33

MQPMR 構造体フィールドの存在 [MQPMO 構造体 (PutMsgRecFields)] 157

MQPRI\_PRIORITY\_AS\_Q\_DEF [MQMD 構造体 (Priority)] 102

MQPUT1 命令 243

MQPUT 命令 233

MQPUT 命令および MQPUT1 命令の引数一覧 296

MQQA\_BACKOUT\_NOT\_HARDENED [キューの属性 (HardenGetBackout)] 265

MQQA\_GET\_ALLOWED [キューの属性 (InhibitGet)] 266

MQQA\_GET\_INHIBITED [キューの属性 (InhibitGet)] 266

MQQA\_NOT\_SHAREABLE [キューの属性 (Shareability)] 274

MQQA\_PUT\_ALLOWED [キューの属性 (InhibitPut)] 267

MQQA\_PUT\_INHIBITED [キューの属性 (InhibitPut)] 266

MQQA\_SHAREABLE [キューの属性 (Shareability)] 274

MQQDT\_PERMANENT\_DYNAMIC [キューの属性 (DefinitionType)] 261

MQQDT\_PREDEFINED [キューの属性 (DefinitionType)] 261

MQQDT\_TEMPORARY\_DYNAMIC [キューの属性 (DefinitionType)] 261

MQQT\_ALIAS [キューの属性 (QType)] 271

MQQT\_CLUSTER [キューの属性 (BaseQName)] 259

MQQT\_CLUSTER [キューの属性 (QType)] 271

MQQT\_LOCAL [キューの属性 (BaseQName)] 259

MQQT\_LOCAL [キューの属性 (QType)] 271

MQQT\_MODEL [キューの属性 (QType)] 271

MQQT\_REMOTE [キューの属性 (BaseQName)] 259

MQQT\_REMOTE [キューの属性 (QType)] 271

MQRC\_ADAPTER\_CONN\_LOAD\_ERROR 330

MQRC\_ADAPTER\_CONV\_LOAD\_ERROR 331

MQRC\_ADAPTER\_DEFS\_ERROR 330

MQRC\_ADAPTER\_DEFS\_LOAD\_ERROR 331

MQRC\_ADAPTER\_DISC\_LOAD\_ERROR 332

MQRC\_ADAPTER\_NOT\_AVAILABLE 342

MQRC\_ADAPTER\_SERV\_LOAD\_ERROR 330

MQRC\_ADAPTER\_STORAGE\_SHORTAGE 330

MQRC\_AIR\_ERROR 373

MQRC\_ALIAS\_BASE\_Q\_TYPE\_ERROR 303

MQRC\_ALREADY\_CONNECTED 303

MQRC\_ANOTHER\_Q\_MGR\_CONNECTED 324

MQRC\_API\_EXIT\_ERROR 372

MQRC\_API\_EXIT\_INIT\_ERROR 372

MQRC\_API\_EXIT\_LOAD\_ERROR 338

MQRC\_API\_EXIT\_NOT\_FOUND 338

MQRC\_API\_EXIT\_TERM\_ERROR 372

MQRC\_APPL\_FIRST 303

MQRC\_APPL\_LAST 303

MQRC\_ASID\_MISMATCH 336

MQRC\_ATTRIBUTE\_LOCKED 377

MQRC\_AUTH\_INFO\_CONN\_NAME\_ERROR 374

MQRC\_AUTH\_INFO\_REC\_COUNT\_ERROR 373

MQRC\_AUTH\_INFO\_REC\_ERROR 373

MQRC\_AUTH\_INFO\_TYPE\_ERROR 373

MQRC\_BACKED\_OUT 303

MQRC\_BACKOUT\_THRESHOLD\_REACHED 370

MQRC\_BAG\_CONVERSION\_ERROR 359

MQRC\_BAG\_WRONG\_TYPE 364

MQRC\_BINARY\_DATA\_LENGTH\_ERROR 377

MQRC\_BO\_ERROR 331

MQRC\_BRIDGE\_STARTED 330

MQRC\_BRIDGE\_STOPPED 330

MQRC\_BUFFER\_ERROR 304

MQRC\_BUFFER\_LENGTH\_ERROR 304

MQRC\_BUFFER\_NOT\_AUTOMATIC 377

MQRC\_CALL\_IN\_PROGRESS 343

MQRC\_CD\_ERROR 356

MQRC\_CF\_NOT\_AVAILABLE 367

MQRC\_CF\_STRUC\_AUTH\_FAILED 367

MQRC\_CF\_STRUC\_ERROR 367

MQRC\_CF\_STRUC\_FAILED 371

MQRC\_CF\_STRUC\_IN\_USE 367

MQRC\_CF\_STRUC\_LIST\_HDR\_IN\_USE 367

MQRC\_CFH\_ERROR 345

MQRC\_CFIL\_ERROR 346

MQRC\_CFIN\_ERROR 346

MQRC\_CFSL\_ERROR 346

MQRC\_CFST\_ERROR 346

MQRC\_CHANNEL\_ACTIVATED 358

MQRC\_CHANNEL\_AUTO\_DEF\_ERROR 345

MQRC\_CHANNEL\_AUTO\_DEF\_OK 345

MQRC\_CHANNEL\_CONV\_ERROR 357

MQRC\_CHANNEL\_NOT\_ACTIVATED 358

MQRC\_CHANNEL\_SSL\_ERROR 371

MQRC_CHANNEL_STARTED	357	MQRC_CORREL_ID_ERROR	343
MQRC_CHANNEL_STOPPED	357	MQRC_CRYPTO_HARDWARE_ERROR	373
MQRC_CHANNEL_STOPPED_BY_USER	356	MQRC_CURRENT_RECORD_ERROR	369
MQRC_CHAR_ATTR_LENGTH_ERROR	304	MQRC_CURSOR_NOT_VALID	377
MQRC_CHAR_ATTRS_ERROR	304	MQRC_DATA_LENGTH_ERROR	305
MQRC_CHAR_ATTRS_TOO_SHORT	304	MQRC_DATA_TRUNCATED	378
MQRC_CHAR_CONVERSION_ERROR	367	MQRC_DB2_NOT_AVAILABLE	367
MQRC_CICS_BRIDGE_RESTRICTION	339	MQRC_DBCS_ERROR	334
MQRC_CICS_WAIT_FAILED	332	MQRC_DEF_XMIT_Q_TYPE_ERROR	341
MQRC_CLIENT_CONN_ERROR	356	MQRC_DEF_XMIT_Q_USAGE_ERROR	342
MQRC_CLUSTER_EXIT_ERROR	354	MQRC_DEST_ENV_ERROR	353
MQRC_CLUSTER_EXIT_LOAD_ERROR	354	MQRC_DEST_NAME_ERROR	354
MQRC_CLUSTER_PUT_INHIBITED	355	MQRC_DH_ERROR	331
MQRC_CLUSTER_RESOLUTION_ERROR	339	MQRC_DISTRIBUTION_LIST_EMPTY	378
MQRC_CLUSTER_RESOURCE_ERROR	355	MQRC_DLH_ERROR	332
MQRC_CMD_SERVER_NOT_AVAILABLE	364	MQRC_DUPLICATE_RECOV_COORD	337
MQRC_CNO_ERROR	332	MQRC_DYNAMIC_Q_NAME_ERROR	305
MQRC_COD_NOT_VALID_FOR_XCF_Q	325	MQRC_ENCODING_ERROR	377
MQRC_CODED_CHAR_SET_ID_ERROR	365	MQRC_ENCODING_NOT_SUPPORTED	360
MQRC_COMMAND_TYPE_ERROR	359	MQRC_ENVIRONMENT_ERROR	305
MQRC_CONFIG_CHANGE_OBJECT	371	MQRC_EXIT_REASON_ERROR	372
MQRC_CONFIG_CREATE_OBJECT	371	MQRC_EXPIRY_ERROR	305
MQRC_CONFIG_DELETE_OBJECT	371	MQRC_FEEDBACK_ERROR	306
MQRC_CONFIG_REFRESH_OBJECT	371	MQRC_FILE_NOT_AUDITED	343
MQRC_CONN_ID_IN_USE	337	MQRC_FILE_SYSTEM_ERROR	343
MQRC_CONN_TAG_IN_USE	356	MQRC_FORMAT_ERROR	326
MQRC_CONN_TAG_NOT_RELEASED	367	MQRC_FUNCTION_ERROR	357
MQRC_CONN_TAG_NOT_USABLE	367	MQRC_FUNCTION_NOT_SUPPORTED	358
MQRC_CONNECTION_BROKEN	305	MQRC_GET_INHIBITED	306
MQRC_CONNECTION_ERROR	356	MQRC_GLOBAL_UOW_CONFLICT	367
MQRC_CONNECTION_NOT_AUTHORIZED	343	MQRC_GMO_ERROR	338
MQRC_CONNECTION QUIESCING	342	MQRC_GROUP_ID_ERROR	352
MQRC_CONNECTION_STOPPING	342	MQRC_HANDLE_IN_USE_FOR_UOW	368
MQRC_CONTEXT_HANDLE_ERROR	323	MQRC_HANDLE_NOT_AVAILABLE	306
MQRC_CONTEXT_NOT_AVAILABLE	323	MQRC_HBAG_ERROR	363
MQRC_CONTEXT_OBJECT_NOT_VALID	378	MQRC_HCONFIG_ERROR	357
MQRC_CONTEXT_OPEN_ERROR	378	MQRC_HCONN_ERROR	306
MQRC_CONVERTED_MSG_TOO_BIG	328	MQRC_HEADER_ERROR	333
MQRC_CONVERTED_STRING_TOO_BIG	339	MQRC_HOBJ_ERROR	307

MQRC\_IIH\_ERROR 334  
MQRC\_INCOMPLETE\_GROUP 346  
MQRC\_INCOMPLETE\_MSG 347  
MQRC\_INCONSISTENT\_BROWSE 352  
MQRC\_INCONSISTENT\_CCSDS 347  
MQRC\_INCONSISTENT\_ENCODINGS 347  
MQRC\_INCONSISTENT\_FORMAT 378  
MQRC\_INCONSISTENT\_ITEM\_TYPE 361  
MQRC\_INCONSISTENT\_OBJECT\_STATE 378  
MQRC\_INCONSISTENT\_OPEN\_OPTIONS 378  
MQRC\_INCONSISTENT\_PERSISTENCE 338  
MQRC\_INCONSISTENT\_UOW 348  
MQRC\_INDEX\_ERROR 362  
MQRC\_INDEX\_NOT\_PRESENT 360  
MQRC\_INHIBIT\_VALUE\_ERROR 307  
MQRC\_INITIALIZATION\_FAILED 357  
MQRC\_INQUIRY\_COMMAND\_ERROR 364  
MQRC\_INSUFFICIENT\_BUFFER 377  
MQRC\_INSUFFICIENT\_DATA 377  
MQRC\_INT\_ATTR\_COUNT\_ERROR 307  
MQRC\_INT\_ATTR\_COUNT\_TOO\_SMALL 307  
MQRC\_INT\_ATTRS\_ARRAY\_ERROR 307  
MQRC\_INVALID\_MSG\_UNDER\_CURSOR 348  
MQRC\_ITEM\_COUNT\_ERROR 362  
MQRC\_ITEM\_TYPE\_ERROR 364  
MQRC\_ITEM\_VALUE\_ERROR 363  
MQRC\_JMS\_FORMAT\_ERROR 370  
MQRC\_JSSE\_ERROR 375  
MQRC\_KEY\_REPOSITORY\_ERROR 372  
MQRC\_LDAP\_PASSWORD\_ERROR 374  
MQRC\_LDAP\_USER\_NAME\_ERROR 374  
MQRC\_LDAP\_USER\_NAME\_LENGTH\_ERR 374  
MQRC\_LOCAL\_UOW\_CONFLICT 368  
MQRC\_MATCH\_OPTIONS\_ERROR 349  
MQRC\_MAX\_CONNS\_LIMIT\_REACHED 308  
MQRC\_MD\_ERROR 308  
MQRC\_MDE\_ERROR 349  
MQRC\_MISSING\_REPLY\_TO\_Q 308  
MQRC\_MISSING\_WIH 365  
MQRC\_MSG\_FLAGS\_ERROR 349  
MQRC\_MSG\_ID\_ERROR 342  
MQRC\_MSG\_NOT\_MATCHED 370  
MQRC\_MSG\_SEQ\_NUMBER\_ERROR 350  
MQRC\_MSG\_TOKEN\_ERROR 365  
MQRC\_MSG\_TOO\_BIG\_FOR\_CHANNEL 343  
MQRC\_MSG\_TOO\_BIG\_FOR\_Q 309  
MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR 309  
MQRC\_MSG\_TOO\_BIG\_FOR\_Q\_MGR (MQMD 構造体 (Feedback)) 86  
MQRC\_MSG\_TOO\_BIG\_FOR\_Q (MQMD 構造体 (Feedback)) 86  
MQRC\_MSG\_TYPE\_ERROR 308  
MQRC\_MULTIPLE\_INSTANCE\_ERROR 359  
MQRC\_MULTIPLE\_REASONS 331  
MQRC\_NAME\_IN\_USE 342  
MQRC\_NAME\_NOT\_VALID\_FOR\_TYPE 340  
MQRC\_NEGATIVE\_LENGTH 378  
MQRC\_NEGATIVE\_OFFSET 378  
MQRC\_NESTED\_BAG\_NOT\_SUPPORTED 364  
MQRC\_NEXT\_OFFSET\_ERROR 369  
MQRC\_NEXT\_RECORD\_ERROR 370  
MQRC\_NO\_BUFFER 377  
MQRC\_NO\_CONNECTION\_REFERENCE 377  
MQRC\_NO\_DATA\_AVAILABLE 372  
MQRC\_NO\_DESTINATIONS\_AVAILABLE 355  
MQRC\_NO\_EXTERNAL\_PARTICIPANTS 329  
MQRC\_NO\_MSG\_AVAILABLE 309  
MQRC\_NO\_MSG\_LOCKED 343  
MQRC\_NO\_MSG\_UNDER\_CURSOR 310  
MQRC\_NO\_RECORD\_AVAILABLE 370  
MQRC\_NONE 303  
MQRC\_NOT\_AUTHORIZED 310  
MQRC\_NOT\_AUTHORIZED (MQMD 構造体 (Feedback)) 86  
MQRC\_NOT\_CONNECTED 378  
MQRC\_NOT\_CONVERTED 328  
MQRC\_NOT\_OPEN 378  
MQRC\_NOT\_OPEN\_FOR\_BROWSE 310

MQRC\_NOT\_OPEN\_FOR\_INPUT 311  
MQRC\_NOT\_OPEN\_FOR\_INQUIRE 311  
MQRC\_NOT\_OPEN\_FOR\_OUTPUT 311  
MQRC\_NOT\_OPEN\_FOR\_PASS\_ALL 322  
MQRC\_NOT\_OPEN\_FOR\_PASS\_IDENT 323  
MQRC\_NOT\_OPEN\_FOR\_SET 311  
MQRC\_NOT\_OPEN\_FOR\_SET\_ALL 323  
MQRC\_NOT\_OPEN\_FOR\_SET\_IDENT 323  
MQRC\_NULL\_POINTER 377  
MQRC\_OBJECT\_ALREADY\_EXISTS 324  
MQRC\_OBJECT\_CHANGED 311  
MQRC\_OBJECT\_DAMAGED 324  
MQRC\_OBJECT\_IN\_USE 311  
MQRC\_OBJECT\_LEVEL\_INCOMPATIBLE 370  
MQRC\_OBJECT\_NAME\_ERROR 334  
MQRC\_OBJECT\_NOT\_UNIQUE 367  
MQRC\_OBJECT\_Q\_MGR\_NAME\_ERROR 335  
MQRC\_OBJECT\_RECORDS\_ERROR 335  
MQRC\_OBJECT\_TYPE\_ERROR 312  
MQRC\_OD\_ERROR 312  
MQRC\_OFFSET\_ERROR 350  
MQRC\_OPEN\_FAILED 332  
MQRC\_OPTION\_ENVIRONMENT\_ERROR 356  
MQRC\_OPTION\_NOT\_VALID\_FOR\_TYPE 312  
MQRC\_OPTIONS\_ERROR 313  
MQRC\_ORIGINAL\_LENGTH\_ERROR 350  
MQRC\_OUT\_SELECTOR\_ERROR 361  
MQRC\_OUTCOME\_MIXED 329  
MQRC\_OUTCOME\_PENDING 329  
MQRC\_PAGESET\_ERROR 340  
MQRC\_PAGESET\_FULL 340  
MQRC\_PARAMETER\_MISSING 363  
MQRC\_PARTIALLY\_CONVERTED 356  
MQRC\_PARTICIPANT\_NOT\_AVAILABLE 329  
MQRC\_PCF\_ERROR 334  
MQRC\_PERSISTENCE\_ERROR 313  
MQRC\_PERSISTENT\_NOT\_ALLOWED 314  
MQRC\_PERSISTENT\_NOT\_ALLOWED (MQMD  
構造体 (Feedback)) 86  
MQRC\_PMO\_ERROR 337  
MQRC\_PMO\_RECORD\_FLAGS\_ERROR 336  
MQRC\_PRIORITY\_ERROR 314  
MQRC\_PRIORITY\_EXCEEDS\_MAXIMUM 314  
MQRC\_PUT\_INHIBITED 314  
MQRC\_PUT\_INHIBITED (MQMD 構造体  
(Feedback)) 86  
MQRC\_PUT\_MSG\_RECORDS\_ERROR 336  
MQRC\_Q\_ALREADY\_EXISTS 358  
MQRC\_Q\_DELETED 315  
MQRC\_Q\_DEPTH\_HIGH 344  
MQRC\_Q\_DEPTH\_LOW 344  
MQRC\_Q\_FULL 315  
MQRC\_Q\_FULL (MQMD 構造体 (Feedback))  
86  
MQRC\_Q\_INDEX\_TYPE\_ERROR 375  
MQRC\_Q\_MGR\_ACTIVE 344  
MQRC\_Q\_MGR\_NAME\_ERROR 316  
MQRC\_Q\_MGR\_NOT\_ACTIVE 344  
MQRC\_Q\_MGR\_NOT\_AVAILABLE 316  
MQRC\_Q\_MGR QUIESCING 337  
MQRC\_Q\_MGR\_STOPPING 337  
MQRC\_Q\_NOT\_EMPTY 315  
MQRC\_Q\_SERVICE\_INTERVAL\_HIGH 344  
MQRC\_Q\_SERVICE\_INTERVAL\_OK 345  
MQRC\_Q\_SPACE\_NOT\_AVAILABLE 316  
MQRC\_Q\_SPACE\_NOT\_AVAILABLE (MQMD 構  
造体 (Feedback)) 86  
MQRC\_Q\_TYPE\_ERROR 316  
MQRC\_RECS\_PRESENT\_ERROR 335  
MQRC\_REFERENCE\_ERROR 379  
MQRC\_REMOTE\_Q\_NAME\_ERROR 338  
MQRC\_REOPEN\_EXCL\_INPUT\_ERROR 376  
MQRC\_REOPEN\_INQUIRE\_ERROR 376  
MQRC\_REOPEN\_SAVED\_CONTEXT\_ERR 377  
MQRC\_REOPEN\_TEMPORARY\_Q\_ERROR 377  
MQRC\_REPORT\_OPTIONS\_ERROR 317  
MQRC\_RESERVED\_VALUE\_ERROR 372  
MQRC\_RESOURCE\_PROBLEM 324

MQRC_RESPONSE_RECORDS_ERROR	335	MQRC_SSL_CONFIG_ERROR	375
MQRC_RFH_COMMAND_ERROR	366	MQRC_SSL_INITIALIZATION_ERROR	375
MQRC_RFH_DUPLICATE_PARM	366	MQRC_SSL_NOT_ALLOWED	375
MQRC_RFH_ERROR	365	MQRC_SSL_PEER_NAME_ERROR	376
MQRC_RFH_PARM_ERROR	366	MQRC_SSL_PEER_NAME_MISMATCH	375
MQRC_RFH_PARM_MISSING	366	MQRC_STOPPED_BY_CLUSTER_EXIT	339
MQRC_RFH_STRING_ERROR	366	MQRC_STORAGE_CLASS_ERROR	325
MQRC_RMH_ERROR	344	MQRC_STORAGE_MEDIUM_FULL	340
MQRC_SCO_ERROR	372	MQRC_STORAGE_NOT_AVAILABLE	318
MQRC_SECOND_MARK_NOT_ALLOWED	317	MQRC_STRING_ERROR	360
MQRC_SECURITY_ERROR	317	MQRC_STRING_LENGTH_ERROR	364
MQRC_SEGMENT_LENGTH_ZERO	351	MQRC_STRING_TRUNCATED	361
MQRC_SEGMENTS_NOT_SUPPORTED	371	MQRC_STRUC_ID_ERROR	377
MQRC_SELECTOR_COUNT_ERROR	317	MQRC_STRUC_LENGTH_ERROR	378
MQRC_SELECTOR_ERROR	318	MQRC_SUPPRESSED_BY_EXIT	325
MQRC_SELECTOR_LIMIT_EXCEEDED	317	MQRC_SYNCPOINT_LIMIT_REACHED	308
MQRC_SELECTOR_NOT_FOR_TYPE	318	MQRC_SYNCPOINT_NOT_AVAILABLE	319
MQRC_SELECTOR_NOT_PRESENT	361	MQRC_SYSTEM_BAG_NOT_ALTERABLE	362
MQRC_SELECTOR_NOT_SUPPORTED	363	MQRC_SYSTEM_BAG_NOT_DELETABLE	365
MQRC_SELECTOR_NOT_UNIQUE	360	MQRC_SYSTEM_ITEM_NOT_ALTERABLE	359
MQRC_SELECTOR_OUT_OF_RANGE	359	MQRC_SYSTEM_ITEM_NOT_DELETABLE	365
MQRC_SELECTOR_TYPE_ERROR	358	MQRC_TARGET_BUFFER_ERROR	333
MQRC_SELECTOR_WRONG_TYPE	361	MQRC_TARGET_CCSID_ERROR	327
MQRC_SERVICE_ERROR	358	MQRC_TARGET_DECIMAL_ENC_ERROR	327
MQRC_SERVICE_NOT_AVAILABLE	357	MQRC_TARGET_FLOAT_ENC_ERROR	328
MQRC_SIGNAL_OUTSTANDING	318	MQRC_TARGET_INTEGER_ENC_ERROR	327
MQRC_SIGNAL_REQUEST_ACCEPTED	318	MQRC_TARGET_LENGTH_ERROR	333
MQRC_SIGNAL1_ERROR	324	MQRC_TERMINATION_FAILED	357
MQRC_SOURCE_BUFFER_ERROR	333	MQRC_TM_ERROR	354
MQRC_SOURCE_CCSID_ERROR	326	MQRC_TMC_ERROR	340
MQRC_SOURCE_DECIMAL_ENC_ERROR	326	MQRC_TRIGGER_CONTROL_ERROR	319
MQRC_SOURCE_FLOAT_ENC_ERROR	327	MQRC_TRIGGER_DEPTH_ERROR	319
MQRC_SOURCE_INTEGER_ENC_ERROR	326	MQRC_TRIGGER_MSG_PRIORITY_ERR	319
MQRC_SOURCE_LENGTH_ERROR	333	MQRC_TRIGGER_TYPE_ERROR	319
MQRC_SRC_ENV_ERROR	353	MQRC_TRUNCATED_MSG_ACCEPTED	320
MQRC_SRC_NAME_ERROR	353	MQRC_TRUNCATED_MSG_FAILED	320
MQRC_SSL_ALREADY_INITIALIZED	374	MQRC_UCS2_CONVERSION_ERROR	367
MQRC_SSL_CERT_STORE_ERROR	376	MQRC_UNEXPECTED_ERROR	340
MQRC_SSL_CERTIFICATE_REVOKED	376	MQRC_UNIT_OF_WORK_NOT_STARTED	345

MQRC\_UNKNOWN\_ALIAS\_BASE\_Q 320  
 MQRC\_UNKNOWN\_AUTH\_ENTITY 358  
 MQRC\_UNKNOWN\_DEF\_XMIT\_Q 341  
 MQRC\_UNKNOWN\_ENTITY 358  
 MQRC\_UNKNOWN\_OBJECT\_NAME 320  
 MQRC\_UNKNOWN\_OBJECT\_Q\_MGR 321  
 MQRC\_UNKNOWN\_Q\_NAME 357  
 MQRC\_UNKNOWN\_REF\_OBJECT 358  
 MQRC\_UNKNOWN\_REMOTE\_Q\_MGR 321  
 MQRC\_UNKNOWN\_REPORT\_OPTION 325  
 MQRC\_UNKNOWN\_XMIT\_Q 341  
 MQRC\_UNSUPPORTED\_CIPHER\_SUITE 376  
 MQRC\_UOW\_CANCELED 358  
 MQRC\_UOW\_ENLISTMENT\_ERROR 369  
 MQRC\_UOW\_IN\_PROGRESS 330  
 MQRC\_UOW\_MIX\_NOT\_SUPPORTED 369  
 MQRC\_UOW\_NOT\_AVAILABLE 351  
 MQRC\_USER\_ID\_NOT\_AVAILABLE 358  
 MQRC\_WAIT\_INTERVAL\_ERROR 322  
 MQRC\_WIH\_ERROR 365  
 MQRC\_WRONG\_CF\_LEVEL 371  
 MQRC\_WRONG\_GMO\_VERSION 351  
 MQRC\_WRONG\_MD\_VERSION 351  
 MQRC\_WRONG\_VERSION 379  
 MQRC\_WXP\_ERROR 369  
 MQRC\_XMIT\_Q\_TYPE\_ERROR 322  
 MQRC\_XMIT\_Q\_USAGE\_ERROR 322  
 MQRC\_XQH\_ERROR 352  
 MQRC\_XWAIT\_CANCELED 325  
 MQRC\_XWAIT\_ERROR 325  
 MQRC\_ZERO\_LENGTH 378  
 MQRMH\_CURRENT\_VERSION [MQRMH 構造体 (Version)] 174  
 MQRMH\_STRUC\_ID [MQRMH 構造体 (StrucId)] 174  
 MQRMH\_VERSION\_1 [MQRMH 構造体 (Version)] 174  
 MQRMHF\_LAST [MQRMH 構造体 (Flags)] 171  
 MQRMHF\_NOT\_LAST [MQRMH 構造体 (Flags)] 172  
 MQRMH 構造体 167  
 MQRO\_ACCEPT\_UNSUP\_IF\_XMIT\_MASK 384  
 MQRO\_ACCEPT\_UNSUP\_MASK 384  
 MQRO\_COA\_WITH\_DATA [MQMD 構造体 (Report)] 112  
 MQRO\_COA\_WITH\_FULL\_DATA [MQMD 構造体 (Report)] 112  
 MQRO\_COA [MQMD 構造体 (Report)] 112  
 MQRO\_COD\_WITH\_DATA [MQMD 構造体 (Report)] 113  
 MQRO\_COD\_WITH\_FULL\_DATA [MQMD 構造体 (Report)] 113  
 MQRO\_COD [MQMD 構造体 (Report)] 112  
 MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID [MQMD 構造体 (Report)] 114  
 MQRO\_DEAD\_LETTER\_Q [MQMD 構造体 (Report)] 115  
 MQRO\_DISCARD\_MSG [MQMD 構造体 (Report)] 115  
 MQRO\_EXCEPTION\_WITH\_DATA [MQMD 構造体 (Report)] 110  
 MQRO\_EXCEPTION\_WITH\_FULL\_DATA [MQMD 構造体 (Report)] 111  
 MQRO\_EXCEPTION [MQMD 構造体 (Report)] 110  
 MQRO\_EXPIRATION\_WITH\_DATA [MQMD 構造体 (Report)] 111  
 MQRO\_EXPIRATION\_WITH\_FULL\_DATA [MQMD 構造体 (Report)] 111  
 MQRO\_EXPIRATION [MQMD 構造体 (Report)] 111  
 MQRO\_NAN [MQMD 構造体 (Report)] 114  
 MQRO\_NEW\_MSG\_ID [MQMD 構造体 (Report)] 114  
 MQRO\_NONE [MQMD 構造体 (Report)] 115  
 MQRO\_PAN [MQMD 構造体 (Report)] 113  
 MQRO\_PASS\_CORREL\_ID [MQMD 構造体 (Report)] 115  
 MQRO\_PASS\_MSG\_ID [MQMD 構造体 (Report)] 114  
 MQRO\_REJECT\_UNSUP\_MASK 384  
 MQRR 構造体 175

MQSCO\_Q\_MGR [キューの属性 (Scope)] 274  
MQSEG\_ALLOWED [MQGMO 構造体 (Segmentation)] 69  
MQSEG\_INHIBITED [MQGMO 構造体 (Segmentation)] 69  
MQSET 命令 250  
MQSET 命令の引数一覧 301  
MQSP\_AVAILABLE [キューマネージャの属性 (SyncPoint)] 285  
MQSP\_NOT\_AVAILABLE [キューマネージャの属性 (SyncPoint)] 285  
MQSS\_LAST\_SEGMENT [MQGMO 構造体 (SegmentStatus)] 69  
MQSS\_NOT\_A\_SEGMENT [MQGMO 構造体 (SegmentStatus)] 69  
MQSS\_SEGMENT [MQGMO 構造体 (SegmentStatus)] 69  
MQTC\_OFF [キューの属性 (TriggerControl)] 275  
MQTC\_ON [キューの属性 (TriggerControl)] 275  
MQTM\_CURRENT\_VERSION [MQTM 構造体 (Version)] 181  
MQTM\_STRUC\_ID [MQTM 構造体 (StrucId)] 180  
MQTM\_VERSION\_1 [MQTM 構造体 (Version)] 181  
MQTM 構造体 177  
MQTT\_DEPTH [キューの属性 (TriggerType)] 277  
MQTT\_EVERY [キューの属性 (TriggerType)] 277  
MQTT\_FIRST [キューの属性 (TriggerType)] 276  
MQTT\_NONE [キューの属性 (TriggerType)] 276  
MQUS\_NORMAL [キューの属性 (Usage)] 277  
MQUS\_TRANSMISSION [キューの属性 (Usage)] 277  
MQWI\_UNLIMITED [MQGMO 構造体 (WaitInterval)] 71  
MQXQH\_CURRENT\_VERSION [MQXQH 構造体 (Version)] 187

MQXQH\_STRUC\_ID [MQXQH 構造体 (StrucId)] 187  
MQXQH\_VERSION\_1 [MQXQH 構造体 (Version)] 187  
MQXQH 構造体 182

## T

TP1/Message Queue の定数値 420

## あ

あて先オブジェクトのオフセット [MQRMH 構造体 (DestNameOffset)] 171  
あて先オブジェクト名長 [MQRMH 構造体 (DestNameLength)] 171  
あて先環境データオフセット [MQRMH 構造体 (DestEnvOffset)] 170  
あて先環境データ長 [MQRMH 構造体 (DestEnvLength)] 170  
あて先キューマネージャ名 [MQDLH 構造体 (DestQMgrName)] 39  
あて先キュー名 [MQDLH 構造体 (DestQName)] 39  
アプリケーション識別子 [MQTM 構造体 (ApplId)] 179  
アプリケーション識別子 [プロセス定義の属性 (ApplId)] 279  
アプリケーション識別データ [MQMD 構造体 (ApplIdentityData)] 76  
アプリケーションタイプ [MQTM 構造体 (ApplType)] 179  
アプリケーションタイプ [プロセス定義の属性 (ApplType)] 279

## い

一致オプション [MQGMO 構造体 (MatchOptions)] 45  
イニシエーションキュー名 [キューの属性 (InitiationQName)] 267  
イベント発生キュー名 [MQTM 構造体 (QName)] 180

## え

エラーコード 302

## お

応答キューマネージャ名 [MQMD 構造体 (ReplyToQMgr)] 108  
応答キュー名 [MQMD 構造体 (ReplyToQ)] 107  
オープンに失敗したキューの数 [MQOD 構造体 (InvalidDestCount)] 133  
オブジェクトインスタンス識別子 [MQRMH 構造体 (ObjectInstanceId)] 172  
オブジェクトキューマネージャ名 [MQOD 構造体 (ObjectQMgrName)] 134  
オブジェクトキューマネージャ名 [MQOR 構造体 (ObjectQMgrName)] 142  
オブジェクトタイプ [MQOD 構造体 (ObjectType)] 137  
オブジェクトタイプ [MQRMH 構造体 (ObjectType)] 172  
オブジェクト名 [MQOD 構造体 (ObjectName)] 134  
オブジェクト名 [MQOR 構造体 (ObjectName)] 142  
オブジェクトレコード数 [MQOD 構造体 (RecsPresent)] 137  
オフセット [MQMDE 構造体 (Offset)] 128  
オフセット [MQMD 構造体 (Offset)] 99

## か

解決したキューマネージャ名称 [MQOD 構造体 (ResolvedQMgrName)] 137  
解決したキュー名称 [MQOD 構造体 (ResolvedQName)] 138  
課金トークン [MQMD 構造体 (AccountingToken)] 75  
課金トークン [MQPMR 構造体 (AccountingToken)] 165  
環境データ [MQTM 構造体 (EnvData)] 180  
環境データ [プロセス定義の属性 (EnvData)] 280  
完了コード 302  
完了コード [MQCFH 構造体 (CompCode)] 17

完了コード [MQRR 構造体 (CompCode)] 175

## き

基本データタイプ [COBOL 言語での実現例] 409  
基本データタイプ [C 言語での実現例] 394  
キューが属するクラスタの名称 [キューの属性 (ClusterName)] 259  
キュー記述子 [キューの属性 (QDesc)] 270  
キュータイプ [キューの属性 (QType)] 271  
キューの属性 256  
キューの有効範囲 [キューの属性 (Scope)] 273  
キュー保持時間 [キューの属性 (RetentionInterval)] 273  
キューマネージャ記述子 [キューマネージャの属性 (QMgrDesc)] 285  
キューマネージャの属性 282  
キューマネージャ名 [キューマネージャの属性 (QMgrName)] 285  
キュー名 [キューの属性 (QName)] 271

## <

グループ識別子 [MQMDE 構造体 (GroupId)] 128  
グループ識別子 [MQMD 構造体 (GroupId)] 90  
グループ識別子 [MQPMR 構造体 (GroupId)] 166

## こ

構造体識別子 [MQDH 構造体 (StruId)] 34  
構造体識別子 [MQDLH 構造体 (StruId)] 42  
構造体識別子 [MQGMO 構造体 (StruId)] 70  
構造体識別子 [MQMDE 構造体 (StruId)] 129  
構造体識別子 [MQMD 構造体 (StruId)] 121  
構造体識別子 [MQOD 構造体 (StruId)] 140  
構造体識別子 [MQPMO 構造体 (StruId)] 162  
構造体識別子 [MQRMH 構造体 (StruId)] 174  
構造体識別子 [MQTM 構造体 (StruId)] 180  
構造体識別子 [MQXQH 構造体 (StruId)] 187  
構造体タイプ [MQCFH 構造体 (Type)] 19  
構造体タイプ [MQCFIL 構造体 (Type)] 21  
構造体タイプ [MQCFIN 構造体 (Type)] 23  
構造体タイプ [MQCFSL 構造体 (Type)] 26

構造体タイプ [MQCFST 構造体 (Type)] 29  
構造体長 [MQCFH 構造体 (StrucLength)] 18  
構造体長 [MQCFIL 構造体 (StrucLength)] 21  
構造体長 [MQCFIN 構造体 (StrucLength)] 22  
構造体長 [MQCFSL 構造体 (StrucLength)] 25  
構造体長 [MQCFST 構造体 (StrucLength)] 28  
構造体長 [MQMDE 構造体 (StrucLength)] 129  
構造体長 [MQRMH 構造体 (StrucLength)] 174  
構造体バージョン番号 [MQCFH 構造体 (Version)] 19  
構造体バージョン番号 [MQDH 構造体 (Version)] 35  
構造体バージョン番号 [MQDLH 構造体 (Version)] 43  
構造体バージョン番号 [MQGMO 構造体 (Version)] 70  
構造体バージョン番号 [MQMDE 構造体 (Version)] 129  
構造体バージョン番号 [MQMD 構造体 (Version)] 123  
構造体バージョン番号 [MQOD 構造体 (Version)] 140  
構造体バージョン番号 [MQPMO 構造体 (Version)] 163  
構造体バージョン番号 [MQRMH 構造体 (Version)] 174  
構造体バージョン番号 [MQTM 構造体 (Version)] 181  
構造体バージョン番号 [MQXQH 構造体 (Version)] 187  
コード変換対象の文字コードセット 447  
コード変換対象の文字セット識別子一覧 447  
コピーファイル [COBOL 言語での実現例] 402  
コンテキスト [MQPMO 構造体 (Context)] 145

## さ

最初の MQPMR 構造体までのオフセット [MQDH 構造体 (PutMsgRecOffset)] 34  
最初の応答レコードのアドレス [MQOD 構造体 (ResponseRecPtr)] 139

最初の応答レコードのアドレス [MQPMO 構造体 (ResponseRecPtr)] 162  
最初の応答レコードまでのオフセット [MQOD 構造体 (ResponseRecOffset)] 138  
最初の応答レコードまでのオフセット [MQPMO 構造体 (ResponseRecOffset)] 161  
最初のオブジェクトレコードのアドレス [MQOD 構造体 (ObjectRecPtr)] 136  
最初のオブジェクトレコードまでのオフセット [MQDH 構造体 (ObjectRecOffset)] 33  
最初のオブジェクトレコードまでのオフセット [MQOD 構造体 (ObjectRecOffset)] 135  
最初の登録メッセージレコードのアドレス [MQPMO 構造体 (PutMsgRecPtr)] 159  
最初の登録メッセージレコードまでのオフセット [MQPMO 構造体 (PutMsgRecOffset)] 158  
最大ハンドル数 [キューマネージャの属性 (MaxHandles)] 284  
最大メッセージ長 [キューの属性 (MaxMsgLength)] 267  
最大メッセージ長 [キューマネージャの属性 (MaxMsgLength)] 284  
最大メッセージ登録数 [キューの属性 (MaxQDepth)] 268  
最大メッセージ優先度 [キューマネージャの属性 (MaxPriority)] 284  
作成時刻 [キューの属性 (CreationTime)] 260  
作成日付 [キューの属性 (CreationDate)] 259

## し

シグナル [MQGMO 構造体 (Signal1)] 70  
受信キューマネージャ名 [MQPMO 構造体 (ResolvedQMgrName)] 160  
受信キュー名 [MQGMO 構造体 (ResolvedQName)] 69  
受信キュー名 [MQPMO 構造体 (ResolvedQName)] 160  
出力オープン数 [キューの属性 (OpenOutputCount)] 269  
使用種別 [キューの属性 (Usage)] 277  
省略時の転送キュー名 [キューマネージャの属性 (DefXmitQName)] 283

省略時の取り出しオプション [キューの属性 (DefInputOpenOption)] 262  
省略時のメッセージ永続性 [キューの属性 (DefPersistence)] 262  
省略時のメッセージ優先度 [キューの属性 (DefPriority)] 263

## せ

制御オプション [MQCFH 構造体 (Control)] 18  
セグメント分割フラグ [MQGMO 構造体 (Segmentation)] 69

## そ

関連識別子 [MQMD 構造体 (CorrelId)] 79  
関連識別子 [MQPMR 構造体 (CorrelId)] 165  
送信に失敗したキューの数 [MQPMO 構造体 (InvalidDestCount)] 145  
送信元オブジェクトのオフセット [MQRMH 構造体 (SrcNameOffset)] 173  
送信元オブジェクト名長 [MQRMH 構造体 (SrcNameLength)] 173  
送信元環境データオフセット [MQRMH 構造体 (SrcEnvOffset)] 173  
送信元環境データ長 [MQRMH 構造体 (SrcEnvLength)] 173

## た

代替セキュリティ識別子 [MQOD 構造体 (AlternateSecurityId)] 131  
代替ユーザ識別子 [MQOD 構造体 (AlternateUserId)] 132

## ち

長大データ高位オフセット [MQRMH 構造体 (DataLogicalOffset2)] 170  
長大データ長 [MQRMH 構造体 (DataLogicalLength)] 169  
長大データ低位オフセット [MQRMH 構造体 (DataLogicalOffset)] 170

## て

定義タイプ [キューの属性 (DefinitionType)] 261

定数 420

デッドレターキュー名 [キューマネージャの属性 (DeadLetterQName)] 282

デフォルトバインディング [キューの属性 (DefBind)] 261

転送キュー名 [キューの属性 (XmitQName)] 278

## と

動的キュー名 [MQOD 構造体 (DynamicQName)] 133

登録アプリケーションタイプ [MQDLH 構造体 (PutApplType)] 40

登録アプリケーションタイプ [MQMD 構造体 (PutApplType)] 104

登録アプリケーション名 [MQDLH 構造体 (PutApplName)] 40

登録アプリケーション名 [MQMD 構造体 (PutApplName)] 103

登録オプション [MQPMO 構造体 (Options)] 146

登録許可 [キューの属性 (InhibitPut)] 266

登録時刻 [MQDLH 構造体 (PutTime)] 41

登録時刻 [MQMD 構造体 (PutTime)] 107

登録日付 [MQDLH 構造体 (PutDate)] 41

登録日付 [MQMD 構造体 (PutDate)] 106

登録メッセージレコード数または応答レコード数 [MQPMO 構造体 (RecsPresent)] 160

登録元データ [MQMD 構造体 (ApplOriginData)] 77

登録元メッセージ記述子 [MQXQH 構造体 (MsgDesc)] 186

登録元メッセージ長 [MQMDE 構造体 (OriginalLength)] 128

登録元メッセージ長 [MQMD 構造体 (OriginalLength)] 100

トランザクション完了待ち最大メッセージ数 [キューマネージャの属性 (MaxUncommittedMsgs)] 284

トランザクション有効性 [キューマネージャの属性 (SyncPoint)] 285

トリガ間隔 [キューマネージャの属性 (TriggerInterval)] 286

トリガ制御 [キューの属性 (TriggerControl)] 274

トリガタイプ [キューの属性 (TriggerType)] 276

トリガデータ [MQTM 構造体 (TriggerData)] 181  
トリガデータ [キューの属性 (TriggerData)] 275  
トリガのためのメッセージ登録数の下限値 [キューの属性 (TriggerDepth)] 275  
トリガのためのメッセージ優先度の下限値 [キューの属性 (TriggerMsgPriority)] 276  
取り出しオプション [MQGMO 構造体 (Options)] 48  
取り出し許可 [キューの属性 (InhibitGet)] 265  
取り出し時の共用性 [キューの属性 (Shareability)] 274

## に

入力オープン数 [キューの属性 (OpenInputCount)] 269

## は

配布リストのサポート [キューの属性 (DistLists)] 264  
配布リストのサポート [キューマネージャの属性 (DistLists)] 283

## ひ

引数構造体の数 [MQCFH 構造体 (ParameterCount)] 18  
引数識別子 [MQCFIL 構造体 (Parameter)] 20  
引数識別子 [MQCFIN 構造体 (Parameter)] 22  
引数識別子 [MQCFSL 構造体 (Parameter)] 25  
引数識別子 [MQCFST 構造体 (Parameter)] 28  
引数の値 [MQCFIL 構造体 (Values)] 21  
引数の値 [MQCFIN 構造体 (Value)] 23  
引数の値の数 [MQCFIL 構造体 (Count)] 20  
引数の値の数 [MQCFSL 構造体 (Count)] 24  
日立コード変換を使用した場合の文字コードの対応 449

## ふ

フォーマット名 [MQDH 構造体 (Format)] 33  
フォーマット名 [MQDLH 構造体 (Format)] 40  
フォーマット名 [MQMDE 構造体 (Format)] 128  
フォーマット名 [MQMD 構造体 (Format)] 86

フォーマット名 [MQRMH 構造体 (Format)] 172  
物理メッセージ [MQGMO 構造体 (Options)] 60  
物理メッセージ [MQPMO 構造体 (Options)] 148  
浮動小数点のマシンコード形式 382  
フラグ [MQDH 構造体 (Flags)] 32  
フラグ [MQMDE 構造体 (Flags)] 127  
フラグ [MQRMH 構造体 (Flags)] 171  
プラットフォーム [キューマネージャの属性 (Platform)] 284  
プロセス記述子 [プロセス定義の属性 (ProcessDesc)] 280  
プロセス定義の属性 279  
プロセス名 [MQTM 構造体 (ProcessName)] 180  
プロセス名 [キューの属性 (ProcessName)] 270  
プロセス名 [プロセス定義の属性 (ProcessName)] 281

## へ

ベースキュー名 [キューの属性 (BaseQName)] 258  
ヘッダファイル [C 言語での実現例] 388

## ほ

報告オプション [MQMD 構造体 (Report)] 109  
報告オプションとメッセージフラグ 384  
報告メッセージ返答コード [MQMD 構造体 (Feedback)] 83  
報告メッセージ返答コード [MQPMR 構造体 (Feedback)] 165

## ま

マシンアーキテクチャのマシンコード形式の要約 383  
マシンコード形式 380  
マシンコード形式 [MQDH 構造体 (Encoding)] 32  
マシンコード形式 [MQDLH 構造体 (Encoding)] 39  
マシンコード形式 [MQMDE 構造体 (Encoding)] 127  
マシンコード形式 [MQMD 構造体 (Encoding)] 80  
マシンコード形式 [MQRMH 構造体 (Encoding)] 171

マシンコード形式の作成と検証 382  
待ち合わせ最大時間 [MQGMO 構造体  
(WaitInterval)] 71

## め

命令識別子 [MQCFH 構造体 (Command)] 17  
メッセージ永続性 [MQMD 構造体 (Persistence)]  
101  
メッセージグループ [MQGMO 構造体 (Options)]  
60  
メッセージグループ [MQPMO 構造体 (Options)]  
149  
メッセージグループフラグ [MQGMO 構造体  
(GroupStatus)] 45  
メッセージシーケンス番号 [MQCFH 構造体  
(MsgSeqNumber)] 18  
メッセージシーケンス番号 [MQMDE 構造体  
(MsgSeqNumber)] 128  
メッセージシーケンス番号 [MQMD 構造体  
(MsgSeqNumber)] 97  
メッセージ識別子 [MQMD 構造体 (MsgId)] 95  
メッセージ識別子 [MQPMR 構造体 (MsgId)] 166  
メッセージタイプ [MQMD 構造体 (MsgType)] 98  
メッセージ通知理由 [MQDLH 構造体 (Reason)]  
41  
メッセージ登録数 [キューの属性  
(CurrentQDepth)] 260  
メッセージ配布順序 [キューの属性  
(MsgDeliverySequence)] 268  
メッセージフラグ [MQMDE 構造体 (MsgFlags)]  
128  
メッセージフラグ [MQMD 構造体 (MsgFlags)] 91  
メッセージ保持時間 [MQMD 構造体 (Expiry)] 81  
メッセージ優先度 [MQMD 構造体 (Priority)] 102

## も

文字セット識別子 447  
文字セット識別子 [MQCFSL 構造体  
(CodedCharSetId)] 24  
文字セット識別子 [MQCFST 構造体  
(CodedCharSetId)] 27

文字セット識別子 [MQDH 構造体  
(CodedCharSetId)] 31  
文字セット識別子 [MQDLH 構造体  
(CodedCharSetId)] 39  
文字セット識別子 [MQMDE 構造体  
(CodedCharSetId)] 127  
文字セット識別子 [MQMD 構造体  
(CodedCharSetId)] 77  
文字セット識別子 [MQRMH 構造体  
(CodedCharSetId)] 169  
文字セット識別子 [キューマネージャの属性  
(CodedCharSetId)] 282  
文字列長 [MQCFSL 構造体 (StringLength)] 25  
文字列長 [MQCFST 構造体 (StringLength)] 28  
文字列の値 [MQCFSL 構造体 (Strings)] 25  
文字列の値 [MQCFST 構造体 (String)] 28

## ゆ

ユーザ識別子 [MQMD 構造体 (UserIdentifier)]  
122  
ユーザデータ [MQTM 構造体 (UserData)] 181  
ユーザデータ [プロセス定義の属性 (UserData)]  
281

## り

リモートキューの数 [MQOD 構造体  
(UnknownDestCount)] 140  
リモートキューの数 [MQPMO 構造体  
(UnknownDestCount)] 163  
リモートキューマネージャ名 [MQXQH 構造体  
(RemoteQMgrName)] 186  
リモートキューマネージャ名 [キューの属性  
(RemoteQMgrName)] 271  
リモートキュー名 [MQXQH 構造体  
(RemoteQName)] 187  
リモートキュー名 [キューの属性  
(RemoteQName)] 272  
理由コード 302  
理由コード [MQCFH 構造体 (Reason)] 18  
理由コード [MQRR 構造体 (Reason)] 175  
リンケージ方法 [COBOL 言語での実現例] 404

## ろ

- ローカルキューの数 [MQOD 構造体 (KnownDestCount)] 133
- ローカルキューの数 [MQPMO 構造体 (KnownDestCount)] 146
- ロールバック回数 [MQMD 構造体 (BackoutCount)] 77
- ロールバック回数記録方法 [キューの属性 (HardenGetBackout)] 265
- 論理メッセージ [MQGMO 構造体 (Options)] 60
- 論理メッセージ [MQPMO 構造体 (Options)] 148
- 論理メッセージフラグ [MQGMO 構造体 (SegmentStatus)] 69