

CICS Transaction Server for z/
OSバージョン 5 リリース 6

CICS での *IBM MQ* の使用



注記

本書および本書で紹介する製品をご使用になる前に、[製品の特記事項](#)に記載されている情報をお読みください。

本書は、IBM® CICS® Transaction Server for z/OS®, バージョン 5 リリース 6 (製品番号 5655-Y305655-BTA)、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典：

CICS Transaction Server for z/OS
Version 5 Release 5
Using IBM MQ with CICS

発行：

日本アイ・ビー・エム株式会社

担当：

トランスレーション・サービス・センター

© Copyright International Business Machines Corporation 1974, 2020.

目次

本書について.....	vii
第 1 章 CICS と IBM MQ.....	1
仕組み: CICS-MQ アダプター.....	3
アラート・モニター (CKAM).....	4
タスク・イニシエーターまたはトリガー・モニター (CKTI).....	6
CICS-MQ アダプターに提供されているリソース定義.....	7
仕組み: CICS-MQ ブリッジ.....	7
CICS-MQ ブリッジでの CICS DPL プログラムの実行方法.....	8
CICS-MQ ブリッジでの CICS 3270 トランザクションの実行方法.....	9
仕組み: IBM MQ classes for JMS with CICS.....	10
CICS-MQ トランザクション・トラッキング・サポート.....	12
MQ サポートのための CICS リソース: MQCONN および MQMONITOR.....	13
z/OS ワークロード・マネージャーの正常性サービスの MQMONITOR への影響.....	15
第 2 章 IBM MQ への接続の構成.....	17
CICS-MQ アダプターのセットアップ.....	17
MQCONN リソースの定義とインストール.....	19
MQMONITOR リソースの定義とインストール.....	20
IBM MQ への接続を開始するための PLTPI プログラムの作成.....	22
DFHMQCOD または独自のプログラムを PLTPI リストで指定する.....	22
CICS-MQ ブリッジのセットアップ.....	23
CICS での CICS-MQ ブリッジ・トランザクションおよびプログラムの定義.....	25
CICS-MQ ブリッジ用の MQMONITOR のセットアップ.....	26
複数の CICS-MQ ブリッジ・モニターのセットアップ.....	26
CICS-MQ ブリッジ・スループットの制御.....	27
第 3 章 CICS-MQ アダプターの管理.....	29
CICS-MQ アダプター制御パネルへのアクセス.....	31
コマンド行からの CKQC コマンド.....	31
CICS アプリケーション・プログラムからの CKQC コマンド.....	32
EXEC CICS LINK インターフェース・メッセージ.....	33
CICS-MQ 接続と MQ モニターのための EXEC CICS コマンドおよび CEMT コマンド.....	34
CICS-MQ 接続と MQ モニターのための CICSplex SM ビュー.....	35
CICS-MQ 接続の開始.....	35
CICS-MQ アダプター制御パネルからの CICS-MQ 接続の開始.....	36
CICS コマンド行からの CICS-MQ 接続の開始.....	37
CICS アプリケーション・プログラムから EXEC CICS SET MQCONN コマンドを使用した CICS-MQ 接続の開始.....	38
CICS アプリケーション・プログラムから DFHMQQCN をリンクすることによる CICS-MQ 接続 の開始.....	39
CICS CEMT トランザクションを介した CICS-MQ 接続の開始.....	40
CICSplex SM Web ユーザー・インターフェースを介した CICS-MQ 接続の開始.....	41
CICS-MQ 接続の停止.....	41
CICS-MQ アダプター制御パネルからの CICS-MQ 接続の停止.....	42
CICS コマンド行からの CICS-MQ 接続の停止.....	43
CICS アプリケーション・プログラムから EXEC CICS SET MQCONN を使用した CICS-MQ 接続 の停止.....	43
CICS アプリケーション・プログラムから DFHMQDSC にリンクすることによる CICS-MQ 接続 の停止.....	44

CICS CEMT トランザクションを介した CICS-MQ 接続の停止.....	44
CICSplex SM Web ユーザー・インターフェースを介した CICS-MQ 接続の停止.....	45
CICS-MQ 接続に関する情報の表示.....	45
CICS-MQ 接続状況と設定の表示.....	46
CICS-MQ の接続時刻と切断時刻の表示.....	48
CICS-MQ 接続の統計と呼び出しタイプの表示.....	48
CICS-MQ 接続を使用しているタスクの表示.....	49
CICS MQ モニターの開始.....	50
CICS アプリケーション・プログラムから EXEC CICS SET MQMONITOR を使用した CICS MQ モニターの開始.....	51
CICSplex SM アプリケーション・プログラムから EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START) を使用した CICS MQ モニターの開始.....	52
CICS CEMT トランザクションを介した CICS MQ モニターの開始.....	52
CICSplex SM Web ユーザー・インターフェースを介した CICS MQ モニターの開始.....	53
CICS MQ モニターの停止.....	54
CICS アプリケーション・プログラムから EXEC CICS SET MQMONITOR を使用した CICS MQ モニターの停止.....	54
CICSplex SM アプリケーション・プログラムから EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP) を使用しての CICS MQ モニターの停止.....	55
CICS CEMT トランザクションを介した CICS MQ モニターの停止.....	55
CICSplex SM Web ユーザー・インターフェースを介した CICS MQ モニターの停止.....	55
CICS MQ モニターに関する情報の表示.....	56
CICS-MQ 接続統計のリセット.....	56
CICS-MQ 接続を使用しているタスクのページ.....	57
CICS-MQ トリガー・モニター.....	57
CKTI インスタンスの開始.....	58
CKTI インスタンスの停止.....	63
現在の CKTI インスタンスの表示.....	65
ユーザー作成の MQ トリガー・モニターと MQ メッセージ・コンシューマーの開発と使用.....	66
第 4 章 CICS-MQ ブリッジの管理.....	69
CICS-MQ ブリッジの開始.....	69
CICS-MQ ブリッジの停止.....	71
第 5 章 CICS-MQ アダプターのセキュリティ.....	73
CICS-MQ アダプター・トランザクションのセキュリティの実装.....	73
CICS-MQ アダプター・ユーザー ID.....	74
MQCONN リソースおよび MQMONITOR リソースのコマンド・セキュリティ.....	74
MQMONITOR リソースの代理ユーザー・セキュリティ.....	75
CICS-MQ アダプターの IBM MQ 接続セキュリティ.....	75
第 6 章 CICS-MQ ブリッジのセキュリティ.....	77
第 7 章 CICS-MQ ブリッジを使用するためのアプリケーションの開発.....	81
CICS-MQ ブリッジ用 DPL メッセージ構造.....	82
例: CICS-MQ ブリッジ経由の DPL プログラム要求メッセージ.....	83
CICS-MQ ブリッジ用の 3270 トランザクション・メッセージ構造.....	84
CICS-MQ ブリッジを使用する 3270 トランザクションのベクトル.....	84
CICS-MQ ブリッジを使用する 3270 要求メッセージ (インバウンド) の構造.....	85
CICS-MQ ブリッジを使用する 3270 応答メッセージ (アウトバウンド) の構造.....	86
例: CICS-MQ ブリッジ経由で CEMT INQUIRE TASK を発行する要求メッセージ.....	87
CICS-MQ ブリッジ用に MQMD および MQCIH 構造体の中に設定する必要があるフィールド.....	88
CICS-MQ ブリッジ・メッセージ用の MQMD フィールド.....	88
DPL プログラム要求メッセージ用の MQCIH フィールド.....	89
3270 トランザクション要求メッセージ用の MQCIH フィールド.....	90
DPL プログラム用の MsgId、CorrelId、および UOWControl フィールド.....	92

3270 トランザクション用の MsgId、CorrelId、および UOWControl フィールド	93
分散環境の CICS-MQ ブリッジのデータ変換.....	95
CICS-MQ ブリッジを使用した基本マッピング・サポート (BMS) アプリケーションの実行.....	96
CICS-MQ ブリッジ用の SEND MAP ベクトルの解釈.....	98
CICS-MQ ブリッジ用の RECEIVE MAP ベクトルの解釈.....	101
ADSDL と ADS の例.....	102
CICS-MQ ブリッジで実行される 3270 トランザクションの例.....	104
最適化なしの厳密なエミュレーション	105
最適化を伴う改良されたエミュレーション	106
第 8 章 CICS-MQ アダプターを使用するためのアプリケーションの開発.....	107
IBM MQ MQI 呼び出しにアクセスするための API スタブ・プログラム.....	107
非同期メッセージ・コンシュームとコールバック・ルーチン.....	108
非同期メッセージ・コンシューム用のサンプル・プログラム.....	110
CICS-MQ API 交差出口.....	114
CICS-MQ API 交差出口の呼び出し方法.....	115
CICS-MQ 交差出口プログラムとの通信.....	115
独自の CICS-MQ API 交差出口プログラムの作成.....	116
サンプル API 交差出口プログラム CSQCAPX.....	117
CICS-MQ API 交差出口の使用可能化.....	118
CICS-MQ API 交差出口の使用不可化.....	119
第 9 章 CICS-MQ アダプターのトラブルシューティング	121
IBM MQ の待機.....	121
CICS-MQ 接続のシャットダウン時の処理.....	121
静止 (または正常) シャットダウン	122
強制シャットダウン.....	122
キュー・マネージャーの停止時の処理.....	123
自動再接続と再同期.....	123
CICS-MQ アダプターの再始動時の処理.....	124
CICS による未確定作業単位の解決方法.....	125
手動による CICS 作業単位の解決方法.....	126
トリガー操作が動作しないとき	128
第 10 章 CICS-MQ ブリッジのトラブルシューティング	129
エラーが発生した場合に CICS-MQ ブリッジによって実行されるアクション	130
CICS-MQ ブリッジのデバッグ.....	132
第 11 章 MQCIH – CICS-MQ ブリッジ・ヘッダー.....	135
初期値および言語宣言.....	136
AbendCode (MQCHAR4).....	137
ADSDescriptor (MQLONG).....	138
AttentionId (MQCHAR4).....	138
Authenticator (MQCHAR8).....	138
CancelCode (MQCHAR4).....	138
CodedCharSetId (MQLONG).....	138
CompCode (MQLONG).....	139
ConversationalTask (MQLONG).....	139
CursorPosition (MQLONG).....	139
Encoding (MQLONG).....	139
ErrorOffset (MQLONG).....	139
Facility (MQBYTE8).....	139
FacilityKeepTime (MQLONG).....	140
FacilityLike (MQCHAR4).....	140
Flags (MQLONG).....	140
Format (MQCHAR8).....	140
Function (MQCHAR4).....	141

GetWaitInterval (MQLONG).....	141
InputItem (MQLONG).....	142
LinkType (MQLONG).....	142
NextTransactionId (MQCHAR4).....	142
OutputDataLength (MQLONG).....	142
Reason (MQLONG).....	142
RemoteSysId (MQCHAR4).....	143
RemoteTransId (MQCHAR4).....	143
ReplyToFormat (MQCHAR8).....	143
Reserved1 (MQCHAR8).....	143
Reserved2 (MQCHAR8).....	143
Reserved3 (MQCHAR8).....	143
Reserved4 (MQLONG).....	143
ReturnCode (MQLONG).....	144
StartCode (MQCHAR4).....	144
StrucId (MQCHAR4).....	145
StrucLength (MQLONG).....	145
TaskEndStatus (MQLONG).....	145
TransactionId (MQCHAR4).....	146
UOWControl (MQLONG).....	146
Version (MQLONG).....	146
CICS-MQ ブリッジ用の IBM MQ C++ メッセージ・ヘッダー.....	147
特記事項.....	151
索引.....	157

本書について

本書は、CICS-MQ アダプターおよび CICS-MQ ブリッジの構成と使用、ならびに CICS-MQ 環境の定義と保守に関する概要情報およびガイダンス情報を提供します。

使用されている用語や表記について詳しくは、IBM Knowledge Center の [CICS 資料で使用されている表記規則および用語](#)を参照してください。

本書の作成日

本書は、2020 年 5 月 28 日に作成されました。

第 1 章 CICS と IBM MQ

CICS は、IBM MQ メッセージを使用して CICS の処理を実行するさまざまな方法を提供しています。Java™ アプリケーションからは、IBM MQ classes for Java または IBM MQ classes for JMS を介して IBM MQ にアクセスできます。その他のアプリケーションからは、CICS で提供される 2 つのインターフェース、CICS-MQ アダプターと CICS-MQ ブリッジを介して IBM MQ にアクセスできます。

z/OS 上の IBM MQ の概念およびアーキテクチャーの要約については、IBM MQ の資料の [IBM MQ 製品資料内の『IBM MQ for z/OS の概念』](#)を参照してください。IBM MQ のどのバージョンが CICS で機能するかを確認するには、[IBM MQ のシステム要件](#)を参照して、z/OS 項目で関係する IBM MQ リリースを探してください。

CICS が IBM MQ と連動する仕組み

CICS は、IBM MQ をサポートするための 2 つのリソースを提供します。

- CICS-MQ 接続の属性を定義する MQCONN
- IBM MQ メッセージ・コンシューマーの属性を定義するための MQMONITOR

これらのリソースについては、13 ページの『[MQ サポートのための CICS リソース: MQCONN および MQMONITOR](#)』で詳しく説明しています。

CICS-MQ アダプターを介した IBM MQ へのアクセス

CICS-MQ アダプターは CICS と共に提供され、CICS リソース・マネージャー・インターフェース (RMI) を使用して、外部リソース・マネージャーとして IBM MQ と通信します。これより、MQ メッセージ・トリガー・メカニズムを介して CICS でユーザー・トランザクションを開始できるため、これは一般にはトリガー・モニターと呼ばれます。トリガーが有効になっているアプリケーション・キューにメッセージを入れることができます。メッセージによってトリガーが解放されると、IBM MQ は指定されたユーザー・トランザクションを開始するためのトリガー・メッセージを CICS に送信します。CICS-MQ アダプターには、以下の 2 つの主要な機能があります。

- システム・プログラマーや管理者がアダプターを管理するために使用する一連の制御機能
- CICS アプリケーションに対する Message Queue Interface (MQI) のサポート

CICS-MQ アダプターは、必要に応じて標準の CICS コマンド・レベル・サービス (**EXEC CICS ASSIGN** や **EXEC CICS ABEND** など) を使用します。CICS-MQ アダプターの一部は、メッセージング要求を発行するトランザクションの制御下で稼働します。そのため、CICS サービスに対するこれらの呼び出しは、トランザクションから発行されたかのように見えます。

CICS の複数領域操作環境やシステム間連絡 (ISC) 環境では、各 CICS アドレス・スペースにキュー・マネージャー・サブシステムに対する独自の接続機構を設定できます。1 つの CICS アドレス・スペースが一度に接続できるキュー・マネージャーは 1 つのみです。ただし、各アドレス・スペースはキュー・マネージャー・サブシステムに接続できます。

詳しくは、3 ページの『[仕組み: CICS-MQ アダプター](#)』を参照してください。

CICS-MQ ブリッジを介した IBM MQ へのアクセス

CICS-MQ ブリッジは CICS で提供されます。これにより、ターゲット・プログラムの名前を含むメッセージを IBM MQ キューに送信することで、IBM MQ アプリケーションから CICS システムのアプリケーションに直接アクセスできます。MQ トリガーとは異なり、CICS-MQ ブリッジは、MQ を認識しない CICS アプリケーションへの直接アクセスを提供します。CICS-MQ ブリッジを使用してプログラムまたはトランザクションを CICS 上で実行するとき、ブリッジは暗黙的な Message Queue Interface (MQI) サポートを使用可能にするため、CICS プログラムには、IBM MQ 呼び出しは含まれません。そのため、既存の CICS アプリケーションを書き換え、再コンパイル、または再リンクすることなく、他オペレーティング・システムから IBM MQ メッセージによって制御できるように、それらのアプリケーションを再設計できます。IBM MQ メッセージには、CICS アプリケーションの制御オプションを提供する IBM MQ CICS 情報ヘッダー (MQCIH 構造) が含まれています。

以下のタイプの CICS アプリケーションは、CICS-MQ ブリッジとともに使用するのに適しています。

- **EXEC CICS LINK** コマンドを使用して呼び出される CICS プログラム (分散プログラム・リンク (DPL) プログラムと呼ばれます)。このプログラムは、CICS API の DPL サブセットに規格が合致している必要があります。つまり、CICS 端末機能または同期点機能を使用することはできません。CICS-MQ ブリッジを使用して、単一の CICS プログラム、または作業単位を形成する一連の CICS プログラムを実行できます。
- 3270 端末から実行するように設計された CICS トランザクション (3270 トランザクションと呼ばれます)。トランザクションは、基本マッピング・サポート (BMS) または端末管理コマンドを使用できます。それらは、会話型か、または疑似会話の一部であることが可能です。また、同期点の発行が許可されています。

詳しくは、[CICS-WebSphere MQ ブリッジについて](#)を参照してください。

Java アプリケーションからの IBM MQ へのアクセス

CICS で実行される Java アプリケーションからは、以下の 2 つのインターフェースを介して IBM MQ にアクセスできます。

- IBM MQ classes for JMS の使用
- IBM MQ classes for Java の使用

Java アプリケーションは、以下のいずれかの方法で IBM MQ に接続します。

- MQ クライアント・モードで、TCP/IP を使用して IBM MQ MQI クライアントとして接続
- MQ バインディング・モードで、Java Native Interface (JNI) を使用して直接 IBM MQ に接続

詳しくは、[Java アプリケーションからの IBM MQ へのアクセス](#)を参照してください。IBM MQ 通信タイプについて詳しくは、[IBM MQ 製品資料内の『IBM MQ classes for JMS の接続モード』](#)を参照してください。

IBM MQ classes for JMS

Java Message Service (JMS) は、Java Enterprise Edition (Java EE) 仕様によって定義された API で、アプリケーションが信頼性のある非同期通信を使用してメッセージを送受信できるようにします。これにより、IBM MQ、WebSphere® Liberty が組み込まれた JMS メッセージング・プロバイダー、サード・パーティー・メッセージング・プロバイダーなどの広範なメッセージング・プロバイダーを使用できます。IBM MQ classes for JMS は、`javax.jms` パッケージで定義されたインターフェースの実装に加えて、JMS API への 2 セットの拡張機能を提供します。Java™ プラットフォーム Standard Edition (Java SE) アプリケーションと Java プラットフォーム Enterprise Edition (Java EE) アプリケーションはどちらも、IBM MQ classes for JMS を使用できます。

バージョン 8.0 から、IBM MQ は JMS 2.0 バージョンの JMS 標準をサポートするようになりました。この実装はクラシック API のすべての機能を提供しますが、要求されるインターフェースが少なくなり、より簡単に使用できます。詳しくは、IBM MQ の資料の [JMS モデル](#)、および [Java.net](#) にある JMS 2.0 仕様を参照してください。

IBM MQ classes for JMS は、以下のサーバーで使用できます。

- OSGi JVM サーバー (制限付き)
- JMS アプリケーションがバインディング・モードまたはクライアント・モードのいずれかの転送を使用してキュー・マネージャーに接続する場合は、CICS 標準モードの Liberty JVM サーバー
- JMS アプリケーションがクライアント・モードの転送を使用してキュー・マネージャーに接続する場合は、CICS 統合モードの Liberty JVM サーバー

新規アプリケーションは、IBM MQ classes for Java ではなく、IBM MQ classes for JMS を使用する必要があります。

通信のタイプ、制約事項などの詳細については、[10 ページの『仕組み: IBM MQ classes for JMS with CICS』](#)を参照してください。

IBM MQ classes for Java

IBM MQ classes for Java には、CICS で実行される Java アプリケーションで使用するための IBM Message Queue Interface (MQI) の Java バリエーションが用意されています。IBM MQ への呼び出し要求は MQI 呼び出しに変換され、既存の CICS-MQ アダプターによって通常どおりに処理されます。変換さ

れた要求は、他のプログラム (COBOL プログラムなど) からの MQI 要求と同じ方式で CICS-MQ アダプターに到着します。そのため、IBM MQ にアクセスする Java プログラムと他のプログラムの間に、動作の違いはありません。

IBM MQ classes for Java は、OSGi JVM サーバーでのみ使用できます。IBM MQ classes for Java を使用する既存のアプリケーションは今後も完全にサポートされますが、新しいアプリケーションでは IBM MQ classes for JMS を使用する必要があります。

詳しくは、[OSGi JVM サーバーでの IBM MQ classes for Java の使用](#)を参照してください。

仕組み: CICS-MQ アダプター

CICS と CICS-MQ アダプターは同じアドレス・スペースを共有します。IBM MQ キュー・マネージャーは、独自のアドレス・スペースで実行されます。

4 ページの図 1 には、CICS、CICS-MQ アダプター、および IBM MQ の間の関係が示されています。

アダプターの一部は、IBM MQ メッセージ・マネージャーと通信する CICS タスク関連ユーザー出口です。CICS 管理モジュールは直接この出口を呼び出します。アプリケーション・プログラムは付属の API スタブ・プログラム (CSQCSTUB) を使用してこの出口を呼び出します。タスク関連ユーザー出口とスタブ・プログラムについて詳しくは、[Task-related user exit programs](#) を参照してください。

各 CKTI トランザクションは、通常は MQGET WAIT 状態になっており、開始キューに入れられたトリガー・メッセージに応答可能な状態になっています。

アダプター管理インターフェースには、[CICS-MQ アダプターの管理](#)で説明されている操作や制御機能があります。

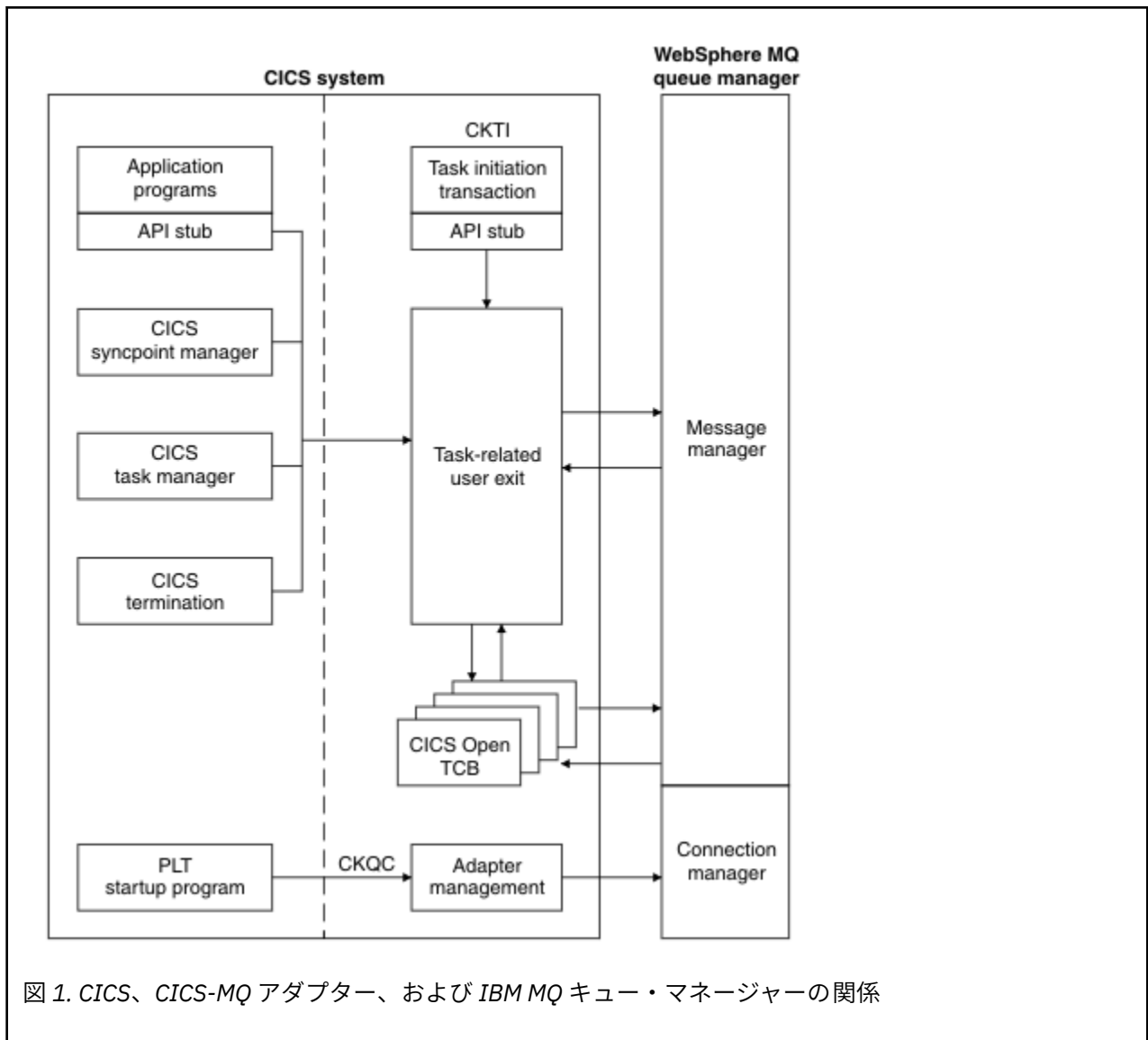


図 1. CICS、CICS-MQ アダプター、および IBM MQ キュー・マネージャーの関係

CICS-MQ アダプター表示機能は、各呼び出しタスクに 2 つの主一時記憶キューを使用して、参照のために出力データを保管します。キューの名前は `ttttCKRT` および `ttttCKDP` であり、`tttt` は表示機能を要求した端末の端末 ID です。これらのキューにアクセスしようとししないでください。

CICS-MQ アダプターは `DFH.genericapplid(8).QMGR` という名前を使用して、接続の開始や停止などの処理中に CICS ENQ 呼び出しと CICS DEQ 呼び出しを発行します。CICS ENQ または DEQ 用に類似する名前を使用しないでください。

アラート・モニター (CKAM)

アラート・モニター・トランザクション CKAM は、2 つの機能を実行します。それは、IBM MQ のインスタンスに対する接続要求または切断要求の結果として生じる保留イベントの処理および MQMONITOR が 1 秒当たり発行できる MQGET 呼び出しの最大数の計算です。

CKAM および保留イベント

保留イベントが発生すると、アラート・モニターはシステム・コンソールに送信されるメッセージを生成します。

保留イベントには、以下の 2 種類があります。

据え置き接続

保留中のイベント (据え置き接続と呼ばれる) は、適切なキュー・マネージャーが開始される前に、CICS が IBM MQ に接続しようとするときアクティブ化されます。適切なキュー・マネージャーとは、接続の設定に応じて、単一の名前付きキュー・マネージャー、または IBM MQ キュー共用グループのメンバーのいずれかです。適切なキュー・マネージャーが使用可能になると、CICS-MQ アダプターにより接続要求が発行され、接続が確立され、保留中のイベントは取り消されます。

複数の据え置き接続が存在する場合があります。適切なキュー・マネージャーが開始されると、その中の 1 つが接続されます。

接続に IBM MQ キュー共用グループを使用している場合に、グループ内のすべてのキュー・マネージャーを使用できない場合、CICS-MQ アダプターは各キュー・マネージャーへの接続を順番に開始するため、結果として複数の接続が据え置かれた状態になります。いずれかのキュー・マネージャーが開始されると、CICS-MQ アダプターによって接続が確立されます。同じ LPAR で複数のキュー・マネージャーが開始すると、CICS-MQ アダプターはそれらの中の 1 つに接続します。

終了通知

IBM MQ に対する接続が正常に行われると、終了通知という保留イベントが作成されます。この保留イベントは、以下のいずれかが発生すると期限切れになります。

- CICS が接続されているキュー・マネージャーが、MODE(QUIESCE) で正常にシャットダウンした。アラート・モニターによって、接続に対する静止要求が発行されます。
- キュー・マネージャーが MODE(FORCE) でシャットダウンしたか、異常終了した。
- CKQC トランザクションから接続がシャットダウンされた。

CICS が処理できる保留イベントの最大数は 99 です。この限度に達すると、1 つ以上の現行イベントの有効期限が切れるまでイベントを作成できなくなります。

すべての保留イベントの有効期限が切れると、アラート・モニターはそれ自身を停止します。新しい接続要求が行われると、自動的に再始動されます。

CKAM および MQMONITOR

z/OS ワークロード・マネージャーの正常性サービスが CICS 領域でアクティブな場合、領域の正常性の状態の変更は、MQMONITOR に影響する可能性があります。この場合、CKAM は以下のようになります。

- 領域の z/OS WLM 正常性値が 100 より小さい場合、CKAM によって、MQMONITOR が 1 秒当たり発行できる MQGET 呼び出しの最大数が計算されます。このスロットルの計算に使用される式、および z/OS ワークロード・マネージャーの正常性サービスが MQMONITOR に与える可能性がある影響について詳しくは、[MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響](#)を参照してください。
- z/OS WLM 正常性のオープン状況が OPENING で、領域の正常性値がゼロより大きい場合 (例えば、CICS ウォームアップ・プロセス中)、CKAM は、AUTOSTART(YES) で定義されたすべての MQMONITOR を開始します。
- z/OS WLM 正常性のオープン状況が CLOSED で、領域の正常性値がゼロの場合 (例えば、CICS クールダウン・プロセスの終了時または **SET WLMHEALTH IMMCLOSE** の発行後)、CKAM は、すべての MQMONITOR を停止します。

CICS によって MXT 条件が検出されると、以下の数式に示すように、CKAM では、この条件が存在する間に MQMONITOR が 1 秒当たり発行できる MQGET 呼び出しの最大数を計算します。この制限により、CICS が MXT にあるときに MQMONITOR によって開始されるタスクの数が制限されます。条件が存在しなくなった場合、CKAM は制限を解除します。

数式: MQGET 呼び出しの最大数は $MXT + 10\%$ です。

MXT ゲーティングが実行されたかどうかを判別するには、[トランザクション統計の解釈](#)を参照してください。

タスク・イニシエーターまたはトリガー・モニター (CKTI)

CKTI は、MQ トリガー・メッセージが読み取られたとき (メッセージが特定のキューに入れられた場合など) に、CICS トランザクションを開始します。

メッセージがアプリケーションのメッセージ・キューに入れられたときに、トリガー条件が満たされるとトリガーが生成されます。次に、キュー・マネージャーは、トリガー・メッセージと呼ばれるユーザー定義データを含むメッセージを、そのメッセージ・キュー用に指定された開始キューに書き込みます。CICS 環境に、開始キューをモニターして、到着するトリガー・メッセージを取得する CKTI のインスタンスをセットアップできます。CKTI は、DEFINE PROCESS コマンドを使用して指定された別の CICS トランザクションを開始します。通常、これはアプリケーション・メッセージ・キューからメッセージを読み取った後、処理します。プロセスは、開始キューではなく、アプリケーション・キュー定義に指定する必要があります。

CKTI の各コピーによって、1 つの開始キューが処理されます。CKTI の複数のインスタンスで 1 つの MQ 開始キューをモニターできます。

CKTI のセットアップ方法

CKTI のインスタンスは、以下の方法でセットアップできます。

MQCONN および MQMONITOR リソースを使用する

CICS 領域の MQCONN リソース定義にデフォルトの開始キューの名前を指定できます。MQCONN リソース定義をインストールすると、CICS はデフォルトの開始キューを表す予約名 DFHMQINI を使用して、MQMONITOR リソースを動的に作成およびインストールします。他の開始キューをモニターするために、さらに MQMONITOR リソースをインストールできます。

MQMONITOR リソースは、MQ キュー・マネージャーへの接続が確立されたときに MQ キューを処理する関連トランザクションを自動的に再始動できるようにします。CEMT または CICS API を使用して、MQMONITOR を手動で開始または停止できます。

MQCONN リソース定義を変更して再インストールし、新しい MQMONITOR リソース定義を作成することで、デフォルトの開始キューの名前を変更できます。また、CICS-MQ 接続を手動で開始する場合、代替のデフォルト開始キューを指定することもできます。

CKQC トランザクションからの端末または DFHMQSSQ にリンクしているユーザー作成プログラムを使用する

CKQC トランザクションを使用して、CICS-MQ アダプターの制御機能にアクセスし、CICS® と IBM MQ の間の接続を開始、管理、および表示します。CKTI のインスタンスを開始または停止するために、**CKQC STARTCKTI** または **CKQC STOPCKTI** コマンドを使用できます。CKTI は、CKQC トランザクションからの端末、またはアダプター・タスク開始プログラム DFHMQSSQ にリンクしている CICS プログラムから開始できます。

CKTI のコピーを開始または停止するには、その CKTI が処理する必要のあるキュー、または現在処理中のキューの名前を指定する必要があります。

開始キューを指定せずに **CKQC STARTCKTI** または **CKQC STOPCKTI** コマンドを発行すると、それらのコマンドは、CICS 領域のデフォルトの開始キュー (MQCONN リソースの INITQNAME 属性で指定されているもの) を参照しているものとして自動的に解釈されます。暗黙的な MQMONITOR が既にインストールされて開始されている場合、要求は失敗します。MQCONN リソース定義をインストールするときに、CICS は、デフォルトの開始キューを表す暗黙的な MQMONITOR リソース定義を作成してインストールします。そのデフォルトの開始キューの名前を変更するには、新しい暗黙的な MQMONITOR リソース定義を作成するために MQCONN リソース定義を変更して再インストールします。また、CICS-MQ 接続を手動で開始する場合、代替のデフォルト開始キューを指定することもできます。

CKQC は、MQMONITOR リソースを制御しません。



注意: CKTI トランザクションのインスタンスを管理するために CKQC トランザクションと MQMONITOR を同時に使用すると、CKQC が MQMONITOR を認識しておらず、MQMONITOR は CKQC を使用して管理される CKTI トランザクションを認識していないため、作成される MQMONITOR の統計と状態が正しくない可能性があります。

注: **CKQC STOPCKTI** が発行されると、開始キューをモニターしているトランザクションは、MQMONITOR に関連付けられているものを含め、すべて停止されます。

CICS-MQ アダプターに提供されているリソース定義

CICS では、DFHLIST の一部としてグループ DFHMQ 内に CICS-MQ アダプター用の CSD 定義が用意されています。サンプル・アプリケーション・プログラムの定義は、IBM MQ によって CSQ4SAMP グループで提供されています。

DFHMQ には、以下の定義が含まれています。

- 提供されるアダプター・プログラム
- 提供されるアダプター管理トランザクション
- 提供される BMS マップ・セット (アダプター・パネルに必要)

プログラム名の形式は DFHMQxxx であり (互換性のためにいくつかの CSQCxxxx 別名も提供)、トランザクション名の形式は CKxx です。

CICS-MQ アダプターが IBM MQ 製品とともに出荷されたとき、IBM MQ は CSQCAT1 および CSQCKB CSD グループを提供していました。CSQCAT1 および CSQCKB グループを CICS TS for z/OS, バージョン 5.6 にインストールしないでください。これらのグループの処理方法について詳しくは、[CICS-MQ アダプターのセットアップ](#)を参照してください。

サンプル・アプリケーション・プログラムの定義が含まれている CSQ4SAMP サンプルを CICS CSD にインストールできます。このサンプルをインストールするための JCL については、[CICS-MQ アダプターのセットアップ](#)を参照してください。リソースを CICS CSD に定義するのではなく、CICS プログラムの自動インストールを使用する場合、自動インストールされた定義が CICS CSD DFHMQ で提供される定義にマップされていることを確認する必要があります。

仕組み: CICS-MQ ブリッジ

CICS-MQ ブリッジを使用すると、CICS 環境で実行されていないアプリケーションで、IBM MQ メッセージを使用して CICS 上のプログラムまたはトランザクションを実行し、応答を受け取ることができます。

非 CICS アプリケーションは、IBM MQ for z/OS を中心に広がる IBM MQ ネットワークにアクセスできるすべての環境から、CICS-MQ ブリッジを使用できます。非 CICS アプリケーションは、次の CICS プログラムを実行する前に、応答が返ってくることを待機する場合があります (同期処理)。また、1 つ以上の CICS プログラムの実行を要求しますが、応答を待機しない場合もあります (非同期処理)。

CICS-MQ ブリッジを使用してプログラムまたはトランザクションを CICS 上で実行するとき、ブリッジは暗黙的な Message Queue Interface (MQI) サポートを使用可能にするため、CICS プログラムには、IBM MQ 呼び出しは含まれません。そのため、既存の CICS アプリケーションを書き換え、再コンパイル、または再リンクすることなく、他のプラットフォームから IBM MQ メッセージによって制御できるように、それらのアプリケーションを再設計することができます。IBM MQ メッセージには、CICS アプリケーションの制御オプションを提供する IBM MQ CICS 情報ヘッダー (MQCIH 構造) が含まれています。

以下のタイプの CICS アプリケーションは、CICS-MQ ブリッジとともに使用するのに適しています。

- **EXEC CICS LINK** コマンドを使用して呼び出される CICS プログラム (DPL プログラムと呼ばれます)。このプログラムは、CICS API の DPL サブセットに規格が合致している必要があります。つまり、CICS 端末機能または同期点機能を使用することはできません。CICS-MQ ブリッジを使用して、単一の CICS プログラム、または作業単位を形成する一連の CICS プログラムを実行できます。
- 3270 端末から実行するように設計された CICS トランザクション (3270 トランザクションと呼ばれます)。トランザクションは、基本マッピング・サポート (BMS) または端末管理コマンドを使用できます。それらは、会話型か、または疑似会話の一部であることが可能です。また、同期点の発行が許可されています。

非 CICS アプリケーションは内部ロジックおよび CICS トランザクションでの制御フローと対話する必要があるため、3270 トランザクションを CICS-MQ ブリッジを介して実行するには、通常、より複雑なアプリケーション・プログラミングが必要になります。CICS アプリケーションのビジネス・ロジックを含む DPL プログラムを実行することをお勧めします。ただし、一部の CICS アプリケーションでは、アプリケーション

ンのビジネス・ロジックが表示ロジックから分離された状態で構造化されていません。そのため、CICS-MQブリッジは、いずれのタイプのアプリケーションとも通信することを許可します。

CICS-MQブリッジでは、CICSとIBM MQの標準のセキュリティー機能を使用されます。要求側のユーザーIDを認証する、信頼する、または無視するように構成できます。IBM MQメッセージのMQCIH構造は、セキュリティー検査用のデータを提供します。

CICS-MQブリッジは、ブリッジ・モニター(デフォルトの名前CKBRを使用するトランザクション)を使用して、CICSアプリケーションを実行するように新たに要求するIBM MQ要求キューをブラウズします。この要求キューは、CICS-MQアダプターに接続されたローカルのz/OSキュー・マネージャーに存在する必要があります。CICS-MQブリッジ・タスクは、通常、ブリッジ・モニターと同じCICS領域で実行します。必要に応じてCICSトランザクション・ルーティングを使用することにより、複数のユーザー・プログラムを同じCICS領域にも、異なる領域にも配置できます。特定のCICS領域で要求を処理する必要がある場合、CICS領域をMQCIHヘッダーに指定できます。

CICS-MQブリッジでのCICS DPLプログラムの実行方法

IBM MQメッセージは、CICS DPLプログラムを実行するために必要なデータを提供します。CICS-MQブリッジは、このデータからCOMMAREAまたはコンテナを構築し、**EXEC CICS LINK**を使用してプログラムを実行します。

8 ページの図 2 は、1つのメッセージを処理してCICS DPLプログラムを実行するために行われるアクションの順序を示しています。

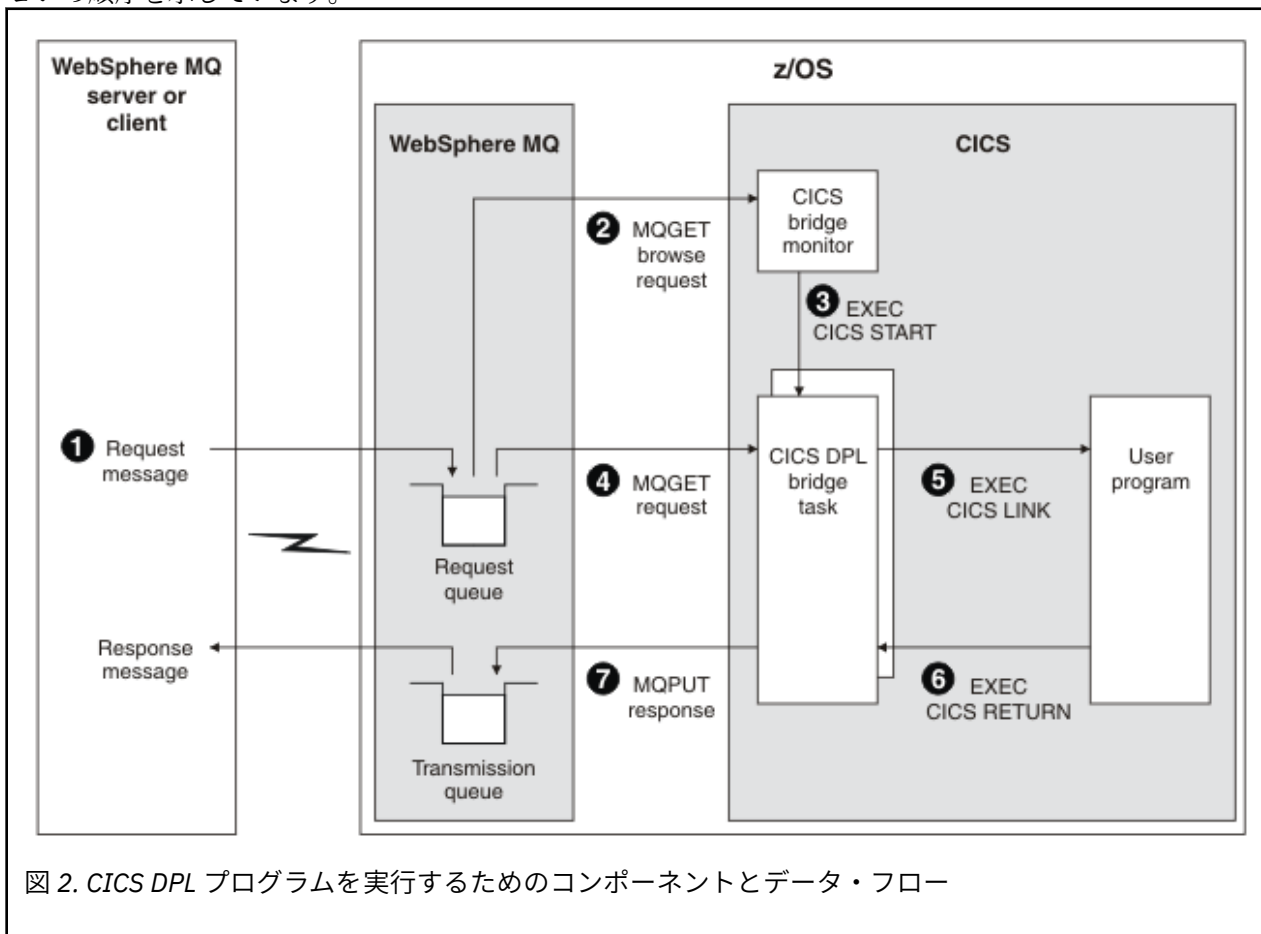


図 2. CICS DPL プログラムを実行するためのコンポーネントとデータ・フロー

1. CICS プログラムの実行要求を含むメッセージが、要求キューに到着します。
2. そのメッセージを、キューを常時ブラウズしているブリッジ・モニター・タスクが調べます。
3. ブリッジ・モニター・タスクが、指定されたトランザクションまたはCKBCのどちらかを使用して、CICS DPLブリッジ・タスクを開始します(コンテナを使用することを選択した場合)。あるいは、デフォルトのCKBPトランザクションを使用できます(COMMAREAを使用することを選択した場合)。CKBCは、COMMAREAではなくチャンネルとコンテナを使用します。使用されるプログラムは

DFHMQBP3 です。CKBP は COMMAREA を使用し、使用されるプログラムは DFHMQBP0 です。独自の指定トランザクションを使用する場合は、使用することに選択した関連プログラムを選択できます。

4. プログラムが DFHMQBP0 であっても DFHMQBP3 であっても、CICS DPL ブリッジ・タスクが、要求キューからメッセージを受信します。
5. DFHMQBP3 プログラムは、メッセージ・ペイロードをチャネル DFHMQBR_CHANNEL のコンテナ DFHREQUEST に入れ、指定されたプログラムに EXEC CICS LINK を発行します。DFHMQBP0 プログラムは、メッセージ・ペイロードを COMMAREA に入れ、指定されたプログラムへの **EXEC CICS LINK** を発行します。
6. DFHMQBP3 プログラムは、アプリケーションがその応答メッセージ・ペイロードをチャネル DFHMQBR_CHANNEL のコンテナ DFHRESPONSE に入れることを予期します。DFHMQBP0 プログラムは、アプリケーションがその応答メッセージ・ペイロードを、要求で使用された COMMAREA に入れることを予期します。

アプリケーションは応答メッセージのペイロードを提供しない場合でも、コンテナ (この場合は空のコンテナ) を返します。その場合も、ヘッダーのみの応答が送信され、ペイロードは送信されないことに注意してください。

応答を送信しないようにクライアントで表明するには、要求メッセージ MQMD の応答先キュー名 (**ReplytoQueue**) を省略します。

7. DFHMQBP3 プログラムは、DFHRESPONSE コンテナの内容を応答メッセージ・ペイロードに入れるとともに、応答メッセージを応答先キューに置きます。DFHMQBP0 プログラムは、COMMAREA の内容を応答メッセージ・ペイロードに入れるとともに、応答メッセージを応答先キューに置きます。

ユーザー・トランザクションは、終了するか、さらに入力を要求します。このフローが疑似会話の中の最後のものではあれば、トランザクションは終了します。メッセージが最後のものでなければ、次のメッセージが受信されるか指定されたタイムアウト間隔が時間切れになるまで、トランザクションは待機します。

複数の CICS プログラム

作業単位は、ただ 1 つのユーザー・プログラムの場合も、複数のユーザー・プログラムの場合もあります。作業単位を構築するために送信できるメッセージの数に、制限はありません。

このシナリオでは、多数のメッセージから成る作業単位も同じように機能します。ただし、最終ステップでは、作業単位内の最後のメッセージでない限り、ブリッジ・タスクは次の要求メッセージを待機します。

CICS-MQ ブリッジでの CICS 3270 トランザクションの実行方法

IBM MQ メッセージは、CICS 3270 トランザクションを実行するために必要なデータを提供します。CICS トランザクションは実際の 3270 端末があるかのように実行されますが、CICS トランザクションと IBM MQ アプリケーションの間の通信には、代わりに 1 つ以上の IBM MQ メッセージが使用されます。

CICS-MQ ブリッジは、CICS Link3270 メカニズムを使用して CICS トランザクションにアクセスします。従来型 3270 エミュレーターとは異なり、CICS-MQ ブリッジは、z/OS Communications Server フローを IBM MQ メッセージに置き換えることによって動作するわけではありません。代わりに、メッセージはベクトルという複数のパーツで構成されます。個々のベクトルは 1 つの **EXEC CICS** 要求に対応しています。したがって、アプリケーションはエミュレーターを介するのではなく、トランザクションで使用されるデータ (アプリケーション・データ構造 (ADS) と呼ばれる) を使用して、CICS トランザクションと直接会話します。

CICS-MQ ブリッジを使用して CICS 3270 トランザクションを実行するためのコンポーネントとデータ・フローを以下に示します。

1. CICS トランザクションの実行要求を含むメッセージが、要求キューに置かれます。
2. キューを常時ブラウズしている CICS-MQ ブリッジ・モニター・タスクが、「作業単位の開始」メッセージが待機していることを認識します (**CorrelId=MQCI_NEW_SESSION**)。
3. 関連する認証検査が行われ、適切な権限で特定のユーザー ID (ブリッジ・モニターの開始に使用されるオプションにより異なる) を使用して CICS 3270 ブリッジ・タスクが開始されます。
4. CICS-MQ ブリッジは、キューからメッセージを除去し、ユーザー・トランザクションを実行するようタスクを変更します。

5. メッセージの中のベクトルは、トランザクション内のすべての端末関連入力 **EXEC CICS** 要求に応えるためのデータを提供します。
6. 端末関連出力 **EXEC CICS** 要求の結果として、出力ベクトルが構築されます。
7. CICS-MQ ブリッジは、すべての出力ベクトルを単一メッセージに組み込み、そのメッセージを応答先キューに置きます。
8. CICS 3270 ブリッジ・タスクが終了します。このフローがトランザクションの中の最後のものであれば、トランザクションは終了します。このメッセージが最後のものでなければ、次のメッセージが受信されるか指定されたタイムアウト間隔が時間切れになるまで、トランザクションは待機します。

複数の CICS トランザクション

多くの場合、CICS アプリケーションは、疑似会話として一体となるようにリンクされた 1 つ以上のトランザクションから成ります。各トランザクションは一般に、3270 端末ユーザーが画面にデータを入力して AID キーを押すことによって開始されます。このモデルのアプリケーションは、CICS-MQ ブリッジを使用する非 CICS アプリケーションによってエミュレートできます。

このモデルでは、メッセージは最初のトランザクション用に構築され、そのトランザクションに関する情報と入力ベクトルを格納します。このメッセージはキューに置かれます。応答メッセージは、出力ベクトル、次に実行されるトランザクションの名前、および疑似会話を表すために使用されるトークンから成ります。IBM MQ アプリケーションは、新しい入力メッセージを構築します。このメッセージでは、トランザクション名が次のトランザクションに設定され、機能トークンおよびリモート・システム ID が直前のメッセージで返された値に設定されます。2 番目のトランザクションのベクトルがメッセージに追加され、メッセージがキューに置かれます。このプロセスは、アプリケーションが終了するまで続けられます。

複数のトランザクションのすべての IBM MQ メッセージを、同じブリッジ・セッションに組み込むことができます。こうすると、モニター・オーバーヘッドが小さくなってパフォーマンスが向上します。

CICS アプリケーションを作成する別の方法として、会話型モデルがあります。このモデルでは、トランザクションを実行するためのすべてのデータが元のメッセージに入っているとは限りません。メッセージ内のどのベクトルでも応えることができない要求をトランザクションが発行する場合は、さらにデータを要求するメッセージが応答先キューに入れられます。IBM MQ アプリケーションはこのメッセージを取得し、要求を満たすベクトルを含んだ新しいメッセージをキューに戻します。

仕組み: IBM MQ classes for JMS with CICS

IBM MQ classes for JMS は、CICS で実行される Java アプリケーションから IBM MQ への優先インターフェースです。(IBM MQ classes for Java は引き続きサポートされますが、新しいアプリケーションでは IBM classes for JMS を使用する必要があります。)IBM MQ classes for JMS は、OSGi JVM サーバー (制限付き) または Liberty JVM サーバーで実行できます。

Java EE プラットフォームでは、IBM MQ classes for JMS は、アプリケーションのコンポーネントと IBM MQ キュー・マネージャーの間の通信として以下の 2 つのタイプをサポートします。

アウトバウンド通信

JMS API を直接使用して、アプリケーション・コンポーネントはキュー・マネージャーへの接続を作成し、それからメッセージを送受信します。

例えば、アプリケーション・コンポーネントはアプリケーション・クライアント、サーブレット、JavaServer Page (JSP)、エンタープライズ Java Bean (EJB)、またはメッセージ・ドリブン Bean (MDB) のいずれであっても構いません。このタイプの通信では、アプリケーション・サーバー・コンテナは、接続プールおよびスレッド管理などのメッセージング操作をサポートする低レベルの機能のみ提供します。

インバウンド通信

宛先に届くメッセージは MDB に送信され、そこでメッセージが処理されます。

Java EE アプリケーションは MDB を使用してメッセージを非同期に処理します。MDB は JMS メッセージ・リスナーとしての役割を果たし、onMessage() メソッドによって実装されます。このメソッドはメッセージの処理方法を定義します。MDB はアプリケーション・サーバーの EJB コンテナにデプロイされます。MDB の厳密な構成方法は、使用するアプリケーション・サーバーに依存しますが、構成情報には、接続先のキュー・マネージャー、キュー・マネージャーへの接続方法、メッセージをモニ

ターする宛先、および MDB のトランザクション動作を指定する必要があります。この情報は EJB コンテナによって使用されます。MDB の選択基準を満たすメッセージが指定された宛先に届くと、EJB コンテナは IBM MQ classes for JMS を使用してキュー・マネージャーからメッセージを取り出し、それから `onMessage()` メソッドを呼び出してメッセージを MDB に配信します。

OSGi JVM サーバーでの IBM MQ classes for JMS

次の制限が適用されます。

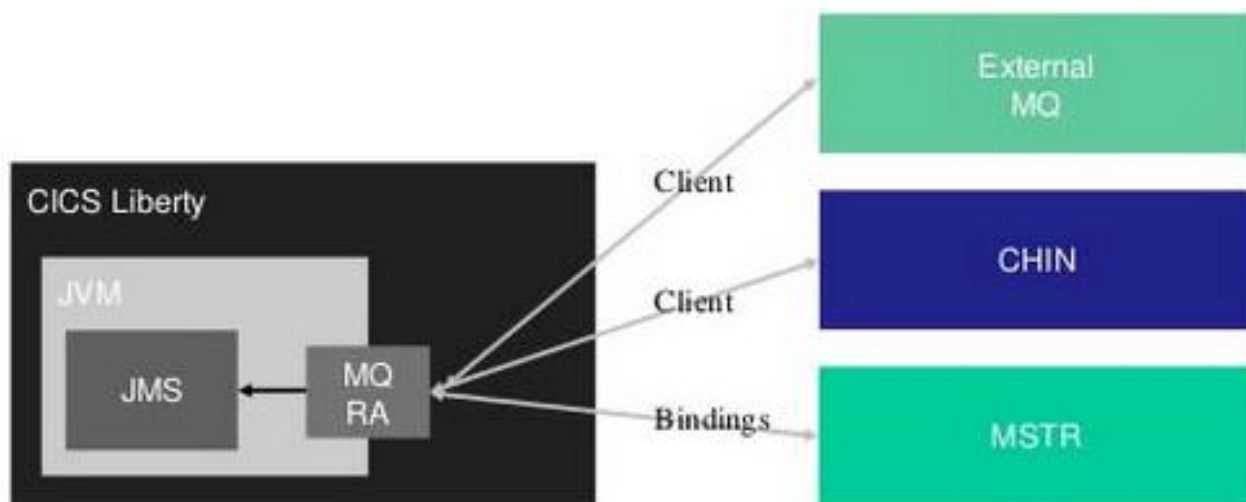
- CICS OSGi JVM サーバーでの IBM MQ classes for JMS の使用は、CICS Version 5.2 以降でのみサポートされます。
- クライアント・モード接続はサポートされていません。
- 接続がサポートされるのは、IBM MQ Version 7.1 または IBM MQ Version 8.0 以降のキュー・マネージャーのみです。接続ファクトリーの `PROVIDERVERSION` 属性は、未指定にするか、7 以上の値にする必要があります。
- XA 接続ファクトリー (例えば `com.ibm.mq.jms.MQXAConnectionFactory`) の使用はサポートされていません。

CICS 標準モード Liberty JVM サーバーでの IBM MQ classes for JMS

JMS アプリケーションがバインディング・モードまたはクライアント・モードのいずれかの接続を使用してキュー・マネージャーに接続する場合は、CICS 標準モードの Liberty JVM サーバー

- IBM MQ リソース・アダプターは、サービス中の任意のバージョンの IBM MQ にクライアント・モードで接続できます。
- IBM MQ リソース・アダプターは、同じ CICS 領域から同じキュー・マネージャーへの CICS 接続 (アクティブな `CICS MQCONN` リソース定義) がない場合、サービス中の任意のバージョンの IBM MQ for z/OS にバインディング・モードで接続できます。

Liberty から IBM MQ に接続するには、Version 9.0.1 以降の IBM MQ リソース・アダプターが必要です。Liberty には IBM MQ リソース・アダプターが含まれていないため、Fix Central から取得する必要があります ([Liberty へのリソース・アダプターのインストール](#) を参照)。

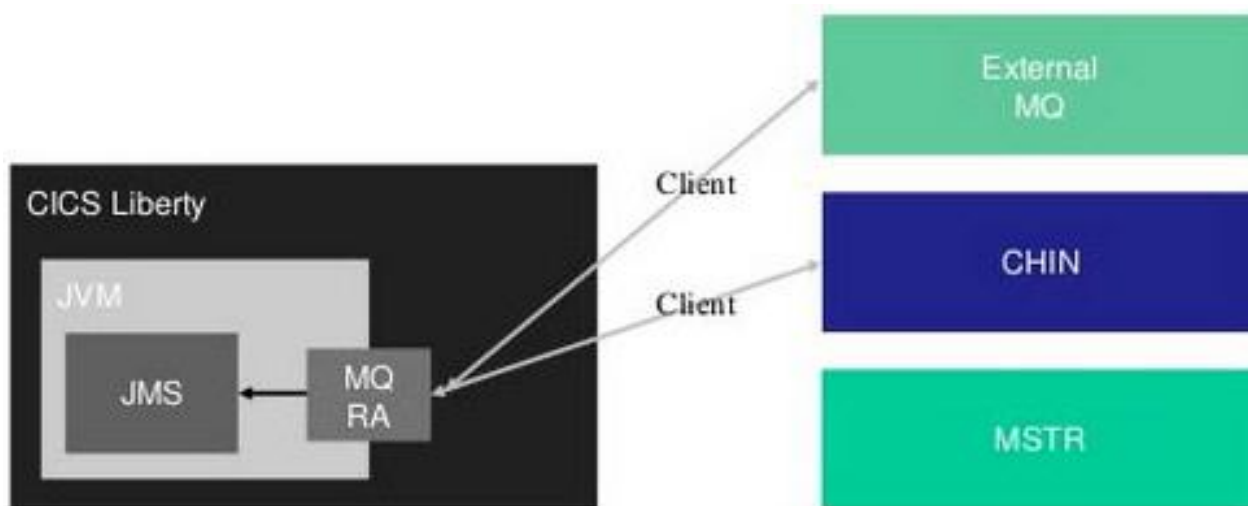


CICS 統合モード Liberty JVM サーバーでの IBM MQ classes for JMS

JMS アプリケーションがクライアント・モードの接続のみを使用してキュー・マネージャーに接続する場合は、CICS 統合モードの Liberty JVM サーバー

- IBM MQ リソース・アダプターは、サービス中の任意のバージョンの IBM MQ にクライアント・モードで接続できます。
- バインディング・モード接続はサポートされていません。

Liberty から IBM MQ に接続するには、Version 9.0.1 以降の IBM MQ リソース・アダプターが必要です。Liberty には IBM MQ リソース・アダプターが含まれていないため、Fix Central から取得する必要があります ([Liberty へのリソース・アダプターのインストール](#)を参照)。



次に行うこと

IBM MQ classes for JMS with CICS を使用したアプリケーションの開発については、CICS 資料の [OSGi JVM サーバーでの IBM MQ classes for JMS の使用](#)または Liberty JVM サーバーでの [IBM MQ classes for Java の使用](#)を参照してください。アプリケーション用の JVM サーバーのセットアップについては、[JMS をサポートするための Liberty JVM サーバーの構成](#)を参照してください。

IBM MQ classes については、IBM MQ の資料の [IBM MQ classes for JMS の使用](#)を参照してください。

CICS-MQ トランザクション・トラッキング・サポート

CICS トランザクション・トラッキング・サポートには、IBM MQ トリガーとブリッジ・モニターの両方によって開始されるタスクが含まれています。

CICS アダプター・トラッキング・サポートには、IBM MQ トリガーまたはブリッジ・モニターによって開始されるタスクが含まれています。

IBM MQ トリガー、またはブリッジ・モニターのタスクは、CICS 領域内で開始されるタスクの元のデータにアダプター・フィールドを設定することで、CICS トランザクション・トラッキングによってサポートされます。次のアダプター・フィールドは、階層順に使用されることが意図されています。アダプター ID により一般情報が指定され、アダプター・データ 3 により最も具体的な情報が指定されます。この方式により、タスクの識別と分離を一貫した方法で行えます。

1. 発信アダプター ID

使用されている IBM MQ キューのバージョンに対する製品 ID。例えば、ID=IBM WebSphere MQ for z/OS V7.0.1 などです。

2. 発信アダプター・データ 1

IBM MQ キュー・マネージャーの名前。例えば、QMGR=RQ38 などです。

3. 発信アダプター・データ 2

トリガー・モニターがトリガー・メッセージを取得し、このタスクを開始した開始キューの名前。例えば、INITQ=ALIINIT1 などです。

開始キューの名前は CICS-MQ ブリッジのシナリオには適用されないため、アダプター・データ 2 はデフォルト値に設定されます。例えば、INITQ=Not Applicable などです。

4. 発信アダプター・データ 3

タスクが開始された結果としてメッセージが配置されるアプリケーション・メッセージ・キューの名前。例えば、QNAME=ALITRIG1 などです。

リモート・トランザクションの場合、最初の接続時にミラー・トランザクションに適用されるアダプター・データは、そのミラーを使用する後続の START コマンドすべてで、個々の START コマンドの設定内容にかかわらず使用されます。

MQ サポートのための CICS リソース: MQCONN および MQMONITOR

MQCONN は、CICS と IBM MQ の間の接続の属性を定義する CICS リソースです。MQMONITOR は、トリガー・モニター・トランザクション CKTI などの MQ メッセージ・コンシューマーの属性を定義する CICS リソースです。

MQCONN について

CICS と IBM MQ の間の接続を開始するには、MQCONN リソースをインストールする必要があります。

1 つの CICS 領域に一度にインストールできる MQCONN リソース定義は、1 つのみです。MQCONN リソースをインストールできるのは、CICS が IBM MQ に接続されていない場合のみです。そのため、MQCONN 定義を変更して、再インストールする場合、最初に接続を停止する必要があります。

MQCONN リソースは、CICS と IBM MQ の間の接続に関して、以下の設定を指定します。

- CICS の接続先となる、単一の MQ キュー・マネージャーまたは MQ キュー・マネージャーのキュー共有グループの名前 (MQNAME 属性)。
- 接続が未処理の作業単位を伴って終了した場合に、CICS が採用する再同期戦略 (RESYNCMEMBER 属性)。
- 接続のデフォルト開始キューの名前 (INITQNAME 属性)。

CICS Explorer[®]、CEDA、CEMT、または SPI コマンドを使用して MQCONN リソースを管理できます。また、CICSplex[®] SM を使用して管理することもできます。

MQMONITOR について

MQMONITOR リソースは、CICS MQ モニターの以下の設定を指定します。

- MQMONITOR の自動開始戦略。これにより、MQ キュー・マネージャーへの接続が確立されたときに MQMONITOR を自動的に開始できます (AUTOSTART 属性)。
- リソースの状況 (ENABLESTATUS 属性)。
- キューをモニターしているトランザクション (CICS-MQ ブリッジなど) に渡されるデータ (ある場合) (MONDATA 属性)。
- モニター対象の MQ キューの名前 (QNAME 属性)。
- MQ キューをモニターしているタスクが使用するユーザー ID (MONUSERID 属性)。
- MQ モニターの状態 (MONSTATUS 属性)。
- MQ キューをモニターしているタスクのトランザクション ID (TRANSACTION 属性)。
- 他のソースから入手可能な適切なユーザー ID がいない場合に、アプリケーション・トランザクションを開始するためにデフォルトで使用するユーザー ID (USERID 属性)。

MQ キュー (開始キューを含む) の着信メッセージを処理するために、任意の数の MQMONITOR リソースをインストールできます。複数の MQMONITOR リソースを使用して、1 つの MQ キューの着信メッセージを処理できます。

MQ 開始キューの着信メッセージを処理するために MQMONITOR リソースを使用する場合は、TRANSACTION 属性に CKTI を設定し (またはブランクのままにし)、QNAME 属性に MQ 開始キューの名前を設定して、MQMONITOR リソースを定義する必要があります。

デフォルトの CKTI ではなくユーザー作成の MQ モニター・プログラムを使用して MQMONITOR リソースを処理する場合、ユーザー作成のプログラムでアプリケーション・キューから直接メッセージを取得して必須のロジックを実行することが必要になります。ユーザー自身がプログラムをコーディングしている場合は、ユーザー作成の MQ トリガー・モニターと MQ メッセージ・コンシューマーの開発と使用に説明されているガイドラインに必ず従ってください。

CICS Explorer、CEDA、CEMT、または SPI コマンドを使用して MQMONITOR リソースを管理できます。また、CICSplex SM を使用して管理することもできます。

セキュリティ検査がアクティブな場合、CICS は MQ モニター状態を開始済みに設定しようとしているトランザクションに関連付けられたユーザー ID に対してセキュリティ検査を行います。このユーザー ID が、MONUSERID に定義されているユーザー ID の代理であり、MONUSERID に関連付けられたトランザクションを開始する権限を持っていることを確認する必要があります。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPIUSR ユーザー ID (指定されている場合) を使用してください。

MQMONITOR への影響

CICS 領域のいくつかの側面は、MQMONITOR が発行できる MQGET 呼び出しの数に影響を与える可能性があります。

アクティブな z/OS ワークロード・マネージャーの正常性サービス

z/OS ワークロード・マネージャーの正常性サービスが CICS 領域でアクティブになっている場合、MQMONITOR は、領域の正常性の状態に反応し、z/OS ワークロード・マネージャーの正常性サービスのオープン状況の設定の影響を受ける可能性があります。詳細については、[MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響](#)を参照してください。

MXT 条件下の CICS

CICS によって MXT 条件が検出されると、この条件が存在する場合に MQMONITOR が 1 秒当たり発行できる MQGET 呼び出しの数の制限が適用されます。詳しくは、[4 ページの『アラート・モニター \(CKAM\)』](#)を参照してください。

動的にインストールされる MQMONITOR リソース DFHMQINI について

INITQNAME 属性の値を指定する MQCONN リソースをインストールすると、CICS もデフォルトの開始キューを表す予約済みの名前 DFHMQINI を使用して、動的に MQMONITOR リソースを作成してインストールします。

DFHMQINI には以下の属性があります。

QNAME

開始キュー名を指定します。

TRANSACTION

CKTI (デフォルト)

MONUSERID

この属性は、以下のように取得されます。

- **PLTPIUSR** システム 初期設定パラメーターから (使用可能な場合)
- それ以外の場合は、CICS 領域ユーザー ID から

USERID

この値は、CICS のデフォルト・ユーザー ID です。

DFHMQINI は、MQ 接続が確立されると自動的に開始されます。セキュリティ検査がアクティブな場合は、CKTI トランザクションに関連付けられたユーザー ID が **MONUSERID** から取得されます。

EXEC CICS INQUIRE MQMONITOR コマンドまたは **CEMT INQUIRE MQMONITOR** コマンドを使用して、動的に作成された MQMONITOR リソースの **QNAME** 属性について照会することができます。この **QNAME** 値を変更したい場合は、最初に MQCONN リソースの **INITQNAME** 属性を変更してから、MQCONN リソースを再インストールする必要があります。**INITQNAME** 属性の設定が含まれている MQCONN リソースを破棄すると、動的に作成された MQMONITOR リソースおよびすべてのユーザー定義 MQMONITOR リソースも破棄されます。



注意: CKTI トランザクションのインスタンスを管理するために CKQC トランザクションと MQMONITOR を同時に使用すると、CKQC が MQMONITOR を認識しておらず、MQMONITOR は CKQC を使用して管理される CKTI トランザクションを認識していないため、作成される MQMONITOR の統計と状態が正しくない可能性があります。

z/OS ワークロード・マネージャーの正常性サービスの MQMONITOR への影響

z/OS ワークロード・マネージャーの正常性サービスが CICS 領域でアクティブな場合、CICS はシステム初期化の完了後にウォームアップ・プロセスを実行し、シャットダウンの前にクールダウン・プロセスを実行できます。システムのウォームアップまたはクールダウンのどちらの間でも、CICS は領域の正常性値を調整し、領域への処理のフローを制御します。領域の正常性の状態の変更は、MQMONITOR に影響する可能性があります。例えば、属性 AUTOSTART(YES) が設定された MQMONITOR が開始または停止されたときなどです。正常性値が 100 未満の場合、開始されるすべての MQMONITOR は、1 つの MQMONITOR が処理できるメッセージを制限するスロットルの対象となります。

MQMONITOR 開始時の影響

MQMONITOR が自動的に開始されるように設定されている場合、WLMHEALTH のオープン状況が OPEN または OPENING であり、領域の正常性値がゼロよりも大きい場合に MQMONITOR は開始されます。

例えば、システムの初期化が完了すると、CICS は事前設定された間隔で z/OS WLM Health API (IWM4HLTH) の呼び出しを開始し、領域の正常性値をゼロから 100% に到達するまで増やします。そのため、最初の間隔の後に正常性値はゼロよりも大きくなり、属性 AUTOSTART(YES) が設定された MQMONITOR が開始されます。間隔と調整値は **WLMHEALTH** システム初期設定パラメーターに指定されています。

注：z/OS ワークロード・マネージャーの正常性サービスが CICS 領域で無効になっている場合（つまり、**WLMHEALTH** が OFF に設定されている場合）、上記のような MQMONITOR は、システム初期化の終了後に制御が CICS に渡されるとすぐに開始されます。

MQMONITOR 停止時の影響

属性 AUTOSTART(YES) が指定された MQMONITOR は、WLMHEALTH のオープン状況が CLOSED になり、領域の正常性値がゼロになると停止されます。例えば、**SET WLMHEALTH OPENSTATUS(CLOSE)** が発行されると、領域の正常性値が一定の間隔で減らされ、ゼロになると、領域内の該当する MQMONITOR は停止されます。

AUTOSTART(YES) が指定された MQMONITOR は、**SET WLMHEALTH OPENSTATUS(IMMCLOSE)** が発行されるとすぐに停止されます。

1 秒当たりの MQGET 呼び出し数に対するスロットル

領域の z/OS WLM 正常性値が 100 より小さい場合、MQMONITOR が 1 秒当たりに発行できる MQGET 呼び出しの数に対するスロットルが設定され、開始されるトリガー・タスクの数が制御されます。このスロットルは、領域内で開始されるすべての MQMONITOR に影響します。領域の正常性値が 100 に達すると、MQGET 呼び出しに対するスロットルは除去されます。スロットルは、**CKAM** によって以下のように計算されます。

1 秒当たりの MQGET 呼び出しの最大数は、**MXT** の値を 2 倍にした値に、領域の z/OS WLM 正常性値を掛けた数値です。

領域に、WLMHEALTH=(20,25) と MXT=400 が指定されていると仮定します。以下の例は、1 秒当たりの MQGET 呼び出し最大数が、システム全体のウォームアップ・プロセス中のそれぞれの間隔の後、どのように変化するかを示しています。

表 1. 例		
時間	正常性値 (%)	1 秒当たりの最大 MQGET 呼び出し
0 (システム 初期化終了時)	0	0
20	25	200
40	50	400
60	75	600

表 1. 例 (続き)		
時間	正常性値 (%)	1 秒当たりの最大 MQGET 呼び出し
80 (システム・ウォームアップ終了時)	100	スロットル除去

同様に、システムのクールダウン・プロセス中に、領域の正常性値が一定の間隔で減らされ、MQGET 呼び出しに対するスロットルが有効になります。MQMONITOR が 1 秒当たりに発行できる MQGET 呼び出しの最大数は、間隔ごとに徐々に削減されます。

第 2 章 IBM MQ への接続の構成

MQ メッセージを使用してワークロードを CICS に取り込めるように、CICS には、CICS-MQ アダプターおよび CICS-MQ ブリッジという 2 つのインターフェースが用意されています。このセクションでは、アダプターおよびブリッジを CICS 領域で使用可能にするためのセットアップ手順について説明します。

CICS-MQ アダプターのセットアップ

CICS-MQ アダプターを CICS 領域でできるようにするには、以下の作業を完了してください。

始める前に

CICS でのリソース定義に精通していない読者は、以下の情報を参照してください。

- 「[Setting up shared data sets, CSD and SYSIN](#)」には、CICS 領域のセットアップ、および CICS システム初期設定パラメーターの指定に関する情報が記載されています。
- 「[CICS リソース](#)」には、CICS でのリソースの定義、および PLT 項目のコーディングに関する情報が記載されています。

手順

1. IBM MQ ライブラリー `thlqual.SCSQAUTH` を CICS プロシージャの STEPLIB 連結に組み込みます。正しいコードが確実に使われるようにするために、CICS ライブラリーの後にこのライブラリーを組み込んでください。

`thlqual` は、IBM MQ ライブラリーの高位修飾子です。

2. 以下の IBM MQ ライブラリーを CICS プロシージャの DFHRPL 連結に組み込みます。正しいコードが確実に使われるようにするために、CICS ライブラリーの後にこのライブラリーを組み込んでください。

```
thlqual.SCSQCICS  
thlqual.SCSQLOAD  
thlqual.SCSQAUTH
```

`thlqual` は、IBM MQ ライブラリーの高位修飾子です。

CICS-MQ API 交差出口 (CSQCAPX) を使用している場合は、プログラムのロード・モジュールを含んでいるライブラリーの名前も追加してください。

SCSQCICS ライブラリーは、IBM MQ に付属のサンプルを実行する場合に限って必要です。そうでない場合は、CICS プロシージャからこれを除去することができます。

3. CICS 領域用の MQCONN リソースを定義してインストールします。
MQCONN リソースは、CICS と IBM MQ との接続の属性を指定します。これには、接続のデフォルト IBM MQ キュー・マネージャーまたはキュー共用グループの名前が含まれます。説明は、[MQCONN リソースの定義とインストール](#)を参照してください。
4. オプション: CICS 領域用の MQMONITOR リソースを定義してインストールします。
MQMONITOR リソースは、CICS-MQ トリガー・モニター CKTI などの MQ メッセージ・コンシューマーの属性を定義します。CKTI のインスタンスを制御する方法としてお勧めするのが、MQMONITOR リソースを使用することです。他の MQ キューをモニターする MQMONITOR リソースをインストールすることもできます。詳しくは、[20 ページの『MQMONITOR リソースの定義とインストール』](#)を参照してください。
5. CICS 初期設定中に CICS-MQ アダプターを使って IBM MQ に自動接続するためには、以下のいずれかの方式から選んで指定してください。
 - CICS システム 初期設定パラメーター **MQCONN=YES** を指定します。IBM MQ に自動接続するには、この方法が推奨されます。

- 『Configuring』の『Specifying DFHMQCOD or your own program in the PLTPI list』の指示に従って、DFHMQCOD プログラムを CICS PLTPI リストに含めます。PLTPI リストは、プログラム・リスト・テーブル (PLT) に、**PLTPI** システム初期設定パラメーターで指定された接尾部を加えたものです。
- WebSphere MQ への接続を開始するための PLTPI プログラムの作成の指示に従って独自のプログラムを作成した後、『Configuring』の『Specifying DFHMQCOD or your own program in the PLTPI list』の指示に従って CICS PLTPI リストにそれを含めます。

CICS が IBM MQ への接続を開始するためには、その前に MQCONN リソース定義をインストールしておく必要があります。CICS の初期設定時に接続を自動開始する際、初期始動またはコールド・スタートの場合には、**GRPLIST** システム初期設定パラメーターで指定されたリスト内のいずれかのグループに MQCONN リソース定義が含まれる必要があります。CICS のウォーム始動または緊急開始の場合は、以前の CICS 実行の終了までに MQCONN リソース定義がインストール済みでなければなりません。

6. リモート CICS システムとの領域間通信 (IRC) を行う CICS システムで CICS-MQ アダプターを使用する場合には、CICS システム初期設定パラメーター IRCSTRT=YES を指定することにより、アダプター開始前に IRC 機能を必ず OPEN してください。

IRC アクセス方式が仮想記憶間として定義されている場合 (つまり ACCESSMETHOD(XM) である場合)、IRC 機能が OPEN している必要があります。

7. ご使用のキュー・マネージャーおよび CICS で使用されるコード化文字セット ID (CCSID)、および UTF-8 と UTF-16 のコード・ページが、z/OS 変換サービスに対して構成されていることを確認します。

CICS コード・ページは、**LOCALCCSID** システム初期化パラメーターで指定されます。

8. 次のようにして CICS CSD を更新します。

- a) 以前のリリースの CICS との間で CSD を共用しない場合は、グループ CSQCAT1 と CSQCKB、およびこれらのグループ (またはグループ項目) のコピーを CSD からすべて削除します。また、CKQQ TDQUEUE をグループ CSQCAT1 から削除する必要もあります。

これで、CKQQ の定義が CICS CSD グループ DFHDCTG で提供されるようになります。

- b) 以前の CICS リリースとの間で CSD を共用する場合は、CSQCAT1 と CSQCKB、およびこれらのグループ (またはその内容) のコピーが CICS TS V4.1 用にインストールされていないことを確認してください。また、CKQQ TDQUEUE をグループ CSQCAT1 から削除する必要もあります。

これで、CKQQ の定義が CICS CSD グループ DFHDCTG で提供されるようになります。

9. サンプル・アプリケーション・プログラム用の定義を含んでいる CSQ4SAMP サンプルを CSD の中にインストールする場合、次のような JCL 断片を CSD アップグレード (DFHCSDUP) ジョブに追加します。

```
//SYSIN DD DSN=thlqual.SCSQPROC(CSQ4S100),DISP=SHR
//      DD *
//      ADD GROUP(CSQ4SAMP) LIST(yourlist)
/*
```

ここで *yourlist* は、システムのコールド・スタート中に CICS によってインストールされるグループのリストです。これは CICS のシステム初期設定パラメーター **GRPLIST** で指定されます。

注: CEDA トランザクションを使用して、再定義されたアダプター・リソースをアクティブ CICS システムにインストールする場合には、まずアダプターをシャットダウンした後、アラート・モニターの作業が完了するのを待つ必要があります。

10. デッド・レター・キュー、デフォルト伝送キュー、および CICS-MQ アダプター・オブジェクトに関する IBM MQ 定義を更新します。

サンプル CSQ4INYG を使用できますが、CICS 領域の MQINI リソース定義内のデフォルト開始キュー名に一致するよう、開始キュー名を変更する必要があるかもしれません。このメンバーをキュー・マネージャー始動プロシーチャーの CSQINP2 DD 連結で使用できます。または、必要な DEFINE コマンドを発行するために、CSQUTIL ユーティリティの COMMAND 関数の入力としてこのメンバーを使用することもできます。CSQUTIL ユーティリティを使用すると、それ以降 IBM MQ を再始動するたびにこれらのオブジェクトを再定義する必要がなくなるため、この方法をお勧めします。

MQCONN リソースの定義とインストール

CICS が IBM MQ への接続を開始する前に、MQCONN リソースをインストールする必要があります。

このタスクについて

MQCONN リソースは、CICS と IBM MQ の間の接続の属性を指定します。

手順

1. Ways of defining CICS resources にリストしているいずれかの方法を使用し、任意の名前とグループを指定して MQCONN リソースを作成します。
CICS の初期設定時に CICS-MQ との接続を自動的に開始することを計画している場合は、CICS **GRPLIST** システム 初期設定パラメーターで指定された 1 つ以上のリストに含まれるいずれかのグループ内に MQCONN リソースを配置します。
2. **MQNAME** 属性として、単一の IBM MQ キュー・マネージャー、または複数の IBM MQ キュー・マネージャーから成るキュー共用グループの名前を 1 文字から 4 文字で指定します。
キュー共用グループを指定した場合、CICS は単一のキュー・マネージャーを待機する代わりに、グループ内の任意のキュー・マネージャーを使用します。キュー共用グループを使用することで、IBM MQ への再接続時の柔軟性が高まり、複数の CICS 領域および z/OS イメージの間でこの CICS セットアップを標準化するうえで役立ちます。
3. IBM MQ キュー・マネージャーのキュー共用グループを既に指定した場合、**RESYNCMEMBER** 属性を使用して、接続が失われた場合の CICS 再同期ストラテジーを適切に選択してください。
 - デフォルト値である **RESYNCMEMBER(YES)** は、接続で作業単位が残存する場合に CICS が同じキュー・マネージャーに再接続する (その際、必要に応じて待機する) ことを指定します。このストラテジーを使用すると、未確定の作業単位が解決される可能性が高くなりますが、IBM MQ への再接続にかかる時間が長くなるかもしれません。
 - **RESYNCMEMBER(NO)** は、CICS が同じキュー・マネージャーに再接続する試行を一度行うことを指定します。その試行が失敗した場合、CICS はグループ内の別の適格なキュー・マネージャーに接続し、未確定の作業単位が元のキュー・マネージャーに存在していればメッセージ DFHMQ2064 を出します。キュー・マネージャーは、CICS 領域と同じ LPAR 上で現在アクティブであれば、CICS 領域に接続できます。このストラテジーにより、IBM MQ への再接続にかかる時間が短くなる可能性があります。ただし、CICS が別のキュー・マネージャーに接続した場合、未確定の作業単位を自動的に解決することはできず、手動で解決する必要があります。
 - **RESYNCMEMBER(GROUPRESYNC)** は、未処理で未確定の作業単位があるかどうかにかかわらず、CICS がキュー共用グループの任意のメンバーに接続できることを指定します。IBM MQ は CICS が接続するキュー・マネージャーを選択し、キュー共用グループのすべての適格キュー・マネージャーの代わりに未確定の作業単位を解決するように CICS に依頼します。この機能のことをグループ・リカバリー単位といいます。このオプションを使用できるのは、CICS のグループ・リカバリー単位をサポートしているリリースの IBM MQ を実行していて、キュー・マネージャーでグループ・リカバリー単位を有効にした場合に限られます。
4. **INITQNAME** 属性として、接続のデフォルト開始キューの名前を 1 文字から 48 文字までで指定します。
CSQ4INYG オブジェクト・サンプルで定義されている開始キューの名前は CICS01.INITQ です。
5. MQCONN リソースをインストールします。

タスクの結果

MQCONN リソースのインストールに加えて、CICS はデフォルトの開始キューを表す予約名 DFHMQINI を使用して MQMONITOR リソースを動的にインストールします。DFHMQINI について詳しくは、MQMONITOR リソースを参照してください。

次のタスク

MQCONN リソースを変更して再インストールする場合には、CICS と IBM MQ との接続を停止する必要があります。CICS が IBM MQ と接続されていないときにのみ、MQCONN リソースをインストールできます。

動的にインストールされている MQMONITOR の **QNAME** 属性を変更する場合、まず MQCONN リソースの **INITQNAME** 属性を変更してから、MQCONN リソースを再インストールする必要があります。

開始キューなどの MQ キューをモニターするための追加の MQMONITOR リソースをインストールできます。MQ キューをモニターする MQMONITOR リソースを複数指定できます。

MQMONITOR DFHMQINI のデフォルト設定に関して懸念がある場合は (例えば DFHMQINI へのマイグレーションが予想より複雑になるなど)、DFHMQINI という名前のユーザー定義 MQMONITOR リソースをインストールすることができます。これにより、属性 AUTOSTART、STATUS、MONUSERID、および USERID をユーザー定義値に柔軟に設定できるので、後方互換性が確保され、マイグレーションが容易になります。TRANSACTION 属性は CKTI でなければなりません。

MQMONITOR リソースの定義とインストール

MQ キュー (開始キューなど) での着信メッセージを処理するための MQMONITOR リソースをインストールすることができます。複数の MQMONITOR リソースで MQ キューでの着信メッセージを処理できます。CICS-MQ トリガー・モニター CKTI のインスタンスの制御方法としてお勧めする方法が、MQMONITOR リソースを使用することです。

始める前に

すでに MQCONN リソースをインストールしている場合にのみ、MQMONITOR リソースをインストールすることができます。

このタスクについて

MQMONITOR リソースを定義およびインストールして、CICS-MQ トリガー・モニター CKTI を制御できます。

また、ユーザー作成のトリガー・モニターや MQ メッセージ・コンシューマーのための MQMONITOR リソースを定義してインストールすることもできます。デフォルトの CKTI ではなくユーザー作成の MQ モニター・プログラムを使用して MQMONITOR リソースを処理する場合、ユーザー作成のプログラムでアプリケーション・キューから直接メッセージを取得して必須のロジックを実行することが必要になります。ユーザー自身がプログラムをコーディングしている場合は、ユーザー作成の MQ トリガー・モニターと MQ メッセージ・コンシューマーの開発と使用に説明されているガイドラインに必ず従ってください。

注：CICS 提供の MQ トリガー・モニター・プログラム DFHMQTSK は、CICS-MQ トリガー・モニターおよびタスク・イニシエーター・トランザクション CKTI で使用するために予約されています。ユーザー・トランザクションとして DFHMQTSK を呼び出そうとすると、ユーザー・トランザクションは異常終了し、異常終了コード AMQO が示されます。

CICS-MQ アダプターが MQ に接続されていても、いつでも新規の MQMONITOR をインストールできます。

MQMONITOR が使用不可の状態、これを使用しているトランザクションがない場合にのみ、既存の MQMONITOR を (置き換えることによって) 再インストールできます。 **SET MQMONITOR DISABLED STOPPED** コマンドを使用して、関連タスクを停止し、リソースを使用不可にします。

MQMONITOR は、使用不可になっていて関連タスクがない場合にのみ廃棄できます。 **SET MQMONITOR DISABLED STOPPED** コマンドを使用して、関連タスクを停止し、リソースを使用不可にします。

INITQNAME 属性の値を指定する MQCONN リソースをインストールすると、CICS もデフォルトの開始キューを表す予約済みの名前 DFHMQINI を使用して、動的に MQMONITOR リソースを作成してインストールします。

DFHMQINI について詳しくは、[MQMONITOR リソース](#)を参照してください。

手順

1. [Ways of defining CICS resources](#) にリストされているいずれかの方法を使用して、MQMONITOR リソースを定義します。

例えば、**CREATE MQMONITOR** コマンドを使用して MQMONITOR リソースを定義できます。

[21 ページの表 2](#) のリストは、MQMONITOR リソースを作成するための重要な属性を示しています。

表 2. 重要な MQMONITOR 属性			
属性	必須/オプション	デフォルト	説明
AUTOSTART	オプション	YES	<p>この属性は、MQ モニターの自動再始動を制御します。</p> <p>AUTOSTART(YES) TRANSACTION 属性で指定したトランザクションを、MQ キュー・マネージャーへの接続が確立されたときに自動で再始動できるようにします。</p> <p>AUTOSTART(NO) MQ モニターは自動的に開始されません。MQ キュー・マネージャーへの接続が確立された後、MQ モニターを手動で始動する必要があります。</p>
MONUSERID	必須	-	<p>MQ キューをモニターしているトランザクションに関連付けるユーザー ID を指定します。この属性は、セキュリティチェックがアクティブである (つまり、SEC システム初期設定パラメーターが YES に設定されている) 場合にのみ有効です。セキュリティチェックが無効にされている (つまり、SEC が NO に設定されている) 場合、MQ モニター・トランザクションに関連付けるユーザー ID は、MQMONITOR リソースの状態を「開始」に設定したトランザクションのユーザー ID です。</p>
QNAME	オプション	QNAME を省略した場合、この値は、リソースのインストール時にデフォルトで &APPLID..INITIATION.QUEUEUE に設定されます。	<p>モニター対象の MQ キューの名前を指定します。</p>
STATUS	オプション	ENABLED	<p>この属性により、領域でリソースを使用できるようにします。</p>
TRANSACTION	オプション	CKTI	<p>MQ キューをモニターする CICS トランザクションの 4 文字の ID を指定します。</p> <p>MQ 開始キューをモニターするための MQMONITOR を定義する場合は、CKTI を指定するか、ブランクのままにします。</p>

MQMONITOR に関連付けたトランザクションを CICS で自動的に始動するためには、MQMONITOR リソースに属性 AUTOSTART(YES) および STATUS(ENABLED) を指定してください。

2. MQMONITOR リソースをインストールします。

MQMONITOR リソースの定義の検査

MQMONITOR を定義すると、CICS は、同じグループあるいはリスト内の他のリソース定義との整合性を検査します。

MQMONITOR オブジェクトでは、CICS は以下の検査を実行します。

- 同じリストに名前の等しい 2 つの MQMONITOR が出現しないこと。表示されている場合、警告メッセージが出されます。

- MQCONN がグループまたはリストに存在すること。存在しない場合、警告メッセージが出されます。MQCONN は別の グループの他の場所に存在できるため、エラーでない場合がありますが、MQMONITOR をインストールするためには、MQCONN がインストールされていることが必要です。

IBM MQ への接続を開始するための PLTPI プログラムの作成

サンプル DFHMQLPT に基づいて CICS-MQ 接続を開始するための独自の PLTPI プログラムを作成できます。このサンプルは、SDFHSAMP ライブラリーにあります。

このタスクについて

このサンプルはアセンブラー言語で作成されていますが、CICS でサポートされる任意の言語で独自のプログラムを作成できます。CICS PLTPI プログラムの作成方法の詳細については、[Writing initialization and shutdown programs](#) を参照してください。

作成した PLTPI プログラムは、CICS-MQ アダプター・プログラムである DFHMQQCN にリンクし、使用する接続値を指定したパラメーター・リストを渡す必要があります。パラメーター・リストについては、[CICS アプリケーション・プログラムからの DFHMQQCN へのリンクによる接続の開始](#)で説明しています。

手順

- PLTPI プログラムは、次のような LINK コマンドを発行して、DFHMQQCN にリンクしてパラメーター・リストを渡す必要があります。

```
EXEC CICS LINK PROGRAM('DFHMQQCN')  
          COMMAREA(CONNPL) LENGTH(length of CONNPL)
```

この例では、パラメーター・リストの名前は、CONNPL です。

CICS 始動のこの段階では端末は利用できないため、COMMAREA オプションを使用してパラメーターを渡す必要があります。

タスクの結果

MQ モニターを使用している場合は、MQ 接続を開始すると MQ モニターが自動的に開始されます (属性 AUTOSTART(YES) および STATUS(ENABLED) を指定して MQMONITOR リソースがインストールされている必要があります)。

DFHMQCOD または独自のプログラムを PLTPI リストで指定する

DFHMQCOD または独自のプログラムを使用して、CICS 初期化の際に IBM MQ への接続を自動的に開始する場合、以下の手順に従ってプログラムを CICS PLTPI リストに追加します。

このタスクについて

PLTPI リストは、プログラム・リスト・テーブル (PLT) に、PLTPI システム初期設定パラメーターで指定された接尾部を加えたものです。CICS システム初期設定パラメーター **MQCONN=YES** を指定した場合、DFHMQCOD または独自のプログラムを CICS PLTPI リストに追加する必要はありません。**MQCONN** パラメーターを使用すると、CICS-MQ 接続の開始には必ず DFHMQCOD プログラムが使用されます。別の名前のユーザー提供の接続プログラムを使うようにカスタマイズすることはできません。**MQCONN** パラメーターを使用しない場合は、以下の手順を続けてください。

手順

1. 現在の PLTPI リストに DFHDELIM のエントリーが含まれていない場合は、追加します。
DFHMQCOD またはこれに相当する独自のプログラムは、CICS 初期設定の 3 番目の段階で実行するため、DFHDELIM のエントリーの後に追加する必要があります。
2. CICS DFHPLT マクロを使用して、DFHMQCOD または独自のプログラムを PLTPI リストに追加します。
[23 ページの図 3](#) を参照してください。
3. PLTPI リストの接尾部が **PLTPI** システム初期設定パラメーターで正しく指定されていることを確認します。

4. MQCONN リソースが、**GRPLIST** システム 初期設定パラメーターで指定した 1 つ以上のリストに含まれているグループのいずれかに存在することを確認します。

IBM MQ への接続を開始するには、MQCONN リソースが必要です。

タスクの結果

MQ モニターを使用している場合は、MQ 接続を開始すると MQ モニターが自動的に開始されます (属性 AUTOSTART(YES) および STATUS(ENABLED) を指定して MQMONITOR リソースがインストールされている必要があります)。

例

次の例は、DFHPLT41 という CICS PLT で、DFHMQCOD のエントリーを DFHDELIM のエントリーの後にコーディングする方法について説明しています。

```
DFHPLT41 DFHPLT TYPE=INITIAL,SUFFIX=41
         DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
         DFHPLT TYPE=ENTRY,PROGRAM=DFHMQCOD
         DFHPLT TYPE=FINAL
         END
```

図 3. CICS-MQ アダプターで使用するためのサンプル PLT

このサンプル PLT を使用するには、**PLTPI** システム 初期設定パラメーターで接尾部 41 を指定します。PLT エントリーのコーディングの詳細については、[Program list table \(PLT\)](#)を参照してください。

CICS-MQ ブリッジのセットアップ

CICS-MQ ブリッジを CICS 領域で使用できるようにするには、これらのタスクを実行してください。

始める前に

CICS-MQ ブリッジを使用するには、IBM MQ と CICS の両方が同じ z/OS イメージで実行されている必要があります。

手順

CICS でステップ 23 ページの『1』から 24 ページの『4』までを実行します。

1. CICS で、17 ページの『CICS-MQ アダプターのセットアップ』の指示に従って CICS-MQ アダプターをセットアップします。
2. CICS で、CICS-MQ ブリッジ・コンポーネントのリソース定義に次の変更を加えます。
 - a) 接頭部 CKB を持つ CICS 一時ストレージ ID がリカバリー不能として定義されていることを確認します。
CICS-MQ ブリッジは、これらの一時ストレージ ID を使用します。
 - b) CICS-MQ ブリッジを使用して 24 ビットの CICS DPL プログラムを実行する場合は、トランザクション CKBP または CKBC、あるいはその両方の定義の TASKDATALOC 属性を BELOW に変更してください。
CICS DPL プログラムは、デフォルトではトランザクション・コード CKBP で実行されます。
TASKDATALOC 属性を変更せずに 24 ビットのプログラムを実行すると、CICS 異常終了 AEZC が発行される場合があります。
 - c) CICS DPL プログラムを別のトランザクション・コードで実行する場合は、CKBP または CKBC、あるいはその両方の定義のコピーをインストールし、そのトランザクション名を任意の名前に変更してください。

24 ビット・プログラムを実行する場合は、必ず TASKDATALOC 属性を BELOW に変更してください。
IBM MQ 要求メッセージの MQCIH ヘッダーに、代わりのトランザクション・コードを指定する必要があります。

グループ DFHMQ にインストールされている CICS-MQ ブリッジ・コンポーネントの完全なリストについては、25 ページの『[CICS での CICS-MQ ブリッジ・トランザクションおよびプログラムの定義](#)』を参照してください。

3. オプション: CICS MQ モニターをセットアップして、ブリッジを制御します。

MQMONITOR を使用すると、IBM MQ マネージャーへの接続が確立したときにブリッジが自動的に再始動するように設定できます。これは、CICS-MQ ブリッジを制御するための推奨される方法の 1 つです。セットアップ手順については、26 ページの『[CICS-MQ ブリッジ用の MQMONITOR のセットアップ](#)』を参照してください。

注: CICS-MQ ブリッジを制御する代替方法は、ブリッジ要求キューを開始キューに関連付け、メッセージがブリッジ要求キューに到着したときにトランザクション CKBR が開始されるようにするプロセスを実装する方法です。

4. 有効期限切れのマークが付いたメッセージを DLQ に転送するかどうかを決定する ROUTEMEM ブリッジ開始パラメーターを使用する場合は、共用ブリッジ・キューと非共用ブリッジ・キューのどちらを使用するかに応じて、オプションの 2 つの TSMODEL リソース定義のいずれかを CICS にインストールします。

定義は以下のとおりです。

a. 非共用ブリッジ・キュー

```
DEFINE TSMODEL(DFHMQMDL) GROUP(DFHMQ)
DESCRIPTION(CICS owned MQ-CICS bridge for RouteMEM non-shared)
PREFIX(CKBR) LOCATION(AUXILIARY) RECOVERY(NO) SECURITY(NO)
```

b. 共用ブリッジ・キュー

```
DEFINE TSMODEL(DFHMQMDL) GROUP(DFHMQ)
DESCRIPTION(CICS owned MQ-CICS bridge - for RouteMEM shared)
PREFIX(CKBR) LOCATION(AUXILIARY) RECOVERY(NO)
SECURITY(NO) POOLNAME(MQPOOL)
```

ROUTEMEM ブリッジ開始パラメーターの詳細については、[CICS-MQ ブリッジの管理](#) を参照してください。

IBM MQ でステップ 24 ページの『5』 から 25 ページの『8』 までを実行します。

5. IBM MQ で、要求メッセージ用のローカル・キューを定義します。

この IBM MQ 要求キューは、CICS-MQ ブリッジのローカルに存在しなければならず、他のアプリケーションで使用することはできません。

サンプル thlqual.SCSQPROC(CSQ4CKBM) を使用して SYSTEM.CICS.BRIDGE.QUEUE というデフォルトのキューを定義することも、独自のキューを定義することもできます。

独自のローカル・キューを定義した場合は、ブリッジ・モニター・トランザクション CKBR を開始するときに必ずその名前を指定する必要があります。

独自のローカル・キューを定義する場合は、以下の属性を設定します。

- ブリッジ・モニターとブリッジ・タスクの両方がキューを読み取れるように **SHARE** 属性を設定します。
- リカバリーが必要な場合は、属性 **DEFPSIST(YES)** を設定して、デフォルトでメッセージがキュー上で持続するように定義します。
- FIFO の順序でメッセージを処理する場合は、**MSGDLVSQ(FIFO)** 属性を設定します。
これを設定しないと、メッセージは優先順位の順序で処理されます。
- QSGDISP(SHARED)** を指定して要求キューを定義する場合は、**INDXTYPE(CORRELID)** も指定する必要があります。

この設定は、非共用キューにもお勧めします。

- e) トリガーによってブリッジ・モニターを開始する場合、属性 **TRIGGER**
TRIGTYPE(FIRST)PROCESS(procid) を設定します。 *procid* は **APPLICID(CKBR)** を指定するプロセスです。
- f) **BOQNAME** および **BOTHRESH** を要求キューに指定することを検討してください。 **BOTHRESH** 属性で指定した回数までメッセージが処理され、バックアウトされた場合に、メッセージを送達不能キューに入れるのではなく、 **BOQNAME** 属性で指定したバックアウト・キューに入れることができます。
 バックアウト・キューを指定する場合は、そのキューのメッセージを処理するプロセスを用意してください。
- g) コンテナを使用する場合は、キューの **MaxMsgLen** 属性を使用してブリッジ入力メッセージのサイズを制限することを検討してください。1つのキューに入れられる最大メッセージ長は、その **MaxMsgLen** 属性によって定義されます。
6. IBM MQ で、応答を保持するためのキューを1つ以上定義します。
 応答キューは、ローカルまたはリモートのいずれでもかまいません。
 応答キューがリモートの場合は、応答キューに伝送されるまで応答を入れておく伝送キューを定義する必要があります。
7. IBM MQ で、送達不能キューが定義されていること、およびそのキューのメッセージを処理するためのプロシージャが定義されていることを確認します。
8. IBM MQ からリモートのブリッジにアクセスする場合は、要求キューに対するチャネル定義と伝送キュー定義、およびリモート・キュー定義を作成します。
 リモート・キューの使用について詳しくは、[IBM MQ 製品資料](#)を参照してください。

CICS での CICS-MQ ブリッジ・トランザクションおよびプログラムの定義

ブリッジの定義は、グループリスト DFHLIST に含まれているグループ DFHMQ に用意されています。

CKBC	チャネルおよびコンテナ DPL ブリッジ用のブリッジ ProgramLink トランザクション
DFHMQBP3	チャネルおよびコンテナで使用する CICS-MQ Bridge DPL プログラム
CKBP	COMMAREA DPL ブリッジ用のブリッジ ProgramLink トランザクション。デフォルトでは、CICS DPL プログラムはこのトランザクション・コードで実行されます。
DFHMQBP0	COMMAREA で使用する CICS-MQ Bridge DPL プログラム
CKBR	ブリッジ・モニター・トランザクション
DFHMQDCI	データ変換出口
DFHMQBRO	ブリッジ・モニター・プログラム
DFHMQBP1	CICS-MQ Bridge DPL 異常終了ハンドラー
DFHMQBRO	CICS-MQ Bridge モニター・プログラム
CSQCBP00	DPL プログラムの CICS-MQ Bridge 別名
CSQCBR00	モニター・プログラムの CICS-MQ Bridge 別名
CSQCBP53	MQ V531 Bridge、DPL プログラムの別名
CSQCBP10	MQ V531 Bridge、DPL 異常終了ハンドラー
CSQCBR53	MQ V531 Bridge、モニター・プログラムの別名
DFHMQDCI	CICS-MQ Bridge データ変換出口
CSQCBDCI	データ変換出口の CICS-MQ Bridge 別名
DFHCKBR	routemem 処理のための CICS-MQ Bridge tsmodel

CICS-MQ ブリッジ用の MQMONITOR のセットアップ

オプションで、CICS-MQ ブリッジを制御する MQMONITOR リソースを定義してインストールすることができます。MQMONITOR リソースを使用すると、IBM MQ マネージャーへの接続が確立したときにブリッジが自動的に再始動するように設定できます。これは、CICS-MQ ブリッジを制御するための推奨される方法の 1 つです。

手順

1. 次の属性を指定して MQMONITOR リソースを定義します。

表 3. CICS-MQ ブリッジを制御するための MQMONITOR リソース属性			
属性	必須/オプション	デフォルト	説明
AUTOSTART	オプション	YES	<p>この属性は、MQ モニターの自動再始動を以下のように制御します。</p> <p>AUTOSTART(YES) IBM MQ キュー・マネージャーへの接続が確立されたときに、ブリッジが自動で再始動するようにします。</p> <p>AUTOSTART(NO) MQ モニターは自動的に開始されません。IBM MQ キュー・マネージャーへの接続が確立された後、MQ モニターを手動で始動する必要があります。</p>
MONDATA	必須	-	CKBR トランザクションに渡されるパラメーターを指定します。
MONUSERID	必須	-	<p>CKBR に関連付けるユーザー ID を指定します。</p> <p>この属性は、セキュリティチェックがアクティブである (つまり、SEC システム初期設定パラメーターが YES に設定されている) 場合にのみ有効です。</p>
STATUS	オプション	ENABLED	この属性により、領域でリソースを使用できるようにします。
TRANSACTION	必須	CKBR	CICS-MQ ブリッジの 4 文字のトランザクション ID を指定します。

注: AUTOSTART(YES) および STATUS(ENABLED) を使用すると、MQ キュー・マネージャーへの接続が確立されたときに、ブリッジを自動で再始動できます。

2. MQMONITOR リソースをインストールします。

複数の CICS-MQ ブリッジ・モニターのセットアップ

CICS-MQ ブリッジを介した CICS アプリケーションに対する大量の要求が预期される場合は、追加の CICS 領域で追加のブリッジ・モニターをセットアップすることを検討してください。追加のブリッジ・モニターを使用して、より細かいセキュリティを実現することもできます。

このタスクについて

ブリッジ・モニターのデフォルトのトランザクション ID は CKBR ですが、これを変更することも、追加のトランザクション ID を定義することもできます。

複数のブリッジ・モニターをセットアップする場合、ブリッジ・モニターごとに専用の個別の要求キューを作成できます。この方法の利点は、ブリッジ・モニターごとに異なるサービス特性を指定できる点です。ただし、欠点として、アプリケーションが CICS-MQ ブリッジを介して要求を行うときに、使用すべき適切な要求キューの名前を認識している必要があります。この理由から、複数のブリッジ・モニターで同じ要求キューを共用することが望ましい場合があります。

複数のブリッジ・モニターで 1 つの共用の要求キューを使用する場合は、以下の要件に従う必要があります。

手順

- すべてのブリッジ・モニターが WebSphere バージョン 6 以降のキュー・マネージャーに関連付けられている必要があります。WebSphere MQ バージョン 5.3.1 より前のシステムに接続されたブリッジ・モニターで共用キューを使用して複数のブリッジ・モニターを実行することは避けてください。予期しない結果が生じます。
- ブリッジ・モニターを実行する各 CICS 領域の SYSID は固有でなければなりません。
- 3270 疑似会話の中のすべてのトランザクションは、最初のトランザクションから戻されたりリモート SYSID を、その疑似会話内のすべての後続メッセージに指定する必要があります。このリモート SYSID は、CICS トランザクション・ルーティング機能を使用して、他の CICS 領域にトランザクションを送信する場合であっても必要です。
- ブリッジ・モニターが実行されるすべての CICS 領域間に CICS ISC リンクを定義する必要があります。
- ユーザー検証でパスチケットを使用する場合は、キューのすべてのブリッジ・モニターが始動時に PASSTKTA パラメーターを使用して同じ APPLID を指定する必要があります。

次のタスク

複数の CICS ブリッジ・モニターを使用する場合の問題判別では、ブリッジによって生成された エラー・メッセージを見つけるために、すべての CICS 領域のすべてのログを調べなければならない場合があります。各キュー・マネージャーでコマンド `DISPLAY QSTATUS(queueename) TYPE(HANDLE)` を使用して、キューが開かれている CICS 領域を調べることができます。

CICS-MQ ブリッジ・スループットの制御

ブリッジ・プログラム・トランザクション CKBP を独自のクラスに入れて、要件に合わせて CLASSMAXTASK を設定することで、ブリッジのスループットを制御できます。

ブリッジ・モニター CKBR で参照された要求メッセージは、マークが付けられ、Mark Browse Interval (MarkBint) で指定した期間にわたって隠されます。これにより、CKBR によって開始された後に CKBP がメッセージを取得するための時間が確保されます。CKBP が MarkBint 内にメッセージを取得しなかった場合は、再処理のためにメッセージがもう一度使用可能になります。CKBP が何らかの理由でキューからメッセージを取得しない場合は、再処理が無限に続けられます。これは、この状況でのブリッジのデフォルトのアクションです。

このデフォルトのアクションは、ブリッジ開始データに `ROUTEMEM=Y` パラメーターを指定して変更できます。このパラメーターを使用した場合は、マークの期間が終了して再処理のためにメッセージが可視になると、それらのメッセージは送達不能キュー (DLQ) に転送されます。ROUTEMEM ブリッジ開始パラメーターの詳細については、[CICS-MQ ブリッジの管理](#) を参照してください。

第 3 章 CICS-MQ アダプターの管理

CICS-MQ アダプターの制御機能を使用することにより、CICS と IBM MQ の間の接続を開始して、管理および表示することができます。CICS-MQ モニターを使用して、CICS-MQ トリガー・モニターの CKTI を制御できます。

始める前に

CICS-MQ アダプターを操作するためには、CICS 領域を適切に構成し、MQCONN リソース定義をインストールする必要があります。

INITQNAME 属性の値を指定する MQCONN リソースをインストールすると、CICS もデフォルトの開始キューを表す予約済みの名前 DFHMQINI を使用して、動的に MQMONITOR リソースを作成してインストールします。

DFHMQINI は、MQ 接続が確立された時に自動的に開始されます。

オプションで、追加の MQMONITOR リソースをインストールして、トリガー・モニター・トランザクションの CKTI などの IBM MQ メッセージ・コンシューマーの属性を定義することができます。

詳しい説明は、[CICS-MQ アダプターのセットアップ](#) を参照してください。

このタスクについて

以下のいずれかの方法を使用して、CICS-MQ アダプターを管理することができます。

CICS-MQ アダプターの制御機能を使用する

制御機能は、CKQC トランザクションか、**EXEC CICS SPI** と CEMT コマンド、CICSplex SM、または CICS Explorer を使用することによって利用できます。

- CICS 環境でコマンドを実行したり、アプリケーション・プログラム内で **EXEC CICS** システム・プログラミング・インターフェース (SPI) コマンドを使用したりするには、CEMT トランザクションを使用することができます。
- CKQC トランザクションは、CICS-MQ アダプター制御パネルを介してか、コマンド行またはアプリケーション・プログラムから使用できます。
- CICS 領域が CICSplex SM によって管理されている場合は、CICSplex SM Web ユーザー・インターフェースを使用したり、アプリケーション・プログラムで EXEC CPSM アプリケーション・プログラミング・インターフェースを使用したり、または CMCI プログラミング・インターフェースを使用したりするなどのいくつかの方法で接続を管理できます。これらの方法は、事実上、CICS システム・プログラミング・インターフェース (SPI) コマンドを使用して、使用可能な CICSplex SM リソース・テーブルを更新します。
- CICS Explorer には、CICS-MQ 接続および MQ モニター用の「SM Operations (SM 操作)」ビューと「SM Administration (SM 管理)」ビューがあります。これらは、**EXEC CICS SPI** コマンドと同じ機能を提供します。CICS Explorer を使用して CICS リソースを扱う方法については、CICS Explorer に付属のヘルプ情報を参照してください。

CICS-MQ モニターを使用して、CICS-MQ トリガー・モニターである CKTI のインスタンスを制御する
MQMONITOR リソースは、CKTI の推奨制御法です。以下の利点があります。

- 1 つの MQ 開始キューをモニターするために CKTI の複数インスタンスをセットアップできます。
- MQ キュー・マネージャーへの接続が確立された時に CKTI を自動的に開始できます。
- MQMONITOR は、CKTI またはメッセージ・コンシューム・トランザクションが、事前設定されたユーザー ID で実行されることを許可します。



注意: CKQC を使用して CKTI のインスタンスを開始したり停止したりすることもできます。ただし、領域が MQ モニターをインストールしていて、CKTI の複数インスタンスが実行されている場合、CKQC を使用して CKTI を停止すると、予測不能の結果になる可能性があります。

オペレーター・アクションの要約

次の表では、CICS-MQ 接続に関して実行できるオペレーター・アクションと、それらのアクションを **EXEC CICS** と CEMT コマンド、CKQC トランザクション、CICS Explorer、または CICSplex SM を使用して実行できるかどうかを要約しています。

表 4. CICS-MQ 接続のオペレーター・アクション			
オペレーター・アクション	EXEC CICS、CEMT	CKQC	CICS Explorer または CICSplex SM
CICS-MQ 接続の開始	はい。 SET MQCONN を使用。ただし、デフォルトの開始キュー名は指定できない。	はい	はい
CICS-MQ 接続の停止	はい。 SET MQCONN を使用。	はい	はい
接続状況と設定の表示	はい。 INQUIRE MQCONN を使用。	はい	はい
接続時刻と切断時刻の表示	はい。CICS 統計コマンドを使用。	いいえ	はい
呼び出しのタイプを含む詳細な接続統計の表示とリセット	はい。CICS 統計コマンド (すべての統計のリセット) を使用。	はい (CICS-MQ 接続統計のリセットのみ)	いいえ
CICS-MQ 接続を使用しているタスクの表示	はい。ただし、タスク数のみ。 INQUIRE MQCONN を使用。	はい。タスクの詳細な一覧表示。	いいえ
CICS-MQ 接続を使用している個々のタスクのページ	はい。 SET TASK FORCEPURGE を使用。	いいえ	はい
CICS-MQ API 交差出口を使用可能または使用不可に設定	いいえ	はい	いいえ
CKTI (CICS-MQ トリガー・モニターまたはタスク・イニシエーター) のインスタンスの開始	いいえ	はい。ただし、 MQMONITOR を使用して CKTI を制御することが推奨されます	いいえ
CKTI のインスタンスの停止	いいえ	はい。ただし、 MQMONITOR を使用して CKTI を制御することが推奨されます	いいえ
MQ モニター (MQMONITOR リソース) の開始	はい	いいえ	はい
MQ モニター (MQMONITOR リソース) の停止	はい	いいえ	はい

表 4. CICS-MQ 接続のオペレーター・アクション (続き)			
オペレーター・アクション	EXEC CICS、CEMT	CKQC	CICS Explorer または CICSplex SM
すべての呼び出しタイプを含めた、詳細な MQMONITOR 統計の表示とリセット	はい	いいえ	はい CICSplex SM は、MQMONITOR レコードで CICS 統計を返します。統計をリセットするには、CICSRGN リソース・テーブルを使用できます。

CICS-MQ アダプター制御パネルへのアクセス

アダプター制御パネルにアクセスするには、CICS トランザクション CKQC を使用します。

このタスクについて

アダプター制御パネルには、キュー・マネージャーを開始せずにアクセスできます。また、接続も開始できますが、キュー・マネージャーが開始されるまではアクティブになりません。

手順

1. CKQC と入力し、Enter キーを押します。
CICS-MQ アダプター制御初期パネルが表示されます。
2. メニュー・バーで、Tab キーを使用して、3 つのオプション、「**Connection (接続)**」、「**CKTI**」、および「**Task (タスク)**」の間を移動します。Enter キーを押して項目を選択します。
3. 選択項目の番号を入力してメニューの 1 つからアクションを選択し、次に確認のために Enter キーを押します。取り消す場合は、ファンクション・キー F12 を押します。
4. 表示されたパネルまたは 2 次ウィンドウで、必要に応じてフィールドに新しい値を入力します。
5. パネルまたはウィンドウでヘルプを表示するには、ファンクション・キー F1 を押します。

コマンド行からの CKQC コマンド

コマンド行から直接 CKQC トランザクションを使用し、コマンド・パラメーターを指定することにより、CICS-MQ アダプター制御パネルをバイパスすることができます。

例えば、以下のコマンドを入力して、デフォルト接続値を使用して CICS-MQ 接続を CICS コマンド行から開始することができます。

```
CKQC START
```

また、z/OS コマンドを使用して、CKQC トランザクション・コマンドをコンソールから発行することもできます。これらのコマンドは、以下の形式を取ります。

```
MODIFY CICS-job-name CKQC command-line-command
```

CKQC トランザクション・コマンドは以下のとおりです。

CKQC START

IBM MQ への接続を開始します。このコマンドの構文と例については、[37 ページの『CICS コマンド行からの CICS-MQ 接続の開始』](#)を参照してください。

CKQC STOP

IBM MQ への接続を停止します。このコマンドの構文と例については、[43 ページの『CICS コマンド行からの CICS-MQ 接続の停止』](#)を参照してください。

CKQC DISPLAY

IBM MQ への接続に関する重要な情報を、CICS メッセージの形式で返します。このコマンドにはオプションはありません。例については、[46 ページの『CICS-MQ 接続状況と設定の表示』](#)を参照してください。

CKQC STARTCKTI

CICS-MQ トリガー・モニターである CKTI のインスタンスを開始します。このコマンドの構文と例については、[61 ページの『CICS コマンド行からの CKTI の開始』](#)を参照してください。

CKQC STOPCKTI

CKTI のインスタンスを停止します。このコマンドの構文と例については、[64 ページの『端末からの CKTI インスタンスの停止』](#)を参照してください。

CKQC MODIFY

CICS-MQ 接続統計をリセットし、CICS-MQ API 交差出口を使用可能または使用不可にします。このコマンドの例については、[CICS-WebSphere MQ API 交差出口の使用可能化](#)、[CICS-WebSphere MQ API 交差出口の使用不可化](#)、および [56 ページの『CICS-MQ 接続統計のリセット』](#)を参照してください。このコマンドの完全な構文は以下のとおりです。

```
CKQC MODIFY Y|N E|D
```

ここで、

Y|N

次のいずれかを指定します。

Y

接続統計をリセットします。

N

接続統計をリセットしません。

このパラメーターは必須です。

E|D

次のいずれかを指定します。

E

CICS-MQ API 交差出口を使用可能にします。

D

CICS-MQ API 交差出口を使用不可にします。

CICS アプリケーション・プログラムからの CKQC コマンド

EXEC CICS LINK コマンドを使用して、CICS アプリケーション・プログラムから、ほとんどの CICS-MQ アダプター制御機能にアクセスできます。

接続の開始などのアクションを実行するために、アプリケーション・プログラムは適切な CICS-MQ アダプター・プログラムにリンクし、CKQC コマンドを発行する必要があります。[32 ページの表 5](#)は、これらのプログラムおよびその機能をリストしています。

表 5. CICS-MQ アダプター・プログラム			
CICS-MQ アダプター・プログラム	互換性の点で受け入れられる代替名	機能	CKQC コマンド
DFHMQQCN	CSQCQCON	IBM MQ への接続の開始	CKQC START
DFHMQDSC	CSQCDSC	接続の停止	CKQC STOP
DFHMQRS	CSQCRST	接続の変更 (統計のリセットまたは API 交差出口の使用可能化)	CKQC MODIFY

表 5. CICS-MQ アダプター・プログラム (続き)			
CICS-MQ アダプター・プログラム	互換性の点で受け入れられる代替名	機能	CKQC コマンド
DFHMQSSQ	CSQCSSQ	CKTI インスタンスの開始または停止	CKQC STARTCKTI または CKQC STOPCKTI
DFHMQDSL	CSQCDSPL	接続の状況の表示	CKQC DISPLAY

アプリケーション・プログラムでのコマンド構文

アプリケーション・プログラムによって発行されるコマンドの一部には、末尾スペースを埋め込んで、コマンドの長さを 10 文字にしなければならないものがあります。

コマンドの後に引数がある場合は、区切り文字として、余分にスペース文字を追加する必要があります。この制限事項の影響を受けるコマンド、および各コマンドに必要な末尾スペースの数は、次のとおりです。

コマンド	末尾スペースの数 (区切り文字は含まない)
START	5
MODIFY	4
STARTCKTI	1
STOPCKTI	2

これ以外のすべてのコマンドでは、スペースの埋め込みはオプションです。

以下の例では、アダプター・コマンドの埋め込み方法が示されています。MODIFY コマンドでは、4 つの末尾スペースに加えて、区切り文字としてさらにもう 1 つのスペースを埋め込まなければなりません。MODIFY の M の文字から数えて、引数 Y は 12 番目の文字になります。

```
EXEC CICS LINK PROGRAM('DFHMQRS ')
      INPUTMSG('CKQC MODIFY      Y')
              ↑                ↑
              1                12
```

CICS アプリケーション・プログラムからのパラメーターの引き渡し

以下の規則を使用して、CICS と IBM MQ の間でパラメーターを受け渡す方法を決定します。

- CICS トランザクションは、接続された端末で実行する必要があります。この条件が満たされていないと、すべての IBM MQ コマンドは無視されます。
- 接続された端末の CICS アプリケーション・プログラムが IBM MQ に接続されている場合、パラメーターを渡すために、**EXEC CICS LINK** に INPUTMSG オプションを使用する必要があります。ただし、PLTPI 時は除きます。
- PLTPI 時に IBM MQ に接続する場合は、COMMAREA オプションを使用してパラメーターを渡す必要があります。INPUTMSG オプションを使用すると、コマンドは無視されます。ただし、アダプター STOP コマンド (CKQC STOP および CKQC STOP FORCE) は、INPUTMSG オプションまたは COMMAREA オプションを使用しているかどうかに関わらず、PLTPI 時には実行できません。

EXEC CICS LINK インターフェース・メッセージ

EXEC CICS LINK を使用して、アプリケーション・プログラムからアダプター操作機能の START および STOP を呼び出すと、結果として出されたメッセージが、システム・コンソールと、CKQQ という名前の一時データ・キュー (TDQ) の両方に書き込まれます。

アプリケーション・プログラムは、LINK から戻されると、**EXEC CICS READQ TD QUEUE(CKQQ)** を繰り返すことにより、キューが空になるまでメッセージを読み取ることができます。次の制限が適用されます。

- TDQ キュー名は CKQQ で、この名前を変更することはできません。CKQQ の定義は、CICS CSD グループの DFHDCTG に提供されています。
- キューは、書き込まれる前に消去されません。
- 複数のアプリケーションで TDQ に書き込みを行うと、メッセージはシリアルライズされません。呼び出し側プログラム自体でシリアルライズを行う必要があります。
- 同じメッセージ・セットがシステム・コンソールにも表示されます。

CICS-MQ 接続と MQ モニターのための EXEC CICS コマンドおよび CEMT コマンド

EXEC CICS システム・プログラミング・インターフェース (SPI) コマンドと CEMT コマンドを使用して、MQCONN と MQMONITOR のリソース定義を管理し、CICS-MQ 接続と MQ モニターを開始および停止し、接続および MQ モニターの情報と統計を表示し、接続を使用しているタスクをページすることができます。

EXEC CICS SPI コマンドは、CICS アプリケーション・プログラム内で発行できます。CEMT コマンドは、アプリケーション・プログラムを使用せずに、CICS CEMT トランザクションを介して直接発行できます。

以下の表は、**EXEC CICS** SPI コマンドと CEMT コマンド、およびそれらの機能を示しています。

表 6. CICS-MQ 接続と MQ モニターのための SPI コマンドおよび CEMT コマンド		
EXEC CICS コマンド	CEMT コマンド	機能
CREATE MQCONN	使用不可	MQCONN リソース定義のセットアップ
DISCARD MQCONN	CEMT DISCARD MQCONN	MQCONN リソース定義の破棄 注: INITQNAME 属性の設定を含む MQCONN リソースを破棄すると、動的に作成された MQMONITOR リソースとユーザー定義の MQMONITOR リソースも破棄されます。
INQUIRE MQCONN	CEMT INQUIRE MQCONN	MQCONN リソース定義属性、接続の状況、および接続を使用しているタスクの数の照会
SET MQCONN	CEMT SET MQCONN	接続の開始と停止、およびデフォルトのキュー・マネージャーまたはキュー共用グループ名および再同期戦略の変更
CREATE MQMONITOR	使用不可	MQMONITOR リソース定義のセットアップ
DISCARD MQMONITOR	CEMT DISCARD MQMONITOR	MQMONITOR リソース定義の破棄
INQUIREMQMONITOR	CEMT INQUIRE MQMONITOR	MQMONITOR リソース定義属性、およびモニターの現在の状況の照会 (例えば、ゼロでないタスク ID で MONSTATUS(STARTED))
SET MQMONITOR	CEMT SET MQMONITOR	MQMONITOR の開始および停止
INQUIRE SYSTEM	CEMT INQUIRE SYSTEM	インストールされている MQCONN リソース定義の名前の検出

表 6. CICS-MQ 接続と MQ モニターのための SPI コマンドおよび CEMT コマンド (続き)

EXEC CICS コマンド	CEMT コマンド	機能
SET TASK FORCEPURGE	CEMT SET TASK FORCEPURGE	CICS-MQ 接続を使用しているタスクのページ
EXTRACT STATISTICS MQCONN	使用不可	CICS-MQ 接続のオンラインでの統計の表示
SET STATISTICS	CEMT SET STATISTICS	CICS-MQ 接続用の統計を含めたすべての統計のリセット

CICS-MQ 接続と MQ モニターのための CICSplex SM ビュー

CICSplex SM ビューを使用して、MQCONN リソース定義と MQMONITOR リソース定義の作成と変更、CICS-MQ 接続の開始と停止、および接続とモニターの情報と統計の表示を行うことができます。

CICSplex SM 内では、CICS-MQ 接続と MQ モニター用に以下のビュー・セットが使用可能です。

「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ 接続**」

MQCONN ビュー・セット内のビューには、接続状況、接続時刻、および切断時刻を含め、接続に関する情報が表示されます。メイン・ビューには接続の開始と停止に使用できるアクション・ボタンがあり、また接続の設定を変更することができます。

「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ 開始キュー**」

MQINI ビュー・セット内のビューには、リソース・シグニチャーを含め、CICS システムの、インストールされている MQINI リソース定義の属性が表示されます。このリソース定義には何のアクションも実行できません。

「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ 接続統計**」

MQCONN ビュー・セット内のビューには、接続に関する情報と統計が表示されます。MQCONN ビュー・セットは、CICS-MQ 接続用に CICS リソース定義が導入される前から存在していたビュー・セットです。したがって、これらのビューには、接続を開始および停止するためのアクション・ボタンがなく、これらのビューで接続の設定を変更することはできません。

「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ モニター**」

MQMON ビュー・セット内のビューは、CICS 領域に定義されている MQ モニターの状況情報および統計を表示します。

「**管理**」 > 「**ベーシック CICS リソース管理**」 > 「**リソース定義**」 > 「**WebSphere MQ 接続定義**」

MQCONDEF ビュー・セット内のビューには、MQCONN リソース定義の属性が表示されます。それらは、CICSplex SM 内で MQCONDEF BAS オブジェクトによって表されます。これらのビューを使用して、MQCONN リソース定義を作成、変更、破棄、またはインストールできます。

「**管理**」 > 「**ベーシック CICS リソース管理**」 > 「**リソース・グループ内のリソース定義**」 > 「**リソース・グループ内の MQCONN リソース**」

MQCINGRP ビュー・セットは、リソース・グループ (RESGROUP) 内の MQCONN リソース定義のメンバーシップに関する情報を表示します。

「**管理**」 > 「**ベーシック CICS リソース管理**」 > 「**リソース・グループ内のリソース定義**」 > 「**リソース・グループ内の MQMON リソース**」

MQMINGRP ビュー・セットは、リソース・グループ (RESGROUP) 内の MQMONITOR リソース定義 (MQMONDEF) のメンバーシップに関する情報を表示します。

CICS-MQ 接続の開始

CICS 初期設定時に CICS-MQ 接続が開始されない場合、いくつかの方法で接続できます。

始める前に

CICS-MQ 接続を開始するためには、CICS 領域を適切に構成し、MQCONN リソース定義をインストールする必要があります。この方法については [CICS-MQ アダプターのセットアップ](#) で説明しています。

このタスクについて

CICS-MQ 接続を設定して、今後 CICS 初期設定を開始するには、CICS システム 初期設定パラメーター **MQCONN=YES** を指定するか、DFHMQCOD プログラムまたはユーザー独自の同等のプログラムを CICS PLTPI リストに組み込む必要があります。これを行う手順については、[CICS-MQ アダプターのセットアップ](#)を参照してください。

手順

CICS-MQ 接続を開始するには、以下のいずれかの方法を使用できます。

- [CICS-MQ アダプター制御パネルの使用](#)

CICS-MQ アダプター制御トランザクションを実行し、次に CICS-MQ アダプター制御パネルから CICS-MQ 接続を開始することができます。

- [CICS コマンド行からの **CKQC START** の発行](#)

CICS-MQ 接続は、コマンド行から **CKQC START** コマンドを発行することにより、開始できます。

- [EXEC CICS SET MQCONN を発行する CICS アプリケーション・プログラムの使用](#)

CICS アプリケーション・プログラムで **EXEC CICS SET MQCONN** コマンドを発行することにより、IBM MQ への接続を開始することができます。

- [リンク先を DFHMQQCN に指定した CICS アプリケーション・プログラムの使用](#)

CICS アプリケーション・プログラムから CICS-MQ アダプター接続プログラム DFHMQQCN (またはこのプログラムの以前の名前である CSQCQCON) にリンクすることにより IBM MQ への接続を開始できます。接続を開始する方法は、互換性のために維持されていますが、CICS アプリケーション・プログラムからの接続を開始するには、**EXEC CICS SET MQCONN** コマンドの使用が推奨されます。

- [CICS CEMT トランザクションの使用](#)

CICS CEMT トランザクションで **SET MQCONN** コマンドを発行することにより、IBM MQ への接続を開始することができます。

- [CICSplex SM Web ユーザー・インターフェースの使用](#)

IBM MQ への接続は、CICSplex SM Web ユーザー・インターフェース (WUI) の「**WebSphere MQ 接続**」ビューから開始できます。

- CICS Explorer の使用

CICS-MQ 接続は、CICS Explorer から開始できます。CICS Explorer は、CICSplex SM Web ユーザー・インターフェースおよび CEMT トランザクションと機能的に同等のものを提供します。

タスクの結果

CICS と MQ の接続が確立された後、MQCONN リソースを CONNECTED に設定したタスクに関連付けられているユーザー ID に関連するトランザクションを開始するための十分な権限がある場合、CICS は、属性 AUTOSTART(YES) および STATUS(ENABLED) を指定してインストールされた MQ モニターを開始します。

CICS-MQ アダプター制御パネルからの CICS-MQ 接続の開始

CICS-MQ アダプター制御トランザクションを実行し、次に CICS-MQ アダプター制御パネルから CICS-MQ 接続を開始することができます。

手順

1. CICS-MQ アダプター制御初期パネルで、メニュー・バーから「**Connection**」を選択します。
2. メニューから「**Start**」アクションを選択します。
3. 「**Start a Connection**」2 次パラメーター・ウィンドウに表示される接続値は、CICS 領域用にインストールされた MQCONN リソース定義と MQINI リソース定義から取られます。キュー・マネージャー名は、単一のキュー・マネージャーであるか、キュー共用グループであるかのいずれかです。これらの設定を変更するには、設定を上書きします。

設定を変更する場合、インストールされた MQCONN リソース定義および MQINI リソース定義の対応する属性 (MQNAME および INITQNAME) は、指定した設定に置換されます。元の設定に復帰するには、リソース定義を再インストールします。

4. Enter を押して確認します。

タスクの結果

接続の開始試行が成功したか失敗したかを示すメッセージが、アダプター・メッセージ・パネル CKQCM1 に表示されます。

例

Connection	CKTI	Task
Select an action.	CS Adapter Control -- Initial panel	
1 1. Start...		sing Tab key. Then press Enter.
2. Stop...		
3. Modify...		
4. Display		
F1=Help F12=Cancel		
		<div> Start a Connection Type parameters. Then press Enter. 1. Queue Manager Name (SN) . . . QMGR 2. Initiation Queue Name (IQ) CICS.INITIATION.QUEUE1 F1=Help F12=Cancel </div>

F1=Help F3=Exit

図 4. 接続の開始

CICS コマンド行からの CICS-MQ 接続の開始

CICS-MQ 接続は、コマンド行から **CKQC START** コマンドを発行することにより、開始できます。

始める前に

デフォルトでは、CICS は、キーワードとパラメーターの両方について、小文字の入力を大文字入力に変換します。したがって、デフォルトでは以下のコマンドは同等です。

CKQC START Y CSQ1	CICS01.INITQ
ckqc start y csq1	cics01.initq

小文字の IBM MQ キュー名を使用する場合は、以下のようにします。

1. アダプター制御機能を開始する端末の TYPETERM 定義に、UCTRAN(TRANID) を指定します。
2. すべての「CKxx」トランザクションで使用されるトランザクション・プロファイルに、UCTRAN(NO) を指定します。

これ以後、アダプターはキュー名を除くすべての小文字の引数を大文字の引数に変換します。

手順

- キュー・マネージャー名とデフォルト開始キュー名についてデフォルト接続値を使用して接続を開始するには、パラメーターを指定せずに、コマンド **CKQC START** を発行します。
トランザクションは、CICS 領域についてインストールされた MQCONN リソース定義および MQMONITOR リソース定義からのデフォルトの接続値を使用します。
- 明示的に定義する接続値を使用して接続を開始するには、以下のコマンドを入力します。

```
CKQC START Y|N <subsystem ID> <filler> <initiation queue name>
```

ここで、

Y|N

次のいずれかを指定します。

Y

デフォルトの接続値を使用します。つまり、ブランクの引数がある場合、インストールされた MQCONN リソース定義および MQMONITOR リソース定義から取られたデフォルト値に置換します。

N

デフォルト接続値を使用しません。

<subsystem ID>

CICS の接続先の IBM MQ キュー・マネージャーの名前。キュー・マネージャー名は、単一のキュー・マネージャーであるか、キュー共用グループであるかのいずれかです。

<filler>

互換性のために、廃止された 3 桁のトレース番号に取って代わる 3 文字の充てん文字。パラメーターは定位置パラメーターであるため、充てん文字が必要です。

<initiation queue name>

接続用のデフォルトの開始キューの名前。

これらのパラメーターは定位置パラメーターであり、デフォルトをオーバーライドしたい場合には、各フィールドの最大長まで入力する必要があります。

接続値をこのように指定すると、インストールされた MQCONN リソース定義の INITQNAME 値およびインストールされた MQMONITOR リソース定義の QNAME 値は、指定する設定に置換されます。元の設定に復帰する場合は、リソース定義を再インストールします。

CICS アプリケーション・プログラムから EXEC CICS SET MQCONN コマンドを使用した CICS-MQ 接続の開始

CICS アプリケーション・プログラムで **EXEC CICS SET MQCONN** コマンドを発行することにより、IBM MQ への接続を開始することができます。

このタスクについて

SET MQCONN CONNECTED コマンドは、属性 AUTOSTART(YES) および STATUS(ENABLED) を持つ、インストールされているすべての MQMONITOR リソースを開始します。これには、MQCONN 属性 INITQNAME が有効なデータを含んでいる時に動的に作成される MQMONITOR リソース DFHMQINI も含まれます。

手順

- デフォルト接続値を使用して CICS アプリケーション・プログラムから接続を開始するには、以下のコマンドを発行します。

```
EXEC CICS SET MQCONN CONNECTED
```

このコマンドは、キュー・マネージャー名にデフォルト接続値を使用して接続を開始します。この値 (単一のキュー・マネージャーまたはキュー共用グループの名前) は、CICS 領域に対する、インストールされている MQCONN リソース定義から取られます。

- CICS アプリケーション・プログラムから接続を開始する時に独自の接続値を指定するには、以下の例のようなコマンドを発行します。

```
EXEC CICS SET MQCONN CONNECTED
              MQNAME(qqqq)
              RESYNC
```

ここで、*qqqq* は、IBM MQ キュー・マネージャーまたはキュー共用グループの、1 文字から 4 文字の名前です。

RESYNC または NORESYNC を指定して、接続が失われた場合の再同期動作を選択します。再同期について詳しくは、[自動再接続と再同期](#)を参照してください。

タスクの結果

要求されたキュー・マネージャーがアクティブの場合は、CICS と IBM MQ が接続された時に制御が返されます。要求先のキュー・マネージャーがアクティブになっていなければ、CICS は、RESP2=8 の NORMAL 応答 (CICS-MQ アダプターが接続中の状態になっていることを示す応答) を返し、要求先のキュー・マネージャーがアクティブになった時点ですぐに IBM MQ に接続します。

次のタスク

MQNAME に独自の値を指定すると、インストールされている MQCONN リソース定義の MQNAME 属性に指定されたキュー・マネージャー名またはキュー共用グループは、このコマンドに指定された名前に置き換えられます。元のキュー・マネージャーまたはキュー共用グループに戻すには、MQNAME を再度設定してください。

CICS アプリケーション・プログラムから DFHMQQCN をリンクすることによる CICS-MQ 接続の開始

CICS アプリケーション・プログラムから CICS-MQ アダプター接続プログラム DFHMQQCN (またはこのプログラムの以前の名前である CSQCQCON) にリンクすることにより IBM MQ への接続を開始できます。接続を開始する方法は、互換性のために維持されていますが、CICS アプリケーション・プログラムからの接続を開始するには、**EXEC CICS SET MQCONN** コマンドの使用が推奨されます。

このタスクについて

ご使用の CICS アプリケーション・プログラムは、C 言語、COBOL 言語、PL/I 言語、またはアセンブラー言語で作成することができます。

手順

1. ご使用のプログラムは、使用する接続値を指定するパラメーター・リストを渡す必要があります。パラメーター・リストは以下のとおりです。

CKQC

4 文字のトランザクション ID で、「CKQC」である必要があります。

DISPMODE

1 バイトのフィールドで、1 つのブランクを含む必要があります。

CONNREQ

10 文字のフィールドで、「START」を含む必要があります。

DELIM1

1 バイトの区切り文字フィールドで、1 つのブランクを含む必要があります。

MQDEF

この接続が CICS 領域についてインストールされた MQCONN リソース定義および MQMONITOR リソース定義からのデフォルトの接続値を使用するかどうかを指定する 1 文字のフィールド。CICS TS 4.1 より前では、このフィールドは INITP と呼ばれていました。有効な値は、以下のとおりです。

Y

CICS 領域についてインストールされた MQCONN リソース定義および MQMONITOR リソース定義からのキュー・マネージャー名およびデフォルトの開始キューの値を使用します。

N

デフォルト値を使用しません。「N」を指定する場合、キュー・マネージャー名を CONNSSN フィールドに提供する必要があり、オプションでデフォルトの開始キュー名を CONNIQ フィールドに指定することができます。これらの値を提供すると、インストールされた MQCONN リソース定義の INITQNAME 値およびインストールされた MQMONITOR リソース定義の QNAME 値は、指定する設定に置換されます。元の設定に復帰するには、リソース定義を再インストールします。

..

'Y'と同じです。

DELIM2

1 バイトの区切り文字フィールドで、1つの空白を含む必要があります。

CONNSSN

4 文字のフィールドで、ターゲット IBM MQ キュー・マネージャーの z/OS サブシステム 名を指定するために使用されます。単一キュー・マネージャーまたはキュー共用グループのいずれかを指定できます。

DELIM3

5 バイトの区切り文字フィールドで、空白を含む必要があります。以前のリリースではトレース・ポイント・オーバーライドがサポートされていました。これは今では無視されます。

CONNIQ

48 文字からなるフィールドで、デフォルトの開始キューの名前を指定します。

2. ご使用のプログラムは、アダプター接続プログラム DFHMQQCN にリンクするのに以下のようなコマンドを発行する必要があります。

```
EXEC CICS LINK PROGRAM('DFHMQQCN')  
          COMMAREA(CONNPL) LENGTH(length of CONNPL)
```

この例では、パラメーター・リストの名前は CONNPL です。

タスクの結果

DFHMQQCN からの出力メッセージは、システム・コンソールに表示されます。

CICS CEMT トランザクションを介した CICS-MQ 接続の開始

CICS CEMT トランザクションで **SET MQCONN** コマンドを発行することにより、IBM MQ への接続を開始することができます。

このタスクについて

SET MQCONN CONNECTED コマンドは、属性 AUTOSTART(YES) および STATUS(ENABLED) を持つ、インストールされているすべての MQMONITOR リソースを開始します。これには、MQCONN 属性 INITQNAME が有効なデータを含んでいる時に動的に作成される MQMONITOR リソース DFHMQINI も含まれます。

CEMT トランザクションの開始方法と使用方法の詳細については、[CEMT - マスター端末](#)を参照してください。

手順

1. CICS コマンド行に、コマンド CEMT SET MQCONN を入力します。
接続状況とデフォルト接続値が表示されます。これらの値は、CICS 領域の、インストールされている MQCONN リソース定義から取られます。
2. オプション: 接続の IBM MQ キュー・マネージャーまたはキュー共用グループの名前を変更する場合は、「Mqname」フィールドの値を別の名前の上書きします。
この値を変更すると、インストールされている MQCONN リソース定義の MQNAME 属性に指定されたキュー・マネージャー名またはキュー共用グループ名が、このコマンドに指定した名前に置き換えられます。元のキュー・マネージャーまたはキュー共用グループに戻すには、値を再び変更します。
3. オプション: 再同期の動作を変更したい場合は、「Resyncmember」フィールドを、「Resync」、「Noresync」、または「Groupresync」のいずれかで上書きしてください。
再同期について詳しくは、[自動再接続と再同期](#)を参照してください。
4. 接続を開始するには、値 Notconnected を値 Connected で上書きし、Enter キーを押します。
CICS は、指定された新しい設定を使用して接続を開始します。

CICSplex SM Web ユーザー・インターフェースを介した CICS-MQ 接続の開始

IBM MQ への接続は、CICSplex SM Web ユーザー・インターフェース (WUI) の「**WebSphere MQ 接続**」ビューから開始できます。

このタスクについて

CICS Explorer では、MQ 接続 (MQCON) ビューが、WUI 内のこのビューと同等の機能を提供します。

手順

1. CICSplex SM WUI メインメニューから、「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ 接続**」を選択します。
このビューセット内の情報の完全なリストおよび説明については、[WebSphere MQ 接続 - MQCON](#) を参照してください。
2. 接続を開始する CICS システムの名前をクリックします。
CICSplex SM に、「**WebSphere MQ 接続 - 一般情報**」ビューが表示されます。
このビューには、接続の現在の状況、およびインストールされている、CICS システムの MQCONN リソース定義から取られたデフォルト接続値が表示されます。
3. オプション: 接続の IBM MQ キュー・マネージャーまたはキュー共用グループの名前を変更したい場合は、「**MQ キュー・マネージャーまたは QSG の名前**」フィールドに別の名前を入力します。
この値を変更すると、インストールされている MQCONN リソース定義の MQNAME 属性に指定されたキュー・マネージャー名またはキュー共用グループ名が、このコマンドに指定した名前に置き換えられます。元のキュー・マネージャーまたはキュー共用グループに戻すには、値を再び変更します。
4. オプション: 再同期の動作を変更したい場合は、「**再同期メンバー**」フィールドから代替りのオプションを選択します。
再同期について詳しくは、[自動再接続と再同期](#)を参照してください。
5. 接続の設定に変更を行った場合は、「**変更の適用**」ボタンをクリックします。
6. 接続を開始するには、「**CICS-MQ 接続の開始**」ボタンをクリックします。
CICS システムが、指定された設定を使用して接続を開始します。

CICS-MQ 接続の停止

CICS-MQ 接続はいくつかの方法で停止できます。接続を停止するときに、静止シャットダウンまたは強制シャットダウンのいずれを実行するか選択できます。

このタスクについて

静止シャットダウンでは、インターフェースがクローズする前に、各 CICS トランザクションを終了させます。強制シャットダウンでは、キュー・マネージャーに接続された CICS トランザクションが異常終了します。CICS-MQ アダプターが静止シャットダウンおよび強制シャットダウンに対して取るアクションの説明については、[CICS-WebSphere MQ 接続のシャットダウン時の処理](#)を参照してください。

接続を停止してから、CICS 領域の変更された MQCONN リソース定義をインストールする必要があります。

強制シャットダウンの後で接続を再開したときに、未確定の作業単位が生じるおそれがあります。未確定の作業単位については、[自動再接続と再同期](#)を参照してください。未確定の作業単位を、必要に応じて手動で解決する手順については、[CICS-WebSphere MQ アダプターの再始動時の処理](#)を参照してください。

手順

CICS-MQ 接続を停止するには、以下のいずれかの方法を使用できます。

- [CICS-MQ アダプター制御パネルの使用](#)

CICS-MQ アダプター制御トランザクション CKQC を実行し、次に CICS-MQ アダプター制御パネルを使用して、CICS-MQ 接続を停止することができます。

- [CICS コマンド行からの CKQC STOP の発行](#)

コマンド行から **CKQC STOP** コマンドを発行することにより CICS-MQ 接続を停止できます。

- **EXEC CICS SET MQCONN**を発行する CICS アプリケーション・プログラムの使用

CICS アプリケーション・プログラムから **EXEC CICS SET MQCONN** コマンドを発行することにより、IBM MQ への接続を停止することができます。

- リンク先を DFHMQDSC に指定した CICS アプリケーション・プログラムの使用

CICS アプリケーション・プログラムから CICS-MQ アダプター・プログラム DFHMQDSC (またはこのプログラムの以前の名前である CSQCDSC) にリンクすることにより、IBM MQ への接続を停止できます。接続を停止するこの方法は、互換性のために維持されていますが、CICS アプリケーション・プログラムからの接続を停止するには、**EXEC CICS SET MQCONN** コマンドの使用が推奨されます。

- CICS CEMT トランザクションの使用

CICS CEMT トランザクションで **SET MQCONN** コマンドを発行することにより、IBM MQ への接続を停止することができます。

- CICSplex SM Web ユーザー・インターフェースの使用

IBM MQ への接続は、CICSplex SM Web ユーザー・インターフェース (WUI) の「**WebSphere MQ 接続**」ビューから停止できます。

- CICS Explorer の使用

CICS-MQ 接続は、CICS Explorer から停止できます。CICS Explorer は、CICSplex SM Web ユーザー・インターフェースおよび CEMT トランザクションと機能的に同等のものを提供します。

タスクの結果

MQ モニターを使用している場合、MQ 接続が停止されると、MQ モニターが自動的に停止されます。

CICS-MQ アダプター制御パネルからの CICS-MQ 接続の停止

CICS-MQ アダプター制御トランザクション CKQC を実行し、次に CICS-MQ アダプター制御パネルを使用して、CICS-MQ 接続を停止することができます。

手順

1. CICS-MQ アダプター制御初期パネルで、メニュー・バーから「**Connection**」を選択します。
2. メニューから「**Stop**」アクションを選択します。
3. 「**Stop Connection**」2 次パラメーター・ウィンドウを使用して、必要なシャットダウンのタイプ (「静止」または「強制」) を選択します。

タスクの結果

接続の停止に関連するメッセージがシステム・コンソールに表示されます。

例

F1=Help F3=Exit

図 5. CKQC 初期パネルからの接続の停止

CICS コマンド行からの CICS-MQ 接続の停止

コマンド行から **CKQC STOP** コマンドを発行することにより CICS-MQ 接続を停止できます。

手順

- ・ 静止シャットダウンを開始するには、コマンド **CKQC STOP** を発行します。
接続は、最後のタスクがその作業を完了した後でのみシャットダウンします。
- ・ 強制シャットダウンを開始するには、コマンド **CKQC STOP FORCE** を発行します。
接続は、未了タスクの状態とは無関係に、即時にシャットダウンします。

CICS アプリケーション・プログラムから EXEC CICS SET MQCONN を使用した CICS-MQ 接続の停止

CICS アプリケーション・プログラムから **EXEC CICS SET MQCONN** コマンドを発行することにより、IBM MQ への接続を停止することができます。

手順

- 接続が停止するまで制御がアプリケーションに返されない静止シャットダウンを実行するには、以下のコマンドを発行します。

EXEC CICS SET MQCONN NOTCONNECTED

BUSY(WAIT) オプションはデフォルトであり、要求を同期します。

- 接続が停止する前に制御がアプリケーションに返される静止シャットダウン (非同期要求) を実行するには、以下のコマンドを発行します。

EXEC CICS SET MQCONN NOTCONNECTED NOWAIT

- 強制シャットダウンを実行するには、以下のコマンドを発行します。

EXEC CICS SET MOCONN NOTCONNECTED FORCE

強制シャットダウンでは、接続が停止するまで制御はアプリケーションに返されません。

CICS アプリケーション・プログラムから DFHMQDSC にリンクすることによる CICS-MQ 接続の停止

CICS アプリケーション・プログラムから CICS-MQ アダプター・プログラム DFHMQDSC (またはこのプログラムの以前の名前である CSQCDSC) にリンクすることにより、IBM MQ への接続を停止できます。接続を停止する方法は、互換性のために維持されていますが、CICS アプリケーション・プログラムからの接続を停止するには、**EXEC CICS SET MQCONN** コマンドの使用が推奨されます。

このタスクについて

EXEC CICS LINK コマンドを発行して DFHMQDSC にリンクする際、このプログラムには端末関連タスクが必要です。

手順

- 静止シャットダウンで CICS アプリケーション・プログラムからの接続を停止するには、次のコマンドを発行します。

```
EXEC CICS LINK PROGRAM('DFHMQDSC ')
          INPUTMSG('CKQC STOP')
```

QUIESCE パラメーターがデフォルトです。

- 強制シャットダウンで CICS アプリケーション・プログラムからの接続を停止するには、次のコマンドを発行します。

```
EXEC CICS LINK PROGRAM('DFHMQDSC ')
          INPUTMSG('CKQC STOP FORCE')
```

タスクの結果

DFHMQDSC からの出力メッセージは、システム・コンソールに表示されます。

CICS CEMT トランザクションを介した CICS-MQ 接続の停止

CICS CEMT トランザクションで **SET MQCONN** コマンドを発行することにより、IBM MQ への接続を停止することができます。

このタスクについて

CEMT トランザクションの開始方法と使用方法の詳細については、[CEMT - マスター端末](#)を参照してください。

手順

1. CICS コマンド行に、コマンド CEMT SET MQCONN を入力します。
接続の現在の状況、およびインストールされている、CICS 領域の MQCONN リソース定義から取られたデフォルト接続値のリストが表示されます。
2. 接続を停止するには、値 Connected を上書きして、以下のようにシャットダウン・タイプを選択します。
 - a) 静止シャットダウンを実行する場合は、Notconnected と入力します。
 - b) 強制シャットダウンを実行する場合は、Forcenotcon と入力します。
3. Enter キーを押します。
CICS が接続を停止します。

CICSplex SM Web ユーザー・インターフェースを介した CICS-MQ 接続の停止

IBM MQ への接続は、CICSplex SM Web ユーザー・インターフェース (WUI) の「**WebSphere MQ 接続**」ビューから停止できます。

このタスクについて

CICS Explorer では、MQ 接続 (MQCON) ビューが、WUI 内のこのビューと同等の機能を提供します。

手順

1. CICSplex SM WUI メインメニューから、「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ 接続**」を選択します。
このビューセット内の情報の完全なリストおよび説明については、[WebSphere MQ 接続 - MQCON](#) を参照してください。
2. 接続を停止する CICS システムの名前をクリックします。
CICSplex SM に、「**WebSphere MQ 接続 - 一般情報**」ビューが表示されます。
3. 接続を停止するには、「**CICS-MQ 接続の停止**」ボタンをクリックします。
CICSplex SM に、「**CICS-MQ 接続の停止**」2 次パネルが表示されます。
4. 「**ビジー値**」ボックスを使用して、以下のように切断のタイプを選択します。
 - 接続が停止されるまで CICSplex SM が一時停止する静止シャットダウン (同期要求) を実行するには、「**WAIT**」を選択します。
 - CICSplex SM がシャットダウン要求を発行し、その後、接続が停止しているかどうかに関係なく作業を続行する静止シャットダウン (非同期要求) を実行するには、「**NOWAIT**」を選択します。
 - 強制シャットダウンを実行するには、「**FORCE**」を選択します。
5. 「**切断を実行しますか? (Perform 'Disconnect?')**」で、「**はい**」ボタンをクリックします。
選択された CICS システムが接続を停止します。

CICS-MQ 接続に関する情報の表示

必要とする情報および現在作業している環境に応じて、CICS-MQ 接続に関する基本情報または詳細な統計情報を取得できます。

接続に関する情報を表示するには、CEMT または **EXEC CICS INQUIRE** コマンド、CICS-MQ アダプター制御パネル、CKQC DISPLAY コマンド、または CICSplex SM を使用できます。各方法は、環境に関して選択された一連の情報を生成します。以下の表は、各方法で使用可能な情報の項目を示しています。

表 7. CICS-MQ 接続に関する情報の表示				
情報	CEMT またはアプリケーション・プログラムで使用する INQUIRE MQCONN	CICS-MQ アダプター制御パネル	コマンド行からの、またはアプリケーションでの CKQC DISPLAY	CICSplex SM ビュー
接続の状況 (例えば、接続済み)	はい	はい	はい	はい
接続用に指定されたキュー・マネージャーまたはキュー共有グループの名前	はい	はい	はい	はい
実際に接続されているキュー・マネージャーの名前	はい	はい	はい	はい
デフォルト開始キューの名前	INQUIRE MQMONITOR を使用	はい	いいえ	はい

表 7. CICS-MQ 接続に関する情報の表示 (続き)

情報	CEMT またはアプリケーション・プログラムで使用する INQUIRE MQCONN	CICS-MQ アダプター制御パネル	コマンド行からの、またはアプリケーションでの CKQC DISPLAY	CICSplex SM ビュー
接続を使用する未了タスク数	はい	はい	はい	はい
接続を使用するトリガー・モニター (CKTI) タスク数	はい	はい	いいえ	はい
接続を使用するタスクの詳細 (トランザクション ID やタスク数など)	いいえ	はい	いいえ	いいえ
CICS-MQ API 交差出口の状況	いいえ	はい	いいえ	いいえ
再同期動作	はい	いいえ	いいえ	はい
IBM MQ リリース番号	はい	いいえ	いいえ	はい
MQCONN リソース定義のリソース・シグニチャー	はい	いいえ	いいえ	はい
インストールされた MQCONN リソース定義の名前	INQUIRE SYSTEM を使用	いいえ	いいえ	はい
接続時間または切断時間	EXTRACT STATISTICS (EXEC CICS のみ) を使用	いいえ	いいえ	はい
MQI 呼び出しの番号などの詳細統計	EXTRACT STATISTICS (EXEC CICS のみ) を使用	はい	いいえ	はい

CICS-MQ 接続状況と設定の表示

CICS-MQ 接続の状況、および接続のキュー・マネージャーまたはキュー共用グループの名前などの基本情報を知るには、CEMT または **EXEC CICS INQUIRE** コマンド、CICS-MQ アダプター制御パネル、**CKQC DISPLAY** コマンド、または CICSplex SM を使用することができます。

このタスクについて

必要な情報と現在の環境に応じて、CICS-MQ 接続に関する基本情報を取得するための最も適切な方法を選択します。それぞれの方法で取得できる情報のリストについては、[45 ページの『CICS-MQ 接続に関する情報の表示』](#)を参照してください。

手順

- CICS コマンド行から接続状況と設定を表示するには、以下のコマンドのいずれかを入力します。

- CEMT INQUIRE MQCONN**

このコマンドは、CEMT トランザクションを開始し、画面上に接続情報を表示します。同じ方法で **CEMT INQUIRE MQMONITOR** コマンドを使用して、デフォルト開始キューの名前を検出します。または、**CEMT INQUIRE SYSTEM** コマンドを使用して、インストールされている MQCONN リソース定義の名前を検出します。

- CKQC DISPLAY**

このコマンドは、接続に関する最も重要な情報のみを生成します。このコマンドを発行すると、以下に示すように、CICS は接続情報と共にメッセージ DFHMQ0453I を一時データ・キュー CKQQ に書き込みます。

```
DFHMQ0453I date time applid
Status of connection to qmgr-name is {Connecting |
Pending |Connected | Quiescing | Stopping-Force | Disconnected |
Inactive |Unknown}. number tasks are in flight.
```

- CICS アプリケーション・プログラムで接続状況および設定を取得するには、以下のいずれかの方法を選択します。
- **EXEC CICS INQUIRE MQCONN** コマンドを発行します。このコマンドは、アプリケーション・プログラムに情報を返します。同じ方法で **INQUIRE MQMONITOR** コマンドを使用して、デフォルト開始キューの名前を検出します。または、**INQUIRE SYSTEM** コマンドを使用して、インストールされている MQCONN リソース定義の名前を検出します。
- 以下の例のように、**EXEC CICS LINK** コマンドを発行して CICS-MQ アダプター・プログラム DFHMQDSL (または互換性のために受け入れられる CSQCDSPL) にリンクし、**CKQC DISPLAY** コマンドを発行します。

```
EXEC CICS LINK PROGRAM('DFHMQDSL') INPUTMSG('CKQC DISPLAY')
```

INPUTMSG の代わりに COMMAREA オプションを使用できますが、それは、プログラムが PLT 時に実行されている場合のみになります。コマンド行から **CKQC DISPLAY** コマンドを発行した時と同様に、CICS は、メッセージ DFHMQ0453I を一時データ・キュー CKQQ に書き込みます。

- CICS-MQ アダプター制御パネルで接続状況と設定を表示するには、初期パネルのメニュー・バーから「**Connection (接続)**」を選択し、次にメニューから「**Display (表示)**」アクションを選択します。ここに示す表示接続パネルには、接続情報が含まれています。ファンクション・キー F1 を押すと、このパネル内の特定のフィールドの説明が表示され、Enter キーを押すと、パネルが最新表示されます。

```
CKQCM2                                Display Connection panel

Read connection information. Then press F12 to cancel.

  CICS Applid = VICIC14  Connection Status = Connected  QMgr name = MQDD
  Mqname = MQDD          Tracing = On                  API Exit = Off
  Initiation Queue Name = VICIC14.INITIATION.QUEUE
----- STATISTICS -----
Number of in-flight tasks = 1          Total API calls = 43912
Number of running CKTI = 1
  APIs and flows analysis              Syncpoint              Recovery
-----
Run OK          43874  MQINQ          6806  Tasks          26  Indoubt        0
Futile          0     MQSET          0     Backout         0  UnResol       0
MQOPEN          6833  ----- Flows -----  Commit          10  Commit         0
MQCLOSE         6823  Calls          43952  S-Phase         10  Backout        0
MQGET           10032  SyncComp       43922  2-Phase         0
  GETWAIT        3399  SuspReqd       0
MQPUT           13399  Msg Wait       7
MQPUT1          5     Switched      43940

F1=Help  F12=Cancel  Enter=Refresh
```

- CICSplex SM で接続情報を表示するには、メインメニューから「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作**」 > 「**WebSphere MQ 接続**」を選択します。
このビュー・セット内の情報のリストと説明については、[WebSphere MQ 接続 - MQCON](#) を参照してください。
デフォルト開始キューの名前を表示するには、「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ 開始キュー**」を選択します。

CICS-MQ の接続時刻と切断時刻の表示

CICSplex SM または CICS 統計を使用して、CICS-MQ 接続の接続時刻と切断時刻を表示することができます。CKQC トランザクションでは、接続時刻と切断時刻は提供されません。

手順

1. CICS で接続時刻と切断時刻を表示するには、CICS アプリケーション・プログラムからコマンド **EXEC CICS EXTRACT STATISTICS MQCONN** を発行します。
CICS が、アプリケーションに統計を返します。
2. CICSplex SM で接続時刻と切断時刻を表示するには、メインメニューから「**CICS 操作**」>「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」>「**WebSphere MQ 接続**」を選択します。
このビュー・セット内の情報のリストと説明については、[WebSphere MQ 接続 - MQCON](#) を参照してください。

CICS-MQ 接続の統計と呼び出しタイプの表示

接続を使用して行われた各 MQI 呼び出しタイプの数、および未確定作業単位の数と状態など、CICS-MQ 接続についての詳細な統計を表示するには、CICS 統計、CICS-MQ アダプター制御パネル、または CICSplex SM を使用します。

手順

1. CICS で接続統計を表示するには、CICS アプリケーション・プログラムからコマンド **EXEC CICS EXTRACT STATISTICS MQCONN** を発行します。
CICS が、アプリケーションに統計を返します。
これらの統計のリストと説明については、[CICS-MQ 接続統計](#) を参照してください。
2. CICS-MQ アダプター制御パネルで接続統計を表示するには、初期パネルのメニュー・バーから「**Connection (接続)**」を選択し、次にメニューから「**Display (表示)**」アクションを選択します。
以下に示す「Display Connection (接続の表示)」パネルには、接続統計が含まれています。

```
CKQCM2                                Display Connection panel

Read connection information. Then press F12 to cancel.

      CICS Applid = VICIC14      Connection Status = Connected      QMgr name = MQDD
      Mqname = MQDD              Tracing = On                      API Exit = Off
      Initiation Queue Name = VICIC14.INITIATION.QUEUE

----- STATISTICS -----
Number of in-flight tasks = 1      Total API calls = 43912
Number of running CKTI = 1
      APIs and flows analysis      Syncpoint      Recovery
-----
Run OK          43874      MQINQ          6806      Tasks          26      Indoubt        0
Futile          0         MQSET          0         Backout         0      UnResol       0
MQOPEN          6833      ----- Flows -----      Commit         10      Commit         0
MQCLOSE         6823      Calls          43952      S-Phase        10      Backout        0
MQGET           10032      SyncComp       43922      2-Phase        0
  GETWAIT       3399      SuspReqd       0
MQPUT           13399      Msg Wait       7
MQPUT1          5        Switched       43940

F1=Help  F12=Cancel  Enter=Refresh
```

未了タスク、CKTI のインスタンス、および API 呼び出しについて表示される統計は、統計が最後にリセットされてからの、現行接続の合計です。APIs and flows analysis、Syncpoint、および Recovery という各見出しの下に表示される統計は、アダプターによって生成された統計です。ファンクション・キー F1 を押すと、このパネル内の特定のフィールドの説明が表示され、Enter キーを押すと、パネルが最新表示されます。

3. CICSplex SM で接続統計を表示するには、メインメニューから「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ 接続統計**」を選択します。

このビュー・セット内の情報のリストと説明については、[WebSphere MQ 接続統計 - MQCONN](#) を参照してください。

CICS-MQ 接続を使用しているタスクの表示

CICS-MQ アダプター制御パネルを使用して、トランザクション ID とタスク番号を含め、MQI 呼び出しを使用している CICS タスクに関する詳細情報を表示することができます。

このタスクについて

接続を使用しているタスクの数だけを表示したい場合は、[CICS-MQ 接続状況と設定の表示](#)に記載されているいずれかの方法を使用できます。**INQUIRE MQCONN** コマンドおよび CICS-MQ アダプターの「**Display Connection (接続の表示)**」パネルは、接続を使用しているトリガー・モニター (CKTI) タスクの数も指定します。

CICS-MQ アダプターの「**Display Task (タスクの表示)**」パネルには、以下に示すように、接続を使用している各 CICS タスクの詳細が提供されます。

- トランザクション ID (名前)
- ユーザー ID
- CICS タスク番号
- タスク状況
- スレッド状況
- このタスクによって発行された API 呼び出しの数
- このタスクに対してリソース・セキュリティ検査がアクティブになっているかどうか
- このタスクが現在、CICS-MQ API 交差出口内にあるかどうか
- このタスクによって発行された最新の API 呼び出し
- IBM MQ によって使用されるスレッド ID

50 ページの図 6 は、この情報のレイアウトを示しています。各属性の詳細については、CKQC ヘルプ・パネル (PF1 を押してアクセス) を参照してください。

手順

CICS-MQ アダプター制御パネルを使用してタスク情報を表示するには、以下の手順を実行します。

- a) CKQC を入力し、Enter キーを押して CICS-MQ アダプター制御パネルにアクセスします。
- b) メニュー・バーから「**Task (タスク)**」を選択します。
- c) 現在アクティブであるすべてのタスクに関する情報を取得するには、メニューから「**List all tasks (すべてのタスクをリスト)**」アクションを選択します。
- d) 表示する最初のタスクの開始番号を指定するには、メニューから「**List from task (次のタスクからリスト)**」アクションを選択します。

例

```

CKQCM3                                Display Task panel

Read task status information. Then press F12 to cancel.

Tasks   1 to   3 of   3

Tran   User   Task   Task   Thread   Total   Res   API   Last   Thread
Id      Id     Num   Status Status      APIs   Sec  Exit  MQ call  ID
-----
PUTQ  CICSUSER 00065  Normal In Queue      102 No   No   MQPUT1 00012420
GETQ  CICSUSER 00067  Normal Between      22 No   No   MQOPEN 00012620
CKTI  CICSUSER 00123  Normal Msg Wait       2 No   No   MQGET  00012C20

F1=Help  F7=Backward  F8=Forward  F12=Cancel  Enter=Refresh

```

図 6. CICS-MQ アダプターの「Display Task (タスクの表示)」パネル

CICS MQ モニターの開始

正しくセットアップされている場合、CICS MQ モニターは、MQ 接続が確立された時に自動的に開始できます。また、いくつかの方法で MQ モニターを手動で開始することもできます。

始める前に

- CICS 領域で使用するには、MQMONITOR リソースをインストールして使用可能にする必要があります。詳しくは、[CICS-MQ アダプターのセットアップ](#)を参照してください。
- セキュリティ検査がアクティブな場合（つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合）は、MQ モニターの状態を開始済みに設定しようとするトランザクションに関連付けられたユーザー ID が、**MONUSERID** に定義されているユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションを開始する権限を持っていることを確認してください。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPUIUSR ユーザー ID (指定されている場合) を使用してください。
- z/OS ワークロード・マネージャー (WLM) ヘルス・サービスがアクティブの場合 (WLMHEALTH を参照)、CICS 領域の z/OS WLM **HEALTH** 値がゼロから 100% に増加するたびに、停止している MQ モニターのうち AUTOSTART(YES) で定義されたすべての MQ モニターの開始が試みられます。停止している MQ モニターをいつ開始するかを制御するには、この MQMONITOR リソースに AUTOSTART(NO) を設定する必要があります。詳しくは、[MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響およびアラート・モニター \(CKAM\)](#)を参照してください。

手順

- MQ 接続が確立された時に CICS MQ モニターを自動的に開始するには、属性 AUTOSTART(YES) および STATUS(ENABLED) を指定して MQMONITOR リソースを定義する必要があります。
CICS と MQ の接続が確立された後、MQCONN リソースを CONNECTED に設定したタスクに関連付けられているユーザー ID に関連するトランザクションを開始するための十分な権限がある場合、CICS は、属性 AUTOSTART(YES) および STATUS(ENABLED) を指定してインストールされた MQ モニターを開始します。
- CICS-MQ モニターを手動で開始するには、以下のいずれかの方法を使用できます。
 - **EXEC CICS SET MQMONITOR** を発行する CICS アプリケーション・プログラムを使用する

CICS アプリケーション・プログラムで **EXEC CICS SET MQMONITOR** コマンドを発行することにより、MQ モニターを開始できます。

- **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START)** を発行する CICSplex SM アプリケーション・プログラムを使用する

CICSplex SM アプリケーション・プログラムで **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START)** コマンドを発行することにより、MQ モニターを開始できます。

- CICS CEMT トランザクションを使用する

CICS CEMT トランザクションで **SET MQMONITOR** コマンドを発行することにより、MQ モニターを開始することができます。

- CICSplex SM Web ユーザー・インターフェースを使用する

CICSplex SM Web ユーザー・インターフェース (WUI) の「**WebSphere MQ モニター**」ビューから、MQ モニターを開始することができます。

- CICS Explorer を使用する

CICS Explorer から CICS-MQ モニターを開始できます。CICS Explorer は、CICSplex SM Web ユーザー・インターフェースおよび CEMT トランザクションと機能的に同等のものを提供します。

- CKQC トランザクションと STARTCKTI コマンドを使用して、MQMONITOR リソースで定義されている開始キューに対して CKTI のインスタンスを開始しないでください。CKQC は MQMONITOR を認識せず、MQMONITOR は CKQC によって管理される CKTI トランザクションを認識しません。CKQC STARTCKTI コマンドの使用は、MQMONITOR を介して管理されない CKTI トランザクションに限定してください。MQMONITOR の場合は、前述の MONITOR SPI コマンドを使用してください。

CICS アプリケーション・プログラムから EXEC CICS SET MQMONITOR を使用した CICS MQ モニターの開始

CICS アプリケーション・プログラムで **EXEC CICS SET MQMONITOR** コマンドを発行することにより、MQ モニターを開始できます。

始める前に

- CICS 領域で使用するには、MQMONITOR リソースをインストールして使用可能にする必要があります。詳しくは、CICS-MQ アダプターのセットアップを参照してください。
- セキュリティ検査がアクティブな場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合) は、MQ モニターの状態を開始済みに設定しようとするトランザクションに関連付けられたユーザー ID が、**MONUSERID** に定義されているユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションを開始する権限を持っていることを確認してください。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPUIUSR ユーザー ID (指定されている場合) を使用してください。
- z/OS ワークロード・マネージャー (WLM) ヘルス・サービスがアクティブの場合 (WLMHEALTH を参照)、CICS 領域の z/OS WLM **HEALTH** 値がゼロから 100% に増加するたびに、停止している MQ モニターのうち AUTOSTART(YES) で定義されたすべての MQ モニターの開始が試みられます。停止している MQ モニターをいつ開始するかを制御するには、この MQMONITOR リソースに AUTOSTART(NO) を設定する必要があります。詳しくは、MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響およびアラート・モニター (CKAM)を参照してください。

手順

MQ モニターを開始するには、CICS アプリケーション・プログラムで以下のコマンドを発行します。

```
SET MQMONITOR (name)
  MONSTATUS(STARTED)
```

ここで、*name* は、開始する MQMONITOR リソース定義の名前です。

CICSplex SM アプリケーション・プログラムから EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START) を使用した CICS MQ モニターの開始

CICSplex SM アプリケーション・プログラムで EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START) コマンドを発行することにより、MQ モニターを開始できます。

始める前に

- CICS 領域で使用するには、MQMONITOR リソースをインストールして使用可能にする必要があります。詳しくは、[CICS-MQ アダプターのセットアップ](#)を参照してください。
- セキュリティ検査がアクティブな場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合) は、MQ モニターの状態を開始済みに設定しようとするトランザクションに関連付けられたユーザー ID が、**MONUSERID** に定義されているユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションを開始する権限を持っていることを確認してください。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPUIUSR ユーザー ID (指定されている場合) を使用してください。
- z/OS ワークロード・マネージャー (WLM) ヘルス・サービスがアクティブの場合 (WLMHEALTH を参照)、CICS 領域の z/OS WLM **HEALTH** 値がゼロから 100% に増加するたびに、停止している MQ モニターのうち AUTOSTART(YES) で定義されたすべての MQ モニターの開始が試みられます。停止している MQ モニターをいつ開始するかを制御するには、この MQMONITOR リソースに AUTOSTART(NO) を設定する必要があります。詳しくは、[MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響およびアラート・モニター \(CKAM\)](#)を参照してください。

手順

MQ モニターを開始するには、CICSplex SM アプリケーション・プログラムで以下のコマンドを発行します。

```
EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START)
```

ここで、MQMON は、開始する MQMONITOR リソース定義の名前です。

CICS CEMT トランザクションを介した CICS MQ モニターの開始

CICS CEMT トランザクションで **SET MQMONITOR** コマンドを発行することにより、MQ モニターを開始することができます。

始める前に

- CICS 領域で使用するには、MQMONITOR リソースをインストールして使用可能にする必要があります。詳しくは、[CICS-MQ アダプターのセットアップ](#)を参照してください。
- セキュリティ検査がアクティブな場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合) は、MQ モニターの状態を開始済みに設定しようとするトランザクションに関連付けられたユーザー ID が、**MONUSERID** に定義されているユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションを開始する権限を持っていることを確認してください。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPUIUSR ユーザー ID (指定されている場合) を使用してください。
- z/OS ワークロード・マネージャー (WLM) ヘルス・サービスがアクティブの場合 (WLMHEALTH を参照)、CICS 領域の z/OS WLM **HEALTH** 値がゼロから 100% に増加するたびに、停止している MQ モニターのうち AUTOSTART(YES) で定義されたすべての MQ モニターの開始が試みられます。停止している MQ モニターをいつ開始するかを制御するには、この MQMONITOR リソースに AUTOSTART(NO) を設定する必要があります。詳しくは、[MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響およびアラート・モニター \(CKAM\)](#)を参照してください。

このタスクについて

CEMT トランザクションの開始方法と使用方法の詳細については、[CEMT - マスター端末](#)を参照してください。

手順

1. CICS コマンド行に以下のコマンドを入力します。

```
CEMT SET MQMONITOR (name)
```

ここで、*name* は、開始する MQMONITOR リソースの名前です。

指定された MQMONITOR リソース定義の AUTOSTART、ENABLESTATUS、および MONSTATUS の各値が表示されます。

2. MQMONITOR リソース定義が使用不可になっている場合は、ENABLESTATUS フィールドの値 DISABLED を値 ENABLED で上書きして、CICS 領域でリソースを使用できるようにします。
3. MONSTATUS フィールド内の値 STOPPED を値 STARTED で上書きして、MQ モニターを開始します。
4. AUTOSTART が使用可能になっていない場合は、AUTOSTART フィールド内の値 NOAUTOSTART を値 AUTOSTART で上書きして、MQMONITOR の AUTOSTART を使用可能にできます。

CICSplex SM Web ユーザー・インターフェースを介した CICS MQ モニターの開始

CICSplex SM Web ユーザー・インターフェース (WUI) の「**WebSphere MQ モニター**」ビューから、MQ モニターを開始することができます。

始める前に

- CICS 領域で使用するには、MQMONITOR リソースをインストールして使用可能にする必要があります。詳しくは、[CICS-MQ アダプターのセットアップ](#)を参照してください。
- セキュリティ検査がアクティブな場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合) は、MQ モニターの状態を開始済みに設定しようとするトランザクションに関連付けられたユーザー ID が、**MONUSERID** に定義されているユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションを開始する権限を持っていることを確認してください。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPUIUSR ユーザー ID (指定されている場合) を使用してください。
- z/OS ワークロード・マネージャー (WLM) ヘルス・サービスがアクティブの場合 ([WLMHEALTH](#) を参照)、CICS 領域の z/OS WLM **HEALTH** 値がゼロから 100% に増加するたびに、停止している MQ モニターのうち AUTOSTART(YES) で定義されたすべての MQ モニターの開始が試みられます。停止している MQ モニターをいつ開始するかを制御するには、この MQMONITOR リソースに AUTOSTART(NO) を設定する必要があります。詳しくは、[MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響](#) および [アラート・モニター \(CKAM\)](#)を参照してください。

このタスクについて

CICS Explorer では、MQ モニター (MQMON) ビューが、WUI 内のこのビューと同等の機能を提供します。

手順

1. CICSplex SM WUI メインメニューから、「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ モニター**」を選択します。
このビューセット内の情報の完全なリストおよび説明については、[WebSphere MQ モニター - MQMON](#)を参照してください。
2. 開始したい WebSphere MQ モニターの「**レコード**」ボックスを選択します。
3. 「**開始**」ボタンをクリックします。

CICS MQ モニターの停止

MQ 接続が停止すると、MQ モニターは自動的に停止します。また、いくつかの方法で MQ モニターを手動で停止することもできます。

z/OS ワークロード・マネージャー (WLM) ヘルス・サービスがアクティブの場合 ([WLMHEALTH](#) を参照)、CICS 領域の z/OS WLM **HEALTH** 値がゼロから 100% に増加するたびに、停止している MQ モニターのうち **AUTOSTART(YES)** で定義されたすべての MQ モニターの開始が試みられます。したがって、特定の MQ モニターをこの期間停止したままにする場合は、これらの MQMONITOR リソースに **AUTOSTART(NO)** を設定する必要があります。詳しくは、[MQMONITOR に対する z/OS ワークロード・マネージャー 正常性サービスの影響およびアラート・モニター \(CKAM\)](#) を参照してください。

手順

MQ モニターを手動で停止するには、以下のいずれかの方法を使用できます。

- [EXEC CICS SET MQMONITOR](#) を発行する CICS アプリケーション・プログラムを使用する

CICS アプリケーション・プログラムで **EXEC CICS SET MQMONITOR** コマンドを発行することにより、MQ モニターを停止できます。

- [EXEC CPSM PERFORM SET OBJECT\(MQMON\) ACTION\(STOP\)](#) を発行する CICSplex SM アプリケーション・プログラムを使用する

CICSplex SM アプリケーション・プログラムで **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP)** コマンドを発行することにより、MQ モニターを停止できます。

- [CICS CEMT トランザクション](#)を使用する

CICS CEMT トランザクションで **SET MQMONITOR** コマンドを発行して MQ モニターを停止することができます。

- [CICSplex SM Web ユーザー・インターフェース](#)を使用する

MQ モニターを停止することができます。

- CICS Explorer を使用する

CICS Explorer から CICS-MQ モニターを停止することができます。CICS Explorer は、CICSplex SM Web ユーザー・インターフェースおよび CEMT トランザクションと機能的に同等のものを提供します。

CICS アプリケーション・プログラムから EXEC CICS SET MQMONITOR を使用した CICS MQ モニターの停止

CICS アプリケーション・プログラムで **EXEC CICS SET MQMONITOR** コマンドを発行することにより、MQ モニターを停止できます。

手順

MQ モニターを停止するには、CICS アプリケーション・プログラムで以下のコマンドを発行します。

```
SET MQMONITOR (name)
  MONSTATUS(STOPPED)
```

ここで、*name* は、停止する MQMONITOR リソース定義の名前です。

CICSplex SM アプリケーション・プログラムから EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP) を使用しての CICS MQ モニターの停止

CICSplex SM アプリケーション・プログラムで EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP) コマンドを発行することにより、MQ モニターを停止できます。

このタスクについて

CICS 領域で使用するには、MQMONITOR リソースをインストールして使用可能にする必要があります。詳しくは、[CICS-MQ アダプターのセットアップ](#)を参照してください。

手順

MQ モニターを停止するには、CICSplex SM アプリケーション・プログラムで以下のコマンドを発行します。

```
EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP)
```

ここで、*MQMON* は、停止する MQMONITOR リソース定義の名前です。

CICS CEMT トランザクションを介した CICS MQ モニターの停止

CICS CEMT トランザクションで SET MQMONITOR コマンドを発行して MQ モニターを停止することができます。

このタスクについて

CEMT トランザクションの開始方法と使用方法の詳細については、[CEMT - マスター端末](#)を参照してください。

手順

1. CICS コマンド行に以下のコマンドを入力します。

```
CEMT SET MQMONITOR (name)
```

ここで、*name* は、停止する MQMONITOR リソースの名前です。

指定された MQMONITOR リソース定義の AUTOSTART、ENABLESTATUS、および MONSTATUS の各値が表示されます。

2. MONSTATUS フィールドの値 STARTED を値 STOPPED で上書きして、MQ モニターを停止します。

CICSplex SM Web ユーザー・インターフェースを介した CICS MQ モニターの停止

CICSplex SM Web ユーザー・インターフェースの「**WebSphere MQ モニター**」ビューから、MQ モニターを停止することができます。

このタスクについて

CICS Explorer では、MQ モニター (MQMON) ビューが、WUI 内のこのビューと同等の機能を提供します。

手順

1. CICSplex SM WUI メインメニューから、「**CICS 操作**」 > 「**DB2、DBCTL、および WebSphere MQ 操作ビュー**」 > 「**WebSphere MQ モニター**」を選択します。
このビューセット内の情報の完全なリストおよび説明については、[WebSphere MQ モニター - MQMON](#)を参照してください。
2. 停止したい WebSphere MQ モニターの「**レコード**」ボックスを選択します。
3. 「**停止**」ボタンをクリックします。

CICS MQ モニターに関する情報の表示

CEMT INQUIRE MQMONITOR コマンドまたは **EXEC CICS INQUIRE MQMONITOR** コマンド、あるいは CICSplex SM を使用して、MQ モニターに関する情報を表示することができます。

このタスクについて

INQUIRE MQMONITOR コマンドを使用して、CICS 領域内にインストールされているすべての MQMONITOR リソースを照会できます。インストールされている MQMONITOR リソースについて以下の情報を知ることができます。

- MQMONITOR リソースが使用可能か、または使用不可か。
- MQMONITOR で自動再始動が使用可能になっているか。
- MQMONITOR が開始されているか、または停止されているか。そして、開始されている場合は、MQ キューを現在モニターしているタスクの ID。このタスクの詳細を表示するには、**CEMT INQUIRE TASK** を使用します。
- MQ キューをモニターしているタスクに渡されるデータ (MONDATA 属性)。例えば、MQ ブリッジ・モニター・トランザクション CKBR に渡されるパラメーターなど。

注:

モニター・タスクによって表示され、取得される場合、MONDATA データの前に以下の 18 バイトが付加されます。

Byte 1: < (left chevron)
バイト 2 から 9: *MQMONITOR resource name*
バイト 10 から 17: *USERID*
Byte 18: > (right chevron)

バイト 19 から 218 には、ユーザーが入力した MONDATA が入ります。

- MQ キューをモニターしているタスクによって使用されるトランザクションの名前。
- MQ モニターによってモニターされている MQ キューの名前。

CICS-MQ 接続統計のリセット

CKQC トランザクションを使用して CICS-MQ 統計だけをリセットするか、標準の CICS 統計リセット・コマンドを使用して、CICS-MQ 統計を含めてすべての統計をリセットすることができます。CICS-MQ 接続統計は、統計間隔の最後にもリセットされます。

手順

- CICS コマンド行から CICS-MQ 接続統計のみをリセットするには、コマンド **CKQC MODIFY Y** を発行します。オプション Y は統計をリセットします。
 - CICS アプリケーション・プログラムを使用して CICS-MQ 接続統計のみをリセットするには、**EXEC CICS LINK** コマンドを発行して、アダプター・リセット・プログラム DFHMQRS (または、互換性のために保持されている CSQCRST) にリンクし、オプション Y を指定して **CKQC MODIFY** コマンドを発行します。
- 以下の例は、この実行方法を示しています。

```
EXEC CICS LINK PROGRAM('DFHMQRS ')
          INPUTMSG('CKQC MODIFY Y')
```

MODIFY コマンドでは、4 つの末尾スペースに加えて、区切り文字としてさらにもう 1 つのスペースを埋め込まなければなりません (32 ページの『[CICS アプリケーション・プログラムからの CKQC コマンド](#)』を参照)。

- CICS-MQ アダプター制御パネルを使用して CICS-MQ 接続統計のみをリセットするには、以下のようになります。
 - a) CKQC を入力し、Enter キーを押して CICS-MQ アダプター制御パネルにアクセスします。

- b) メニュー・バーから「**Connection (接続)**」を選択します。
- c) メニューから「**Modify (修正)**」アクションを選択します。
- d) 「**Modification Options (変更オプション)**」2 次パラメーター・ウィンドウで「**Reset statistics (統計のリセット)**」を選択し、Enter キーを押します。
- CICS-MQ 接続統計を含めたすべての CICS 統計をリセットするには、以下に示す標準の CICS 統計リセット方法のいずれかを選択します。
 - CEMT トランザクションを使用し、RESETNOW オプションを指定して **CEMT SET STATISTICS** コマンドを発行します。
 - CICS アプリケーション・プログラムで、RESETNOW オプションを指定して **EXEC CICS SET STATISTICS** コマンドを発行するか、**EXEC CICS PERFORM STATISTICS RECORD ALL RESETNOW** コマンドを発行します。

CICS-MQ 接続を使用しているタスクのページ

CICS CEMT トランザクションを使用して、CICS-MQ アダプターを使用しているユーザー・タスクをページすることができます。

このタスクについて

アダプターで待機しているタスクは、**CEMT SET TASK FORCEPURGE** コマンドのみに応答します。**CEMT SET TASK PURGE** コマンドは無視されます。**CEMT SET TASK** コマンドの完全な構文については、[CEMT SET TASK](#) を参照してください。

手順

1. CICS-MQ アダプター制御パネルを使用して、CICS-MQ 接続を使用しているタスクの詳細を、そのタスク番号を含めて表示します。ページするタスクの番号をメモしてください。
49 ページの『[CICS-MQ 接続を使用しているタスクの表示](#)』でこの方法を説明しています。
2. 強制ページする各タスクについて、CICS コマンド行でコマンド **CEMT SET TASK(number) FORCEPURGE** を入力します。ここで、*number* はタスクのタスク番号です。

タスクの結果

CICS-MQ アダプターが **FORCEPURGE** コマンドを取り扱う方法は、タスクの待ち状態の種類によって決まります。

- タスクがメッセージの到着を待っている場合、例えば、アプリケーションが **MQGET WAIT** 呼び出しを既に実行している場合、そのタスクは即時に停止して、コード **AEXY** を出します。
- それ以外の場合、アダプターは要求が完了するのを待ってから、タスクの終了が適当かどうかを検査します。
 - タスクがクリティカルな状態にある場合、アダプターは、データおよびシステムの保全性を保持するために、アダプターにタスクを継続させ、タスクをページする試みは無視します。メッセージ **DFHMQ0415I** が表示されます。タスクがクリティカルな状態にあるのは、例えば 2 フェーズ・コミット・シーケンスのフェーズ 2 の完了が進行中であるような場合です。
 - タスクがクリティカルな状態にない場合、アダプターは終了し、コード **AEXY** が出されます。メッセージ **DFHMQ0414I** が表示されます。

CICS-MQ トリガー・モニター

CKTI は、CICS で提供されているトリガー・モニター(またはタスク・イニシエーター) トランザクションです。CKTI は、CICS 環境で使用され、関連付けられたいずれかキューのトリガー条件に合致した時にトランザクションを開始します。CICS-MQ トリガー・モニターである CKTI は、MQ トリガー・モニターによって開始されたトランザクションを追跡する機能を提供します。この機能のメリットにより、管理者は作業を視覚化および追跡することが可能になり、それによりさらなる価値が提供されます。

キュー・マネージャーにとって、トリガー・モニターは、キューを処理する他のアプリケーションと同様です。ただし、トリガー・モニターは開始キューを処理します。

トリガー・モニターは、通常、継続的に稼働しているプログラムです。トリガー・メッセージが開始キューに到着すると、トリガー・モニターはそのメッセージを取り出します。そして、メッセージ内の情報を使用して、アプリケーション・キューにあるメッセージを処理するアプリケーションを開始するためのコマンドを発行します。

トリガー・モニターは、開始するプログラムに十分な情報を渡して、そのプログラムが、適正なアプリケーション・キューに対して適正なアクションを実行できるようにする必要があります。

CKTI

各開始キューに対して CKTI の複数インスタンスを開始できます。CKTI は、**EXEC CICS START TRANSID** によって開始するプログラムにトリガー・メッセージの MQTM 構造体を渡します。開始されたプログラムは、**EXEC CICS RETRIEVE** コマンドを使用してこの情報を取得します。MQTM 構造体について詳しくは、[MQTM トリガー・メッセージ](#)を参照してください。

プログラムは、RTRANSID オプションを指定した **EXEC CICS RETRIEVE** コマンドを使用して、プログラムがどのように開始されたかを判別できます。返された値が CKTI の場合、プログラムは、CICS-MQ トリガー・モニターによって開始されています。

CKTI の使用方法の例については、IBM MQ for z/OS で提供されている信用チェック・サンプル・アプリケーションのモジュール CSQ4CVB2 のソース・コードを参照してください。このサンプルについて詳しくは、[信用小切手サンプル](#)を参照してください。

MQMONITOR を使用した CKTI の管理

MQMONITOR リソースの使用は、CKTI インスタンスの推奨される制御方法です。以下の利点があります。

- 1 つの MQ 開始キューをモニターするために CKTI の複数インスタンスをセットアップできます。
- MQ キュー・マネージャーへの接続が確立された時に CKTI を自動的に開始できます。
- MQMONITOR は、CKTI またはメッセージ・コンシューム・トランザクションが、事前設定されたユーザー ID で実行されることを許可します。



注意: CKTI トランザクションのインスタンスを管理するために CKQC トランザクションと MQMONITOR を同時に使用すると、CKQC が MQMONITOR を認識しておらず、MQMONITOR は CKQC を使用して管理される CKTI トランザクションを認識していないため、作成される MQMONITOR の統計と状態が正しくない可能性があります。

重要: CICS 提供のトリガー・モニターの代わりに、ユーザー独自の MQ トリガー・モニターを使用して CICS 環境で MQ キューを処理する場合は、ユーザー作成の MQ トリガー・モニター・プログラムを設計して実装する際に、66 ページの『[ユーザー作成の MQ トリガー・モニターと MQ メッセージ・コンシューマーの開発と使用](#)』の指示に必ず従ってください。

CKTI インスタンスの開始

CKTI インスタンス (CICS-MQ トリガー・モニターまたはタスク・イニシエーター) は、いくつかの方法で開始できます。MQMONITOR リソースの使用は、CKTI インスタンスの推奨される制御方法です。

このタスクについて

CKTI のユーザー ID

CKTI のインスタンスが MQMONITOR によって開始される場合、CKTI トランザクションに関連付けられるユーザー ID は MQMONITOR リソースの **MONUSERID** 属性から取得されます。

端末から CKQC トランザクションまたは DFHMQSSQ (または互換性のために保持されている CSQCSSQ) にリンクしているユーザー作成プログラムを実行して CKTI を開始した場合に、CKTI が使用するユーザー ID は、CKTI を開始した端末のユーザー ID と同じです。

CKTI がその他の CICS トランザクション (ユーザー作成の CICS アプリケーションなど) を開始する場合、CKTI のユーザー ID はそれらのアプリケーションに伝搬されます。例えば、CKTI がユーザー ID

CIC1 の環境で実行されている場合に、送信側 MCA トランザクションの CKSG を開始する必要があるトリガー・イベントが発生すると、CKSG トランザクションもユーザー ID CIC1 の環境で実行されます。したがって、ユーザー ID CIC1 は、要求された伝送キューへのアクセス権を持っている必要があります。

手順

以下のいずれかの方法を使用して、CKTI を開始できます。

- CICS MQ モニターの使用

MQMONITOR リソースは、正しくセットアップされている場合、MQ キュー・マネージャーへの接続が確立された時に CKTI トランザクションを自動的に開始できるようにします。CKTI を制御する MQMONITOR が自動的に開始されない場合は、いくつかの方法で MQ モニターを開始できます。

- CICS-MQ アダプター制御パネルの使用

CICS-MQ アダプター制御トランザクション CKQC を実行し、次に CICS-MQ アダプター制御パネルから CKTI トランザクションのインスタンスを開始できます。

- CICS コマンド行からの **CKQC STARTCKTI** の発行

CKQC STARTCKTI コマンドを使用して、デフォルト開始キューまたは指定する開始キューを扱う CKTI インスタンスを開始できます。

- CICS アプリケーション・プログラムの使用

CICS アプリケーション・プログラムから CKTI インスタンスを開始するには、アプリケーション・プログラムは、アダプター・タスク開始プログラム DFHMQSSQ (または互換性のために保持されている CSQCSSQ) にリンクする必要があります。CKTI を制御するために MQMONITOR を使用している場合、EXEC CICS SET MQMONITOR (name) STARTED を発行するアプリケーション・プログラムから MQMONITOR を開始する必要があります。

- 自動化製品の使用

特定のユーザー ID のもとでトリガー・モニターの開始を自動化するには、NetView などの自動化製品を使用することができます。これを使用して CICS コンソールにサインオンし、**STARTCKTI** コマンドを発行することができます。

CICS MQ モニターを使用した CKTI の開始

MQMONITOR リソースは、正しくセットアップされている場合、MQ キュー・マネージャーへの接続が確立された時に CKTI トランザクションを自動的に開始できるようにします。CKTI を制御する MQMONITOR が自動的に開始されない場合は、いくつかの方法で MQ モニターを開始できます。

始める前に

- MQMONITOR を使用して CICS 領域内の CKTI のインスタンスを制御する場合は、そのような MQMONITOR リソースを定義し、インストールする必要があります。[60 ページの表 8](#) に、CKTI を制御する MQMONITOR に指定するいくつかの重要な属性をリストします。

属性 AUTOSTART(YES) および STATUS(ENABLED) は、MQ 接続が確立された時に自動的に再開始される MQMONITOR に対して定義することに注意してください。自動再開始を使用不可にし、MQ モニターを手動で開始したい場合は、AUTOSTART(NO) 属性を指定するか、AUTOSTART 属性を指定せずに MQMONITOR リソースを定義してください。

表 8. 開始キューをモニターするための MQMONITOR リソース属性			
属性	必須/オプション	デフォルト	説明
AUTOSTART	オプション	YES	<p>この属性は、MQ モニターの自動再始動を以下のように制御します。</p> <p>AUTOSTART(YES) MQ キュー・マネージャーへの接続が確立された時に、CKTI を自動的に再開始します。</p> <p>AUTOSTART(NO) MQ モニターは自動的に開始されません。MQ キュー・マネージャーへの接続が確立された後、MQ モニターを手動で始動する必要があります。</p>
MONUSERID	必須	-	<p>CKTI に関連付けられるユーザー ID を指定します。</p> <p>この属性は、セキュリティー検査がアクティブである (つまり、SEC システム初期設定パラメーターが YES に設定されている) 場合にのみ有効です。</p>
QNAME	オプション	省略されている場合、デフォルトは &APPLID..INITIATION.QUEUE です。ここで、&APPLID. は、CICS 領域のアプリケーション ID です。	モニターする MQ 開始キューの名前を指定します。
STATUS	オプション	ENABLED	この属性により、領域でリソースを使用できるようにします。
TRANSACTION	オプション	CKTI	<p>CICS 提供のトリガー・モニター (またはタスク・イニシエーター) トランザクションの、4 文字の ID を指定します。</p> <p>注: 独自の MQ トリガー・モニターを使用している場合は、このトランザクションの名前を指定します。CICS 環境での独自の MQ トリガー・モニターの使用に関する考慮事項については、57 ページの『CICS-MQ トリガー・モニター』を参照してください。</p>

- セキュリティー検査がアクティブな場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合) は、MQ モニターの状態を開始済みに設定しようとするトランザクションに関連付けられたユーザー ID が、**MONUSERID** に定義されているユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションを開始する権限を持っていることを確認してください。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPUIUSR ユーザー ID (指定されている場合) を使用してください。

手順

- CKTI を制御する MQ モニターを自動的に開始するには、MQMONITOR リソースに属性 AUTOSTART(YES) および STATUS(ENABLED) が指定されていることを確認してください。

CICS と MQ の接続が確立された後、MQCONN リソースを CONNECTED に設定したタスクに関連付けられているユーザー ID に関連するトランザクションを開始するための十分な権限がある場合、CICS は、属性 AUTOSTART(YES) および STATUS(ENABLED) を指定してインストールされた MQ モニターを開始します。そして、CKTI のインスタンスが実行されます。

- CKTI を制御する MQ モニターを手動で開始するには、50 ページの『CICS MQ モニターの開始』で説明されているいずれかの方法を使用できます。

CICS-MQ アダプター制御パネルからの CKTI の開始

CICS-MQ アダプター制御トランザクション CKQC を実行し、次に CICS-MQ アダプター制御パネルから CKTI トランザクションのインスタンスを開始できます。

手順

1. CICS-MQ アダプター制御初期パネルで、メニュー・バーから「**CKTI**」を選択します。
2. メニューから「**開始 (Start)**」アクションを選択します。
3. 「**タスク・イニシエーターの開始**」2 次ウィンドウから、「**開始キュー名**」フィールドを使用して、この CKTI インスタンスでサービスされる開始キューの名前を指定します。
このフィールドをブランクのままにした場合、デフォルトの開始キュー (定義されている場合) が使用されます。

例

Connection	CKTI	Task
CKQCM0	Select an action.	-- Initial panel
Select menu bar it	1 1. Start... 2. Stop... 3. Display	press Enter.
	F1	Start Task Initiator
	Type Initiation Queue Name. Then press Enter.	
	Initiation Queue Name (IQ) CICS01.INITIATION.QUEUE2	
	F1=Help F12=Cancel	

F1=Help F3=Exit

図 7. CKTI インスタンスの開始

CICS コマンド行からの CKTI の開始

CKQC STARTCKTI コマンドを使用して、デフォルト開始キューまたは指定する開始キューを扱う CKTI インスタンスを開始できます。

このタスクについて

開始キューを指定せずに **CKQC STARTCKTI** コマンドを発行すると、このコマンドは、CICS 領域のデフォルトの開始キュー `regionAPPLID.initiation.queue` を指すものとして自動的に解釈されます。

手順

- デフォルトの開始キュー (定義されている場合) を扱う CKTI インスタンスを開始するには、次のコマンドを使用します。

```
CKQC STARTCKTI
```


- 指定された開始キュー CICS01.INITIATION.QUEUE2 を扱う CKTI インスタンスを開始するには、次のコマンドを使用します。

```
CKQC STARTCKTI CICS01.INITIATION.QUEUE2
```

CICS アプリケーション・プログラムからの CKTI の開始

CICS アプリケーション・プログラムから CKTI インスタンスを開始するには、アプリケーション・プログラムは、アダプター・タスク開始プログラム DFHMQSSQ (または互換性のために保持されている CSQCSSQ) にリンクする必要があります。CKTI を制御するために MQMONITOR を使用している場合、EXEC CICS SET MQMONITOR (*name*) STARTED を発行するアプリケーション・プログラムから MQMONITOR を開始する必要があります。

このタスクについて

DFHMQSSQ にリンクするアプリケーション・プログラムの使用

DFHMQSSQ に **EXEC CICS LINK** を行う場合、このプログラムには端末関連タスクが必要です。

STARTCKTI コマンドは、スペースを埋め込んで 10 文字にする必要があります (33 ページの『アプリケーション・プログラムでのコマンド構文』を参照)。

EXEC CICS SET MQMONITOR (*name*) STARTED を発行するアプリケーション・プログラムの使用

以下の条件を満たす場合、アプリケーション・プログラムは CKTI を制御する MQMONITOR を開始します。

- 現在の tranid が TRANID 属性の値と同じでない。
- MQMONITOR がまだ STARTED 状態でない。
- 現在のトランザクションに関連したユーザー ID は、MQMONITOR に関連したトランザクションの状態を「開始済み」に設定する権限を持っています。

手順

- DFHMQSSQ にリンクして、デフォルトの開始キューを使用する CKTI を開始するには、次のようなコマンドを発行します。

```
EXEC CICS LINK PROGRAM('DFHMQSSQ ')
      INPUTMSG('CKQC STARTCKTI ')
```

- DFHMQSSQ にリンクして、開始キュー CICS01.INITIATION.QUEUE2 を使用する CKTI を開始するには、次のようなコマンドを発行します。

```
EXEC CICS LINK PROGRAM('DFHMQSSQ ')
      INPUTMSG('CKQC STARTCKTI CICS01.INITIATION.QUEUE2')
```

- CKTI を制御する MQMONITOR を開始するには、アプリケーション・プログラムで次のコマンドを発行します。

```
SET MQMONITOR (name)
  MONSTATUS(STARTED)
```

ここで、*name* は CKTI を制御する MQMONITOR の名前です。

タスクの結果

DFHMQSSQ からの出力メッセージは、システム・コンソールに表示されます。

自動化製品を通じて CKTI を自動的に開始

特定のユーザー ID のもとでトリガー・モニターの開始を自動化するには、NetView などの自動化製品を使用することができます。これを使用して CICS コンソールにサインオンし、**STARTCKTI** コマンドを発行することができます。

事前設定のセキュリティー順次端末を使用することもできます。この端末は、**CKQC STARTCKTI** コマンドが入っている順次端末入力を指定して、CRLP 端末をエミュレートするように定義されています。

ただし、アダプター・アラート・モニターが CICS を IBM MQ などに再接続すると、キュー・マネージャーが再始動した後、初期 IBM MQ 接続の時点で指定された CKTI のみが再始動されます。追加の CKTI (ある場合) の開始はユーザー自身で自動化する必要があります。

CKTI インスタンスの停止

CKTI インスタンスはいくつかの方法で停止できます。MQMONITOR リソースの使用は、CKTI インスタンスの推奨される制御方法です。

このタスクについて

CKTI を制御するのに MQ モニターを使用している場合、MQ 接続が停止されると、MQ モニターが自動的に停止されます。

CKTI を制御するのに領域内で MQ モニターを使用している場合、複数の CKTI インスタンスが実行中だと、CKQC を使用して CKTI インスタンスを停止すると予測不能な結果を引き起こすおそれがあります。

手順

- 以下の方法のいずれかを使用して、CKTI インスタンスを停止できます。
 - CKTI を制御する CICS MQ モニターの停止
CKTI インスタンスを制御するのに MQ モニターを使用している場合、MQ モニターはいくつかの方法で停止できます。
 - CICS-MQ アダプター制御パネルの使用
CICS-MQ アダプター制御トランザクション CKQC を実行し、次に CICS-MQ アダプター制御パネルから CKTI トランザクションのインスタンスを停止できます。
 - 端末からの CKQC STOPCKTI コマンドの発行
端末から **CKQC STOPCKTI** コマンドを発行して CKTI インスタンスを停止できます。
 - アプリケーション・プログラムの使用
アダプター・タスク開始プログラム DFHMQSSQ (または互換性のために保持された CSQCSSQ) にリンクすることにより、CKTI インスタンスを停止できます。

CICS-MQ アダプター制御パネルからの CKTI インスタンスの停止

CICS-MQ アダプター制御トランザクション CKQC を実行し、次に CICS-MQ アダプター制御パネルから CKTI トランザクションのインスタンスを停止できます。

手順

1. CICS-MQ アダプター制御初期パネルで、メニュー・バーから「**CKTI**」を選択します。
2. メニューから「**停止 (Stop)**」アクションを選択します。
3. 「**タスク・イニシエーターの終了**」2 次ウィンドウを使用して、この CKTI インスタンスでサービスされる開始キューの名前を指定します。
名前をブランクのままにした場合、デフォルトの開始キュー (定義されている場合) が使用されます。

例

Connection	CKTI	Task
CKQCM0	Select an action.	-- Initial panel
Select menu bar it	2 1. Start... 2. Stop... 3. Display	press Enter.
	F1	Stop Task Initiator
		Type Initiation Queue Name. Then press Enter.
		Initiation Queue Name (IQ) CICS01.INITIATION.QUEUE2
		F1=Help F12=Cancel

F1=Help F3=Exit

図 8. タスク・イニシエーター CKTI インスタンスの停止

端末からの CKTI インスタンスの停止

端末から **CKQC STOPCKTI** コマンドを発行して CKTI インスタンスを停止できます。

このタスクについて

開始キューを指定せずに **CKQC STOPCKTI** コマンドを発行すると、このコマンドは、CICS 領域のデフォルトの開始キュー `regionAPPLID.initiation.queue` を指すものとして自動的に解釈されます。

手順

- デフォルトの開始キューを扱う CKTI インスタンスを停止するには、次のコマンドを使用します。

```
CKQC STOPCKTI
```

- 指定された開始キューを扱う CKTI インスタンスを停止するには、次のコマンドを使用します。

```
CKQC STOPCKTI queue_name
```

ここで、`queue_name` は開始キューの名前です。

注: MQMONITOR リソースを使用している場合、このコマンドは指定された MQ キューをモニターしているすべての MQMONITOR も停止することに注意してください。

例えば、CICS01.INITIATION.QUEUE2 を扱う CKTI インスタンスを停止するには、次のコマンドを発行します。

```
CKQC STOPCKTI CICS01.INITIATION.QUEUE2
```

アプリケーション・プログラムからの CKTI インスタンスの停止

アダプター・タスク開始プログラム DFHMQSSQ (または互換性のために保持された CSQCSSQ) にリンクすることにより、CKTI インスタンスを停止できます。

このタスクについて

以下の例は、CICS プログラムから CKTI インスタンスを停止するための代替 LINK コマンドを示しています。STOPCKTI コマンドは、スペースを埋め込んで 10 文字にする必要があります (33 ページの『アプリケーション・プログラムでのコマンド構文』を参照)。

手順

- このコマンドは、デフォルトの開始キューを扱う CKTI を停止します。

```
EXEC CICS LINK PROGRAM('DFHMQSSQ ')
        INPUTMSG('CKQC STOPCKTI ')
```

- このコマンドは、指定された開始キューを扱う CKTI を停止します。

```
EXEC CICS LINK PROGRAM('DFHMQSSQ ')
        INPUTMSG('CKQC STOPCKTI CICS01.INITIATION.QUEUE2')
```

現在の CKTI インスタンスの表示

CICS-MQ アダプター制御パネルを使用して、現在の CKTI インスタンスの詳細を表示できます。同等の機能は CICS コマンド行からも、CICS アプリケーション・プログラムからも使用できません。

手順

- CICS-MQ アダプター制御初期パネルで、メニュー・バーから「**CKTI**」を選択します。
- プルダウン・メニューから、「**Display (表示)**」アクションを選択します。
65 ページの図 9 は、各 CKTI インスタンスについて提供される詳細を示しています。
 - CICS タスク番号
 - タスク状況
 - スレッド状況
 - 実行した API 呼び出しの数
 - 最新実行の API 呼び出し
 - 扱っている開始キューの名前
- パネルの各フィールドについてヘルプ情報を表示するためには、ファンクション・キー F1 を押してください。

例

```
CKQCM4                      Display CKTI panel
Read CKTI status information. Then press F12 to cancel.
CKTI  1 to  1 of  1

Task Num   Task Status   Thread Status   Num of APIs   Last API
-----
0000123    Normal             Msg Wait                2      MQGET
Initiation Queue Name: CICS01.INITIATION.QUEUE1
```

F1=Help F7=Backward F8=Forward F12=Cancel Enter=Refresh

図 9. CKQC の CKTI 表示パネル

ユーザー作成の MQ トリガー・モニターと MQ メッセージ・コンシューマーの開発と使用

CICS で提供されているトリガー・モニター CKTI の代わりに、独自の MQ トリガー・モニターを作成し、それらを使用して CICS 環境で MQ キューを処理することができます。独自の MQ メッセージ・コンシューマーを作成し、アプリケーション入力キューから直接メッセージを取得して必須のロジックを実行することもできます。このトピックでは、ユーザー作成の MQ トリガー・モニターおよび MQ メッセージ・コンシューマーの開発と使用に関する重要な手順について説明します。

ユーザー作成の MQ モニター・プログラムまたは MQ メッセージ・コンシューマー・プログラムの責任

MQMONITOR を使用して MQ キューを処理している場合、MQMONITOR が開始されると、**EXEC CICS START** コマンドは、FROM データとして **<MQMONITOR_resource_nameUSERID>** が前に付加された **MONDATA** を使用してトランザクションを開始します。したがって、ユーザー作成のプログラムは、MQ の呼び出しを発行する前にこのデータを取得し、**MONDATA** 属性の記述に一致する構造体にこのデータを取り込む必要があります。

次に、MQ キューをオープンする前に、プログラムは取得したデータに指定されている MQ モニターの状態を **started** に設定する必要があります。

CICS に戻る前に、プログラムは MQ モニターの状態を **stopped** に設定する必要があります。

MQ メッセージ・コンシューマーを使用して作業をリモート領域に経路指定するときのセキュリティ上の考慮事項

MQ メッセージ・コンシューマーを使用して作業をリモート領域に経路指定する場合は、リモート領域のトランザクションに関連付けられるユーザー ID の決定にシステム間連絡定義のセキュリティ構成が与える可能性のある影響を考慮する必要があります。詳しくは、[START](#) の **START** コマンドで開始された、経路指定されたトランザクションに関する情報を参照してください。

プロシージャー

ユーザー作成の MQ モニター・プログラムまたは MQ メッセージ・コンシューマー・プログラムは、以下の手順を、示された順序どおりに実行する必要があります。

1. **EXEC CICS ASSIGN STARTCODE** コマンドを使用して、トランザクションがデータを指定して開始されたことを確認します。
2. **EXEC CICS RETRIEVE** コマンドを発行して、トランザクションが開始されたときに渡された FROM データを構造に取り込みます。
以下のステップでは、このデータのバイト 2 から 9 で渡された MQMONITOR 名を使用します。
3. **EXEC CICS SET MQMONITOR(MQMONITOR_name) STARTED** を発行して、取得したデータに指定された MQ モニターの状態を **started** に設定します。
4. MQ キューを開きます。
5. 入力キュー上のメッセージを取得して、必要なアプリケーション・ロジックを実行します。
6. 何らかの理由でプログラムを終了する際には、**EXEC CICS SET MQMONITOR(MQMONITOR_name) STOPPED** を発行して MQ モニターの状態を **stopped** に設定します。MQMONITOR が開始されたときに収集された CICS 統計情報を書き出します。
7. CICS に戻ります。



注意：

- ユーザー作成の MQ トリガー・モニター・プログラムまたは MQ メッセージ・コンシューマー・プログラムが上記のプロシージャーで示されたとおりの順序でステップを実行できなかった場合は、予測不能の結果になる可能性があります。例えば、記録される統計が予測不能になったり、MQMONITOR の MONSTATUS 値が予測不能の状態になったりします。
- CICS 提供の MQ トリガー・モニター・プログラム DFHMQTSK は、CICS-MQ トリガー・モニターおよびタスク・イニシエーター・トランザクション CKTI で使用するために予約されています。ユ

ユーザー・トランザクションとして DFHMQTSK を呼び出そうとすると、ユーザー・トランザクションは異常終了し、異常終了コード AMQO が示されます。

注：ユーザー作成の MQ トリガー・モニター・プログラムまたは MQ メッセージ・コンシューマー・プログラムが **EXEC CICS START** 要求を発行する場合に、**EXEC CICS START** 要求を制限するポリシーがデプロイされていて、同じ領域でアクティブになっていると、ポリシーはそのユーザー作成プログラムを異常終了させることがあります。この状態では、**EXEC CICS START** 要求を制限するポリシーをアクティブにすることは適切ではありません。ポリシー・ルールについて詳しくは、[ポリシーのタスク・ルール](#)を参照してください。

詳細情報

- [CICS Developer Center: MQMONITOR を使用して CICS-MQ トリガー・モニターと MQ メッセージ・コンシューマーの管理を単純化する](#)

このブログでは、MQMONITOR を使用し、参照用のサンプル・コードを提供する、MQ メッセージ・コンシューマー・プログラムの開発方法を説明します。

第 4 章 CICS-MQ ブリッジの管理

ブリッジを開始するには、CICS-MQ ブリッジ・モニター・トランザクションを実行します。トランザクションのデフォルト名は CKBR ですが、独自の代替トランザクションを定義できます。

このタスクについて

CICS-MQ ブリッジを制御するには、CICS-MQ モニターを使用することを推奨します。MQMONITOR リソースにより、ブリッジは、IBM MQ マネージャーへの接続が確立された時点で自動的に再始動できます。セットアップ手順については、[CICS-MQ ブリッジ用の MQMONITOR リソースのセットアップ](#)を参照してください。

CKBR を実行する際に、以下のオプション・パラメーターを指定できます。

Q=qqq

qqq は、CICS-MQ ブリッジ用の IBM MQ 要求キューの名前です。独自の要求キューを定義しておいた場合、それをここで指定する必要があります。IBM MQ でのオブジェクト名は大文字と小文字が区別されるので注意してください。キュー名を指定しない場合、CKBR はデフォルトの要求キュー SYSTEM.CICS.BRIDGE.QUEUE を使用します。

AUTH=LOCAL/IDENTIFY/VERIFY_UOW/VERIFY_ALL

使用する認証のレベルを指定します。デフォルトは LOCAL です。LOCAL を指定すると、ブリッジ・タスクによって実行された CICS プログラムは、ブリッジ・モニターが開始されたときのユーザー ID で開始されるため、これはブリッジ・モニターの開始方法の選択に影響を与える可能性があることに注意してください。その他の認証レベルでは、CICS プログラムは、要求メッセージから取り出されたユーザー ID で実行されます。認証レベルについての詳細は、[CICS-WebSphere MQ ブリッジ用のセキュリティ](#)を参照してください。

WAIT=nnn

nnn は、多数のユーザー・プログラムを実行する作業単位をブリッジ・タスクが処理する際に、ブリッジ・タスクがタイムアウトになるまで後続の要求を待機する秒数です。この値の範囲は 0 から 999 までです。このパラメーターを指定しない場合、デフォルトは MQWI_UNLIMITED (値が -1) です。待ち時間を指定することを推奨します。これは、待ち時間を指定しないと、ブリッジが CICS またはキュー・マネージャーのシャットダウンを禁止するおそれがあるためです。

MSG=CSMT/LOG/BOTH

CICS-MQ ブリッジによって生成されたメッセージが CICS ジョブ・ログ、CICS マスター端末、またはその両方のいずれに送信されるかを決定します。デフォルトは BOTH です。

PASSTKTA=applid

applid は、PassTicket の検証に使用されるアプリケーション ID を指定します。デフォルトは、CICS 領域のアプリケーション ID です。PassTicket については、[IBM MQ 製品資料内の『z/OS でのセキュリティのセットアップ』](#)を参照してください。

ROUTEMEM=Y/N

マーク有効期限切れメッセージが DLQ に送付されるかどうかを決定します。パラメーターを指定しない場合、デフォルトは N (送付なし) です。

SMFMQGET=number

システム初期設定パラメーター **MNSYSNC=NO** が指定され、CICS-MQ ブリッジの開始時にパフォーマンス・クラス・モニターがアクティブである場合、ブリッジは、指定された数の MQGET 要求がブリッジによって発行されるたびに SMF タイプ 110 サブタイプ 1 モニター・レコードを書き込みます。有効な数は 1 から 99 999 までの範囲です。

CICS-MQ ブリッジの開始

CICS MQ モニターを使用してブリッジを制御しており、MQMONITOR リソースが正しく構成されている場合、ブリッジは、MQ キュー・マネージャーへの接続が確立された時に自動的に開始されます。MQMONITOR リソースの使用は、推奨されるブリッジ制御方法です。また、CKBR トランザクションを使用して手動でブリッジを開始することもできます。

始める前に

- MQ モニターを使用してブリッジを制御する場合は、MQ ブリッジを制御するために、MQMONITOR リソースを定義およびインストールする必要があります。詳しい説明は、[CICS-MQ ブリッジ用の MQMONITOR リソースのセットアップ](#)を参照してください。
- それに加え、セキュリティチェックがアクティブな場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合) は、MQ モニターの状態を **started** に設定しようとするトランザクションに関連付けられたユーザー ID が、**MONUSERID** に定義されているユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションを開始する権限を持っていることを確認してください。CICS Explorer などの CICSplex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPIUSR ユーザー ID (指定されている場合) を使用してください。

手順

- MQ キュー・マネージャーへの接続が確立された時に CICS-MQ ブリッジを自動的に開始するには、MQ モニターを使用してブリッジを制御し、この MQMONITOR リソースに、属性 AUTOSTART(YES) および STATUS(ENABLED) が指定されていることを確実にします。
CICS と MQ の接続が確立された後、MQCONN リソースを CONNECTED に設定したタスクに関連付けられているユーザー ID に関連するトランザクションを開始するための十分な権限がある場合、CICS は、属性 AUTOSTART(YES) および STATUS(ENABLED) を指定してインストールされた MQ モニターを開始します。
- MQ モニターを使用してブリッジを制御している場合は、いくつかの方法で MQ モニターを手動で開始できます。[50 ページの『CICS MQ モニターの開始』](#)の説明に従ってください。
- CKBR トランザクションを使用してブリッジを手動で開始する場合は、以下のいずれかの方法で CKBR タスクを開始します。
 - (3270 またはその他の) 端末から、次の 1 行を入力します。

```
CKBR Q=<queue name>,AUTH=<auth option>,WAIT=nnn,MSG=<msg option>,PASSTKA=<applid>,ROUTEMEM=<routemem option>,SMFMQGET=<number>
```

以下に例を示します。

```
CKBR Q=MyQueue,AUTH=IDENTIFY,WAIT=30,MSG=LOG,PASSTKA=APP1,ROUTEMEM=Y,SMFMQGET=1
```

この入力の後、端末はアンロックされ、他の作業に使用できるようになります。

- パラメーターをデータとして指定して、CKBR トランザクションに対して **EXEC CICS START** コマンドを発行します。
CICS PLTPI 処理の一部として実行されてこのコマンドを発行するプログラムを作成でき、ブリッジ・モニター・トランザクションの実行に使用されるユーザー ID をこのプログラムに指定できます。
- COMMAREA にデータをパラメーターとして指定して **EXEC CICS LINK** を、プログラム DFHMQBRO (CSQCBROO と呼ばれる) に対して発行します。
DFHMQBRO は、長時間実行のタスクであり、このプログラムは、ブリッジが停止した時にのみ戻ります。
- USERDATA の任意のパラメーターと共に APPLICID(CKBR) を指定して、ブリッジ要求キューで TRIGGER TRIGTYPE(FIRST) を使用してプロセスを開始します。
USERDATA には **Q=qqq** パラメーターを指定できないため、独自の要求キューを定義している場合、この方法は適さないことに留意してください。

次のタスク

1 つのキューを共用する複数のブリッジ・モニターを実行している場合、メッセージをブリッジ要求キューに書き込むことによって、それらのモニターのいずれかを開始することができます。ただし、専用ローカル・キューの場合、生成されるのは 1 つのトリガー・メッセージのみであるため、ブリッジ・モニターは 1 つの CICS 領域でのみ開始されます。したがって、追加のブリッジ・モニターを開始するには、例えば、

必要なパラメーターが指定されたトランザクションを開始するために CICS 始動 PLT 処理でプログラムを使用したり、トランザクションを開始するための自動化製品を使用したりするなどの代替方法を検討する必要があります。要求キューの属性を変更してブリッジをシャットダウンする場合は、ブリッジがシャットダウンされた後に GET(ENABLED) をリセットすることを忘れないでください。

CICS-MQ ブリッジの停止

GET(DISABLED) を設定して要求キューの属性を変更するか、CICS をシャットダウンするか、またはキュー・マネージャーをシャットダウンすることにより、CICS-MQ ブリッジをシャットダウンできます。CICS MQ モニターを使用してブリッジを制御している場合は、MQ モニターを停止することによりブリッジをシャットダウンできます。

このタスクについて

MQ モニターを使用してブリッジを制御している場合、ブリッジは、MQ キュー・マネージャーへの接続が停止された時に自動的に停止されます。

手順

- MQ モニターを使用してブリッジを制御している場合は、いくつかの方法で MQ モニターを手動で停止できます。54 ページの『[CICS MQ モニターの停止](#)』の説明に従ってください。

第 5 章 CICS-MQ アダプターのセキュリティ

CICS-MQ アダプターは、IBM MQ に対して、特に IBM MQ セキュリティーで使用される情報を提供します。提供される情報は、次のとおりです。

- CICS リソース・レベル・セキュリティはこのトランザクションに対してアクティブかどうか。詳しくは、[リソース定義のセキュリティ](#)を参照してください。
- ユーザー ID。
 - ユーザーがサインオンしていない端末タスクの場合、ユーザー ID は端末に関連した CICS ユーザー ID で、以下のいずれかです。
 - CICS **DFLTUSER** システム 初期設定パラメーターで指定されたデフォルトの CICS ユーザー ID。
 - 端末の定義に事前設定されているセキュリティ・ユーザー ID。
 - 非端末タスクの場合、アダプターはユーザー・ドメインの呼び出しを使用してユーザー ID を取得します。

CICS-MQ アダプター・トランザクションのセキュリティの実装

CICS-MQ アダプターをユーザーに管理させる場合は、適切な CICS トランザクションに対する権限をそのユーザーに付与する必要があります。

必要に応じて、アダプターの特定の機能に対するアクセスを制限できます。例えば、ユーザーにアダプターの現在の状況を表示することは許可するが、それ以外のものは許可したくない場合、そのユーザーには、CKQC、CKBM、CKRT、および CKDP のアクセス権のみを与えます。

CICS に対してこれらのトランザクションを RESSEC(NO) と CMDSEC(NO) で定義します。詳しくは、[リソース定義のセキュリティ](#)および [CICS command security](#) を参照してください。

トランザクション	機能
CKAM	アラート・モニター
CKBM	アダプター 機能を制御する
CKCN	接続
CKDL	行モード表示
CKDP	フルスクリーン表示
CKQC	アダプター 機能を制御する
CKRS	統計
CKRT	アダプター 機能を制御する
CKSD	切断
CKSQ	CKTI 開始/停止
CKTI	トリガー・モニター

管理者と同様に、IBM MQ に接続するユーザー ID、**PLTPIUSR** システム 初期設定パラメーターで設定されたユーザー ID、および [CICSplex SM MAS エージェント・ユーザー ID](#) にも CKTI トランザクションおよび CKAM トランザクションを実行する権限が必要です。

CICS-MQ アダプター・ユーザー ID

CICS-MQ アダプターに関連付けられたユーザー ID は、IBM MQ にアクセスする呼び出し側トランザクションに関連付けられたユーザー ID です。

IBM MQ リソースに関するユーザー ID 検査

CICS-MQ リソース (MQCONN または MQMONITOR) を使用して IBM MQ にアクセスする場合、IBM MQ が使用するユーザー ID は、MQI コマンドを発行するトランザクションのユーザー ID です。

- PLTPI プログラムの場合、これは **PLTPIUSR** システム初期設定パラメーターです。
- PLTSD プログラムの場合、これはシャットダウン・トランザクションに関連付けられたユーザー ID です。
- その他のプログラムの場合、これは実行中のトランザクションに関連付けられたユーザー ID です。

CICS-MQ トリガー・モニターのユーザー ID

セキュリティ検査がアクティブな場合は、常に MQMONITOR を使用して CKTI インスタンスを開始することをお勧めします。MQMONITOR を使用する場合は、CKQC を使用しないでください。MQMONITOR を使用すると、CKTI インスタンスにおいて単一のユーザー ID が常に使用されます。つまり、その開始方法に関係なく MQMONITOR の **MONUSERID** 属性が使用されます。

MQMONITOR を使用しない CICS-MQ トリガー・モニターのユーザー ID

MQMONITOR を使用せずに CKTI を開始することもできますが、推奨されていません。

MQMONITOR を使用しないで CKTI のインスタンスを開始する場合、CKTI トランザクションに関連付けられるユーザー ID は、CKTI を開始するトランザクションのユーザー ID になります。

- PLTPI プログラムの場合、これは **PLTPIUSR** システム初期設定パラメーターです。
- CKQC ユーザーの場合、これはトランザクションを実行するサインオン・ユーザー ID です。
- CKQC を実行するために順次端末を使用する場合、これは順次端末の実行に使用されるユーザー ID です。このユーザー ID は、デフォルトの CICS ユーザー ID ではなく、事前設定したユーザー ID でなければなりません。 [DFHTCT TYPE=TERMINAL](#) マクロを参照してください。

MQCONN リソースおよび MQMONITOR リソースのコマンド・セキュリティ

MQCONN リソース定義および MQMONITOR リソース定義に対してユーザーが SPI コマンドを発行する機能を制御するには、CICS コマンド・セキュリティを使用します。例えば、それを使用して、どのユーザーが CICS 領域の MQCONN リソース定義に対して CREATE コマンドおよび DISCARD コマンドを発行できるかを制御できます。

コマンド・セキュリティがトランザクションに対して有効にされている場合、外部セキュリティ・マネージャーは、そのトランザクションに関連付けられているユーザー ID が必要に応じて MQCONN リソースまたは MQMONITOR リソースでコマンドの使用を許可されているかどうかを検査します。リソース・セキュリティは、MQCONN リソースおよび MQMONITOR リソースには使用できません。

CICS コマンド・コマンド・セキュリティは、**EXEC CICS CREATE MQCONN**、**DISCARD MQCONN**、**SET MQCONN**、**INQUIRE MQCONN**、**CREATE MQMONITOR**、**DISCARD MQMONITOR**、**SET MQMONITOR**、および **INQUIRE MQMONITOR** の各コマンドを対象としています。コマンド・セキュリティの説明および CICS 領域に対してコマンド・セキュリティをセットアップする手順については、[CICS command security](#) を参照してください。各コマンドに必要な権限レベルのリストについては、[Resource and command check cross-reference](#) を参照してください。

コマンド・セキュリティがアクティブな場合、IBM MQ への接続を開始するために **EXEC CICS SET MQCONN** コマンドを発行する実行中トランザクションのユーザー ID には、以下の権限が必要です。

1. **EXEC CICS SET MQCONN** コマンドを使用する権限。この権限がないと、接続を開始するときに応答 NOTAUTH (RESP2 は 100) が返され、失敗します。

2. **EXEC CICS EXTRACT EXIT** コマンドを使用する権限。この権限がないと、接続を開始するときに応答 INVREQ (RESP2 は 9) が返され、失敗します。この場合、CICS はメッセージ DFHXS1111 および DFHMQ0302 を発行します。

また、MQMONITOR が使用されている場合、MQMONITOR を実行するユーザー ID (MQMONITOR 定義の **MONUSERID** パラメーターで指定) には、コマンド・セキュリティの許可が必要です。これは、CICS-MQ トリガー・モニター、CICS-MQ ブリッジ、またはユーザー作成の MQMONITOR プログラムを制御するために使用される MQMONITOR に適用されます。MONUSERID には、以下の権限が付与されている必要があります。

1. **EXEC CICS SET MQMONITOR** コマンドを使用して MQMONITOR の状況を STARTED または STOPPED に設定する権限。この権限がないと、MQMONITOR タスクは失敗します。CICS-MQ トリガー・モニターの場合は、CICS によってメッセージ DFHMQ0125 が発行されます。
2. CICS-MQ トリガー・モニターの場合、トリガー・メッセージに指定されているトランザクションが **TRANSID** オプションに設定された **EXEC CICS START** コマンドを使用する権限。この権限がないと、CICS はメッセージ DFHMQ0102 を発行し、トリガー・メッセージは送達不能キューに送信されます。

MQMONITOR リソースの代理ユーザー・セキュリティ

CICS 代理ユーザー・セキュリティは、CICS-MQ トリガー・モニターを開始できるトランザクションを制御するときに使用します。これは、MQMONITOR によってトリガー・モニター・インスタンスが開始された場合にのみ適用されます。

セキュリティチェックがアクティブであり、システム初期設定パラメーターとして **XUSER=YES** が指定されている場合に、USERID オプションを指定した **EXEC CICS START** コマンドを使用してトランザクションを開始すると、CICS® は代理ユーザー検査を実行します。MQMONITOR を使用して CICS-MQ トリガー・モニターを開始すると、MQMONITOR 定義の情報を使用してユーザー・トランザクションが開始します。トリガー・モニターは、MQMONITOR の **USERID** 属性から取得した値を指定した USERID オプションを指定して **EXEC CICS START** コマンドを発行します。

CICS では、**START** 要求を発行するトランザクションに関連付けられたユーザー ID が、開始されたトランザクションに関連付けられたユーザー ID の代理になる必要があります。MQMONITOR で開始された CICS-MQ トリガー・モニターは、必ず MQMONITOR の **MONUSERID** 属性で指定されたユーザー ID で実行されます。そのため、トリガー・メッセージに指定されたユーザー・トランザクションを CKTI で開始できるようにするには、MONUSERID を、開始されたユーザー・タスクに関連付けられたユーザー ID の代理にする必要があります。他のソースから適切なユーザー ID を入手できない場合、デフォルトでは、MQMONITOR 定義で指定された USERID を使用してユーザー・トランザクションが開始するので、MONUSERID はその USERID の代理にもする必要があります。代理セキュリティ検査で不合格になると、CICS はメッセージ DFHMQ0102 を発行し、トリガー・メッセージはデッド・レター・キューに送信されます。

CICS-MQ アダプターの IBM MQ 接続セキュリティ

アプリケーション・プログラムが MQCONN 要求または MQCONNX 要求を発行してキュー・マネージャーに接続しようとする場合、またはチャネル・イニシエーターあるいは CICS-MQ アダプターが接続要求を発行する場合に、IBM MQ は接続セキュリティ検査を実行します。

キュー・マネージャー・レベルのセキュリティを使用している場合は、特定のキュー・マネージャーに対する接続セキュリティ検査をオフにすることができますが、そのようにすると、どのユーザーでもそのキュー・マネージャーに接続できます。

接続セキュリティ検査には、CICS アドレス・スペース・ユーザー ID のみ使用され、個々の CICS 端末ユーザー ID は使用されません。

IBM MQ の接続セキュリティ検査は、キュー・マネージャー・レベルまたはキュー共用グループ・レベルのいずれかでオン/オフにすることができます。

第 6 章 CICS-MQ ブリッジのセキュリティ

CICS-MQ ブリッジを開始する際、認証のレベルを指定できます。ブリッジ・モニターは、IBM MQ の要求メッセージから抽出したユーザー ID とパスワードの検査が要求されていれば、その検査を実行してから、その要求メッセージで指定されている CICS プログラムを実行します。

CICS-MQ ブリッジ・モニター・トランザクション (例えば、CKBR またはユーザーのトランザクション名) を実行する際に、**AUTH** パラメーターを指定して以下のいずれかの認証レベルを選択できます。

LOCAL

このレベルがデフォルトです。ブリッジ・モニターは CICS のデフォルト・ユーザー ID を使用してブリッジ・タスクを開始します。ブリッジ・タスクが実行する CICS ユーザー・プログラムには、このユーザー ID に関連付けられた権限があります。メッセージ内のユーザー ID またはパスワードは無視されるため、IBM MQ 要求メッセージは、より高い権限を要求できません。ブリッジ・タスクが保護リソースへのアクセスを試行する CICS プログラムを実行する場合、CICS プログラムは失敗する可能性があります。

IDENTIFY

要求メッセージ内のメッセージ記述子 (MQMD) がユーザー ID を指定する場合、ブリッジ・モニターはそのユーザー ID を使用してブリッジ・タスクを開始します。ブリッジ・タスクが実行する CICS ユーザー・プログラムには、そのユーザー ID に関連付けられた権限があります。ユーザー ID は信頼できると見なされます。つまり、ブリッジ・モニターはパスワードまたはパスチケット情報の使用による ID の認証を行いません。MQMD がユーザー ID を指定しない場合、ブリッジ・モニターは CICS のデフォルト・ユーザー ID を使用して、LOCAL オプションと同じ方法でブリッジ・タスクを開始します。

VERIFY_UOW

以下のすべての条件が該当する場合、ブリッジ・モニターはパスワードまたはパスチケットを使用してユーザー ID を認証します。

- 要求メッセージ内のメッセージ記述子 (MQMD) はユーザー ID を指定します。
- 要求メッセージには IBM MQ CICS 情報ヘッダー (MQCIH) が含まれています。
- MQCIH の「認証子 (Authenticator)」フィールドには、パスワードまたはパスチケットが含まれています。

認証が成功すると、ブリッジ・モニターはそのユーザー ID を使用してブリッジ・タスクを開始します。認証が失敗すると、ブリッジ・モニターは要求に対処できず、MQCRC_SECURITY_ERROR 戻りコードが出されます。

前にリストされた条件のうちのいずれか 1 つが満たされない場合、ブリッジ・モニターは CICS のデフォルト・ユーザー ID を使用して、LOCAL オプションと同じ方法でブリッジ・タスクを開始します。作業単位内の最初の要求メッセージのみが検査されます。同じ作業単位を構成する後続のメッセージのユーザー ID とパスワードまたはパスチケットの情報は無視されます。

VERIFY_ALL

このレベルは VERIFY_UOW と同じです。ただし、ブリッジ・タスクは、ユーザー ID が同じ作業単位内のすべての要求メッセージで同一であることも確認し、要求メッセージに含まれるパスワードまたはパスチケットを使用して、要求メッセージごとにユーザー ID を再認証する点は異なります。

さまざまなアプリケーション用に別々の認証レベルが必要な場合は、複数のブリッジ・モニターを別々のトランザクション ID で使用します。CICS 代理セキュリティーを使用して、ブリッジ・モニター・トランザクションとユーザー ID が開始できるトランザクションとユーザー ID の組み合わせを制限できます。

78 ページの表 9 は、ブリッジ・モニターの開始ユーザー ID を示しています。ユーザー ID は、ブリッジ・モニター・トランザクションの実行に使用するメソッド (通常は CKBR) によって異なります。

表 9. CICS-MQ ブリッジ・モニターのセキュリティ		
ブリッジ・モニターの開始メソッド	サインオンした端末かどうか	ブリッジ・モニターのユーザー ID
端末、またはプログラム内の EXEC CICS LINK から	はい	サインオン・ユーザー ID
端末、またはプログラム内の EXEC CICS LINK から	いいえ	CICS デフォルト・ユーザー ID
ユーザー ID が指定された EXEC CICS START	–	START のユーザー ID
ユーザー ID が指定されていない EXEC CICS START	–	START のユーザー ID
CICS-MQ トリガー・モニター CKTI	–	START のユーザー ID
CICS MQ モニター (MQMONITOR)	–	<ul style="list-style-type: none"> • セキュリティ検査が CICS 領域に対してアクティブの場合 (つまり、SEC システム初期設定パラメーターが YES に設定されている場合)、MQMONITOR リソースの MONUSERID 属性 • セキュリティ検査が CICS 領域に対して無効の場合 (つまり、SEC が NO に設定されている場合)、MQMONITOR リソースを開始したユーザー ID

要求メッセージ内のユーザー ID とパスワード

IDENTIFY、VERIFY_UOW、または VERIFY_ALL の認証オプションを使用する場合、ブリッジ・タスクおよびそれが実行する CICS プログラムは、要求メッセージ内のメッセージ記述子 (MQMD) で指定されたユーザー ID で開始されます。VERIFY_UOW オプションおよび VERIFY_ALL オプションを使用して、ブリッジ・モニターは要求メッセージ内の IBM MQ CICS 情報ヘッダー (MQCIH) で指定されたパスワードも検査します。

これらの認証レベルを使用するには、IBM MQ アプリケーションは MQMD でユーザー ID を提供し、さらに MQCIH (パスワードを含む) を提供する必要があります。これらのユーザー ID を RACF® に対して定義する必要があります。使用されるユーザー ID を制御するには、IBM MQ アプリケーションは、MQOO_SET_IDENTITY_CONTEXT が含まれるオープン・オプションを使用して、ブリッジ・モニターの要求キューを開く必要があります。また、アプリケーションはメッセージの書き込みオプションに MQPMO_SET_IDENTITY_CONTEXT の値を含める必要があります。

ブリッジ・モニターが要求メッセージ内のユーザー ID またはパスワードに問題があることを検出した場合、以下のように処理されます。

- 認証レベルが IDENTIFY の場合、取り消されたユーザー ID がメッセージに含まれると、異常終了 AICO が発生する可能性があります。このエラー応答には、戻りコード MQCRC_BRIDGE_ERROR と理由 MQFB_CICS_BRIDGE_FAILURE があります。
- 認証レベルが VERIFY_UOW または VERIFY_ALL の場合、ユーザー ID またはパスワードが無効であると、要求は失敗し、戻りコード MQCRC_SECURITY_ERROR が出されます。
- 要求メッセージでユーザー ID またはパスワードのいずれかが省略される場合、ブリッジ・モニターが他のいずれかの認証オプションを使用して開始した場合でも、ブリッジ・モニターは LOCAL 認証レベルで

ブリッジ・タスクを実行します。このような状況では、ブリッジ・タスクによって開始された CICS プログラムは、ブリッジ・モニターの開始ユーザー ID で実行されます。

パスワードの代わりにパスチケットを使用して、パスワードがメッセージ内をフローする必要がないようにすることができます。

- IBM MQ アプリケーションは、要求メッセージ内の MQCIH でパスチケットを提供する必要があります。
- パスチケットを生成するには、アプリケーション ID が必要です。デフォルトのアプリケーション ID は CICS APPLID です。CICS-MQ ブリッジ・モニター・トランザクション (例えば、CKBR またはユーザーのトランザクション名) を開始する際に、**PASSTKTA** パラメーターを使用して、代替アプリケーション ID を指定できます。
- 同じ要求キューに複数のブリッジ・モニターを使用する場合、**PASSTKTA** パラメーターを使用して、ブリッジ・モニターごとに同じアプリケーション ID を指定する必要があります。

CICS サービスはユーザーが APPLID を指定することを許可しないため、パスチケットの検証は、**EXEC CICS VERIFY**ではなく、IBM MQ サービスを使用して実行されます。パスチケットの詳細については、[セキュア・サインオンのためのパスチケットの生成と使用および「IBM MQ 製品資料内の『z/OS でのセキュリティのセットアップ』」](#)を参照してください。

権限

CICS-MQ ブリッジで使用するユーザー ID に以下の権限を与える必要があります。ユーザー ID には、ブリッジ・モニター・トランザクションの開始ユーザー ID (78 ページの表 9 にリストされている) と、IBM MQ アプリケーションが要求メッセージ内で指定するユーザー ID が含まれます。

- ブリッジ・モニター・トランザクションの開始ユーザー ID には、CICS DPL プログラムの CKBP トランザクションと CKBC トランザクション、および IBM MQ アプリケーションが要求メッセージの MQCIH 構造の「TransactionId」フィールドで指定する代替のトランザクションを開始するための権限が必要です。
- IBM MQ アプリケーションが要求メッセージでユーザー ID を指定している場合、ブリッジ・モニター・トランザクションの開始ユーザー ID を、要求メッセージで使用されるすべてのユーザー ID の代理として RACF に対して定義する必要があります。代理ユーザーとは、他のユーザーのパスワードを知らなくてもそのユーザーに代わって作業を開始する権限を持っているユーザーのことです。代理ユーザー・セキュリティの詳細については、[代理ユーザー・セキュリティ](#)を参照してください。
- ブリッジ・モニターのユーザー ID およびすべてのブリッジ・タスクのユーザー ID には、要求キューからメッセージを取得するための権限が必要です。
- ブリッジ・タスクのユーザー ID には、その応答先キューにメッセージを書き込むための権限が必要です。
- エラー応答が受信されるようにするために、ブリッジ・モニター・トランザクションの開始ユーザー ID には、メッセージをすべての応答先キューに書き込むための権限が必要です。
- ブリッジ・タスクのユーザー ID には、送達不能キューにメッセージを書き込むための権限が必要です。
- ブリッジ・モニター・トランザクションの開始ユーザー ID には、メッセージを送達不能キューに書き込むための権限が必要です (エラーが発生した場合にブリッジを停止させたくない場合)。
- ブリッジ・モニターのユーザー ID およびすべてのブリッジ・タスクのユーザー ID には、バックアウト・リキュー・キュー (定義されている場合) にメッセージを書き込むための権限が必要です。

第 7 章 CICS-MQ ブリッジを使用するためのアプリケーションの開発

非 CICS アプリケーションは、IBM MQ メッセージを CICS-MQ ブリッジ上で送受信することによって、CICS のプログラムまたはトランザクションと通信できます。CICS アプリケーションによって必要とされるデータは、要求メッセージに含まれます。また、CICS-MQ ブリッジは、応答メッセージを使用して、CICS アプリケーションによって提供されるデータを戻します。

以下のタイプの CICS アプリケーションは、CICS-MQ ブリッジとともに使用するのに適しています。

- **EXEC CICS LINK** コマンドを使用して呼び出される CICS プログラム (DPL プログラムと呼ばれます)。このプログラムは、CICS API の DPL サブセットに規格が合致している必要があります。つまり、CICS 端末機能または同期点機能を使用することはできません。CICS-MQ ブリッジを使用して、単一の CICS プログラム、または作業単位を形成する一連の CICS プログラムを実行できます。
- **3270** 端末から実行するように設計された CICS トランザクション (3270 トランザクションと呼ばれます)。トランザクションは、基本マッピング・サポート (BMS) または端末管理コマンドを使用できます。それらは、会話型か、または疑似会話の一部であることが可能です。また、同期点の発行が許可されています。

非 CICS アプリケーションは内部ロジックおよび CICS トランザクションでの制御フローと対話する必要があるため、3270 トランザクションを CICS-MQ ブリッジを介して実行するには、通常、より複雑なアプリケーション・プログラミングが必要になります。CICS アプリケーションのビジネス・ロジックを含む DPL プログラムを実行することをお勧めします。ただし、一部の CICS アプリケーションでは、アプリケーションのビジネス・ロジックが表示ロジックから分離された状態で構造化されていません。そのため、CICS-MQ ブリッジは、いずれのタイプのアプリケーションとも通信することを許可します。

非 CICS アプリケーションは、構造化された IBM MQ メッセージを CICS-MQ ブリッジの要求キューに送信することによって、CICS アプリケーションを開始します。CICS アプリケーションで必要なすべてのデータを要求メッセージに含めることができます。データは次のとおりです。

- DPL プログラムの場合、必要なデータは、CICS アプリケーションが使用する CICS 通信域 (COMMAREA) データです。CICS アプリケーションが COMMAREA データを一切必要としない場合、メッセージ・データは DPL プログラムの名前のみで構成されます。
- 3270 トランザクションの場合、必要なデータは、CICS アプリケーションで使用されるアプリケーション・データ構造 (ADS) を記述しているベクトルによって構成されます。

要求メッセージには、通常、CICS アプリケーションの制御オプションを提供する IBM MQ CICS 情報ヘッダー (MQCIH 構造) も含まれています。デフォルトのトランザクション・コードで単一の DPL プログラムを実行していて、そのプログラムが何も許可を必要としない場合、MQCIH は必要ありません。その他のすべての場合、およびすべての 3270 トランザクションでは、MQCIH は必要です。

CICS-MQ ブリッジは、応答キューに送信される IBM MQ メッセージで非 CICS アプリケーションに戻される出力 CICS アプリケーション送信データとして生成されるデータを取得します。

- DPL プログラムの場合、戻されるデータは、CICS アプリケーションによる出力として生成される COMMAREA データです。
- 3270 トランザクションの場合、戻されるデータは、CICS アプリケーションによって出力として生成されるアプリケーション・データ構造 (ADS) を記述しているベクトルで構成されます。非 CICS アプリケーションは、これらのベクトルを解釈し、応答する必要があります。

単一または複数の DPL プログラムや 3270 トランザクション実行するために、CICS-MQ ブリッジが IBM MQ メッセージを処理する方法について詳しくは、[CICS-WebSphere MQ ブリッジでの CICS DPL プログラムの実行方法](#)および [CICS-WebSphere MQ ブリッジでの CICS 3270 トランザクションの実行方法](#)を参照してください。

3270 トランザクションを実行する計画を立てる場合、CICS トランザクションのロジックを分析するのに役立つ CICS SupportPac CA1E CICS 3270 ブリッジ・パススルーのコピーを取得すると役立つ場合があります。SupportPac を使用して、インバウンドおよびアウトバウンドのデータ・フローを表示してキャプチ

ャーすることができます。さらに、メッセージがどういう構造でなければならないか、そして MQCIH およびベクトルのフィールドにどんな値を挿入するかについて調べることができます。

CICS-MQ ブリッジ用 DPL メッセージ構造

以下の例は、CICS-MQ ブリッジ経由で CICS DPL プログラムを実行するために非 CICS アプリケーションから出される要求メッセージに使用できるさまざまな構造を示しています。

最も単純なケースでは、メッセージ・データは、実行される DPL プログラムの名前だけで構成されます。データを DPL プログラム (ユーザー・プログラム) が開始時に使用できるようにするには、その後に COMMAREA データまたは DFHREQUEST コンテナ・データを続けます。

作業単位内で複数の DPL プログラムを実行する場合、特定のトランザクション・コードを使用する (デフォルト CKBP に上書きする) 場合、あるいは DPL プログラムを実行したり (プログラム DFHMQBP3 を実行するための CKBC または相当するトランザクションを使用して) チャンネルやコンテナを使用したりするため一定レベルの許可を必要とする場合は、MQCIH で情報を提供する必要があります。MQCIH は、DPL プログラム名と、送信する DPL プログラム・データがあればその前になければなりません。

- 非 CICS アプリケーションがデフォルト 処理オプションを使用して単一 CICS DPL プログラムを実行し、DPL プログラム・データの送受信は行わない場合は、次の構造を使用します。

MQMD		ProgName	
------	--	----------	--

ProgName で指定されたプログラムが、DPL プログラムとして CICS により呼び出されます。

- 非 CICS アプリケーションがデフォルト 処理オプションを使用して単一 DPL プログラムを実行し、COMMAREA データの送受信を行う場合は、次の構造を使用します。

MQMD		ProgName		CommareaData	
------	--	----------	--	--------------	--

- 非 CICS アプリケーションが作業単位内で 1 つ以上の DPL プログラムを実行するか、プログラムを実行するための特定の許可を必要としていて、DPL プログラム・データの送受信は行わない場合は、次の構造を使用します。

MQMD		MQCIH		ProgName	
------	--	-------	--	----------	--

- 非 CICS アプリケーションが作業単位内で 1 つ以上の DPL プログラムを実行するか、プログラムを実行するための特定の許可を必要としていて、COMMAREA データの送受信を行うか、DFHREQUEST コンテナ・データの送信を行い、DFHRESPONSE コンテナ・データを受信する場合は、次の構造を使用します。

MQMD		MQCIH		ProgName		DPL program data	
------	--	-------	--	----------	--	------------------	--

注: CICS-MQ ブリッジにプログラム名だけを送信して COMMAREA データは送信しない場合、プログラム名の長さは 8 文字でなければなりません。右側にスペースを埋め込んだ名前であってはなりません。そのような名前を指定すると、CICS-MQ ブリッジは COMMAREA 負の長さエラーを報告します。COMMAREA データまたは DFHREQUEST コンテナ・データを送信する場合は、プログラム名全体の長さが 8 文字になるように、プログラム名の右側にスペースを埋め込む必要があります。

オプションで、形式名が MQH で始まって標準のリンク・フィールドを含んだ追加ヘッダーを、MQCIH ヘッダーの前に置くこともできます。このようなヘッダーは、ヘッダー内のデータが CICS-MQ ブリッジで使用されることがないため、変更されないまま出力メッセージに返されます。

応答メッセージの構造

DPL プログラムを実行している CICS-MQ ブリッジ・タスクが異常終了した場合、障害が起こる前のインバウンド・メッセージに MQCIH が入っているかないかにかかわらず、次の構造でメッセージが応答キューに返されます。

MQMD		MQCIH		DFHMQ* message	
------	--	-------	--	----------------	--

DFHMQ message* は、エラー・タイプを示すエラー・メッセージを表します。最終宛先が別の CCSID およびエンコードを使用している場合にメッセージが適切に変換されるように、MQCIH.Format フィールドの値は MQFMT_STRING に設定されます。MQCIH には、問題を診断するために使用できるその他のフィールドも入ります。

例: CICS-MQ ブリッジ経由の DPL プログラム要求メッセージ

この C 言語コード・フラグメントは、メッセージ・バッファを構成して IBM MQ CICS 情報ヘッダー (MQCIH) を組み込むことによって、COMMAREA データを使用して CICS DPL プログラムを CICS-MQ ブリッジ経由で呼び出す方法を示しています。

呼び出される DPL プログラムは、DPL サブセット規則に従っていなければなりません。詳しくは、[CICS DPL のアプリケーション・プログラミング](#)を参照してください。

```
/* #defines */
#define PGMNAME "DPLPGM" /* DPL program name */
#define PGMNAMELEN 8
#define CALEN 100 /* Commarea length */
:
/* Data declarations */
MQMD mqmd; /* Message descriptor */
MQCIH mqcih; /* CICS information header */
MQCHAR *Commarea; /* Commarea pointer */
MQCHAR *MsgBuffer; /* Message buffer pointer */
:
/* allocate storage for the buffers */

Commarea = malloc(CALEN * sizeof(MQCHAR));
MsgBuffer = malloc(sizeof(MQCIH) + PGMNAMELEN + CALEN);
:
/* Initialize commarea with data */
:
/* Initialize fields in the MQMD as required, including:

memcpy(mqmd.MsgId, MQMI_NONE, sizeof(mqmd.MsgId));
memcpy(mqmd.CorrelId, MQCI_NEW_SESSION, sizeof(mqmd.CorrelId));

/* Initialize fields in the MQCIH as required
:
/* Copy the MQCIH to the start of the message buffer

memcpy(MsgBuffer, &mqcih, sizeof(MQCIH));

/* Set 8 bytes after the MQCIH to spaces

memset(MsgBuffer + sizeof(MQCIH), ' ', PGMNAMELEN);

/* Append the program name to the MQCIH. If it is less than
/* 8 characters, it is now padded to the right with spaces.

memcpy(MsgBuffer + sizeof(MQCIH), PGMNAME, PGMNAMELEN);

/* Append the commarea after the program name

memcpy(MsgBuffer + sizeof(MQCIH) + PGMNAMELEN, &Commarea, CALEN);

/* The message buffer is now ready for the MQPUT
/* to the Bridge Request Queue.
:
:
```

CICS-MQ ブリッジ用の 3270 トランザクション・メッセージ構造

このセクションに記載した例は、非 CICS アプリケーションが CICS-MQ ブリッジ経由で CICS 3270 トランザクションを実行するときの要求メッセージと応答メッセージの構造を示しています。CICS-MQ ブリッジ・ベクトルは、トランザクション内の EXEC CICS コマンドを表すためにメッセージ内で使用されます。

CICS-MQ ブリッジは、端末ユーザーのエミュレーションや EXEC CICS コマンドの発行や受信を含め、CICS トランザクションを開始するいくつかの方法をエミュレートできます。IBM MQ を使用せずに、以下のようないくつかの方法で CICS トランザクションを開始できます。

- 端末ユーザーは、トランザクション名を入力し、オプションでその後にデータを続けることができます。トランザクションは、開始時に EXEC CICS RECEIVE コマンドを発行することにより、ID の後続くデータを取得できます。
- EXEC CICS RETURN TRANSID(*transid*) が発行されると、端末での先行トランザクションが停止します。端末は 3270 データ・ストリームを送信し、新しいトランザクションを開始します。この方法で開始されるトランザクションは、BMS (基本マッピング・サポート) マッピングと端末制御のどちらを使用しているかによって、EXEC CICS RECEIVE MAP コマンドか EXEC CICS RECEIVE コマンドを発行し、3270 データ・ストリーム内のデータを取得します。
- アプリケーションが EXEC CICS START コマンドを発行します。開始されたトランザクションは EXEC CICS RETRIEVE コマンドを発行し、START コマンドで指定されたデータを取得します。

端末で開始されたトランザクションはその後、端末ユーザーとの会話または疑似会話で、EXEC CICS CONVERSE、EXEC CICS SEND MAP、および EXEC CICS RECEIVE MAP などのコマンドを発行できます。

CICS-MQ ブリッジは、CICS トランザクションを開始するこれらのどの方法もエミュレートできます。また、トランザクションからのデータの画面を送受信する端末ユーザーもエミュレートできます。これらのエミュレーションは、CICS-MQ ブリッジ・ベクトルを使用することによって実現されます。これらのベクトルは、エミュレートされる EXEC CICS コマンドを表し、必要とされるすべてのデータを提供します。インバウンド・メッセージには CICS トランザクションが必要とするデータが伴い、アウトバウンド・メッセージには CICS-MQ ブリッジ・アプリケーションが必要とするデータが伴います。

CICS-MQ ブリッジを使用する 3270 トランザクションのベクトル

CICS-MQ ブリッジは、最小限の機能である BMS を含め、CICS 端末 API のすべての機能をエミュレートします。CICS-MQ ブリッジを使用して 3270 トランザクションを実行する場合、**EXEC CICS** コマンドを表すベクトルを要求メッセージと応答メッセージで使用します。

ベクトルは、ブリッジ・メッセージの中でベクトル記述子という数字のストリングによって識別されます。各ベクトル記述子は、それが表す EXEC CICS コマンドの CICS EIBFN 値です。例えば、0402 は **EXEC CICS RECEIVE** の EIBFN 値であると同時に、RECEIVE ベクトルのベクトル記述子でもあります。ベクトルが (CICS-MQ ブリッジへの) インバウンドか (CICS-MQ ブリッジからの) アウトバウンドかを示すために、ベクトルはさらに文字 I と O で修飾されます。複数の連結ベクトルがメッセージに含まれる場合があります。

データなしでトランザクションを開始する場合を除き、3270 トランザクションを実行するために CICS-MQ ブリッジに送信されるインバウンド・メッセージには、MQCIH の後にベクトル構造体があればなりません。CICS トランザクションからの将来の RECEIVE MAP 要求に備えて、複数の RECEIVE MAP ベクトルをインバウンド・メッセージに格納することができます。アプリケーションは、メッセージを構成できるように、トランザクション内の制御のフローを認識している必要があります。

アウトバウンド・メッセージには、要求ベクトルと応答ベクトルを格納することができます。これは、アウトバウンド・メッセージが、要求キューか応答キューのどちらかに入れられるという意味ではありません。すべてのアウトバウンド・メッセージは、応答キューに入れられます。CICS トランザクションは、要求ベクトルを使用して、仮想端末の機能を果たしている非 CICS アプリケーションにデータを要求します。CICS トランザクションは、データが返される必要がない場合は応答ベクトルを使用します。インバウンド・ベクトルでは、このような区別はありません。

アウトバウンド・メッセージには複数のベクトルを格納することができます。例えば、トランザクションによって発行される連続した **EXEC CICS SEND MAP** コマンドの結果としてのような場合です。CICS トランザクションは、アウトバウンド・メッセージに単一のベクトルを格納するか複数のベクトルを格納す

るかは制御しません。要求ベクトルを生成するコマンドをトランザクションが発行する場合、要求ベクトルは必ずメッセージ内の最後のものになります。

以下のベクトルが使用可能です。各ベクトルが表す CICS コマンドにするには、ベクトル名に接頭部 EXEC CICS を付けます。

インバウンド・ベクトル:

- RECEIVE
- RECEIVE MAP
- CONVERSE
- RETRIEVE

アウトバウンド応答ベクトル (次のインバウンド・メッセージでこれ以上のデータは不要):

- SEND
- SEND CONTROL
- SEND MAP
- SEND TEXT
- ISSUE ERASEAUP

アウトバウンド要求ベクトル (次のインバウンド・メッセージでさらにデータが必要):

- RECEIVE
- RECEIVE MAP
- CONVERSE

これらのベクトルはそれぞれ設計された構造体であり、後に可変長データが続きます。すべてのベクトルが共通ヘッダーを持ちますが、各ベクトルの構造は異なります。構造について詳しくは、[Link3270 メッセージ形式](#)を参照してください。

ベクトル構造体定義は、C 言語ヘッダー・ファイル `dfhbrmqh.h` および COBOL コピーブック `DFHBRMQO` にあります。これらの定義を、ブリッジを使用するすべてのアプリケーションに組み込んでください。これらのメンバーは、z/OS 上の CICS Transaction Server でのみ提供されています。別のプラットフォーム上でアプリケーションを作成する場合は、その環境にこれらのメンバーをコピーしてください。

RETRIEVE ベクトル

CICS 3270 トランザクションがその開始データを取得するために **EXEC CICS RETRIEVE ... QUEUE(data-area)** コマンドを発行した場合、非 CICS アプリケーションは、C で `brmq_retrieve` と定義された RETRIEVE ベクトル構造体が入ったメッセージを CICS-MQ ブリッジに送信しなければなりません。この構造体には長さが 8 バイトの文字フィールドがあります。非 CICS アプリケーションは、取り出されるデータが入っている一時記憶域キューの名前を、その文字フィールドで指定する必要があります。RETRIEVE ベクトルが入っているメッセージは、会話または疑似会話を表す交換において常に最初のメッセージです。

CICS-MQ ブリッジを使用する 3270 要求メッセージ (インバウンド) の構造

以下の例は、非 CICS アプリケーションが CICS-MQ ブリッジを使用して 3270 トランザクションに送信する要求メッセージ (インバウンド・メッセージ) の構造を示します。

- 非 CICS アプリケーションがデータなしで CICS トランザクションを呼び出す場合は、次の構造を使用します。

MQMD		MQCIH	
------	--	-------	--

フィールド `MQCIH.TransactionId` を、開始するトランザクションの名前に設定します。MQCIH 内の中その他のフィールドは、アプリケーションに応じて適切な値に設定します。

- データが使用可能であることを期待する EXEC CICS コマンドを発行するトランザクションを非 CICS アプリケーションが実行する場合は、次の構造を使用します。

MQMD	MQCIH	BRMQ structure	data
------	-------	----------------	------

BRMQ structure は、インバウンド・ベクトル構造体 RECEIVE、RECEIVE MAP、CONVERSE、または RETRIEVE のいずれかを表します。BRMQ structure そのものはヘッダーとその後に続くベクトルで構成され、これらのベクトルにデータを含めることができます。

- ゼロ長データが入っている要求メッセージの場合は、次の構造を使用します。

MQMD	MQCIH	BRMQ structure
------	-------	----------------

例えば、インバウンド RECEIVE MAP ベクトルは、ユーザーが PF キーを押しただけというアクションを表すことができます。この場合、BRMQ structure 内のフィールドがどの AID キーが使用されたのかを示しますが、BRMQ structure の後ろにはデータが続きません。

- オプションで、形式名が MQH で始まって標準のリンク・フィールドを含んだ追加ヘッダーを、MQCIH ヘッダーの前に置くこともできます。このようなヘッダーは、ヘッダー内のデータが CICS-MQ ブリッジで使用されることがないため、変更されないまま出力メッセージに返されます。MQCIH の前にヘッダーがある要求メッセージの場合は、次の構造を使用します。

MQMD	MQRFH2	MQCIH
------	--------	-------

CICS-MQ ブリッジを使用する 3270 応答メッセージ (アウトバウンド) の構造

以下の例は、3270 トランザクションに代わって CICS-MQ ブリッジが非 CICS アプリケーションに送信する応答メッセージ (アウトバウンド・メッセージ) の構造を示しています。

CICS-MQ ブリッジからのアウトバウンド・メッセージの構造は、エラーが発生したかどうかによって、3 つの構造のうちの 1 つになります。以下の各例では、ベクトルが 1 つだけ示されていますが、エラーが発生しない限り、メッセージ内でベクトルが複数連結されることがあります。

- CICS-MQ ブリッジ処理が正常終了し、エラーが検出されず、非 CICS アプリケーションにデータが返されない場合は、次の構造が使用されます。

MQMD	MQCIH	BRMQ structure
------	-------	----------------

CICS アプリケーションが異常終了しても、CICS-MQ ブリッジによって正常完了と見なされます。CICS アプリケーションによって発行された異常終了コードは、MQCIH に入ります。

- CICS-MQ ブリッジ処理が正常終了し、エラーが検出されず、非 CICS アプリケーションにデータが返される場合は、次の構造が使用されます。

MQMD	MQCIH	BRMQ structure	data
------	-------	----------------	------

BRMQ structure は、設計されたアウトバウンド応答ベクトル構造体またはアウトバウンド要求ベクトル構造体を表します。

- CICS-MQ ブリッジ・モニターによってエラーが検出された後に CICS-MQ ブリッジ処理が異常終了する場合は、次の構造が使用されます。

MQMD	MQCIH	DFHMQ* message
------	-------	----------------

DFHMQ* message は、エラー・タイプを示すエラー・メッセージを表します。最終宛先が別の CCSID およびエンコードを使用している場合にメッセージが適切に変換されるように、フィールド MQCIH.Format の値は MQFMT_STRING に設定されます。MQCIH には、問題を診断するために使用できるその他のフィールドも入ります。

明示的な同期点要求またはロールバック要求を発行する CICS トランザクションの場合、CICS-MQ ブリッジは応答キューに追加メッセージも生成して、MQCIH 内の TaskEndStatus フィールドで同期点の結果を示します。この追加メッセージは、MQMT_DATAGRAM の MQMD MsgType と共に送信されます。アプリケーションからの入力 は 予期されず、追加メッセージの後に通常のタスク終了メッセージが続きます。

例: CICS-MQ ブリッジ経由で CEMT INQUIRE TASK を発行する要求メッセージ

この例は、アプリケーションが CICS-MQ ブリッジを使用して 3270 トランザクション (ここでは CEMT) を開始するための方法を示しています。このような 3270 トランザクションは、通常その ID といくつかのコマンド行引数を CICS 端末で入力することにより開始されます。

CEMT タスクが開始すると、CEMT タスクはその ID の後に続くコマンド行引数を受け取るために、EXEC CICS RECEIVE コマンドを発行します。したがって、コマンド行呼び出しをエミュレートする非 CICS アプリケーションは、ベクトル構造体に適切な値の入った RECEIVE ベクトルを使用して CEMT を開始する必要があります。また、コマンド行の値を組み込む必要があります。

要求メッセージの組み立て

次の C 言語コード・フラグメントは、要求メッセージを組み立てる方法を示しています。C 言語ヘッダー・ファイル dfhbrmqh.h は、CICS SAMPLIB ライブラリーにあります。

```
/* #includes */
#include cmqc.h /* IBM MQ header */
#include dfhbrmqh.h /* Vector structures */
:
/* #defines */
#define CMDSTRING "CEMT I TASK" /* Command string */
#define RCV_VECTOR "0402" /* Vector descriptor */
#define INBOUND "I " /* Inbound type */
#define VERSION "0000" /* Vector version */
#define YES "Y " /* YES indicator */
#define NO "N " /* NO indicator */
:
/* Data declarations */
/* AID indicator value */
const char AID[ 4 ] = { 0x7d, ' ', ' ', ' ' };
MQMD mqmd; /* Message descriptor */
MQCIH mqcih = {MQCIH_DEFAULT}; /* CICS information header */
brmq_vector_header brvh; /* Standard vector header */
brmq_receive brrcv; /* RECEIVE vector structure */
MQCHAR *MsgBuffer; /* Message buffer pointer */
:
```

応答キューに返されるアウトバウンド・メッセージには、端末管理形式のデータを含んだ SEND 応答ベクトルが入っています。アプリケーションはこの形式を解釈して、受信するデータを分析する必要があります。

メッセージ・バッファ用ストレージの割り振り

```
/* allocate storage for the message buffer. Note that the RECEIVE */
/* vector structure includes space for the standard vector header. */

MsgBuffer = malloc(sizeof(MQCIH) + sizeof(brrcv) + strlen(CMDSTRING));
:
```

MQMD のセットアップ

```
memcpy(mqmd.Format, MQFMT_CICS, sizeof(MQFMT_CICS));
memcpy(mqmd.MsgId, MQMI_NONE, sizeof(MQMI_NONE));
memcpy(mqmd.CorrelId, MQCI_NEW_SESSION, sizeof(MQCI_NEW_SESSION));
mqmd.MsgType = MQMT_REQUEST;
strcpy(mqmd.ReplyToQueue, "MyReplyQueue");
```


MQCIH のセットアップ

```
mqcih.LinkType = MQCLT_TRANSACTION ;
mqcih.GetWaitInterval = 1000 ; /* one second */
mqcih.FacilityKeepTime = 10000 ; /* != 0 says return token */
memcpy(mqcih.Facility, MQCFAC_NONE, sizeof(MQCFAC_NONE)) ;
strcpy(mqcih.TransactionId, "CEMT", strlen("CEMT"));
strcpy(mqcih.FacilityLike, " ", strlen(" "));
mqcih.UOWControl = MQCUOWC_FIRST;
memcpy(mqcih.AttentionId,AID,sizeof(mqcih.AttentionId); /* enter pressed */
```

BRMQ のセットアップ

```
brvh.brmq_vector_length = sizeof(brrcv) + strlen(CMDSTRING) ;
strcpy(brvh.brmq_vector_descriptor, RCV_VECTOR, strlen(RCV_VECTOR)) ;
strcpy(brvh.brmq_vector_type, INBOUND, strlen(INBOUND)) ;
strcpy(brvh.brmq_vector_version, VERSION, strlen(VERSION)) ;
/* Initialize fields in the RECEIVE vector structure: */
strcpy(brrcv.brmq_re_transmit_send_areas, YES, strlen(YES)) ;
strcpy(brrcv.brmq_re_buffer_indicator, NO, strlen(NO)) ;
strcpy(brrcv.brmq_re_aid, AID, sizeof(brrcv.brmq_re_aid)) ;
brrcv.brmq_re_data_len = strlen(CMDSTRING) ;
```

メッセージの作成

```
/* Copy the MQCIH to the start of the message buffer */
memcpy(MsgBuffer, &mqcih, sizeof(MQCIH)) ;
/* Append the RECEIVE vector to the CIH */
memcpy(MsgBuffer + sizeof(MQCIH), brrcv, sizeof(brrcv)) ;
/* Overlay the standard vector header on the RECEIVE vector */
memcpy(MsgBuffer + sizeof(MQCIH), brvh, sizeof(brvh)) ;
/* Append the command string to the vector structure */
strcpy(MsgBuffer + sizeof(MQCIH) + sizeof(brrcv), CMDSTRING, strlen(CMDSTRING)) ;
/* the message is now ready for the MQPUT with length of */
/* sizeof(MQCIH) + sizeof(brrcv) + strlen(CMDSTRING) */
```

CICS-MQ ブリッジ用に MQMD および MQCIH 構造体の中に設定する必要があるフィールド

非 CICS アプリケーションは、CICS-MQ ブリッジのために、要求メッセージ内の MQMD 構造体および MQCIH 構造体の中にいくつかのフィールドを設定する必要があります。

CICS-MQ ブリッジ・メッセージ用の MQMD フィールド

CICS-MQ ブリッジの操作に影響を与えることができる MQMD 内のフィールドは、アプリケーション・プログラム内で初期化されなければなりません。

MQMD.CorrelId

要求キューに対する MQPUT コマンドの場合、作業単位内の最初のメッセージまたはメッセージが 1 つしかない場合はそのメッセージで、値を MQCI_NEW_SESSION に設定します。作業単位内の後続メッセージでは、最初のメッセージを要求キューに入れたときに IBM MQ がメッセージ記述子の中に設定した MQMD.MsgId に値を設定します。

応答キューに対する MQGET コマンドの場合は、最新のメッセージを要求キューに入れたときに IBM MQ がメッセージ記述子の中に設定した MQMD.MsgId 値を使用するか、または MQCI_NONE を指定します。92 ページの『DPL プログラム用の MsgId、CorrelId、および UOWControl フィールド』および 93 ページの『3270 トランザクション用の MsgId、CorrelId、および UOWControl フィールド』も参照してください。

MQMD.Expiry

アプリケーションに応答を待機させる時間に基づいてメッセージの有効期限時間を設定します。応答メッセージに残りの有効期限時間を伝搬するには、MQCIH フラグを設定します。

MQMD.Format

DPL プログラムの場合、ブリッジ要求キューに送信するメッセージに MQCIH を組み込むときは、値を MQCICS に設定します。そうでなければ、後続データの形式に設定します。

3270 トランザクションの場合、値は MQFMT_CICS でなければなりません。

MQMD.MsgId

要求キューに対する MQPUT コマンドの場合は、MsgId は作業単位の固有値または MQMI_NONE に設定します。

応答キューに対する MQGET コマンドの場合は、最初のメッセージを要求キューに入れたときに IBM MQ がメッセージ記述子の中に設定した MQMD.MsgId 値を使用します。92 ページの『DPL プログラム用の MsgId、CorrelId、および UOWControl フィールド』および 93 ページの『3270 トランザクション用の MsgId、CorrelId、および UOWControl フィールド』も参照してください。

MQMD.Persistence

要求キューに入れられたメッセージを永続または非永続にすることができます。応答メッセージの永続性は、要求メッセージと同じになります。

MQMD.ReplyToQ

ブリッジが応答メッセージを送信する宛先キューの名前に値を設定します。

MQMD.UserIdentifier

このフィールドが使用されるのは、許可レベル IDENTIFY、VERIFY_UOW、または VERIFY_ALL でブリッジ・モニターが実行されているときだけです。これらの許可レベルのいずれかを使用する場合は、CICS DPL プログラムへのアクセス権限が検査されるユーザー ID に値を設定してください。VERIFY_* オプションのいずれかを指定してこのフィールドを使用する場合は、MQCIH.Authenticator フィールドも初期化する必要があります。ユーザー ID に関連付けられたパスワードまたはパスチケットの値に設定してください。

注：許可レベル IDENTIFY、VERIFY_UOW、または VERIFY_ALL を使用する場合は、ブリッジ要求キューを開くときにオープン・オプションに値 MQOO_SET_IDENTITY_CONTEXT を追加する必要があるほか、キューにメッセージを送信するときにメッセージ書き込みオプションに値 MQPMO_SET_IDENTITY_CONTEXT を追加する必要があります。

DPL プログラム要求メッセージ用の MQCIH フィールド

CICS-MQ ブリッジを使用して CICS DPL プログラムを実行する要求メッセージを送信するときは、以下の重要な入力フィールドをアプリケーション・プログラム内で初期化する必要があります。

MQCIH.Authenticator

このフィールドが適用されるのは、許可レベルとして VERIFY_UOW または VERIFY_ALL を使用する場合のみです。MQMD.UserIdentifier フィールドのユーザー ID に関連付けられるパスワードまたはパスチケットに値を設定します。これらの値は、DPL プログラムにリンクすることをユーザーが許可されているかどうかを判別する目的で、外部セキュリティー・マネージャーによって使用されます。パスチケットを使用する場合、パスチケットの生成に使用されるアプリケーション ID は、ブリッジ・モニターの開始時に使用される PASSTKTA キーワード値と同じでなければなりません。

MQCIH.Flags

- 残りの有効期限時間を応答メッセージに渡す場合は、MQCIH_PASS_EXPIRATION に設定します。
- 応答メッセージから末尾のヌル文字 ('00'X) を除去する場合は、MQCIH_REPLY_WITHOUT_NULLS に設定します。
- MQCIH_SYNC_ON_RETURN に設定して、EXEC CICS LINK コマンドに SYNCONRETURN オプションを指定します。

これらの値は、加算して組み合わせることができます。

MQCIH.Format

MQCIH 構造体の後続くデータの形式を指定します。データが文字データの場合は MQFMT_STRING を使用します。変換が必要ない場合は MQFMT_NONE を使用します。

MQCIH.GetWaitInterval

このフィールドをデフォルトにすると、作業単位内のメッセージのブリッジ・タスク GET WAIT 間隔は、ブリッジ・モニターの開始時に WAIT パラメーターで指定された値になります。WAIT パラメーターもデフォルトにした場合、GET WAIT 間隔は無制限になります。

MQCIH.LinkType

CICS DPL ブリッジを使用している場合は、MQCLT_PROGRAM を指定します。

MQCIH.OutputDataLength

このフィールドは CICS DPL ブリッジにのみ適用され、プログラムが返すデータの長さを設定します。チャンネルおよびコンテナー・インターフェースを使用する DPL 要求の場合、MQCIH.OutputDataLength は無視されます。これらの要求の場合、出力 (すなわち応答) の長さは、DFHRESPONSE コンテナーのサイズになります。

MQCIH.RemoteSysId

特定の CICS システムで要求を処理する必要がなければ、このフィールドはブランクにしておいてください。

MQCIH.ReplyToFormat

アプリケーションとブリッジが同じ CCSID およびエンコード環境で動作する場合は、このフィールドを MQFMT_NONE (デフォルト値) に設定します。それ以外の場合は、返される COMMAREA データの形式に値を設定します。チャンネルおよびコンテナー・インターフェースを使用する DPL 要求の場合、MQCIH.replytoformat は無視されます。これらの要求の場合、応答先形式は DFHRESPONSE コンテナーの内容に基づいて設定されます。つまり、内容が文字データの場合、応答先形式は MQFMT_STRING になります。内容がバイナリー・データの場合、応答先形式は MQFMT_NONE になります。

MQCIH.TransactionId

CICS DPL ブリッジ・プログラムが実行されるトランザクション・コードを指定します。提供される以下のトランザクションのうちの 1 つを指定できます。

- CKBP (COMMAREA を使用してプログラム DFHMQBP0 を実行)
- CKBC (チャンネルおよびコンテナーを使用してプログラム DFHMQBP3 を実行)
- DFHMQBP0 または DFHMQBP3 を実行するための独自のトランザクション・コード

このフィールドをブランク (スペース 4 つ) にしておくと、デフォルトで提供トランザクション・コード CKBP になります。

MQCIH.UOWControl

このフィールドは、ブリッジが実行する作業単位処理を制御します。詳しくは、[92 ページの『DPL プログラム用の MsgId、CorrelId、および UOWControl フィールド』](#)を参照してください。

3270 トランザクション要求メッセージ用の MQCIH フィールド

CICS-MQ ブリッジを使用して 3270 トランザクションを実行する要求メッセージを送信するときは、以下の重要な入力フィールドをアプリケーション・プログラム内で初期化する必要があります。

MQCIH.ADSDescriptor

このフィールドは、BMS SEND MAP および RECEIVE MAP 呼び出しを使用するトランザクションに適用されます。この場合、CICS-MQ ブリッジ要求メッセージを送信するアプリケーションがワークステーション上にある場合は、この値を MQCADSD_SEND + MQCADSD_RECV + MQCADSD_MSGFORMAT に設定してください。この設定により、ワークステーションとメインフレームの異なる CCSID およびエンコード・スキーム間で、要求メッセージおよび応答メッセージの中のベクトルが正しく変換されるようになります。

MQCIH.AttentionId

このフィールドは、トランザクションが予期する AID キーがある場合は、それを表す値に設定します。ない場合は、デフォルト値のスペース 4 つを受け入れます。このデフォルト値は、CICS トランザクションには ENTER AID キーとして現れます。

インバウンド・ベクトルの RECEIVE、RECEIVE MAP、および CONVERSE にも、AID 値を指定できるフィールドがあります。MQCIH の中の値は、アプリケーションの開始時に EIBAID が設定される値です。この値は、トランザクションを開始するために使用されるファンクション・キーを表しています。インバウンド・ベクトルの中の値は、データが入力されるときに使用される値です。例えば、この値は、EXEC CICS RECEIVE MAP 命令後は EIBAID の値になります。

注:

1. 会話型トランザクションの場合、MQCIH の初期値とベクトルの値は別個の値です。
2. 非 CICS アプリケーションが要求ベクトルに応答してメッセージを送信する場合、MQCIH の中の値は無視されます。
3. 疑似会話型トランザクションの場合は、MQCIH と最初のベクトルに同じ値を入力してください。

このフィールドの最初のバイトは、CICS コピーブック DFHAID 中の値に設定されます。

MQCIH.Authenticator

このフィールドが適用されるのは、許可レベルとして VERIFY_UOW または VERIFY_ALL を使用する場合のみです。MQMD.UserIdentifier フィールドのユーザー ID に関連付けられるパスワードまたはパスチケットに値を設定します。これらの値は、3270 トランザクションを開始することをユーザーが許可されているかどうかを判別する目的で、外部セキュリティー・マネージャーによって使用されます。パスチケットを使用する場合、パスチケットの生成に使用されるアプリケーション ID は、ブリッジ・モニターの開始時に使用される PASSTKTA キーワード値と同じでなければなりません。

MQCIH.ConversationalTask

この CICS トランザクションに必要な入力ベクトルがすべてこの入力メッセージで提供される場合は、値 MQCCT_NO を指定します。トランザクションの入力ベクトルを複数のメッセージを使用して提供できる場合は、MQCCT_YES を指定します。

MQCIH.Facility

疑似会話の中の最初のメッセージで値 MQCFAC_NONE を指定するとともに、MQCIH.FacilityKeepTime フィールドをゼロ以外の値に設定します。CICS-MQ ブリッジは、最初のメッセージで機能トークンを返します。この値は、疑似会話の後続インバウンド・メッセージのすべてで使用する必要があります。

MQCIH.FacilityKeepTime

1 つの疑似会話の中で複数のメッセージを送信する場合は、CICS-MQ ブリッジが機能トークンを返すように、最初のメッセージでこのフィールドをゼロ以外の値 (秒単位) に設定します。疑似会話の中の一連のトランザクションは、このように機能トークンが設定された後は、その同じ機能トークンを使用して、関連端末領域 (例えば TCTUA) が疑似会話の期間中保持されるようにしなければなりません。

MQCIH.FacilityLike

デフォルト値のスペース 4 個を使用するか、またはインストール済み端末の名前を指定します。インストール済み端末の名前を検出するには、CICS 端末で CICS コマンド CEMT I TASK または CEMT I TERM を入力します。

MQCIH.Flags

残りの有効期限時間を応答メッセージに渡す場合は、値を MQCIH_PASS_EXPIRATION に設定します。

MQCIH.Format

値を DFHMQDCI (または CSCQBDCI) に設定します。この値は、MQCIH に続くデータはブリッジへのインバウンドであり、変換しなければならない可能性があることを CICS-MQ ブリッジに通知します。ブリッジは、応答キューに返されるアウトバウンド・メッセージで MQCIH.Format の値を DFHMQDCO (または CSCQBDCO) に設定します。

MQCIH.GetWaitInterval

このフィールドをデフォルトにすると、作業単位内のメッセージのブリッジ・タスク GET WAIT 間隔は、CICS-MQ ブリッジ・モニターの開始時に WAIT パラメーターで指定された値になります。WAIT パラメーターもデフォルトにした場合、GET WAIT 間隔は無制限になります。

MQCIH.LinkType

3270 トランザクションの MQCLT_TRANSACTION を指定します。

MQCIH.RemoteSysid

要求を特定の CICS システム上で処理することが必要でない限り、疑似会話の最初のメッセージではこのフィールドをブランクに設定します。疑似会話内の後続メッセージでは、このフィールドは最初の応答メッセージで返された値に設定します。

注: 以前のバージョンの CICS-MQ ブリッジでは、RemoteSysId フィールドは使用されていませんでした。しかし現在では、複数の CICS-MQ ブリッジ・モニター用の機能を使用できるようにするために、会話全体を通じてこのフィールドが渡されるようにする必要があります。CICS プログラミングの標準的なスタイルは疑似会話型です。つまり、一体となるようにリンクされて完全なアプリケーションを形成する一連の独立したトランザクションです。CICS-MQ ブリッジを使用する場合、疑似会話のトランザクション間のリンクは、シーケンスの最初のトランザクションによって返される Facility トークンと RemoteSysId 値を会話の後続メッセージに渡すことにより維持されます。

MQCIH.StartCode

3270 トランザクションを開始するときに、このフィールドの値をデフォルト値の MQCSC_NONE から変更する必要があります。使用する値は、トランザクションの性質によって異なります。データなしで EXEC CICS START コマンドによってトランザクションが開始される場合は、値 MQCSC_START を使

用します。そうすれば EXEC CICS RETRIEVE コマンドは発行されません。データ付きの EXEC CICS START コマンドによってトランザクションが開始される場合は、値 MQCSC_STARTDATA を使用します。そうすれば EXEC CICS RETRIEVE コマンドが発行されます。それ以外の場合は、値 MQCSC_TERMINPUT を使用します。

MQCIH.TransactionId

この値は、ブリッジ・タスクによって実行される 3270 トランザクションのトランザクション ID です。最初のメッセージでは、最初に開始するトランザクションを指定する必要があります。後続のメッセージでは、先行応答メッセージで返された MQCIH.NextTransactionId の値にこのフィールドを設定します。

MQCIH.UOWControl

この値は、CICS-MQ ブリッジによって行われる作業単位の処理を制御します。93 ページの『3270 トランザクション用の MsgId、CorrelId、および UOWControl フィールド』も参照してください。

DPL プログラム用の MsgId、CorrelId、および UOWControl フィールド

非 CICS アプリケーションが CICS-MQ ブリッジを使用して単一 DPL プログラムを実行する場合は、MQCIH.UOWControl フィールドの値を MQCUOWC_ONLY に設定します。ただし、アプリケーションが複数メッセージを送受信する場合は、作業単位が正しく処理されるよう代わりのオプションを設定する必要があります。

1 つの作業単位内で複数のユーザー・プログラムを実行するには、MQCIH.UOWControl フィールドの値を次のように設定します。

- 最初の要求で MQCUOWC_FIRST
- 中間のすべての要求で MQCUOWC_MIDDLE
- 最後の要求で MQCUOWC_LAST

アプリケーションは、応答メッセージを受信する前に、1 つの作業単位内で複数の要求メッセージを送信できます。最初のメッセージ後のいつでも、MQCIH.UOWControl フィールドを MQCUOWC_COMMIT または MQCUOWC_BACKOUT に設定したメッセージを送信することにより、その作業単位を停止することができます。

最初のメッセージでは MQMD.CorrelId = MQCI_NEW_SESSION を指定し、後続のメッセージでは MQMD.CorrelId フィールドを最初のメッセージのメッセージ ID に設定する必要があります。CICS-MQ ブリッジを使用して複数のユーザー DPL プログラムを実行する場合、要求メッセージ内の MsgId フィールドはキュー・マネージャーによって設定され (ここで示す例では M1 に設定)、その後 CorrelId フィールドにコピーされます。

次の図は、CICS-MQ ブリッジを使用して実行する標準的な CICS DPL プログラムの場合に、MQMD および MQCIH 内の主なフィールドで使用されることになる値を要約したものです。

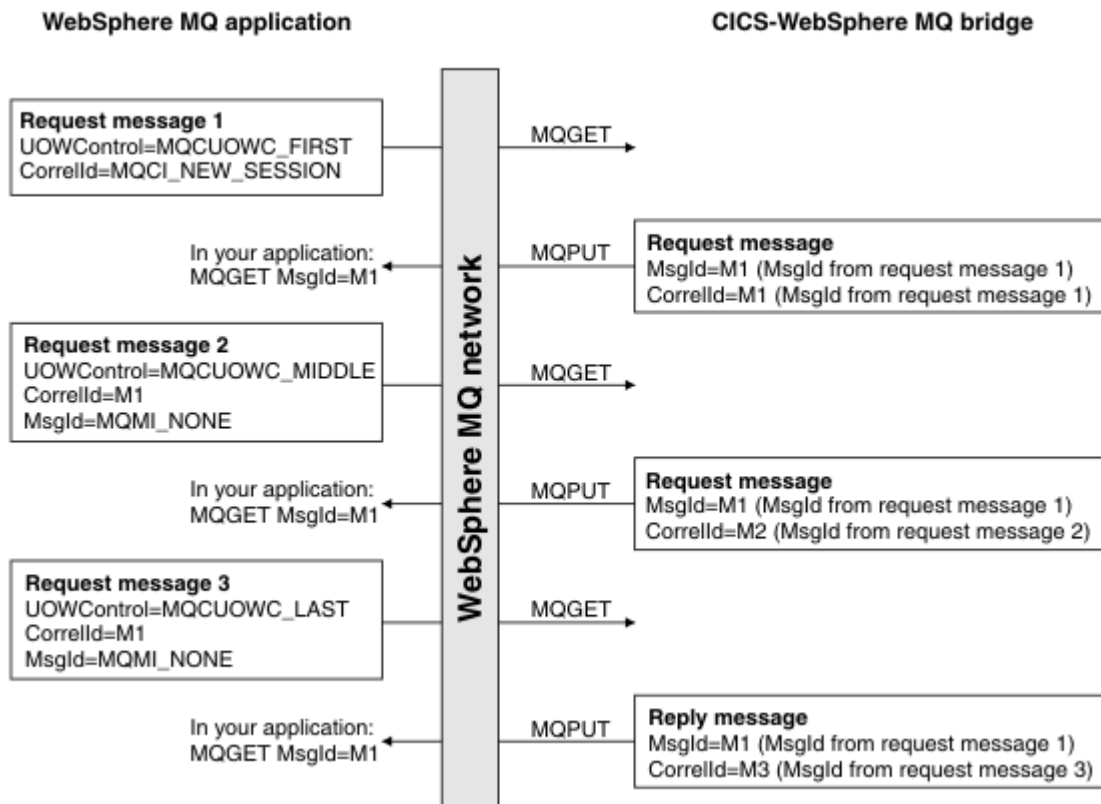


図 10. 1つの作業単位内に CICS ユーザー・プログラムが多数ある場合の主なフィールドの設定 (ブリッジの観点から見た場合)

3270 トランザクション用の MsgId、CorrelId、および UOWControl フィールド

一連の CICS 3270 トランザクションは通常、疑似会話で一体となるようにリンクされて完全なアプリケーションを形成します。非 CICS アプリケーションが CICS-MQ ブリッジを使用して 3270 トランザクションを実行する場合、MQCIH.UOWControl フィールドを使用することにより、トランザクションを作業単位にどのようにグループ化するかを制御できます。

注：MQCIH.UOWControl フィールドと MQMD.Correlid フィールドを設定すると同時に、MQCIH.Facility フィールドと MQCIH.RemoteSysId フィールドを使用して、シーケンスの最初のトランザクションによって返された Facility トークンと RemoteSysId 値を会話の後続メッセージに渡すことも重要です。詳しくは、90 ページの『3270 トランザクション要求メッセージ用の MQCIH フィールド』を参照してください。

複数の 3270 トランザクションを単一の CICS-MQ ブリッジ・セッションにグループ化するには、MQCIH.UOWControl フィールドの値を次のように設定します。

- ブリッジ・セッションの最初のメッセージでは、MQCUOWC_FIRST を使用します。
- ブリッジ・セッションの後続メッセージでは、追加データをトランザクションに提供する場合でも、新しいトランザクションを開始する場合でも、MQCUOWC_MIDDLE を使用します。
- セッション終了提案を示すには、MQCUOWC_LAST を使用します。94 ページの『セッションの終了』を参照してください。

MQMD.Correlid フィールドの値は、次のように設定します。

- ブリッジ・セッションの最初のメッセージでは、MQCI_NEW_SESSION を使用します。
- ブリッジ・セッションのすべての後続メッセージでは、トランザクションの最初のメッセージに対して生成されたメッセージ ID を使用します。

MQCIH.Facility フィールドの値は、次のように設定します。

- ブリッジ・セッションの最初のメッセージでは、MQCFAC_NONE を指定するとともに、MQCIH.FacilityKeepTime フィールドをゼロ以外の値に設定します。

- ブリッジ・セッションのすべての後続メッセージでは、最初の応答メッセージで返された機能トークンを使用します。

MQCIH.RemoteSysid フィールドの値は、次のように設定します。

- 要求を特定の CICS システム上で処理することが必要でない限り、ブリッジ・セッションの最初のメッセージではブランクを使用します。必要がある場合はその CICS システムを指定します。
- ブリッジ・セッションのすべての後続メッセージでは、最初の応答メッセージで返された値を使用します。

セッションの終了

MQCIH.UOWControl フィールドの値が MQCUOW_LAST に設定されてセッション終了提案が示されたら、アプリケーションは応答メッセージのメッセージ・タイプを検査して、さらに必要なアクションを決定する必要があります。

- Msgtype フィールドで示される応答メッセージ・タイプが MQMT_REPLY の場合は、CICS トランザクションがそれ以上データを要求せずに終了することを意味するので、セッションは終了します。
- 応答メッセージ・タイプが MQMT_REQUEST の場合は、CICS トランザクションがさらにデータを要求していることを意味するので、データを提供し、MQCIH.UOWControl フィールドを MQCUOWC_LAST と MQCUOWC_CONTINUE のいずれかに設定して次のメッセージを送信することができます。
- 応答メッセージ・タイプが MQMT_REQUEST であってもセッションを継続しない場合は、MQCIH.UOWControl フィールドを MQCUOWC_COMMIT に設定したメッセージを送信してセッションを終了します。

実行中の 3270 トランザクションを終了する必要がある場合は、MQCIH.CancelCode フィールドを 4 文字の異常終了コードに設定します。

トランザクションの間でセッションを終了する場合は、MQCIH.UOWControl フィールドを MQCUOWC_COMMIT に設定します。

次の図は、標準的な会話型 3270 トランザクションとのやり取りで MQMD および MQCIH の中の主なフィールドで使用され、予期される値を要約したものです。

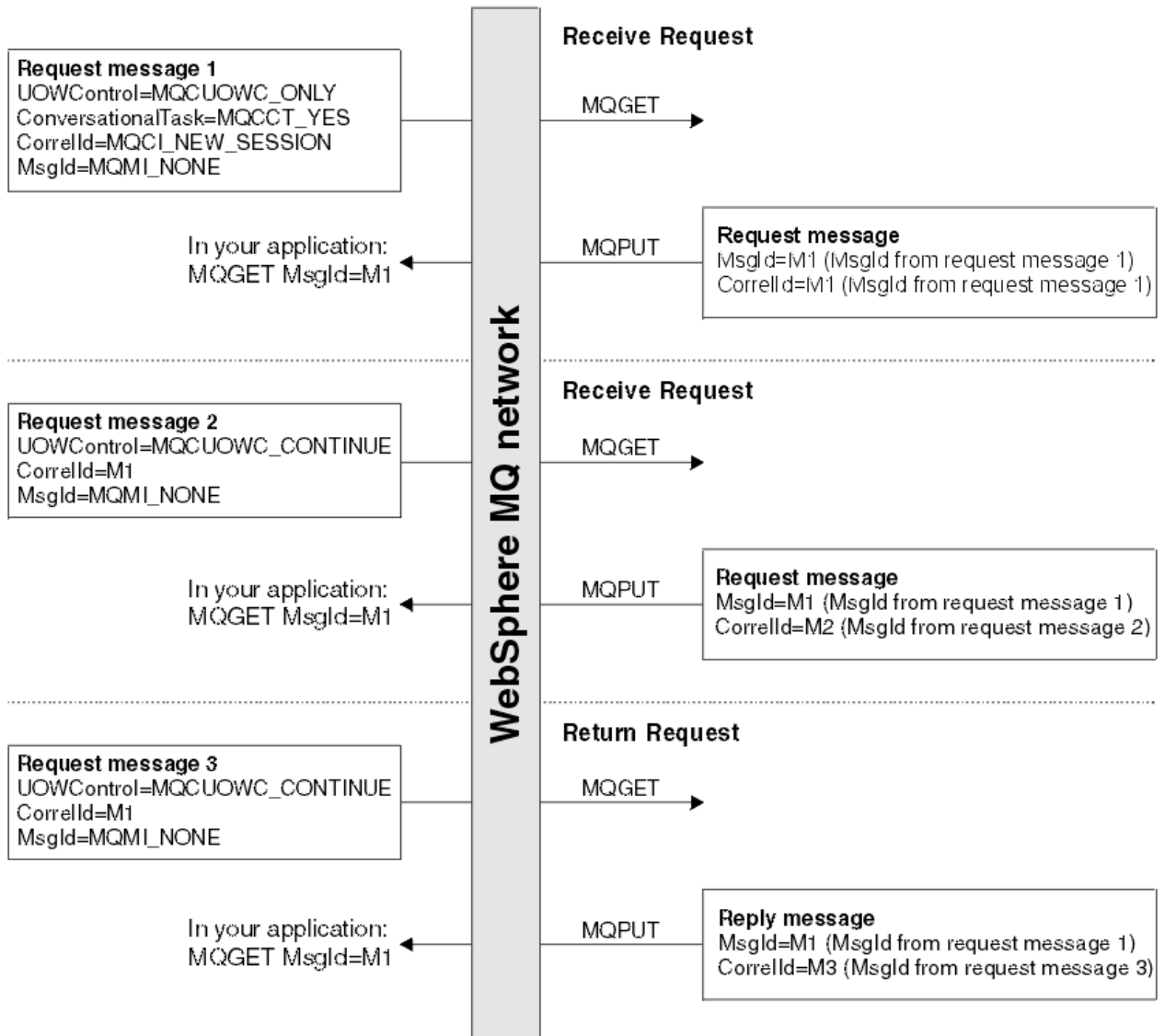


図 11. 主なフィールドの設定: ブリッジの観点から見た IBM MQ 会話型 3270 トランザクション

分散環境の CICS-MQ ブリッジのデータ変換

クライアント・アプリケーションがワークステーション上にある場合、CICS DPL プログラムおよび 3270 トランザクションを CICS-MQ ブリッジ経由で駆動できます。分散環境向けプログラミングにおける主な考慮事項として、ワークステーションと z/OS の互いに異なるエンコード・スキームと CCSID 値の間のデータ変換があります。

変換は 2 つの異なるルーチンによって行われます。1 つは MQCIH 構造体のためのルーチン、もう 1 つはメッセージで提供されるデータまたはベクトルのためのルーチンです。メッセージ内に MQCIH がある場合は、MQCIH の直前のヘッダーの形式フィールドで MQFMT_CICS を指定する必要があります。MQCIH の前にヘッダーがない場合は、MQMD.Format フィールドで MQFMT_CICS を指定する必要があります。データおよびベクトルの変換では、さらに考慮が必要となります。

COMMAREA データもコンテナー・データも受信したり返したりしない DPL プログラムを駆動する場合や、COMMAREA データまたはコンテナー・データが純粋に文字データである場合は、形式値として MQFMT_STRING を指定することによってデータ変換を行えます。COMMAREA データまたはコンテナー・データが純粋にバイナリーである場合、データ変換は不要なので、形式値として MQFMT_NONE を指定します。COMMAREA データまたはコンテナー・データが文字データとバイナリー・データの混合である場合

は、独自の変換ルーチンを作成して、適切な形式値を指定する必要があります。変換ルーチンの作成については、[IBM MQ 製品資料](#)を参照してください。

メッセージに MQCIH が含まれる場合は、MQCIH.Format で形式値を指定します。メッセージに MQCIH が含まれないが他のヘッダーが含まれる場合は、最後のヘッダーで形式値を指定します。メッセージにどのようなヘッダーも含まれない場合は、MQMD で形式値を指定します。

COMMAREA に応答データを生成する DPL プログラムを駆動しており、応答先キューを提供している場合には、その応答データの形式値を要求メッセージの MQCIH.ReplyToFormat で指定する必要があります。コンテナに応答データを生成する DPL プログラムを駆動する場合は、MQCIH.ReplyToFormat を指定する必要はありません。コンテナが自身を文字データまたはバイナリー・データと識別するからです。

3270 トランザクションを駆動する場合、SEND MAP ベクトルまたは RECEIVE MAP ベクトルを変換するには、以下のようにします。

- DFHMSD マクロで DSECT=ADSD を指定してマップをアセンブルします。オリジナルのマップ・セット定義がない場合は、CICS DFHBMSUP ユーティリティを使用してマップを再作成してください。
- フィールド MQCIH.ADSDescriptor で値 MQCADSD_SEND+MQCADSD_MSGFORMAT を指定します。ADSD または ADSD を使用して RECEIVE MAP ADS を作成する場合は、このフィールドに値 MQCADSD_REC を追加します。
- すべてのインバウンド・メッセージのフィールド MQCIH.Format で、値 DFHMQDCI または CSCQBDCI を指定します。入力形式が CSQCBDCI の場合、出力形式は CSQCBDCO になります。入力形式が DFHMQDCI の場合、出力形式は DFHMQDCO になります。
- z/OS とワークステーション間のチャンネルに対して CONVERT=YES が指定されていること、およびチャンネル・イニシエーターにデータ変換プログラム DFHMQDCI がインストールされていることを確認します。

ワークステーションとメインフレーム間のベクトルの形式の変換は、SEND MAP (アウトバウンド) と RECEIVE MAP (インバウンドとアウトバウンドの両方) 以外はサポートされていません。

3270 トランザクションからの応答メッセージを非 z/OS システムに送信する場合、送信側チャンネルは CONVERT(YES) オプションを指定する必要があります。

CICS-MQ ブリッジを使用した基本マッピング・サポート (BMS) アプリケーションの実行

CICS 3270 トランザクションが BMS マップを使用する場合、非 CICS アプリケーションはマップ・アセンブリ時に作成されたコピーブックを使用して、ベクトル内のデータの解釈に役立てることができます。コピーブックへのアクセス権限がない場合、アプリケーションはアプリケーション・データ構造 (ADS) 記述子を利用して、データを間接的に分析できます。

CICS 基本マッピング・サポート (BMS) は、CICS アプリケーションがさまざまな端末タイプをサポートできるようにするためのものです。アプリケーションが EXEC CICS SEND MAP コマンドを発行すると、BMS は端末固有の制御データとアプリケーション・データを組み合わせて、端末で表示可能な 3270 データ・ストリームを作成します。アプリケーションが EXEC CICS RECEIVE MAP コマンドを発行すると、インバウンド 3270 データ・ストリームからアプリケーション・データが取り出されてアプリケーションに返されます。

CICS アプリケーション用の BMS マップは、表示に必要なフィールドの特性を定義する一連の BMS マクロをアセンブルすることによって作成されます。マップ・アセンブリからの出力の 1 つは、表示フィールドを ADS にマップするコピーブックです。このコピーブックは、マップ内のフィールドをシンボリックにアドレッシングできるように、CICS アプリケーションのデータ定義の中に組み込む必要があります。SEND MAP 内にあって RECEIVE MAP が予期するアプリケーション・データは、コピーブック内の ADS に直接マップされます。

CICS ブリッジ環境でトランザクションが実行されると、EXEC CICS SEND MAP コマンドおよび EXEC CICS RECEIVE MAP コマンドによって、アウトバウンド・メッセージの中に SEND MAP ベクトルおよび RECEIVE MAP ベクトルが生成されます。これらのベクトルには、3270 データ・ストリームではなく、CICS アプリケーションがマップの中のフィールドをアドレッシングするために使用するものと等価の ADS が格納されます。

ADS の形式は、マップごとに固有です。形式は、マップ生成の一部として作成されるコピーブックによって記述されます。このコピーブックがなければ、データを解釈するのは困難です。通常、IBM MQ ブリッジを使用する非 CICS アプリケーションには、RECEIVE MAP データの作成と SEND MAP データの解釈ができるように、BMS コピーブックが組み込まれます。ただし、特定の BMS コピーブックを使用せずにアプリケーションを作成することもできます。データ形式は、ADS 記述子 (ADSD) という構造体によって記述されます。ADSD は SEND MAP ベクトルの最後に追加され、ベクトルの中の ADS の形式を記述します。ADSD の内容には、ADS の中のフィールドの名前、位置、および長さが含まれます。ADSD を RECEIVE MAP 要求で送信することもできます。この情報を会話型アプリケーションで使用して、CICS アプリケーションが要求する ADS の構造体を非 CICS アプリケーションに通知することができます。次に非 CICS アプリケーションは、この ADS を使用して RECEIVE MAP ベクトルを作成し、それを新しい要求として送信できます。

アプリケーション・プログラマーは、ADS、ADSD、または ADSDL (長い形式のアプリケーション・データ構造記述子) のどれを使用してメッセージの中のベクトル・データを解釈するかを選択できます。ADS を直接解釈するには、マップ・アセンブリーからのコピーブックを CICS-MQ ブリッジ・アプリケーションに組み込みます。組み込みたいがコピーブックまたはマップへのアクセス 権限がない場合は、CICS ユーティリティー DFHBMSP を使用してマップを再作成します。

例えば、すべてのマップを扱う汎用アプリケーションを作成する場合など、ADSD または ADSDL を使用して ADS を間接的に解釈するには、ブリッジ・アプリケーションにコピーブックを組み込む必要はありません。その代わりに、アウトバウンド SEND MAP 要求ベクトルおよび RECEIVE MAP 要求ベクトルに ADSD または ADSDL を必要に応じて組み込むよう通知する制御情報を、ブリッジに送信します。

アプリケーションを分散環境で実行しなければならない場合は、アウトバウンド SEND MAP ベクトルに ADSDL を組み込みます。こうすることで、IBM MQ はアウトバウンド・メッセージ内のデータを変換できるようになります。

インバウンド・メッセージの中の MQCIH.ADSDDescriptor フィールドで適切な値を設定することにより、次のいずれかのオプションを指定できます。

- SEND MAP ベクトルに ADSD (短い形式のアプリケーション・データ構造記述子) を組み込むには、次のように設定します。

```
MQCIH.ADSDDescriptor = MQCADSD_SEND
```

この設定の場合は、短い形式の ADS (アプリケーション・データ構造) も SEND MAP ベクトルに組み込まれます。

- RECEIVE MAP ベクトルに ADSD を組み込むには、次のように設定します。

```
MQCIH.ADSDDescriptor = MQCADSD_RECV
```

アウトバウンド RECEIVE MAP 要求ベクトルに ADS が存在することはありません。

- SEND MAP ベクトルまたは RECEIVE MAP ベクトルに ADSDL を組み込むには、次のように設定します。

```
MQCIH.ADSDDescriptor = MQCADSD_MSGFORMAT
```

この設定の場合は、長い形式の ADS も SEND MAP ベクトルに組み込まれます。

- SEND MAP ベクトルまたは RECEIVE MAP ベクトルに ADS 記述子を組み込まないようにするには、次のように設定します。

```
MQCIH.ADSDDescriptor = MQCADSD_NONE
```

この設定はデフォルトです。この設定の場合は、短い形式の ADS が SEND MAP ベクトルに組み込まれません。

次のように MQCADSD_* 値を同時に追加することにより、SEND MAP ベクトルと RECEIVE MAP ベクトルの両方に、長い形式のアプリケーション・データ構造記述子を組み込むことができます。

```
MQCIH.ADSDDescriptor = MQCADSD_SEND + MQCADSD_RECV + MQCADSD_MSGFORMAT
```

この場合、SEND MAP ベクトルには、長い形式の ADS も組み込まれます。

CICS-MQ ブリッジ用の SEND MAP ベクトルの解釈

アウトバウンド SEND MAP ベクトルには、アプリケーション・データ構造 (ADS) と、短い形式のアプリケーション・データ構造記述子 (ADSD) または長い形式のアプリケーション・データ構造記述子 (ADSDL) を入れることができます。

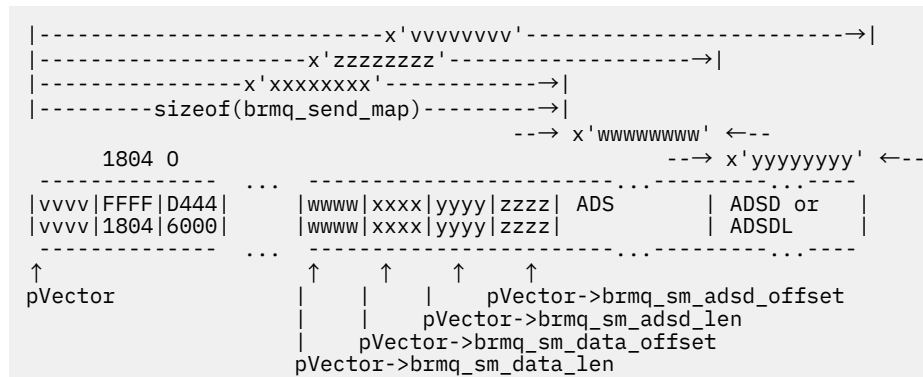
SEND MAP ベクトルを解釈するには、次のようにします (ADS と ADSD または ADSDL の両方がメッセージに入っているとします)。

1. SEND MAP ベクトルが入っているメッセージをブリッジ応答キューから取得して、それをメッセージ・バッファに入れます。
2. メッセージ・バッファの中でアウトバウンド SEND MAP ベクトルの開始位置を見つけます。このベクトルは MQCIH に追加されるので、メッセージ・バッファの最初から MQCIH の長さと同じオフセットの位置にあります。次のコード・フラグメントをモデルとして使用できます。

```
/* #includes */
#include cmqc.h /* IBM MQ header */
#include dfhbrmqh.h /* Vector structures */
:
/* #defines */
:
MQCHAR * MsgBuffer ; /* Message buffer pointer */
brmq_send_map * pVector ; /* Vector pointer */
:
/* Get message from reply queue */
:
/* Set the vector pointer to the start of the vector */
pVector = MsgBuffer + ((MQCIH *) MsgBuffer)->StrucLength ;
```

3. SEND MAP ベクトルから、アプリケーション・データ構造 (ADS) およびアプリケーション・データ構造記述子 (ADSD または ADSDL) の開始アドレスを識別します。

次の図は、アウトバウンド SEND MAP ベクトルの構造を示しています。ここでは、上記のコード・フラグメントのとおり、brmq_send_map ベクトルの開始位置をアドレッシングするための pVector というポインターが設定されているとします。



この図の中に、次のような値があります。

```
ABCD
1234
```

これは、hex on を指定した ISPF エディターで表示される 16 進値を示しています。この値は、16 進値 X'A1B2C3D4' と等価です。

フィールド pVector->brmq_sm_data_offset と pVector->brmq_sm_data_len は ADS のオフセットと長さ、フィールド pVector->brmq_sm_adsd_offset と pVector->brmq_sm_adsd_len は ADSD または ADSDL のオフセットと長さを表します。

フィールド brmq_sm_adsd_offset と brmq_sm_adsd_len は、メッセージに ADSD も ADSDL も組み込まれていない場合は、両方ともゼロに設定されます。

4. ADSD または ADSDL 中のフィールドを識別します。

ADSD と ADSDL は両方とも、CICS ライブラリー <hlq>.SDFHC370 で配布されるヘッダー・ファイル dfhbrarh.h で定義された構造にマップされます。その構造定義を調べることで、フィールドのレイアウトを知ることができます。ADSD のフィールドは、Link3270 ADS 記述子にも記載されています。

ブリッジ・アプリケーションをワークステーション上でコンパイルするには、ファイル `dfhbrarh.h` をその環境にコピーしてください。

ASDとASDLは両方とも、2つのタイプの構造体によって表されます。最初の構造体は記述子です。これは、ASDまたはASDLの先頭に1回だけ出現します。これらのタイプは、次のように定義されています。

ads descriptor

ADSD (短い形式) 用の記述子

ads long descriptor

ADSDL (長い形式) 用の記述子

2 番目の構造体はフィールド記述子です。マップ内のフィールドごとに 1 回ずつ繰り返されます。これらのタイプは、次のように定義されています。

ads field descriptor

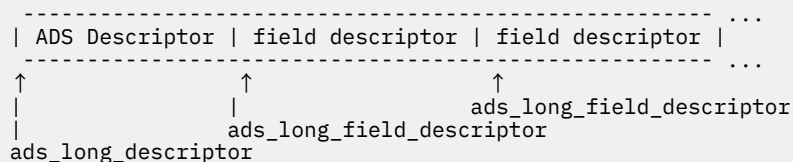
ADSD (短い形式) 用のフィールド記述子

ads_long_field_descriptor

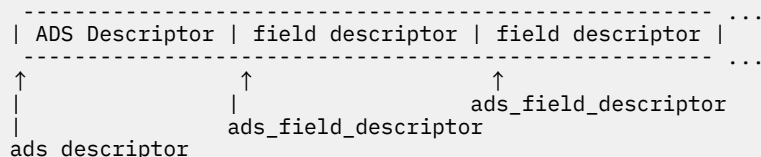
ADSDL (長い形式) 用のフィールド記述子

ADSDL と ADSD についてこの構造を図で示すと、次のようになります。

ADSDL:



ADSD:



記述子の中のフィールド `adsd_field_count` と `adsdl_field_count` は、ADSD および ADSDL の中のフィールド記述子の数を示します。

次のコード・フラグメントを使用して、ADSD または ADSDL 構造体の開始位置を指すポインタを設定し、フィールド記述子を順次処理できます。pVector は既に brmq_send_map ベクトルの開始位置をアドレッシングしており、インバウンド・メッセージからの CIH が入っている mqcih という名前の MOCIH 構造体があるとします。

```

/* #includes */
#include cmqc.h /* IBM MQ header */
#include dfhbrmqh.h /* Vector structures */
#include dfhbrarh.h /* ADSD structures */
:
/* Ptr to ADSD descriptor */
ads_descriptor * pADSD_D ;
/* Ptr to ADSDL descriptor */
ads_long_descriptor * pADSDL_D ;
/* Ptr to ADSD field descriptor */
ads_field_descriptor * pADSD_FD ;
/* Ptr to ADSDL field descriptor */
ads_long_field_descriptor * pADSDL_FD ;
:
/* Initialize the pointer to the ADSDL descriptor or the
/* ADSD descriptor depending on macih.ADSDescriptor */

```

```

if (mqcih.ADSDescriptor && MQCADSD_MSGFORMAT)
{
    pADSDL_D = pVector->brmq_sm_adsd_offset; /* Long form */
    pADSDL_FD = pADSDL_D + sizeof(ads_long_descriptor) ;
    /* Enter a loop where we process all field descriptors */
    /* in the ADSDL sequentially */

    do
    {
        /* Perform some processing */
        pADSDL_FD += sizeof(ads_long_field_descriptor) ;
    }
    while (pADSDL_FD < pADSDL_D->adsdl_length ) ;
}

else /* Short form */
{
    pADSD_D = pVector->brmq_sm_adsd_offset; /* Short form */
    pADSD_FD = pADSD_D + sizeof(ads_descriptor) ;
    /* Enter a loop where we process all field descriptors */
    /* in the ADSD sequentially */

    do
    {
        /* Perform some processing */
        pADSD_FD += sizeof(ads_field_descriptor) ;
    }
    while (pADSD_FD < pADSD_D->adsd_length ) ;
}

```

5. ADS 中のフィールドを識別します。

ADS は、マップをアセンブルするときに生成される構造体にマップされます。マップ・セット定義マクロに DSECT=ADSDL という keyword=parameter 値を組み込むと、ADS は長い形式になります。マップ・アセンブリからの出力は、入力構造体と出力構造体の 2 つの構造体の共用体です。次の例は、そのような共用体の一部を示しています (各構造体の最初のフィールド定義だけが示されており、マップ・アセンブリの後にコメントが追加されています)。

```

union
{
    struct {
        char    dfhms1[12]; /* 12 reserved bytes */
        int     dfhms2;      /* Offset to next field */
        int     tranidl;     /* Data length of this field */
        int     tranidf;     /* Flag or attribute value */
        int     dfhms3[7];   /* Extended attributes array */
        char    tranidi[4];  /* Data value of field */
        ...
    } bmstmp1i; /* Input structure */

    struct {
        char    dfhms56[12]; /* 12 reserved bytes */
        int     dfhms57;      /* Offset to next field */
        int     dfhms58;      /* Data length of this field */
        int     tranida;      /* Flag or attribute value */
        int     tranidc;      /* Extended attribute */
        int     tranidp;      /* Extended attribute */
        int     tranidh;      /* Extended attribute */
        int     tranidv;      /* Extended attribute */
        int     tranidu;      /* Extended attribute */
        int     tranidm;      /* Extended attribute */
        int     tranidt;      /* Extended attribute */
        char    tranido[4];   /* Data value of field */
        ...
    } bmstmp1o; /* Output structure */
} bmstmp1; /* Union */

```

この 2 つの構造体は機能的にまったく同じですが、入力構造体には要素が 7 つの配列に拡張属性値が含まれており、出力構造体には個別の名前付きフィールドがあります。

次のコード・フラグメントを使用して、ADS の開始位置を指すポインターを設定できます。DSECT の例で示されている構造体名を、例として使用しています。2つのポインターが設定されます。最初のポインターはインバウンド・データをアドレッシングするためのもの、2番目のポインターはアウトバウンド・データをアドレッシングするためのものです。pVector は、既に brmq_send_map ベクトルの開始位置をアドレッシングしているものとしします。

```

/* #includes */
#include cmqc.h /* IBM MQ header */
#include dfhbrmqh.h /* Vector structures */
#include dfhbrarh.h .. /* ADSD structures */
#include mydsect.h /* DSECT from map assembly */
:
bmstmp1i * pADSI ; /* Pointer to the inbound ADS */
bmstmp1o * pADSO ; /* Pointer to the outbound ADS */
bmstmp1i * pADSI_An ; /* Inbound ADS Anchor */
bmstmp1o * pADSO_An ; /* Outbound ADS Anchor */
:
/* We are dealing with an outbound vector, so we will */
/* initialize the outbound pointer to address the ADS */

pADSO = pVector->brmq_sm_adsd_offset ;

/* Save initial value as anchor */

pADSO_An = pADSO ;

/* Move to the start of the first field */

pADSO += pADSDL_FD->adsdl_field_offset ;

/* Enter a loop where we process all fields in the ADS */
/* sequentially. It is assumed that the value of pADSDL_FD */
/* is being augmented to the next field descriptor in the */
/* ADSDL with every loop. A model for this is shown in the */
/* previous code fragment. Note that adsdl_field_offset */
/* contains the absolute offset of the field from the start */
/* of the ADS. */

do
{
    /* Perform some processing */
    :
    /* Add offset of next field to ADS Anchor value */
    /* to address the next field */

    pADSO = pADSO_An + pADSDL_FD->adsdl_field_offset ;
}
while (pADSDL_FD < pADSDL_D->adsd_length ) ;
:

```

CICS-MQ ブリッジ用の RECEIVE MAP ベクトルの解釈

RECEIVE MAP 要求は、次の入力メッセージで RECEIVE MAP を提供するようにクライアントに求める要求です。

SEND MAP ベクトルとは異なり、アウトバウンド RECEIVE MAP 要求ベクトルに ADS が格納されることはありません。このベクトルには、MQCIH.ADSDescriptor の中で MQCADSD_RECV が指定されていれば、次のインバウンド RECEIVE MAP ベクトルで必要となる ADS データを記述した ADSD または ADSDL が格納されます。RECEIVE MAP ベクトルの構造は、SEND MAP ベクトルの構造とは異なります。主な違いは、ADS のオフセットと長さを示すフィールドがないという点です。

RECEIVE MAP ベクトルを解釈するには、次のようにします (ADSD または ADSDL がメッセージに格納されているとします)。

1. RECEIVE MAP 要求ベクトルが格納されるメッセージをブリッジ応答キューから取得して、それをメッセージ・バッファに入れます。
2. メッセージ・バッファの中でアウトバウンド RECEIVE MAP ベクトルの開始位置を見つけます。このベクトルは MQCIH に追加されるので、メッセージ・バッファの最初から CIH の長さと同じオフセットの位置にあります。次のコード・フラグメントをモデルとして使用できます。

```

/* #includes */
#include cmqc.h /* IBM MQ header */

```

```

#include dfhbrmqh.h          /* Vector structures          */
/* #defines                  */
MQCHAR * MsgBuffer ;        /* Message buffer pointer */
brmq_receive_map_request * pVector ; /* Vector pointer */
/* Get message from reply queue */
/* Set the vector pointer to the start of the vector */
pVector = MsgBuffer + ((MQCIH *) MsgBuffer)->StrucLength ;

```

3. RECEIVE MAP ベクトルから開始アドレス ADSD または ADSDL を識別します。

次の図は、アウトバウンド RECEIVE MAP 要求ベクトルの構造を示しています。この図では、上記のコード・フラグメントのとおり、brmq_receive_map_request ベクトルの開始位置をアドレッシングするための pVector というポインターが設定されているとします。

```

|-----x'vvvvvvvv'----->|
sizeof(brmq_receive_map_request)
|----->|

      1802 0          --> x'wwwwwww' <--
-----
|vvvv|FFFF|D444| ... |www| ADSD or |
|vvvv|1802|6000| ... |www| ADSDL  |
-----
↑                ↑
pVector          pVector->brmq_rmr_adsd_len

```

この図の中に、次のような値があります。

```

ABCD
1234

```

これは、hex on を指定した ISPF エディターで表示される 16 進値を示しています。この値は、16 進値 X'A1B2C3D4' と等価です。

フィールド pVector->brmq_rmr_adsd_len は、ADSD または ADSDL の長さを表します。ADSDL は brmq_receive_map_request ベクトルに直接追加されているので、オフセットはありません。

4. ADSD または ADSDL の中のフィールドを識別します。

これを行うには、98 ページの『CICS-MQ ブリッジ用の SEND MAP ベクトルの解釈』で説明した SEND MAP ベクトルの場合と総じて同じようにします。ただし、次のコード・フラグメントを使用して、ADSD または ADSDL の開始位置を指すポインターを設定します。

```

:
if (mqcih.ADSDDescriptor && MQCADSD_MSGFORMAT)
{
    pADSDL_D = pVector + sizeof(brmq_receive_map_request) ;
    :
}
else
/* Short form */
{
    pADSDL_D = pVector + sizeof(brmq_receive_map_request) ;
    :
}
:

```

RECEIVE MAP ベクトルの中の ADSD または ADSDL の構造は、SEND MAP ベクトルの中の場合とまったく同じです。したがって、ADSD または ADSDL の開始アドレスを特定した後は、SEND MAP ベクトルについての説明のとおりに進むことができます。

ADSDL と ADS の例

ADSDL と ADS の一部を示す例を説明します。

図の中に次のような値があります。

Offset to next field	Start of next field
12 bytes reserved	Value of field

ここで示されている値の意味は、次のとおりです。

12 bytes reserved

短い形式でも長い形式でもすべての ADS の先頭にある予約されたスペース。

Offset to next field

現在のフィールドの情報の長さは、このフルワード長さ値の開始位置から 0x2C バイトです。

Value of field

ADSDL の中で名前が TRANID として識別されるフィールドの値は BAAA です。長い形式の ADS の場合のデータのオフセットは、フィールドの開始位置から常に 0x28 バイトです。

Start of next field

ADS の中の次のフィールドの情報の開始位置。

この例では、フィールド情報はフルワードの整数倍です。情報が整数倍でない場合は、次のフィールドがフルワード境界から始まるように、データ値と次のフィールドの間に埋め込みバイトが追加されます。埋め込みバイトは *offset to next field* 値に含められます。

フィールドのいくつかの属性値と拡張属性値はここで示されていませんが、それらは次のフィールドへのオフセットを表すフルワードとフィールド値そのものの間にあります。

CICS-MQ ブリッジで実行される 3270 トランザクションの例

この例では、CICS 3270 トランザクションが 3270 端末とやり取りするときに生じるデータ・フローと、CICS-MQ ブリッジ・アプリケーションとやり取りするときに生じるデータ・フローの違いを中心に説明します。

この例では、トランザクションの ID は BAAA です。トランザクションは BMS マップを使用します。このマップは、トランザクションを CICS-MQ ブリッジ環境で実行されるように適応させるためのものです。

CICS 環境では、CICS 3270 端末でトランザクションの名前を入力して Enter キーを押すことによって、トランザクションを開始します。疑似会話でトランザクションが最初に呼び出されると、トランザクションのロジックによって EXEC CICS SEND MAP コマンドが発行され、その後コマンド EXEC CICS RETURN TRANSID(BAAA) が発行されてトランザクションが終了します。

ユーザーは、端末に表示されるマップのフィールドに値を入力した後、AID キーを押します。トランザクションに対して 2 回目の呼び出しが行われると、トランザクションのロジックによって EXEC CICS RECEIVE MAP コマンドが発行され、トランザクションがマップを受け取ります。トランザクションは、自身のアプリケーション・データ構造の中の値を変更することによってマップ内の特定のフィールドを更新した後、EXEC CICS SEND MAP コマンドを発行してユーザーの端末にマップを再表示します。

これによりユーザーは、再表示されたマップの中のフィールドを更新して、RECEIVE MAP - SEND MAP サイクルを再開できます。このロジックを図で示すと、次のようになります (EC は EXEC CICS を表します)。

Terminal user		3270 Transaction
BAAA <ENTER>	----->	<Initial start>
	<-----	<business logic>
		EC SEND MAP FROM(ads)
		EC RETURN TRANSID(BAAA)
Update fields	----->	EC RECEIVE MAP INTO(ads)
<ENTER>	<-----	<business logic>
		EC SEND MAP
		EC RETURN TRANSID(BAAA)
Update fields	----->	EC RECEIVE MAP
<ENTER>		
⋮		

このトランザクションを CICS-MQ ブリッジを使用して実行すると、物理端末は非 CICS アプリケーションに置き換えられます。3270 トランザクションのロジックは変わらず、3270 トランザクションが受け取るアプリケーション・データも同じですが、流れるデータとデータの伝送方法が異なります。MQCIH 構造体、ベクトル構造体、および必要に応じてアプリケーション・データ構造が表現されたものを格納する IBM MQ メッセージが、3270 データ・ストリームの代わりに使用されます。

105 ページの『最適化なしの厳密なエミュレーション』は、上記のスキームが厳密にエミュレートされたときに生じるフローを示しています。106 ページの『最適化を伴う改良されたエミュレーション』に示すように、インバウンド・メッセージに複数のベクトルを組み込むことによって、フローを最適化できます。

MOCADSD SEND + MOCADSD RECV + MOCADSD MSGFORMAT

分かりやすくするために、ここではメッセージングの詳細は省略しています。CICS-MQブリッジが使用するキューイング・モデルの説明については、**IBM MQ 製品資料**を参照してください。

この例は、CICS-MQブリッジを使用して3270トランザクション例のオリジナル・ロジックを厳密にエミュレートする場合のデータ・フローを示しています。

1. アプリケーションからの初期フローには、MQCIH だけが入っています。MQCIH には、開始するトランザクションを指定する制御情報が組み込まれています。
2. 3270 トランザクションからの戻りフローには、MQCIH が入っています。この MQCIH には、後続のすべてのフローで使用される機能トークンと、エラーが発生した場合には診断情報が含まれています。また、マップそのものに関連する制御情報とマップを表すデータを格納する SEND MAP ベクトル構造体も含まれます。開始側アプリケーションが要求した場合は、アプリケーション・データ構造記述子も組み込まれます。
3. ブリッジ・アプリケーションは、トランザクションをもう一度開始するための制御情報が入った MQCIH だけを格納したメッセージを送り返します。
4. 3270 トランザクションは、レガシー環境の場合と同様に、EXEC CICS RECEIVE MAP コマンドを発行します。ただし、ブリッジ環境では、マップ・データを即時に入手することはできません。呼び出しは、

アウトバウンド RECEIVE MAP 要求ベクトルを格納したメッセージに変換されます。メッセージには、アプリケーション・データ構造記述子も含まれます。この例では、ブリッジ・アプリケーションからメッセージが戻ってくるまでの間、トランザクションは待機します。このモデルは、レガシー環境の場合とは少し異なります。しかし、ブリッジ・アーキテクチャーでは、メッセージに複数のベクトルを格納することができるので、1つのインバウンド・メッセージで複数の要求に応えることができます。

5. アプリケーション・データ構造の中のフィールドを更新した後、ブリッジ・アプリケーションは、アウトバウンド要求に応えるために、インバウンド RECEIVE MAP 応答ベクトルを送信します。
6. 3270 トランザクションは、EXEC CICS SEND MAP (SEND MAP ベクトルに変換される) を発行し、サイクルが繰り返されます。

キャプチャーされるフローを調べるとき、そのようなトランザクションがパススルー・ツール (CICS SupportPac CA1E CICS 3270 ブリッジ・パススルー) を使用して実行される場合には、用意されているオンライン資料を活用して構造を識別することができます。

最適化を伴う改良されたエミュレーション

この例は、CICS-MQ ブリッジを使用した 3270 トランザクション例の最適化されたデータ・フローを示しています。

CICS-MQ Bridge Application	3270 Transaction
MQPUT to Bridge RequestQ	--MQCIH-----→ <Initial start> <business logic>
MQGET from Bridge ReplyQ	←--MQCIH+b1mq_send_map+ADS+ADSDL----- EC SEND MAP FROM(ads) EC RETURN TRANSID(BAAA)
MQPUT to Bridge RequestQ	----MQCIH+b1mq_receive_map+ADS-----→ <Start> EC RECEIVE MAP INTO(ads) <business logic>
MQGET from Bridge ReplyQ	←--MQCIH+b1mq_send_map+ADS+ADSDL----- EC SEND MAP FROM(ads) EC RETURN TRANSID(BAAA) ⋮

このシーケンスを 105 ページの『最適化なしの厳密なエミュレーション』に示されている最適化されていないフローと比べると、CICS トランザクションは RECEIVE MAP 要求ベクトルを送信する必要がないことが分かります。インバウンド RECEIVE MAP ベクトルは要求を既に予期していて、インバウンド・マップがトランザクションで既に使用できるようになっているからです。

第 8 章 CICS-MQ アダプターを使用するためのアプリケーションの開発

CICS-MQ アダプターは、CICS アプリケーション・プログラムによって使用される IBM MQ Message Queue Interface (MQI) を実装します。また、このアダプターは、CICS-MQ API 交差出口をサポートします。この出口を使用して、MQI 呼び出しの実行中に、モニター、テスト、保守、またはセキュリティ目的で、その呼び出しを代行受信することができます。

トランザクションの整合性の点では、このアダプターは CICS リカバリー・マネージャーの制御下で同期点をサポートするので、必要に応じて作業単位をコミットしたりバックアウトしたりできます。また、アダプターでは、Security Server (以前の RACF) など適切なセキュリティ管理製品とともに使用すると、IBM MQ リソースのセキュリティ検査もサポートされます。このアダプターにより、キュー・マネージャー終了後の自動再接続の高可用性と、再始動後の自動リソース再同期の高可用性が提供されます。さらに、キュー・マネージャーのシャットダウンなどの、スケジュールされていないイベントに応答するアラート・モニター機能も用意されています。

IBM MQ MQI 呼び出しにアクセスするための API スタブ・プログラム

IBM MQ にアクセスするために、CICS の下で実行するアプリケーション・プログラムは、動的呼び出しを使用していない限り、提供されている API スタブ・プログラム (CSQCSTUB) とリンク・エディットする必要があります。CSQCSTUB は、すべての MQI 呼び出しに対するアクセス権をアプリケーションに付与します。

CICS スタブの動的呼び出しについては、[IBM MQ 製品資料内の『アプリケーションの開発』](#)を参照してください。

CSQCSTUB は、SDFHLOAD ロード・ライブラリーのモジュール DFHMQSTB として、また SDFHAUTH ロード・ライブラリーの DFHMQSTX としてシップされます。両方のモジュールは、以下の別名で定義されます。

CSQCBFMH
CSQCCB
CSQCCLOS
CSQCCONN
CSQCCONX
CSQCCTL
CSQCCTMH
CSQCDISC
CSQCDTMH
CSQCDTMP
CSQCINQ
CSQCIQMP
CSQCMHBF
CSQCOPEN
CSQCGET
CSQCPUT
CSQCPUT1
CSQCSET
CSQCSTAT
CSQCSTMP
CSQCSTUB
CSQCSUB
CSQCSUBR

別名 CSQBFMH、CSQCCB、CSQCCTL、CSQCCTMH、CSQCDTMH、CSQCDTMP、CSQCIQMP、CSQCMHBF、CSQCSTAT、CSQCSTMP、CSQCSUB、および CSQCSUBR は、CICS がバージョン 7 以降の IBM MQ に接続されているときにのみ使用されます。

既存の CICS-MQ アプリケーションは、CICS 付属の CICS-MQ アダプターにより、変更せずに実行できます。それらのアプリケーションを再度コンパイルする必要はなく、再度リンク・エディットする必要もありません。新しいアプリケーションまたは変更したアプリケーションの場合、既存のリンク・エディット手順を使用できます。また、アプリケーションがバージョン 7 の IBM MQ に追加された新しい API 呼び出しを使用しない限り、CICS または IBM MQ に付属のスタブを使用できます。バージョン 7 の API 呼び出しは、IBM MQ に付属のスタブではなく CICS に付属のスタブを使用する場合にのみ、CICS でサポートされます。新しい API 呼び出しは、MQBUFMH、MQCB、MQCTL、MQCRTMH、MQDLTMH、MQDLTMP、MQINQMP、MQMHBUF、MQSETMP、MQSTAT、MQSUB、および MQSUBRQ です。

非同期メッセージ・コンシュームとコールバック・ルーチン

IBM MQ のバージョン 7 以降で使用可能な非同期メッセージ・コンシュームは、MQCB および MQCTL MQI 呼び出しを使用します。これらの機能を CICS アプリケーション・プログラムで使用する場合、CICS は処理のいくつかの面を実行します。そのため、アプリケーションをコーディングするときに、IBM MQ のプログラミング資料に加え、以下に示す情報を使用する必要があります。

IBM MQ 非同期メッセージ・コンシュームを使用するため、キューやトピックを含む複数のメッセージ宛先にコールバック・ルーチンと呼ばれるプログラムを登録できます。適切なメッセージが宛先に送信されると、それはコールバック・ルーチンに渡されます。キュー・マネージャーの静止などの状態を通知するイベント・ハンドラーをセットアップすることもできます。

CICS コマンド・レベル・アプリケーション・プログラムをコールバック・ルーチンとして登録するとき、CICS は IBM MQ によって渡されるメッセージを取得し、EXEC CICS LINK を使用してコールバック・ルーチンにリンクします。CICS は、チャンネル上にある一連のコンテナのコールバック・ルーチンにメッセージ・データを渡します。

コールバック・ルーチンとして使用するプログラムは、以下の要件を満たす必要があります。

- コールバック・ルーチンでは、CICS によってサポートされるすべてのプログラミング言語を使用できますが、コールバック・ルーチンに Java を使用すると、Java 環境と非 Java 環境の間で過度の TCB 切り替えが発生し、パフォーマンスに影響を与えることに注意してください。
- RENT オプション、および AMODE(31) と RMODE(ANY) のオプションを指定してコールバック・ルーチンをコンパイルおよびリンク・エディットします。
- スレッド・セーフ標準に適合するようにコールバック・ルーチンをコーディングします。また、属性 CONCURRENCY(THREADSAFE) および API(CICSAPI) を指定して CICS にコールバック・ルーチンを定義し、TCB 切り替えの回数を最適化します。API(OPENAPI) を使用すると、ストレージ保護がコールバック・ルーチンに対してアクティブな場合、過度の TCB 切り替えが発生するため、パフォーマンスに影響を与えます。

CONCURRENCY(REQUIRED) および API(CICSAPI) を使用してプログラムを定義すると、スレッド・セーフ標準に適合するようにコーディングされたプログラムが、オープン TCB のプログラムの開始から実行できるようになります。L8 オープン TCB は、実行キーに関係なく使用されます。

- リモート・プログラムまたは動的プログラムとしてではなく、ローカル・プログラムとして、コールバック・ルーチンを定義します。CICS がコンテナに渡すデータ構造には、領域内のローカル・ストレージへのポインターが含まれるため、リモート・プログラムおよび動的プログラムは、コールバック・ルーチンとしてサポートされません。リモートとして定義されたコールバック・ルーチンを登録しようとすると、理由コード MQRC_MODULE_NOT_FOUND で失敗します。

コールバック・ルーチン用の CICS コンテナ

CICS がコールバック・ルーチンに渡すコンテナは、MQ_ASYNC_CONSUME という名前のチャンネルを使用します。チャンネルの名前およびコンテナの名前は、すべて 16 文字の長さです。そのため、ここで使用されるコンテナの名前には、ブランクが埋め込まれます。コンテナは以下のとおりです。

表 10. コールバック・ルーチンに渡されるメッセージ・データ・コンテナ	
コンテナ名	コンテンツ
MsgDesc	MQMD2 - メッセージ記述子バージョン 2
GetMsgOpts	MQGMO - Get メッセージ・オプション
Buffer	メッセージ。切り捨てや変換など MQGMO で指定されるオプションには、MQGET 呼び出しで実行するのと同じメッセージへの効果があります。
Context	MQCBC - コールバック・コンテキスト・バージョン 2

IBM MQ 製品資料には、MQMD、MQGMO、および MQCBC のデータ構造について説明されています。[IBM MQ 製品資料](#)を参照してください。

コールバック・ルーチンでデータをメッセージ・データ・コンテナから取得する際は、将来的にデータ構造のサイズを変更できるように、GET CONTAINER INTO コマンドではなく、GET CONTAINER SET コマンドを使用します。空のコンテナがないかどうかをチェックするには、FLENGTH キーワードをコーディングして、その値がゼロになっていないか調べます。例えば、コールバック・ルーチンのイベントの呼び出しでは、MsgDesc、GetMsgOpts、および Buffer のコンテナが空です。

コールバック・ルーチンでの同期点、異常終了、および静止の処理

コールバック・ルーチンが CICS-MQ アダプターと正常に対話するようにするため、以下の要件が満たされていることを確認します。

1. メッセージ・コンシューマー・コールバック・ルーチンがメッセージに対して開始される場合、そのコールバック・ルーチンのみで EXEC CICS SYNCPOINT および EXEC CICS SYNCPOINT ROLLBACK コマンドを使用します。これらのコマンドは、イベント・ハンドラーでは使用しないでください。また、登録、登録解除、開始、または停止のためにコールバック・ルーチンが呼び出される際にも使用しないようにしてください。
2. EXEC CICS ABEND コマンドをコールバック・ルーチンで使用する場合、異常終了が処理され、制御が呼び出し元に戻されるため、MQCTL コマンドは理由コード MQRC_CALLBACK_ERROR(2452) で失敗します。1つのコールバック・ルーチンでの 1 回の異常終了で、MQCTL 要求を発行した CICS タスクによって管理されているすべてのコールバック・ルーチンの呼び出しが停止します。
3. イベント・ハンドラー・コールバック・ルーチンでは、キュー・マネージャーまたは接続が静止するとき、CICS-MQ インターフェースが正常に静止できるように、MQCTL STOP コマンドを発行します。このコマンドを発行しない場合、アクティブなイベント・ハンドラーのために、CICS-MQ インターフェースは静止できません。CICS-MQ インターフェースを強制的に閉じる必要があります。
4. MQCB を使用してコールバック・ルーチンを登録する場合、以下のようになります。
 - a. CallBackDesc (MQCBD) パラメーターで、オプション・フィールドに MQCBDO_FAIL_IF QUIESCING を設定しません。新しい作業の到着を停止するため、CICS で MQCB コマンドを処理の一部として発行する必要があります。
 - b. コールバック・ルーチンが正常に静止するようにするため、GetMsgOpts (MQGMO) パラメーターで、オプション・フィールドに MQGMO_FAIL_IF QUIESCING を設定します。この静止オプションを正しく設定しない場合、アクティブなイベント・ハンドラーのために、CICS-MQ インターフェースは静止できません。CICS-MQ インターフェースは強制的に閉じる必要があります。
 - c. CallBackDesc (MQCBD) パラメーターで、CallbackArea フィールドを使用してストレージ域をコールバック・ルーチンに渡す場合、CallbackArea フィールドはプログラムの動的ストレージではなく、EXEC CICS GETMAIN を使用して取得したストレージのアドレスに設定します。コールバック・ルーチンは、コールバック・コンテキスト構造 MQCBC の CallbackArea フィールドを使用して、このストレージにアクセスします。また、プログラム終了時に、ストレージまたはそのストレージに固定されているストレージ域を解放するのではなく、タスクの終了時に CICS がストレージを FREEMAIN するようにしてください。これにより、CICS がタスク処理の終了時に登録解除のためにコールバック・ルーチンを呼び出すと、ストレージが使用可能になります。

5. MQCTL を使用してコールバック・ルーチンの呼び出しを制御するとき、ControlOpts (MQCTLO) パラメーターで、ConnectionArea フィールドを使用してすべてのコールバック・ルーチンで共用されるストレージ域を渡す場合、ConnectionArea フィールドはプログラムの動的ストレージではなく、EXEC CICS GETMAIN を使用して取得したストレージのアドレスに設定します。コールバック・ルーチンは、コールバック・コンテキスト構造 MQCBC の ConnectionArea フィールドを使用して、このストレージにアクセスします。また、プログラム終了時に、ストレージまたはそのストレージに固定されているストレージ域を解放するのではなく、タスクの終了時に CICS がストレージを FREEMAIN するようにしてください。これによって、CICS がタスク処理の終了時にコールバック・ルーチンを呼び出して登録解除する場合、ストレージを使用可能にすることができます。

非同期メッセージ・コンシューム用のサンプル・プログラム

COBOL でのサンプル・プログラムのこのセットでは、IBM MQ 非同期メッセージ・コンシュームおよびコールバック・ルーチンが CICS 環境でどのように機能するかを示します。IBM MQ 非同期メッセージ・コンシュームを使用するには、バージョン 7 以降の IBM MQ が必要です。サンプル・プログラムは、IBM MQ で提供されます。

サンプルの登録クライアント・プログラムを実行して 3 つのコールバック・ルーチン (1 つのイベント・ハンドラーと 2 つのメッセージ・コンシューマー) を登録し、非同期メッセージ・コンシュームを開始します。その後、2 つのメッセージ・コンシューマーによる非同期コンシュームのために、サンプルのメッセージング・クライアント・プログラムを使用して CICS 端末から IBM MQ のキューとトピックにメッセージを送信できます。これらのメッセージを使用して、いずれかのメッセージ・コンシューマーにコールバック・ルーチンの登録解除、再開、または中断を指示するか、あるいは非同期メッセージ・コンシュームを停止させます。

表 11. 非同期メッセージ・コンシューム用のサンプル COBOL プログラム		
サンプル・プログラム	機能	アクション
CSQ4CVRG	登録クライアント	イベント・ハンドラーおよびメッセージ・コンシューマーを登録します。CICS トランザクション MVRG の下で実行します。
CSQ4CVPT	メッセージング・クライアント	メッセージをトピックに公開するか、または制御メッセージをキューに送信します。CICS トランザクション MVMP の下で実行します。
CSQ4CVCN	基本メッセージ用のメッセージ・コンシューマー	トピック News/Media/Movies で公開されている IBM MQ メッセージをコンシュームします。
CSQ4CVCT	制御メッセージ用のメッセージ・コンシューマー	SAMPLE.CONTROL.QUEUE キューからの制御メッセージをコンシュームし、該当する MQCB または MQCTL コマンドを発行して、非同期メッセージ・コンシュームを停止するか、またはコールバック・ルーチンを登録解除、再開、または中断します。
CSQ4CVEV	イベント・ハンドラー	キュー・マネージャーの停止または静止など、コールバック環境全体に影響する条件が発生したときに、IBM MQ から通知を受け取ります。

非同期メッセージ・コンシューム用のサンプルの設計

サンプル COBOL プログラムには、CICS トランザクションの下で実行する登録クライアントとメッセージング・クライアント、および CICS-MQ からのメッセージやイベントをコンシュームする 3 つのコールバック・ルーチンが含まれます。それぞれのサンプル・プログラムでは、COBOL DISPLAY ステートメントを使用することにより適切なタイミングでメッセージを表示して、その動作を観察できるようにします。

サンプル・プログラムからのメッセージは、CEEMSG データ・セットである一時データ・キュー CESE に送信されます。イベント・ハンドラーおよびメッセージ・コンシューマーによって発行されるメッセージには、START や REGISTER など、プログラムを駆動する CICS-MQ の呼び出しタイプが示されます。

登録クライアント・プログラム CSQ4CVRG

登録クライアントは、CICS 端末から CICS トランザクション MVRG の下で開始されますが、入力は何も取得しません。登録クライアントは、イベント・ハンドラー CSQ4CVEV、メッセージ・コンシューマー CSQ4CVCN、およびメッセージ・コンシューマー CSQ4CVCT を IBM MQ を使用するコールバック・ルーチンとして登録します。また、CSQ4CVCT に登録された 3 つのコールバック・ルーチンすべての名前と 2 つのメッセージ・コンシューマーに関連付けられたオブジェクト処理を含むデータ構造を渡します。

登録クライアントは、コールバック・ルーチンを登録したときに、MQCTL START_WAIT コマンドを発行して非同期メッセージ・コンシュームを開始します。その後、制御が戻るまで、中断します。

コールバック・ルーチンの 1 つが MQCTL STOP コマンドを発行して非同期メッセージ・コンシュームを停止すると、制御は登録クライアントに戻されます。

メッセージング・クライアント・プログラム CSQ4CVPT

メッセージング・クライアントは、CICS 端末から CICS トランザクション MVMP の下で開始され、コマンド行入力を取得します。メッセージング・クライアントには、以下の 2 つの機能があります。

- 基本テキスト・メッセージをトピック News/Media/Movies に公開して、CSQ4CVCN によってコンシュームされるようにします。
- 制御メッセージをキュー SAMPLE.CONTROL.QUEUE に配置して、CSQ4CVCT によってコンシュームされるようにします。制御メッセージにより、一連のサンプル・プログラムの動作を変更するためのコマンドが CSQ4CVCT から発行されます。また、サンプル・プログラムによって表示されるメッセージを通して結果を確認できます。

113 ページの『CSQ4CVPT (サンプル・メッセージング・クライアント)』では、CSQ4CVPT を使用してメッセージを作成する方法が示されています。

基本メッセージ用のメッセージ・コンシューマー・プログラム CSQ4CVCN

CSQ4CVCN は、メッセージング・クライアントを使用してトピック News/Media/Movies の下に公開する基本 IBM MQ メッセージをコンシュームするメッセージ・コンシューマーです。

CSQ4CVCN が IBM MQ の呼び出しタイプ MSG_REMOVED で呼び出されると、インバウンド・メッセージを取得し、それを CICS ジョブ・ログにエコー出力します。

制御メッセージ用のメッセージ・コンシューマー・プログラム CSQ4CVCT

CSQ4CVCT は、キュー SAMPLE.CONTROL.QUEUE に配置する IBM MQ メッセージをコンシュームするメッセージ・コンシューマーです。IBM MQ の呼び出しタイプ MSG_REMOVED で CSQ4CVCT が呼び出されると、CSQ4CVCT は、登録クライアントから渡されるインバウンド・メッセージとデータ構造を取り出します。

メッセージング・クライアントを使用して、CSQ4CVCT に以下のアクションを実行するように指示する制御メッセージを作成できます。CSQ4CVCT は、適切な MQCB または MQCTL コマンドを発行して、要求するアクションを実行します。

- 指定したコールバック・ルーチンの登録解除
- 指定したコールバック・ルーチンの再開
- 指定したコールバック・ルーチンの中断
- MQCTL STOP コマンドを発行して非同期メッセージ・コンシュームを停止し、登録クライアントに制御を戻します

CSQ4CVCT にそれ自体を中断するか、または登録解除するように指示することもできますが、そのようにすると、そのサンプル・プログラムの動作をそれ以上制御できなくなります。すべてのコールバック・ルーチンを中断するか、または登録解除すると、制御は登録クライアントに戻され、タスクが終了します。

イベント・ハンドラー・プログラム CSQ4CVEV

イベント・ハンドラーとして、キュー・マネージャーの停止または静止など、コールバック環境全体に影響する条件が発生したときに、CSQ4CVEV は IBM MQ の通知を受け取ります。

理由コード CONNECTION_QUIESCING を受け取る場合、CSQ4CVEV は MQCTL STOP コマンドを発行して、非同期メッセージ・コンシュームを停止し、制御を登録クライアントに戻します。

非同期メッセージ・コンシューム用のサンプルのセットアップ

サンプル・プログラムをセットアップして、IBM MQ 非同期メッセージ・コンシュームおよびコールバック・ルーチンをデモンストレーションします。

このタスクについて

サンプル・プログラム CSQ4CVRG、CSQ4CVPT、CSQ4CVCN、CSQ4CVCT、および CSQ4CVEV は、IBM MQ で提供されます。IBM MQ バージョン 7.0.1 の場合、APAR PM06722 の PTF を適用してサンプルを取得します。サンプルのソースは IBM MQ ライブラリー SCSQCOBS にあります。また、ロード・モジュールは IBM MQ ライブラリー SCSQCICS にあります。

サンプル・プログラムの CICS リソース定義は、IBM MQ グループ CSQ4SAMP で提供されます。CSQ4SAMP には、プログラム CSQ4CVRG (登録クライアント) および CSQ4CVPT (メッセージング・クライアント) に対してそれぞれ使用する CICS トランザクション MVRG および MVMP のリソース定義も含まれます。

手順

1. IBM MQ ライブラリー *thlqual.SCSQCICS* を CICS プロシージャの DFHRPL 連結に組み込みます。
また、高位修飾子 *thlqual* が IBM MQ バージョン 7.0.1 以降に対応していることを確認します。
正しいコードが確実に使われるようにするために、CICS ライブラリーの後にこのライブラリーを組み込んでください。
2. CICS で、以下のようにして、IBM MQ によって提供される、サンプル・プログラム用、およびトランザクション MVRG と MVMP 用のリソース定義をインストールします。
 - a) 既存の CSQ4SAMP グループを CICS 領域で削除します。
 - b) IBM MQ バージョン 7.0.1 用の PTF を適用する場合、PTF に同梱されているライブラリー SCSQPROC のメンバー CSQ4S100 のバージョンを使用します。
このメンバーには、新しいリソース定義が含まれています。
 - c) DFHCSDUP を使用して、CSQ4SAMP グループを取り込むデータ・セット *thlqual.SCSQPROC (CSQ4S100)* で CSD を更新します。
 - d) CSQ4SAMP グループをインストールします。CEDA トランザクションを使用して、CICS-MQ アダプター・リソースをアクティブ CICS システムにインストールする場合には、まずアダプターをシャットダウンした後、アラート・モニターの作業が完了するのを待つ必要があります。
3. メッセージ・コンシューマー CSQ4CVCT によって使用されるキュー SAMPLE.CONTROL.QUEUE を IBM MQ キュー・マネージャーまたは CICS 領域に関連付けられたキュー共用グループに定義します。
CICS 領域の MQCONN リソース定義は、IBM MQ キュー・マネージャーまたはキュー共用グループを指定します。キュー名は、サンプル・プログラムでコーディングします。
4. オプション: トピック News/Media/Movies を IBM MQ キュー・マネージャーまたはキュー共用グループに定義します。
このトピックを定義しない場合、実行時に IBM MQ により、デフォルト管理オブジェクトの下にそれが作成されます。
キュー名のようなトピックをサンプル・プログラムでコーディングします。
5. サンプルの登録クライアント・プログラム CSQ4CVRG を CICS トランザクション MVRG の下で開始します。
登録クライアントは、イベント・ハンドラー CSQ4CVEV、メッセージ・コンシューマー CSQ4CVCN、およびメッセージ・コンシューマー CSQ4CVCT を IBM MQ を使用するコールバック・ルーチンとして登録します。

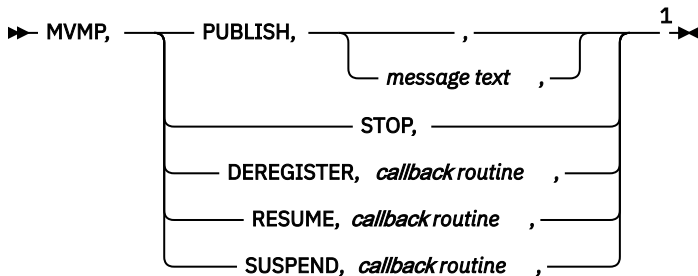
次のタスク

サンプルのメッセージング・クライアント・プログラム CSQ4CVPT を使用して、基本メッセージの公開、制御メッセージの作成、および結果の確認を行えます。CSQ4CVPT を使用する方法については、[113 ページの『CSQ4CVPT \(サンプル・メッセージング・クライアント\)』](#)で説明しています。

CSQ4CVPT (サンプル・メッセージング・クライアント)

サンプルのメッセージング・クライアント・プログラム CSQ4CVPT を CICS 端末から CICS トランザクション MVMP の下で使用します。

メッセージング・クライアント・コマンド



注:

1 キーワードとコールバック・ルーチンの名前は、大/小文字を区別しません。入力フィールドは定位置であり、コンマによって区切られ、終了します。

パラメーター

MVMP

CSQ4CVPT 用の CICS トランザクション。

PUBLISH , { , | message text , }

指定したメッセージ・テキストを CICS-MQ の保持されたメッセージとしてトピック News/Media/Movies の下で公開します。メッセージは、メッセージ・コンシューマー・プログラム CSQ4CVCN によってコンシュームされます。メッセージ・テキストを省略して終了コンマを指定すると、デフォルトのメッセージ・テキストが公開されます。

DEREGISTER , callback routine ,

CICS-MQ メッセージをキュー SAMPLE.CONTROL.QUEUE に配置し、メッセージ・コンシューマー・プログラム CSQ4CVCT に指定したコールバック・ルーチンの登録解除を指示します。

- CSQ4CVCT を登録解除した場合、このサンプル・プログラムの動作を制御できなくなります。
- すべてのコールバック・ルーチンを登録解除すると、制御は登録クライアントに戻されます。

RESUME , callback routine ,

CICS-MQ メッセージをキュー SAMPLE.CONTROL.QUEUE に配置し、メッセージ・コンシューマー・プログラム CSQ4CVCT に指定したコールバック・ルーチンの再開を指示します。

SUSPEND , callback routine ,

CICS-MQ メッセージをキュー SAMPLE.CONTROL.QUEUE に配置し、メッセージ・コンシューマー・プログラム CSQ4CVCT に指定したコールバック・ルーチンの中断を指示します。

- CSQ4CVCT を中断した場合、このサンプル・プログラムの動作をそれ以上制御できなくなります。
- すべてのコールバック・ルーチンを中断すると、制御は登録クライアントに戻されます。

STOP ,

CICS-MQ メッセージをキュー SAMPLE.CONTROL.QUEUE に配置し、メッセージ・コンシューマー・プログラム CSQ4CVCT に MQCTL STOP コマンドを発行して非同期メッセージ・コンシュームを停止するように指示します。このアクションは、制御を登録クライアント CSQ4CVRG に戻します。

CSQ4CVCN によるコンシューム用のデフォルト・メッセージ・テキストを公開するには、次のコマンドを使用します。

```
MVMP,PUBLISH,,
```


CSQ4CVCN によるコンシューム用のメッセージ・テキスト「A short message」を公開するには、次のコマンドを使用します。

```
MVMP,publish,A short message,
```

非同期メッセージ・コンシュームを停止するには、次のコマンドを使用します。

```
MVMP,STOP,
```

イベント・ハンドラー CSQ4CVEV の登録を解除するには、次のコマンドを使用します。

```
MVMP,DEREGISTER,CSQ4CVEV,
```

メッセージ・コンシューマー CSQ4CVCN を再開するには、次のコマンドを使用します。

```
MVMP,resume,csq4cvcn,
```

イベント・ハンドラー CSQ4CVEV を中断するには、次のコマンドを使用します。

```
MVMP,SUSPEND,CSQ4CVEV,
```

CICS-MQ API 交差出口

CICS は、CICS-MQ アダプターとともに使用する API 交差出口を提供します。この出口を使用して、モニター、テスト、保守、またはセキュリティー目的で、IBM MQ 呼び出しの実行中に、その呼び出しを代行受信することができます。この出口は CICS アドレス・スペースで実行します。

このセクションには、プロダクト・センシティブ・プログラミング・インターフェースに関する情報が含まれています。

以下の目的で、CICS-MQ API 交差出口を使用できます。

- それぞれの MQI 呼び出しの前後に各メッセージの内容を調べることによって、付加的なセキュリティー検査を行う。
- メッセージで提供されたキュー名を別のキュー名に置き換える。
- 呼び出しを取り消し、0 の戻りコードを発行して正常な呼び出しをシミュレートするか、あるいは別の値を発行してその呼び出しが実行されなかったことを示す。
- アプリケーションにおける MQI 呼び出しの使用状況をモニターする。
- 統計を収集する。
- 特定の呼び出しにおける入力パラメーターを変更する。
- 特定の呼び出しの結果を変更する。

CICS-MQ API 交差出口を使用すると、IBM MQ のパフォーマンスは低下します。この出口の使用は、慎重に計画してください。

CICS-MQ API 出口プログラムは、CSQCAPX と呼ばれます。CICS は、サンプル出口プログラムとグループ DFHMQ の CSQCAPX に対するプログラム定義を提供します。この出口を使用するために、このプログラムは、DFHRPL 連結に含まれている必要があります。また、CICS システム定義ファイル (CSD) で定義されている必要があり、さらに使用可能である必要があります。提供されているサンプルの代わりに独自の出口プログラムを作成できます。また、プログラム CSQCAPX を呼び出す必要もあります。

CSQCAPX をロードすると、CICS-MQ アダプター制御パネル (CKQC) またはコンソールに確認メッセージが書き込まれます。ロードできない場合は診断メッセージが表示されますが、それ以外の場合、アプリケーション・プログラムは正常に実行されます。

重要: 提供される CSQCAPX の定義は、パラメーター CONCURRENCY(THREADSAFE) を指定します。独自の出口プログラムを作成する場合、定義するときに CONCURRENCY(THREADSAFE) を指定します。出口では、スレッド・セーフの CICS コマンドのみを使用します。また、この設定を指定し、出口プログラムが呼び出すすべてのプログラムに対しても、スレッド・セーフの CICS コマンドのみを使用します。以前の CICS リリースで作成された API 交差出口を調べ、それらのロジックがスレッド・セーフであることを確認します。

CICS-MQ API 交差出口の呼び出し方法

CICS-MQ API 交差出口を使用すると IBM MQ for z/OS のパフォーマンスが低下します。そのため、これを使用するには慎重に計画してください。使用可能な場合、CICS-MQ API 交差出口は、以下の状況で呼び出されます。

- CSQCAPX は、CICS-MQ アダプターを使用するすべてのアプリケーションによって呼び出されます。
- CSQCAPX は、以下のいずれかの MQI 呼び出しが実行されるたびに呼び出されます。
 - MQCB
 - MQCTL
 - MQCLOSE
 - MQGET
 - MQINQ
 - MQOPEN
 - MQPUT
 - MQPUT1
 - MQSET
 - MQSTAT
 - MQSUB
 - MQSUBRQ

呼び出しの前と後に一度ずつ CSQCAPX が呼び出されます。

- 非同期メッセージ・コンシュームの処理中に、CSQCAPX は 3 つの状況で呼び出されます。
 - メッセージをコールバック・ルーチンに渡す前にデータ変換が必要な場合、IBM MQ ルーチン CSQAVICD を呼び出す前。
 - コールバック・ルーチンの呼び出し前後。

これらの呼び出しの場合、CSQCAPX に渡されるパラメーター・リストの MQXP_PCOPYPARM フィールドによって参照される IBM MQ MQI パラメーター・リストには、コールバック・ルーチンに渡されるのと同じデータ (つまり、MQMD、MQGMO、Buffer、および Context) が含まれます。

CSQCAPX は、メッセージ・プロパティおよびメッセージ処理に関する関数呼び出し (つまり、MQCRTMH、MQDLTMH、MQSETMP、MQINQMP、MQDLTMP、MQMHBUF、および MQBUFMH の呼び出し) の場合、呼び出されません。

TCB 切り替えの過剰な使用を抑えるため、出口プログラムを THREADSAFE として定義していることを確認する必要があります。

呼び出しの実行前に一度、呼び出し実行後に一度、出口プログラムが呼び出されます。「実行前」のタイプの出口呼び出しでは、出口プログラムは MQI 呼び出しに関するすべてのパラメーターを修正し、その呼び出しを完全に抑制するか、またはその呼び出しを処理できます。呼び出しが処理されると、その呼び出しの完了後に出口が再び呼び出されます。

出口プログラムは再帰的ではありません。出口の内部でなされる MQI 呼び出しが出口プログラムの 2 度目の呼び出しを実行することはありません。

CICS-MQ 交差出口プログラムとの通信

CICS-MQ 交差出口プログラムは、呼び出された後で、DFHEICAP と呼ばれるフィールドによって指し示される CICS 連絡域内のパラメーター・リストが渡されます。

CICS EXEC インターフェース・ブロックのフィールド EIBCALEN には、この領域の長さが示されます。この通信域の構造は、交差出口プログラムで以下のように定義されます。

* MQXP_COPYPLIST		
	DSECT	
	DS 0D	Force doubleword alignment
MQXP_PXPB	DS AL4	Pointer to exit parameter block
MQXP_PCOPYPARM	DS 11AL4	Copy of original plist

```

*
      ORG  MQXP_PCOPYPARM
MQXP_PCOPYPARM1  DS  AL4      Copy of 1st parameter
MQXP_PCOPYPARM2  DS  AL4      Copy of 2nd parameter
MQXP_PCOPYPARM3  DS  AL4      Copy of 3rd parameter
MQXP_PCOPYPARM4  DS  AL4      Copy of 4th parameter
MQXP_PCOPYPARM5  DS  AL4      Copy of 5th parameter
MQXP_PCOPYPARM6  DS  AL4      Copy of 6th parameter
MQXP_PCOPYPARM7  DS  AL4      Copy of 7th parameter
MQXP_PCOPYPARM8  DS  AL4      Copy of 8th parameter
MQXP_PCOPYPARM9  DS  AL4      Copy of 9th parameter
MQXP_PCOPYPARM10 DS  AL4      Copy of 10th parameter
MQXP_PCOPYPARM11 DS  AL4      Copy of 11th parameter
*
MQXP_COPYPLIST_LENGTH EQU *-MQXP_PXPB
      ORG  MQXP_PXPB
MQXP_COPYPLIST_AREA  DS  CL(MQXP_COPYPLIST_LENGTH)
*

```

フィールド *MQXP_PXPB* は、出口パラメーター・ブロック *MQXP* を指し示します。

フィールド *MQXP_PCOPYPARM* は、呼び出しパラメーターのアドレスの配列です。例えば、アプリケーションがパラメーター P1、P2、または P3 を指定して MQI 呼び出しを発行すると、連絡域には次のデータが入っています。

```
PXPB,PP1,PP2,PP3
```

ここで、*P* はポインター (アドレス) を示し、*XPB* は、出口パラメーター・ブロックです。

独自の CICS-MQ API 交差出口プログラムの作成

CICS で提供されるサンプル API 交差出口プログラム (CSQCAPX) を、自分の作成するプログラムのフレームワークとして使用できます。スレッド・セーフになるように出口プログラムを作成し、作成した出口プログラムをスレッド・セーフとして宣言する必要があります。

このタスクについて

パフォーマンス上の理由から、アセンブラー言語でプログラムを作成してください。CICS がサポートする他の言語のいずれかでプログラムを作成する場合には、独自のデータ定義ファイルを作成する必要があります。プログラムを AMODE(31) および RMODE(ANY) としてリンク・エディットします。交差出口プログラムに渡される *MQXP* パラメーター・リストのレイアウトは、プログラム内で定義します。

手順

- 出口は、CICS-MQ コードの拡張として作成します。作成した出口が、MQI を使用する IBM MQ for z/OS のプログラムやトランザクションに障害を引き起こさないことを確認します。
これらは、CSQ、DFH、または CK の接頭部で示されます。
- 作成するプログラムでは、CICS タスク関連のユーザー出口プログラムが使用できるすべての API (例えば、IMS、Db2®、および CICS) を使用できます。
出口プログラムおよび出口プログラムから呼び出すすべてのプログラムでは、スレッド・セーフの CICS コマンドのみを使用する必要があります。
- プログラムでは、MQCONN、MQCONNX、および MQDISC を除くすべての MQI 呼び出しを使用できます。
出口プログラム内でなされる MQI 呼び出しが出口プログラムの 2 度目の呼び出しを実行することはありません。
- プログラムは、MQI 呼び出しの後に呼び出されると、その呼び出しによって設定された完了コードと理由コードを調べて、変更できます。
- アプリケーションによって発行される MQI 呼び出しの名前を確認するには、*MQXP* 構造の **ExitCommand** フィールドを調べます。その呼び出しのパラメーターの数を確認するには、**ExitParmCount** フィールドを調べます。

16 バイトの **ExitUserArea** フィールドを使用して、アプリケーションが取得するすべての動的ストレージのアドレスを保管できます。このフィールドは、出口が使用される間、保持されます。存続期間は CICS タスクと同じです。

- ユーザーのプログラムは、MQXCC_SUPPRESS_FUNCTION または MQXCC_SKIP_FUNCTION を **ExitResponse** フィールドに戻すことによって、MQI 呼び出しの実行を抑制できます。
呼び出しを実行する場合 (また、その呼び出しが完了した後で出口プログラムを再び呼び出す場合)、ユーザーの出口プログラムが MQXCC_OK を戻す必要があります。
- 作成するプログラムでは **EXEC CICS SYNCPOINT** コマンドまたは **EXEC CICS SYNCPOINT ROLLBACK** コマンドを発行できます。これらのコマンドは、タスクによって実行されたすべての更新を、使用された出口のポイントまで、コミットまたはロールバックします。
- パラメーター MQXP_EXITCALLPROG は、COMMAREA を介してモジュールに渡されるパラメーター・リストに含まれています。また、このリストには、モジュールを呼び出したプログラムの名前が含まれます。CSQCAPX はユーザー置換可能モジュール (URM) として呼び出されるため、EXEC CICS ASSIGN INVOKINGPROG コマンドはブランクに戻します。呼び出し側プログラムの名前が含まれており、EXEC CICS ASSIGN INVOKINGPROG コマンドから受け取る出力と形式が同じであるため、パラメーター MQXP_EXITCALLPROG を代わりに使用できます。パラメーター MQXP_VERISON の値が 2 の場合に、このパラメーターを使用できます。
- プログラムは、**EXEC CICS RETURN** コマンドを発行して終了する必要があります。XCTL コマンドで制御権を移動することはできません。
- 出口プログラムおよび出口プログラムから呼び出すすべてのプログラムの PROGRAM リソース定義を作成します。

出口プログラム CSQCAPX を呼び出す必要があります。

PROGRAM リソース定義に、以下の設定を指定します。

- a) 出口プログラムおよび出口プログラムから呼び出すすべてのプログラムに対して、CONCURRENCY(THREADSAFE) または CONCURRENCY(REQUIRED) を指定します。
出口プログラムおよび出口プログラムから呼び出すすべてのプログラムでは、スレッド・セーフの CICS コマンドのみを使用する必要があります。
- b) CICS Transaction Server のストレージ保護機能を使用している場合、CICS 実行キーで実行するように、出口プログラムおよび出口プログラムが制御を渡すすべてのプログラムに対して EXECKEY(CICS) を指定します。
詳しくは、出口プログラムおよび CICS ストレージ保護機能を参照してください。
- c) 通信域のパラメーターは、アドレス指定です。そのため、出口プログラムを CICS システムに対して、リモート・プログラムとしてではなく、ローカルとして定義する必要があります。

CICS は、グループ DFHMQ の CSQCAPX に対するサンプル・プログラム定義を提供します。このプログラム定義は、CONCURRENCY(THREADSAFE) および EXECKEY(CICS) を指定します。このサンプルを出口プログラムのプログラム定義の基礎として使用できます。

タスクの結果

CICS システムは、CICS が IBM MQ for z/OS に接続すると、出口プログラムをロードしようとします。この試行が成功する場合、メッセージ DFHMQ0301I が CKQC パネルまたはシステム・コンソールに送られます。ロードが失敗すると (例えば、ロード・モジュールが DFHRPL 連結のどのライブラリーにも存在しない場合)、メッセージ DFHMQ0315E が CKQC パネルまたはシステム・コンソールに送られます。

サンプル API 交差出口プログラム CSQCAPX

サンプル出口プログラムはアセンブラ言語プログラムとして提供されます。ソース・ファイル (CSQCAPX) は *thlqual.SDFHSAMP* で提供されます。ここで、*thlqual* はインストールによって使用される高位修飾子です。このソース・ファイルには、プログラム論理を記述する疑似コードが含まれています。

このサンプル・プログラムには、初期化コードと、独自の出口プログラムを作成する際に使用できるレイアウトが含まれています。

このサンプルでは、以下のタスクを実行する方法を示します。

- 出口パラメーター・ブロックのセットアップ

- 呼び出しと出口パラメーター・ブロックのアドレッシング
- どの MQI 呼び出しに対して出口が呼び出されているかの判別
- 出口の呼び出しが MQI 呼び出しの処理の前か後かの判別
- CICS 一時記憶域キューへのメッセージの書き込み
- 動的ストレージ獲得用マクロ DFHEIENT の使用による、再入可能性の維持
- CICS exec インターフェース 制御ブロックに対する DFHEIBLK の使用
- エラー条件のトラップ
- 呼び出し側への制御の戻し

サンプル出口プログラムでは、CICS 一時記憶域キュー (CSQ1EXIT) にメッセージを書き込み、出口の操作を示します。メッセージは、出口の呼び出しが MQI 呼び出しの前か後かを示します。出口が MQI 呼び出しの後に呼び出される場合、メッセージにはその呼び出しによって戻される完了コードと理由コードが含まれています。このサンプルでは、CMQXA マクロから名前付きの定数を使用して、入り口のタイプ (つまり、その呼び出しの前か後) について検査します。

このサンプルでは、モニターを実行しませんが、処理中の呼び出しのタイプを示す、タイム・スタンプ付きのメッセージを CICS キューに入れます。メッセージには、出口プログラムの本来の機能と共に、MQI のパフォーマンス情報が示されます。

サンプル出口プログラムは、プログラムの実行中に行われる MQI 呼び出しごとに 6 つの EXEC CICS 呼び出しを発行します。この出口プログラムを使用すると、CICS-MQ アプリケーションのパフォーマンスは低下します。

CICS-MQ API 交差出口の使用可能化

CSQCAPX の提供されるプログラム定義は、STATUS(DISABLED) を指定しています。出口プログラムを使用すると、IBM MQ パフォーマンスが著しく低下する可能性があるため、このプログラムは使用不可の状態です。API 交差出口を使用する前に使用可能にする必要があります。

このタスクについて

CICS-MQ アダプター制御パネル、コマンド行、または CICS アプリケーション・プログラムを使用して API 交差出口を一時的にアクティブにすることができます。CICS リソース定義を変更することによって、API 交差出口を永続的に使用可能にすることができます。

手順

- CICS-MQ アダプター制御パネルを使用して API 交差出口を一時的にアクティブにするには、以下の手順を実行します。
 - a) CKQC を入力し、Enter キーを押して CICS-MQ アダプター制御パネルにアクセスします。
 - b) メニュー・バーから「**Connection (接続)**」を選択します。
 - c) メニューから「**Modify (修正)**」アクションを選択します。
 - d) 「**Modification Options (変更オプション)**」2 次パラメーター・ウィンドウで、「**Enable API Exit (API 出口の使用可能化)**」を選択し、Enter キーを押して API 交差出口の状況を「Enabled (使用可能)」に変更します。

画面には、以下の処理が表示されます。

Connection	CKTI	Task
Select an action.	CS Adapter Control -- Initial panel	
3 1. Start...		sing Tab key. Then press Enter.
2. Stop...		
3. Modify...		
4. Display		
F1=Help F12=Cancel		
		<div> <div>Modification Options</div> <div>Select modify option. Then press Enter.</div> <div>4 1. Reset statistics</div> <div>2. Enable API Exit</div> <div>3. Disable API Exit</div> <div>F1=Help F12=Cancel</div> </div>

F1=Help F3=Exit

- CICS コマンド行から API 交差出口を一時的にアクティブにするには、以下の方法のいずれかを選択します。
 - CICS マスター端末からコマンド CEMT S PROGRAM(CSQCAPX) ENABLED を発行する。
 - コマンド行からコマンド CKQC MODIFY N E を発行します。オプション E は、接続を使用可能にします。オプション N は、接続の統計に関係し、必須ですが、影響はありません。
- CICS アプリケーション・プログラムを使用して API 交差出口を一時的にアクティブにするには、EXEC CICS LINK コマンドを発行してアダプター・リセット・プログラム (DFHMQRS (または互換性が保持される CSQCRST)) にリンクし、オプション E を指定した CKQC MODIFY コマンドを発行します。以下の例は、この実行方法を示します。

```
EXEC CICS LINK PROGRAM('DFHMQRS ')
      INPUTMSG('CKQC MODIFY N E ')
```

MODIFY コマンドでは、4 つの末尾スペースに加えて、区切り文字としてもう 1 つのスペースを埋め込む必要があります。(CICS(r) アプリケーション・プログラムの [CKQC コマンドを参照してください](#))。オプション N は、接続の統計に関係し、必須ですが、影響はありません。

- API 交差出口を永続的に使用可能にして実行するには、以下の手順を実行します。
 - CSQCAPX の定義を CICS 提供のグループ DFHMQ から独自のグループにコピーします。
 - 状況を DISABLED から ENABLED に変更するように CSQCAPX 定義を変更し、新規グループをインストールします。
 - どのグループ・リストでも独自グループが DFHMQ の後にインストールされていることを確認します。

CICS-MQ API 交差出口の使用不可化

IBM MQ パフォーマンスを向上させるため、必要ではなくなったときに、CICS-MQ API 交差出口を使用不可にします。

このタスクについて

CICS-MQ アダプター制御パネル、コマンド行、または CICS アプリケーション・プログラムを使用して API 交差出口を使用不可にすることができます。

手順

- CICS-MQ アダプター制御パネルを使用して API 交差出口を使用不可にするには、以下の手順を実行します。
 - CKQC を入力し、Enter キーを押して CICS-MQ アダプター制御パネルにアクセスします。
 - メニュー・バーから「**Connection (接続)**」を選択します。

- c) メニューから「**Modify (修正)**」アクションを選択します。
- d) 「**Modification Options (変更オプション)**」2 次パラメーター・ウィンドウで、「**Disable API Exit (API 出口の使用不可化)**」を選択し、Enter キーを押して API 交差出口の状況を「Disabled (使用不可)」に変更します。

画面には、以下の処理が表示されます。

Connection	CKTI	Task
Select an action.	CS Adapter Control -- Initial panel	
3 1. Start...		sing Tab key. Then press Enter.
2. Stop...		
3. Modify...		
4. Display		
F1=Help F12=Cancel		
		<div> Modification Options </div>
		Select modify option. Then press Enter.
		4 1. Reset statistics
		2. Enable API Exit
		3. Disable API Exit
		F1=Help F12=Cancel

F1=Help F3=Exit

2. CICS コマンド行から API 交差出口を使用不可にするには、以下の方法のいずれかを選択します。
 - CICS マスター端末からコマンド CEMT S PROGRAM(CSQCAPX) DISABLED を発行します。
 - コマンド行からコマンド CKQC MODIFY N D を発行します。オプション D は、接続を使用不可にします。オプション N は、接続の統計に関係し、必須ですが、影響はありません。
 3. CICS アプリケーション・プログラムを使用して API 交差出口を使用不可にするには、EXEC CICS LINK コマンドを発行してアダプター・リセット・プログラム (DFHMQRS (または互換性が保持される CSQCRST)) にリンクし、オプション D を指定した CKQC MODIFY コマンドを発行します。
- 以下の例は、この実行方法を示します。

```
EXEC CICS LINK PROGRAM('DFHMQRS ')
INPUTMSG('CKQC MODIFY      N D ')
```

MODIFY コマンドでは、4 つの末尾スペースに加えて、区切り文字としてもう 1 つのスペースを埋め込む必要があります。(CICS(r) アプリケーション・プログラムの [CKQC コマンドを参照してください](#))。オプション N は、接続の統計に関係し、必須ですが、影響はありません。

第 9 章 CICS-MQ アダプターのトラブルシューティング

CICS-MQ アダプターに問題がある場合、CICS は、原因を診断するのに役立つ幅広い情報を提供します。

必要な情報および現在作業している環境に応じて、CEMT、**EXEC CICS INQUIRE** コマンド、CICS-MQ アダプター制御パネル、CKQC DISPLAY コマンド、または CICSplex SM を使用して、CICS-MQ 接続に関する情報を表示できます。各種の方式については、[CICS-MQ 接続に関する情報の表示](#)を参照してください。

CKQC トランザクション (CICS-MQ アダプター制御パネルからの) は、IBM MQ への接続を使用している個々のタスクの詳細とそれぞれの状態 (GET WAIT など) を表示できます。[CICS-MQ 接続を使用しているタスクの表示](#)では、この情報を表示する方法について説明しています。

CICS 実行診断機能 (CEDF) は、各 MQI 呼び出しの CICS-MQ アダプターへの出入り口をトラップし、さらにすべての CICS API サービスへの呼び出しをトラップします。

CICS-MQ アダプターは、A000 から A1FF までの範囲の AP ドメイン・トレース・ポイントを使用し、トレース項目が標準 CICS トレース宛先に書き込まれます。CICS-MQ アダプターが発行するトレース・ポイントの内容は、[CICS IBM MQ のトレース・ポイント](#)に文書化されています。例外トレース項目は、エラー条件に対して無条件に生成されます。非例外トレース項目は、RI (リソース・マネージャー・インターフェース) および RA (リソース・マネージャー・アダプター) レベル 1 およびレベル 2 トレースによって制御されます。CICS トレースについて詳しくは、[CETR - トレース制御](#)を参照してください。

IBM MQ の待機

タスクがリソース・タイプ MQseries、WMQ_INIT、または WMQCDISC を待機している場合、CICS-MQ アダプターがそのリソースを中断しています。

リソース・タイプ MQseries

MQGET 呼び出しで WAIT オプションが使用され、使用可能なメッセージがなかったため、CICS-IBM MQ MQ アダプター (DFHMQTRU モジュール) はタスクを CICS 待機状態にしました。待機に使用されたリソース名は GETWAIT です。ディスパッチャーの WAIT_MVS 機能がこの待機に使用され、ワークロード管理の待機タイプは OTHER_PRODUCT です。タスクをページできます。

リソース・タイプ WMQ_INIT

DFHMQIN1 (CICS-IBM MQ 初期設定プログラム) は、この待機を発行して DFHMQIN2 が完了するのを待ちます。ディスパッチャーの WAIT_OLDC 機能がこの待機に使用され、ワークロード管理の待機タイプは MISC です。タスクをページできます。

リソース・タイプ WMQCDISC

WAIT オプションまたは FORCE オプションを指定して SET MQCONN NOTCONNECTED コマンドが発行され、DFHMQTM モジュールは IBM MQ を使用するユーザー・タスクのカウン트가ゼロになるのを待ちます。リソース名は、CICS システムのインストール済みの MQCONN リソース定義の名前として示されます。ディスパッチャーの WAIT_OLDC 機能がこの待機に使用され、ワークロード管理の待機タイプは MISC です。タスクをページできます。

CICS-MQ 接続のシャットダウン時の処理

CICS と IBM MQ の間の接続には、2 つのタイプのシャットダウン (静止 (または正常) シャットダウンと強制シャットダウン) があります。接続のシャットダウンは、オペレーター・アクション、CICS のシャットダウン、または IBM MQ キュー・マネージャーに起因する可能性があります。

[122 ページの表 12](#) は、接続がアクティブなときの各種形式のシャットダウンについて、アダプターの処理方法を要約しています。接続が非アクティブのとき (例えば、静止された後) に CICS または IBM MQ がシャットダウンした場合、アクションは何も取られず、メッセージも発行されません。

表 12. CICS アダプター接続のシャットダウン

シャットダウンの方式	アダプターによる 処理方法
接続の静止シャットダウン (EXEC CICS SET MQCONN NOTCONNECTED BUSY(WAIT NOWAIT)、CEMT SET MQCONN NOTCONNECTED、または CKQC STOP)	アダプターの状況に <i>Quiescing</i> のマークを付けます。アクティブ・タスクと待機中のタスクの両方の完了を許可します。同期点を許可します。新しいタスクからの呼び出しは許可しません。最後のタスクが IBM MQ からの切断を開始します。
接続の強制シャットダウン (EXEC CICS SET MQCONN NOTCONNECTED BUSY(FORCE)、CEMT SET MQCONN FORCENOTCON、または CKQC STOP FORCE)	アダプターの状況に <i>StoppingForce</i> のマークを付けます。IBM MQ から切断します。最初に、IBM MQ で待機中のタスク (CKTI のインスタンスを含む) を再開します。次に、WebSphere MQ にアクセスした未完了タスクを強制ページします。
CICS ウォーム・シャットダウン	メッセージ DFHMQ0411I を出します。接続の静止シャットダウンを開始します。
CICS 即時シャットダウン	メッセージ DFHMQ0410 I を出します。IBM MQ を使用している未完了タスクはすべてバックアウトされます。
CICS 異常終了	メッセージ DFHMQ0412 I を出します。
IBM MQ 静止	接続の静止シャットダウンを開始します。
IBM MQ 異常終了または強制シャットダウン	接続の強制シャットダウンを開始します。

静止 (または正常) シャットダウン

接続の静止シャットダウンでは、インターフェースが閉じる前に、各 CICS トランザクションを終了できます。この方式を使用すると、CICS を再接続したときに、未確定の作業単位がないものと想定できます。

静止シャットダウンは、以下の各状況で発生します。

- CICS 端末オペレーターが **EXEC CICS、CEMT SET MQCONN NOTCONNECTED、CKQC STOP** のいずれかのコマンドを発行する。CICS およびキュー・マネージャーはアクティブなままです。
- CICS 端末オペレーターが **CEMT PERFORM SHUTDOWN** コマンドを発行する。
- キュー・マネージャーが次のコマンドによって静止される。

```
+CSQ1 STOP QMGR MODE(QUIESCE)
```

このコマンドは、キュー・マネージャーを停止します。現在識別済みのタスクは通常の処理を続行できますが、新しいタスクはそれ自体をキュー・マネージャーに識別することはできません。CICS はアクティブなままです。

強制シャットダウン

接続の強制シャットダウンにより、キュー・マネージャーに接続されている CICS トランザクションが異常終了することがあります。そのために、システムが再接続されたときに、未確定の作業単位が生じる可能性があります。

強制シャットダウンは、以下の各状況で発生します。

- CICS 端末オペレーターが FORCE オプションを指定した **EXEC CICS SET MQCONN NOTCONNECTED** コマンドか、**CEMT SET MQCONN FORCENOTCON** コマンドまたは **CKQC STOP FORCE** コマンドを発行する。
- CICS 端末オペレーターが次の CICS 即時シャットダウン・コマンドを発行する。

```
CEMT PERFORM SHUTDOWN IMMEDIATE
```

キュー・マネージャーはアクティブなままです。このコマンドについては、[CEMT PERFORM SHUTDOWN](#)を参照してください。

- 次の IBM MQ 強制シャットダウン・コマンドが発行される。

```
+CSQ1 STOP QMGR MODE(FORCE) or +CSQ1 STOP QMGR MODE(RESTART)
```

CICS はアクティブなままです。

- IBM MQ 異常終了が発生する。CICS はアクティブなままです。
- CICS 異常終了が発生する。キュー・マネージャーはアクティブなままです。

キュー・マネージャーの停止時の処理

キュー・マネージャーが正常に停止した場合、IBM MQ はすべてのアクティビティを正常な方法で停止します。IBM MQ は、静止モード、強制モード、または再始動モードを使用して停止できます。

それぞれの影響を [123 ページの表 13](#) に示します。

表 13. QUIESCE、FORCE、および RESTART の各モードでのキュー・マネージャーの停止			
スレッド・タイプ	QUIESCE	FORCE	RESTART
アクティブ・スレッド	完了まで実行	バックアウト	バックアウト
新しいスレッド	開始可能	許可されていない	許可されていない
新しい接続	許可されていない	許可されていない	許可されていない

CICS では、現在のスレッドは作業単位の終わりまでのみ実行されます。CICS では、キュー・マネージャーを静止モードで停止すると CICS-MQ アダプターが停止します。そのため、アクティブ・タスクに複数の作業単位が含まれている場合、そのタスクは必ずしも完了まで実行されません。

キュー・マネージャーを停止するときは、いずれのモードの場合も、以下のような手順になります。

1. 接続が終了します。
2. IBM MQ が、コマンドの受け入れを中止します。
3. IBM MQ は、ページ・セットに対する未解決の更新がすべて完了したことを確認します。
4. IBM MQ によって内部で DISPLAY USAGE コマンドが発行され、再始動相対バイト・アドレス (RBA) が z/OS コンソール・ログに記録されます。
5. シャットダウン・チェックポイントが取られ、IBM MQ ブートストラップ・データ・セット (BSDS) が更新されます。

キュー・マネージャーを強制モードまたは再始動モードで停止した場合、新しいスレッドは割り振られず、接続されたスレッド上の作業はロールバックされます。これらのモードを使用している場合、2つのコミット処理フェーズの間にあるスレッドに未確定の作業単位が生じる可能性があります。未確定の作業単位は、IBM MQ が制御 CICS、IMS、または RRS サブシステムと再接続されると解決します。

静止モードは、未確定の作業単位には影響を与えません。未確定の作業単位は、未確定のままです。

自動再接続と再同期

CICS が IBM MQ に接続されているときにキュー・マネージャーが停止すると、CICS-MQ アダプターは、停止が検出されてから再接続を試行します。接続要求には、前回の接続要求で使用されたものと同じ接続パラメーターが使用されます。

接続にキュー・マネージャーを 1 つしか指定していない場合、停止が検出された後 CICS は 10 秒間待機し、その後で再接続を試みます。キュー・マネージャーが 10 秒以内に再始動しなかった場合、そのキュー・マネージャーが再始動するまで接続要求は延期され、キュー・マネージャーが再始動したときに CICS が自動的に再接続を行います。

接続にキュー共用グループを指定した場合は、停止が検出されてすぐ、CICS は再接続を試みます。CICS が接続を復元するために取るアクションは、最後のキュー・マネージャーに未解決の作業単位があるかどうか、および再同期に対して指定した設定によって異なります。

接続にキュー共用グループを指定した場合は、MQCONN リソース定義の RESYNCMEMBER 属性を使用して、CICS に適合する再同期アクションを選択できます。再同期が行われるのは、IBM MQ への接続が失われ、CICS が最後のキュー・マネージャーに対して未解決の作業単位を保持している場合です。CICS に同じキュー・マネージャーへの再接続を待機させるかどうか、あるいは CICS に同じキュー・マネージャーへの再接続を一度試行させるかどうかを選択できます。ただし、再接続の試行が失敗した場合は、グループ内の別の使用可能なキュー・マネージャーへの接続が行われます。CICS 領域への接続に使用できるのは、現在 CICS 領域と同じ LPAR でアクティブになっているキュー・マネージャーです。

あるいは、IBM MQ が CICS のグループ・リカバリー単位をサポートする場合は、**RESYNCMEMBER(GROUPRESYNC)** オプションを使用できます。GROUPRESYNC オプションを指定すると、CICS は、前にどのキュー・マネージャーに接続していたかに関係なく、キュー共用グループ内の任意のローカル IBM MQ キュー・マネージャーに接続し、未解決のすべての未確定作業単位を解決します。未解決の作業単位を解決するには、CICS は、前の接続で使用されていたものと同じ MQCONN リソース定義を使用して再接続する必要があります。詳しくは、[CICS による未確定の作業単位の解決方法](#)を参照してください。

CICS-MQ アダプターの再始動時の処理

接続が切断されるたびに、アダプターは再接続処理中に再始動フェーズを実行する必要があります。再始動フェーズによってリソースが再同期されます。CICS と IBM MQ の間での再同期により、未確定の作業単位を特定して解決できます。

再同期は、以下の要求が原因で生じる可能性があります。

- 分散キューイング・コンポーネントからの明示的な要求
- IBM MQ への接続時の暗黙の要求

IBM MQ への接続が原因で再同期が生じた場合、イベントの順序は以下のようになります。

1. 接続プロセスは、IBM MQ が未確定と見なす作業単位 (UOW) ID のリストを入手します。
2. UOW ID が DFHMQ0313I メッセージでコンソール上に表示されます。
3. UOW ID が CICS に渡されます。
4. CICS が、未確定 UOW ID ごとに再同期タスク (CRSY) を開始します。
5. 未確定 UOW ごとに、タスクの結果がコンソールに表示されます。

接続プロセス中表示される以下のメッセージを確認する必要があります。

DFHMQ0313I

UOW が未確定であることを示します。

DFHMQ0400I

UOW を識別し、その後以下いずれかのメッセージが続きます。

- DFHMQ0402I または DFHMQ0403I は、UOW が正常に解決された (コミットまたはバックアウトされた) ことを示します。
- DFHMQ0404E、DFHMQ0405E、DFHMQ0406E、または DFHMQ0407E は、UOW が解決されなかったことを示します。

DFHMQ0409I

すべての UOW が正常に解決されたことを示します。

DFHMQ0408I

一部の UOW が正常に解決されなかったことを示します。

DFHMQ0314I

*で強調表示された UOW ID は自動的に解決されていないことを警告します。これらの UOW は、分散キューイング・コンポーネントが再始動時に明示的に解決する必要があります。

DFHMQ0313I メッセージの合計数が、DFHMQ0402I と DFHMQ0403I のメッセージを加算した合計数に等しくなければなりません。合計数が等しくない場合、接続プロセスは一部の UOW を解決できません。解決できないこれらの UOW は、CICS の問題 (例えばコールド・スタート)、IBM MQ の問題、または分散キューイングが原因で生じます。問題が修正されたら、接続を切断してから再接続することにより、新たな再同期を開始できます。

あるいは、ユーザー自身が RESOLVE INDOUBT コマンドと、メッセージ DFHMQ0400I で示された UOW ID を使用して、未解決の各 UOW を解決することもできます。その場合は、切断してから接続を開始し、CICS 内の作業単位の記述子をクリーンアップする必要があります。手動で UOW を解決するには、UOW の正しい結果を知っている必要があります。

未解決の UOW に関連付けられたメッセージはすべて IBM MQ によってロックされ、バッチ、TSO、または CICS の各タスクはアクセスできません。

CICS に障害が発生し、緊急時再始動が必要な場合、CICS システムの GENERIC APPLID を変更しないでください。変更した場合、その後に IBM MQ に再接続したときに、IBM MQ とのデータ保全性を保証できません。IBM MQ は CICS の新規インスタンスを異なる CICS (APPLID が異なるため) として扱うためです。この場合、未確定の解決が、誤った CICS ログに基づいて行われることになります。

IBM MQ で作業単位が未処理になっている場合は、RESYNCMEMBER の設定を変更しないでください。変更すると作業単位を解決できなくなるためです。CICS で保留になっている作業単位は、リソース・マネージャーの修飾子で特定できます。RESYNCMEMBER(GROUPRESYNC) を使用する場合は、キュー共用グループの名前が修飾子になり、そうでない場合は、個々のキュー・マネージャーの名前が修飾子として使用されます。

CICS による未確定作業単位の解決方法

CICS-MQ アダプターの機能の 1 つは、CICS と IBM MQ の間でデータの同期を保つことです。キュー・マネージャーが CICS に接続されている間に異常終了した場合、CICS は作業をコミットまたはバックアウトする可能性があります。IBM MQ はそれを認識しません。キュー・マネージャーが再始動すると、その作業は未確定になります。

IBM MQ は、CICS への接続が再開または再接続されるまでは、このような未確定作業単位を解決する (つまり、IBM MQ リソースに対する変更をコミットまたはバックアウトする) ことができません。

CICS が同じ IBM MQ キュー・マネージャーに再接続した場合、CICS-MQ アダプターの始動中に、未確定作業単位を解決するためのプロセスが開始されます。このプロセスの手順は、以下のとおりです。

- CICS-MQ アダプターが、IBM MQ からこの接続 ID の未確定作業単位のリストを要求します。
- アダプターは、未確定作業単位のリストを受け取り、解決のためにリストを CICS に渡します。
- CICS は、このリストからのエントリーを、独自のログ内のエントリーと比較します。CICS は、独自のリストから、それぞれの未確定作業単位に対して取ったアクションを判別します。

解決されたすべての作業単位について、IBM MQ は必要に応じてキューを更新し、対応するロックを解放します。いずれかの作業単位が未確定で中断されている場合、つまり、CICS 自体では解決できない場合、それらは個別に解決されます。中断された作業単位について詳しくは、[中断された作業単位](#)を参照してください。

未確定作業単位の解決は、CICS リソースには影響を与えません。IBM MQ は、CICS をリカバリー・コーディネーターと見なし、WebSphere MQ が再始動されると、ログ・レコードにコミットの始まりがマークされているかどうかに応じて、自動的に各単位をコミットまたはバックアウトします。未確定オブジェクトが存在しても、WebSphere MQ の再接続中に CICS リソースはロックされません。

未解決の作業単位

接続に IBM MQ キュー共用グループを使用しており、接続の MQCONN 定義で RESYNCMEMBER(NO) を指定した場合、CICS は同じキュー・マネージャーへの再接続を 1 回しか試行しません。その試行に失敗すると、CICS は、キュー共用グループの任意の適格なメンバーに接続します。CICS 領域への接続に使用できるのは、現在 CICS 領域と同じ LPAR でアクティブになっているキュー・マネージャーです。CICS が異なるキュー・マネージャーに接続すると、未確定の作業単位を解決できず、CICS は、未解決の作業単位が残っている場合、警告メッセージ DFHMQ2064 を出します。

場合によっては、CICS は、同じキュー・マネージャーに再接続しても、未確定作業単位を解決するための IBM MQ プロセスを実行できないことがあります。そのような場合、CICS は以下のいずれかのエラー・メッセージを出します。

- DFHMQ0404E
- DFHMQ0405E
- DFHMQ0406E
- DFHMQ0407E

これらのメッセージの後に、メッセージ DFHMQ0408I が続きます。

再始動後に未解決の単位が残っている場合は、[126 ページの『手動による CICS 作業単位の解決方法』](#)に説明されている方法で解決します。

IBM MQ は、CICS に対してグループ・リカバリー単位をサポートしており、WebSphere MQ 7.1 以降を使用している場合は、新規オプション RESYNCMEMBER(GROUPRESYNC) を使用できます。GROUPRESYNC オプションを指定すると、CICS はキュー共用グループ内の任意のローカル IBM MQ キュー・マネージャーに接続します。キュー・マネージャーは IBM MQ によって選択され、どのキュー・マネージャーが未確定作業単位を持っているか関係なく、すべての未確定作業単位を解決するように CICS に要求します。IBM MQ で作業単位が未処理になっている場合は、RESYNCMEMBER の設定を変更しないでください。変更すると作業単位を解決できなくなるためです。CICS で保留になっている作業単位は、リソース・マネージャーの修飾子で特定できます。RESYNCMEMBER(GROUPRESYNC) を使用する場合は、キュー共用グループの名前が修飾子になり、そうでない場合は、個々のキュー・マネージャーの名前が修飾子として使用されます。

- RESYNCMEMBER(GROUPRESYNC) を指定し、前の接続で RESYNCMEMBER(YES) または RESYNCMEMBER(NO) を指定した MQCONN 定義が使用されていた場合、IBM MQ 内に未解決の未確定作業単位があると、これらの未確定作業単位は、前の設定の RESYNCMEMBER に戻さない限り解決できません。CICS は、警告メッセージ DFHMQ2065 を出します。
- RESYNCMEMBER(YES) または RESYNCMEMBER(NO) を指定し、前の接続で RESYNCMEMBER(GROUPRESYNC) を指定した MQCONN 定義が使用されていた場合、IBM MQ 内に未解決の未確定作業単位があると、これらの未確定作業単位は、前の設定の RESYNCMEMBER に戻さない限り解決できません。CICS は、警告メッセージ DFHMQ2066 を出します。

手動による CICS 作業単位の解決方法

CICS-MQ アダプターが異常終了した場合、CICS および IBM MQ は、異常終了の原因になったサブシステムに応じて、動的にまたは再始動時に未確定リストを作成します。CICS が IBM MQ に接続したときに、1 つ以上の作業単位が解決されていない可能性があります。CICS が解決できない作業単位は、すべて IBM MQ コマンドを使用して手動で解決する必要があります。

このタスクについて

一部の作業単位が解決されていない場合、以下のいずれかのメッセージがコンソールに送信されます。

- DFHMQ0404E
- DFHMQ0405E
- DFHMQ0406E
- DFHMQ0407E
- DFHMQ0408I
- DFHMQ2064
- DFHMQ2065
- DFHMQ2066

CICS は、接続の始動中に解決されなかった作業単位の詳細を保持します。エントリーが IBM MQ によって提供されるリストに表示されなくなると、そのエントリーはページされます。

CICS が解決できない作業単位は、すべて IBM MQ コマンドを使用して手動で解決する必要があります。この手作業手順が必要になるのは、操作上のエラーやソフトウェアの問題により自動的に解決できない場合

に限られるため、手作業手順が使用されることはめったにありません。未確定の解決中に検出された不整合をすべて調査する必要があります。

作業単位を解決するには、以下の手順で行います。

手順

1. 次のコマンドを使用して、IBM MQ から作業単位のリストを入手します。

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

以下のメッセージを受け取ります。

```
CSQM201I +CSQ1 CSQMDRTC  DISPLAY CONN DETAILS
CONN(BC85772CBE3E0001)
EXTCONN(C3E2D8C3C7D9F0F940404040404040)
TYPE(CONN)
CONNOPTS(
  MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-04)
UOWLOGTI(10.17.44)
UOWSTDA(2005-02-04)
UOWSTTI(10.17.44)
UOWSTATE(UNRESOLVED)
NID(IYRCSQ1 .BC8571519B60222D)
EXTURID(BC8571519B60222D)
QMURID(0000002BDA50)
URTYPE(CICS)
USERID(MQTEST)
APPLTAG(IYRCSQ1)
ASID(0000)
APPLTYPE(CICS)
TRANSID(GP02)
TASKNO(0000096)
END CONN DETAILS
```

CICS 接続の場合、NID (起点 ID) は、CICS アプリケーション ID と固有の番号 (同期点ログ・エントリーの書き込み時に CICS によって提供される) で構成されます。この固有の番号は、同期点処理時に CICS システム・ログと IBM MQ ログの両方に書き込まれるレコードに保管されます。この値は、CICS では作業トークンと呼ばれます。

2. CICS ログをスキャンして、特定の作業単位に関連したエントリーを検索します。
 - a) タスク関連のインストール済み環境の場合、作業トークン・フィールド (JCSRMTKN) がネットワーク ID から取得した値に等しい PREPARE レコードを検索します。
ネットワーク ID は、IBM MQ によって、DISPLAY CONN コマンドの出力に提供されます。
 - b) 作業単位の CICS ログ内の PREPARE レコードは、CICS タスク番号を提供します。この CICS タスクのログ上のその他のエントリーは、すべてこの番号を使用して見つけることができます。
 - c) ログをスキャンするには、CICS ジャーナル印刷ユーティリティー DFHJUP を使用できます。
このプログラムの使用の詳細については、[バッチ・ジョブ \(DFHJUP\) の使用によるログ・ストリームの読み取り](#)を参照してください。
3. IBM MQ ログをスキャンして、特定の作業単位に関連した NID を持つレコードを検索します。次に、そのレコードからの URID を使用して、この作業単位の残りのログ・レコードを入手します。
ログのスキャンには、IBM MQ ログ・レコード印刷プログラム (CSQ1LOGP) を使用できます。
IBM MQ ログのスキャン時には、IBM MQ の始動メッセージ CSQJ001I で、このセッションの開始 RBA が提供されることに注意してください。
4. 必要な場合、IBM MQ で未確定の解決を実行します。IBM MQ **RESOLVE INDOUBT** コマンドを使用すると、作業単位に対して作業アクションを実行するように IBM MQ に指示できます。特定の *connection-name* に関連したすべてのスレッドをリカバリーするには、NID(*) オプションを使用します。
このコマンドは、次のいずれかのメッセージを生成して、スレッドがコミットされたかバックアウトされたかを示します。

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED
```

タスクの結果

未確定の解決を実行する場合、CICS とアダプターは、作業単位をコミットまたはバックアウトするための IBM MQ へのコマンドを認識しません。コマンドの影響を受けるのは IBM MQ のリソースに限られるためです。ただし CICS は、IBM MQ によって解決されなかった未確定スレッドに関する詳細を保持します。この情報は、提供されたリストが空の場合、または CICS に詳細情報がある作業単位がリストに含まれていない場合、ページされます。

トリガー操作が動作しないとき

一時的な MQ エラー (キュー・マネージャーがリカバリー可能なカップリング・ファシリティ障害など) が原因でトリガー・モニターがキュー・マネージャーからメッセージを取得できない場合、トリガー・モニターは 1 分間隔で最大 1 時間にわたって取得要求を再試行します。障害が 1 時間を超えて続く場合、トリガー・モニターは終了されます。

トリガー・モニターがプログラムを開始できないか、またはキュー・マネージャーがトリガー・メッセージを送達できない場合は、プログラムは起動されません。例えば、プロセス・オブジェクトのアプリケーション ID では、プログラムをバックグラウンドで開始するように指定する必要がありますが、これが指定されていない場合、トリガー・モニターはプログラムを開始できません。

トリガー・メッセージが作成されても開始キューに書き込むことができない場合 (例えば、キューが満杯であるか、トリガー・メッセージの長さがその開始キューに指定されているメッセージの最大長を超えているため)、トリガー・メッセージは代わりに送達不能 (未配布メッセージ) キューに書き込まれます。

送達不能キューへの PUT 操作を正常に完了できない場合、トリガー・メッセージは廃棄され、警告メッセージが z/OS コンソールまたはシステム・オペレーターに送られるか、エラー・ログに書き込まれます。

トリガー・メッセージを送達不能キューに書き込むと、そのキュー用のトリガー・メッセージが生成される場合があります。この 2 つ目のトリガー・メッセージは、メッセージを送達不能キューに追加すると、破棄されます。

プログラムは正常に起動されたがキューからメッセージを受け取る前に異常終了した場合は、トレース・ユーティリティ (例えば、プログラムが CICS 下で実行されている場合は CICS AUXTRACE) を使用して、障害の原因を見つけてください。

第 10 章 CICS-MQ ブリッジのトラブルシューティング

エラーが発生した場合に CICS-MQ ブリッジが取る特定のアクション、およびユーザーがブリッジの問題を識別して解決するために取ることができるアクションがあります。

CICS-MQ ブリッジでのエラーの処理方法

CICS-MQ ブリッジでは、IBM MQ アプリケーションが、MQ 要求メッセージを MQ ブリッジ・キューに置く (MQPUT) ことによって、CICS DPL プログラムまたは CICS 3270 トランザクションを実行することができます。CICS-MQ ブリッジでは、MQGET 要求を発行してキューからメッセージを取り出した後、そのメッセージ・コンテンツ、つまり、MQ メッセージ・ヘッダーのないメッセージをターゲット・プログラムまたはトランザクションに渡します。

一時的な MQ エラー (キュー・マネージャーがリカバリー可能なカップリング・ファシリティ障害など) が原因で CICS-MQ ブリッジがキュー・マネージャーからメッセージを取得できない場合、CICS-MQ ブリッジは 1 分間隔で最大 1 時間にわたって取得要求を再試行します。障害が 1 時間を超えて続く場合、CICS-MQ ブリッジは終了されます。

通常、アプリケーションやトランザクションではメッセージを処理して、その応答を CICS-MQ ブリッジに返します。要求メッセージが応答先キューを指定している場合、CICS-MQ ブリッジでは、この応答に MQ 応答メッセージ・ヘッダーを追加し、MQPUT 要求を介して、この MQ 応答メッセージを応答先キューに入れます。

アプリケーションまたはトランザクションがエラー (例えば、要求内のデータの欠落や正しくないデータ) を検出した場合は、CICS-MQ ブリッジに応答メッセージを返すことができます。CICS-MQ ブリッジは、通常応答を処理する場合と同じ方法で、このエラー応答を処理します。

CICS-MQ ブリッジでは、IBM MQ アプリケーションが処理できない、または処理しないエラーの一部を処理できます。以下に例を示します。

- メッセージ・ヘッダーに誤った文字コーディングが指定されている。
- メッセージに、MQ または MQ ブリッジの要件に準拠しないメッセージ・ヘッダーが含まれている。
- トランザクションまたはそのプログラムがインストールされていない。
- メッセージのセキュリティ資格認定によって、トランザクションの開始が許可されない。
- 例外条件によって、アプリケーションまたはブリッジ自体の異常終了 (ABEND) が発生する。

一般に、CICS-MQ ブリッジでは、このようなエラーの場合に以下の手順を実行します。

1. DFHMQ07nn メッセージを CSMT 一時データ・キュー、CICS ジョブ・ログ、またはその両方に書き込みます。
2. トランザクション・ダンプを実行します。
3. **SYNCPOINT ROLLBACK** を発行します。
4. ブリッジが再試行しない要求メッセージを処理します。

詳細については、130 ページの『エラーが発生した場合に CICS-MQ ブリッジによって実行されるアクション』を参照してください。

MQ ブリッジでの作業単位のロールバックの実行方法

標準の同期点 MQGET は可逆的です。作業単位によって SYNCPOINT が発行されると、MQ はキューからのメッセージの除去を完了します。作業単位によって **SYNCPOINT ROLLBACK** が発行されると、MQ はメッセージをキューに返します。

一時的な状態の場合、このメカニズムにより、CICS-MQ ブリッジは自動的にリカバリーすることができます。MQ は要求メッセージをブリッジ・キューに返し、CICS-MQ ブリッジは、同じメッセージを再度取得して処理します。

ただし、一時的でない状態の場合、このメカニズムは無意味にシステム・リソースを使用し、(例えば) MQ 期限処理によってメッセージが削除されるまで、同じ要求メッセージを繰り返し再処理します。

この問題の解決に役立てるために、CICS-MQ ブリッジで再試行回数を制限することができます。これにより、CICS-MQ ブリッジは、一時的な状態から自動的にリカバリーすることができ、一時的な状態でないときに無限に再試行されることを防止できます。

ブリッジ・キューのバックアウトしきい値 (BOTHRESH) 属性では、再試行の最大回数を指定します。つまり、**SYNCPOINT ROLLBACK** が同じメッセージをブリッジ・キューに返すことができる最大回数です。デフォルトはゼロです。これは、**SYNCPOINT ROLLBACK** がブリッジ・キューにメッセージを返さないことを意味します。

CICS-MQ ブリッジは再試行しないメッセージをどのように処理しますか？

CICS-MQ ブリッジがブリッジ・キューに単にメッセージを戻すことができない場合 (バックアウト・カウントがバックアウトしきい値より大きいメッセージなど)、ブリッジはメッセージを次のように処理します。

1. バックアウト・リキュー・キューがキュー・マネージャーに定義されている場合、ブリッジでは、まず、そのメッセージをバックアウト・リキュー・キューに書き込むを試みます。
2. ステップ 1 に失敗するか、ステップ 1 を行うことが不可能な場合、ブリッジは代わりに送達不能キュー (キュー・マネージャーに定義されている場合) を使用することを試みます。
3. ステップ 2 に失敗するか、ステップ 2 を行うことが不可能な場合、ブリッジはメッセージの永続性属性を検査します。
 - メッセージが永続的ではない場合、ブリッジはメッセージを破棄します。
 - メッセージが永続的である場合、ブリッジは異常終了します。

CICS-MQ ブリッジをデバッグする方法

CICS-MQ ブリッジによってメッセージが予期したとおりに処理されない場合、または、ブリッジ・タスクが異常終了するか予期せず終了した場合は、[132 ページの『CICS-MQ ブリッジのデバッグ』](#)のトラブルシューティングの指針に従ってブリッジをデバッグできます。

エラーが発生した場合に CICS-MQ ブリッジによって実行されるアクション

CICS-MQ ブリッジでは、エラーを検出すると、一連のアクションを実行します。

[131 ページの図 12](#) は、エラーを処理するために CICS-MQ によって実行される一連のアクションを示しています。

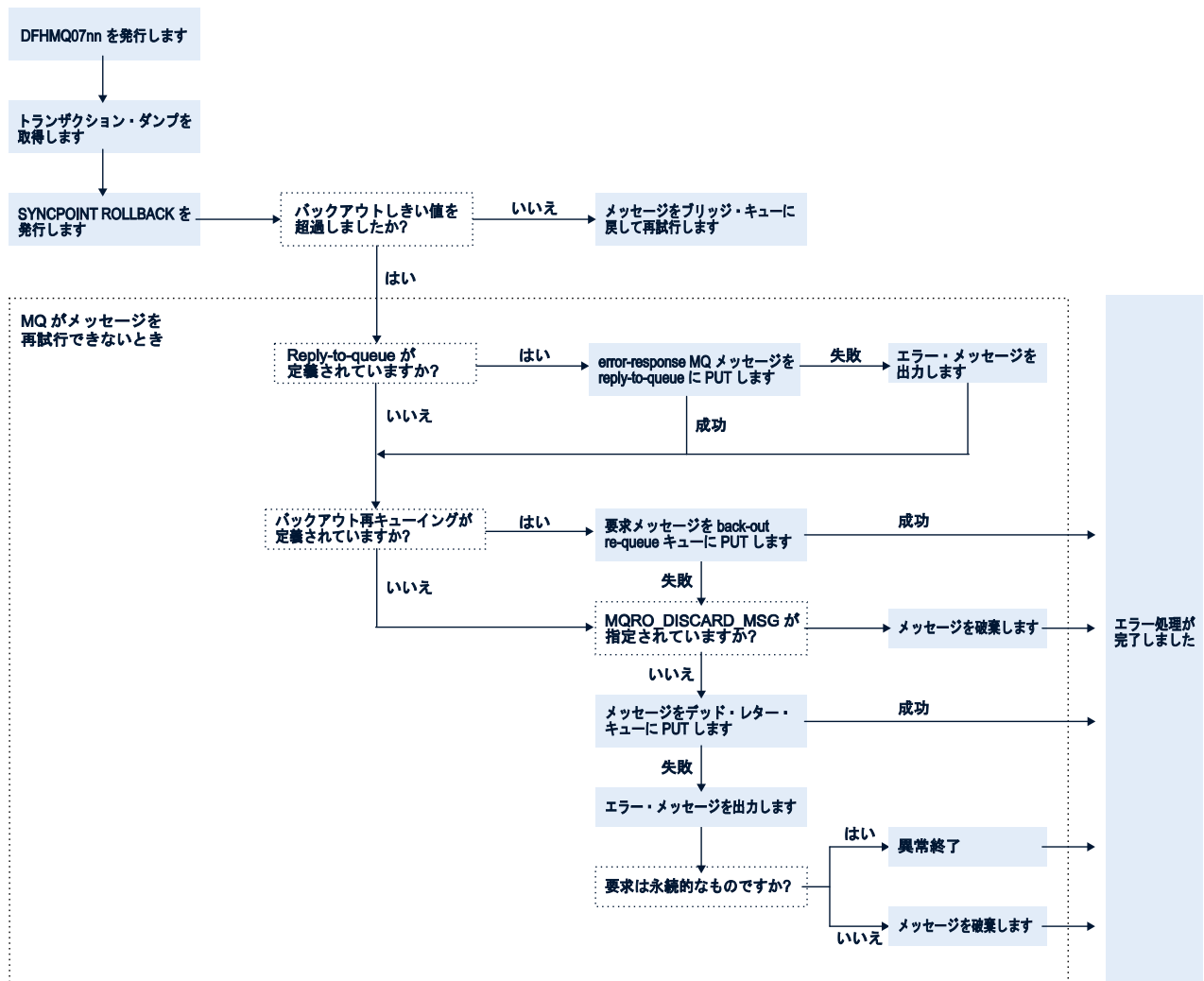


図 12. CICS-MQ のエラー処理ワークフロー

エラーが発生すると、CICS-MQ ブリッジでは以下のアクションを実行します。

1. DFHMQ07nn メッセージを CSMT 一時データ・キュー、CICS ジョブ・ログ、またはその両方に書き込みます。
2. トランザクション・ダンプを実行します。
3. SYNCPOINT ROLLBACK を発行します。
4. メッセージのバックアウト・カウントがまだバックアウトしきい値に達していない場合、SYNCPOINT ROLLBACK 処理によって要求メッセージがブリッジ・キューに返されるため、CICS-MQ ブリッジでは要求メッセージを再び取得し、処理を再試行できます。このエラーの処理が完了しました。

メッセージのバックアウト・カウントがバックアウトしきい値を超える場合は、ステップ [132 ページの『5』](#) およびステップ [132 ページの『6』](#) に進みます。

5. 要求メッセージが応答先キューを指定している場合は、エラー応答 MQ メッセージを作成し、それを応答先キューに入れます。このエラー応答メッセージの PUT 操作が失敗すると、エラー・メッセージが出力されます。このエラーの処理が完了しました。

注：

- 要求メッセージに MQCIH が含まれていない場合にも、このエラー応答 MQ メッセージには MQCIH が含まれます。
 - CICS-MQ ブリッジは、要求メッセージ内の MQCIH_PASS_EXPIRATION オプションおよび MQRO_PASS_DISCARD_AND_EXPIRY オプションをチェックして処理します。
6. ブリッジ・キューに対して定義されたバックアウト・リキュー・キューがある場合は、要求メッセージをバックアウト・リキュー・キューに入れます。PUT 操作が正常に実行された場合、このエラーに対する処理は完了します。

PUT 操作が失敗するか、ブリッジ・キューに対して定義されたバックアウト・リキュー・キューがない場合は、ステップ [132 ページの『7』](#) に進みます。

7. 要求メッセージで MQRO_DISCARD_MSG が指定されていない場合は、メッセージを送達不能キューに入れます。MQRO_DISCARD_MSG は、送達不能キューにメッセージを書き込まないようにブリッジに指示します。

PUT 操作が失敗した場合、ブリッジはエラー・メッセージを出し、要求が持続する場合は異常終了 (ABEND) し、メッセージが非永続の場合はメッセージを破棄します。

このエラーの処理が完了しました。

注：この要求メッセージは、送達不能キューに書き込まれるとき、MQDLH とともに追加されます。結果のメッセージは、送達不能キューに対して長すぎる場合には切り捨てられることがあります。

CICS-MQ ブリッジのデバッグ

CICS-MQ ブリッジによってメッセージが予期したとおりに処理されない場合、または、ブリッジ・タスクが異常終了するか予期せず終了した場合は、これらのトラブルシューティングの指針がブリッジのデバッグに役立ちます。

発生したエラーは何ですか。

- [132 ページの『メッセージはブリッジ要求キューに PUT されるが、ブリッジ・モニターによって処理されない』](#)
- [133 ページの『インバウンド・メッセージはブリッジ・モニターによって要求キューから取り出されるが、CICS DPL プログラムまたは CICS トランザクションの実行に失敗する』](#)
- [133 ページの『CICS-MQ ブリッジ・タスクの異常終了』](#)
- [134 ページの『CICS-MQ ブリッジ・モニターの予期しない終了』](#)

メッセージはブリッジ要求キューに PUT されるが、ブリッジ・モニターによって処理されない

1. ブリッジ・モニターが稼働していることを確認します。**CEMT I TASK** を発行して、CKBR またはブリッジ・モニター用に使用している他のトランザクション ID を検索します。

ブリッジ・モニターは稼働していないが、起動されることを期待している場合は、ブリッジ要求キューのトリガー・オプションが正しいか確認します。キュー属性 TRIGTYPE(FIRST) を使用してください。

ブリッジ・モニターは稼働していたものの、今は稼働していない場合は、CICS CSMT の出力、およびブリッジ・モニターが稼働しているものと予想するすべての CICS 領域のジョブ・ログを調べて、ブリッジ・モニターを終了させる原因になったエラーがないか確認します。

2. ブリッジ要求キューが QSGDISP(SHARED) で定義されている場合、INDXTYPE(CORRELID) も指定されていることを確認します。
3. 処理中ではないインバウンド・メッセージをブラウズして、MQMD.MsgId と MQMD.CorrelId の値が正しいことを確認します。これが作業単位または疑似会話の最初のメッセージである場合、MQMD.CorrelId の値は MQCI_NEW_SESSION に設定され、MQMD.MsgId は MQMI_NONE (2 値ゼロ) に設定されていなければなりません。
4. これが作業単位または疑似会話の最初のメッセージではない場合は、アプリケーションが前の応答メッセージにエラーがないか適切にチェックしたことを確認します。最小限として、アプリケーションは MQCIH 内の以下のフィールドをチェックする必要があります。
 - MQCIH.ReturnCode
 - MQCIH.CompCode
 - MQCIH.TaskEndStatus
 - MQCIH.AbendCode
 - MQCIH.ErrorOffset

[最上部に移動...](#)

インバウンド・メッセージはブリッジ・モニターによって要求キューから取り出されるが、CICS DPL プログラムまたは CICS トランザクションの実行に失敗する

1. CICS MSGUSR ログの出力を確認します。この出力には、ほぼ常に DPL プログラムまたはトランザクションの実行に失敗した理由が報告されます。一般的な理由には、以下のものがあります。
 - プログラムまたはトランザクションが CICS に対して定義されていない。CEDA を使用してプログラムまたはトランザクションを定義し、ブリッジ・アプリケーションを再度実行してください。
 - プログラムまたはトランザクションを実行する権限が不十分である。CICS-MQ ブリッジによって使用される認証レベルの制御方法の詳細は、[CICS-MQ ブリッジのセットアップ](#)に記載されています。
2. ブリッジ・モニターが応答キューに送信したメッセージを確認します。エラーが発生している場合、MQCIH.Format フィールドが MQFMT_STRING に設定されており、エラー・メッセージがベクトルの代わりに MQCIH に追加されている可能性があります。
3. 送達不能キューを調べて、ブリッジ・モニターがそこに応答メッセージを送信していないか確認します。送信されていて、MQMD.MsgId と MQMD.CorrelId の値も正しい場合は、MQDLH.Reason の値を確認します。この値は、通常、障害の理由を示すフィードバック・コードに設定されます。

フィードバック・コードに関する情報 (CICS-MQ ブリッジに固有のものも含む) は、[CICS メッセージ](#)を参照してください。

[最上部に移動...](#)

CICS-MQ ブリッジ・タスクの異常終了

アウトバウンド・メッセージのフィールド MQCIH.AbendCode に、異常終了コードが設定されます。さらに、CICS MSGUSR ログの出力にも、失敗したブリッジ・タスクの異常終了コードが報告されます。

異常終了については、[CICS メッセージ](#)に文書化されています。

一般的な異常終了コードの一部のものは、以下のように対応できます。

ABRG

インバウンド・メッセージに無効なブリッジ機能トークンが指定されました。最初のインバウンド・メッセージでは、必ず、フィールド MQCIH.Facility に値 MQCFAC_NONE を指定し、MQCIH.FacilityKeepTime に非ゼロ値を指定する必要があります。CICS は、機能トークンをフィールド

MQCIH.Facility に返します。この値を疑似会話の後続のすべてのインバウンド・メッセージに使用できません。

ABXH

バッファ・オプションを使用した受信が指定された場合に brmq_re_buffer_indicator が N に設定されたか、バッファ・オプションなしの受信が指定された場合に brmq_re_buffer_indicator が Y に設定されました。

MBRJ

MQCIH に無効なデータがあります。フィールドごとに MQCIH の値を確認して、範囲外の値がないか調べます。MBRJ は、長さの不一致が原因でも発生します。例えば、brmq_vector_length の値がデータ・ベクトルの長さと一致しない場合、または CICS ヘッダーとベクトルに十分なデータが含まれていない場合などです。

MBRN

メッセージが予想された長さより短くなっています。すべてのベクトル構造に 1 つまたは 2 つのデータ長さフィールドがあります。最初のフィールドは、すべてのベクトルの標準ヘッダーの最初のフルワード・フィールドです。このフィールドは、可変長データを含めたベクトル全体の長さと等しくなければなりません。一部のベクトルには、可変長データの長さだけを示す別のフルワードの長さフィールドも含まれています。これらの値が実際より多くのデータを示している場合、ブリッジ・タスクは MBRN で異常終了します。

MBRO および MBRP

ベクトル構造 (可変長データではなく) にエラーがあります。MQCIH フィールド ERROROFFSET に、エラーのあるフィールドのオフセットが示されます。ベクトル内のフィールドの値を許容値 ([Link3270 メッセージ形式で説明](#)) と対比して確認してください。

[最上部に移動...](#)

CICS-MQ ブリッジ・モニターの予期しない終了

一部のエラーは、ブリッジ・モニター・トランザクション CKBR が予期せずに終了する原因になります。起動されたキューを使用してモニターを開始しているときに、ブリッジ要求キューにメッセージが残っている場合、CKTI トランザクションは CKBR の再始動を試みる場合があります。元のエラーが存続している場合、CKBR 障害がループする可能性があります。このループを停止するために、根本の問題を診断して修正するまでの間は、要求キューの TriggerControl 属性をオフに設定してください。

ブリッジ・モニターが失敗する可能性があるのは、キューまたは CICS トランザクションにアクセスするための十分な権限がない場合、送達不能キューに書き込めない場合、あるいは CICS または IBM MQ のサービスの実行中に問題が発生した場合です。

[最上部に移動...](#)

第 11 章 MQCIH – CICS-MQ ブリッジ・ヘッダー

MQCIH 構造体は、IBM MQ for z/OS を介して CICS-MQ ブリッジに送信されるメッセージの先頭に置く情報を記述します。

可用性

AIX®、HP-UX、z/OS、Solaris、Linux®、Windows、および IBM MQ の各システムに接続された、各システムのクライアント。

C++ アプリケーションの場合、ImqCICSBridgeHeader クラスは、MQCIH データ構造体の特定の機能をカプセル化します。[147 ページの『CICS-MQ ブリッジ用の IBM MQ C++ メッセージ・ヘッダー』](#)を参照してください。

形式名

MQFMT_CICS。

バージョン

MQCIH の現行バージョンは MQCIH_VERSION_2 です。サポートされるプログラム言語用に提供されているヘッダー・ファイル、COPY ファイル、および INCLUDE ファイルには、MQCIH の最新バージョンが含まれており、「Version」フィールドの初期値は MQCIH_VERSION_2 に設定されています。

バージョンが MQCIH_VERSION_2 より前のものである場合、フィールド「CursorPosition」、「ErrorOffset」、「InputItem」、および「Reserved4」は存在しません。

文字セットとエンコード方式

MQCIH 構造体およびアプリケーション・メッセージ・データに使用される文字セットとエンコード方式は、以下の特別条件に従います。

- CICS-MQ ブリッジ・キューを所有するキュー・マネージャーに接続するアプリケーションは、そのキュー・マネージャーの文字セットとエンコード方式で記述されている MQCIH 構造体を提供する必要があります。その理由は、この場合、MQCIH 構造体のデータ変換が実行されないためです。
- 他のキュー・マネージャーに接続するアプリケーションでは、サポートされているいずれかの文字セットとエンコード方式で記述されている MQCIH 構造体を提供することができます。その理由は、CICS-MQ ブリッジ・キューを所有するキュー・マネージャーに接続されている受信メッセージ・チャネル・エージェントが、MQCIH 構造体を変換するためです。
- MQCIH 構造体の後続くアプリケーション・メッセージ・データは、MQCIH 構造体と同じ文字セットとエンコード方式で記述されていなければなりません。MQCIH 構造体の CodedCharSetId フィールドおよび Encoding フィールドを使用して、そのアプリケーション・メッセージ・データの文字セットとエンコード方式を指定することはできません。データがキュー・マネージャーにサポートされる組み込み形式のいずれでもない場合、ユーザーは、アプリケーション・メッセージ・データを変換するためにデータ変換出口を提供する必要があります。

使用法

アプリケーションに必要な値が初期値と同じであり、ブリッジが AUTH=LOCAL または AUTH=IDENTIFY の設定で動作している場合は、メッセージから MQCIH 構造体を省略することができます。それ以外の場合はいずれも、この構造体が存在している必要があります。

ブリッジは MQCIH 構造体のバージョン 1 とバージョン 2 のどちらをも受け入れますが、3270 トランザクションについては、バージョン 2 の構造体を使用する必要があります。

要求フィールドと記されているフィールドでは、ブリッジに送信されるメッセージ内の値を、そのアプリケーション側で適切に設定しておく必要があります。これらのフィールドはブリッジに対する入力として使用されます。

応答フィールドと記されているフィールドは、CICS-MQ ブリッジにより、そのブリッジがアプリケーションに送信する応答メッセージ内に設定されます。エラー情報は、「ReturnCode」、「Function」、「CompCode」、「Reason」、および「AbendCode」のフィールドに戻されます。ただし、すべてのケースでこれらのすべてのフィールドが設定されているとは限りません。次の表は、ReturnCode の各種の値に対して設定されるフィールドを示しています。

表 14. MQCIH 構造体のエラー情報フィールドの内容				
ReturnCode	Function	CompCode	Reason	AbendCode
MQCRC_OK	–	–	–	–
MQCRC_BRIDGE_ERROR	–	–	MQFB_CICS_*	–
MQCRC_MQ_API_ERROR MQCRC_BRIDGE_TIMEOUT	MQ 呼び出し名	MQ CompCode	MQ Reason	–
MQCRC_CICS_EXEC_ERROR MQCRC_SECURITY_ERROR MQCRC_PROGRAM_NOT_AVAILABLE MQCRC_TRANSID_NOT_AVAILABLE	CICS EIBFN	CICS EIBRESP	CICS EIBRESP2	–
MQCRC_BRIDGE_ABEND MQCRC_APPLICATION_ABEND	–	–	–	CICS ABCODE

初期値および言語宣言

この表は、CICS-MQ ブリッジ・ヘッダー (MQCIH) 内のフィールドの初期値と、MQCIH 構造体でのそれらの順序を示しています。

表 1. MQCIH 内のフィールドの初期値		
フィールド名	定数の名前	定数の値
StrucId	MQCIH_STRUC_ID	'CIHb'
Version	MQCIH_VERSION_2	2
StrucLength	MQCIH_LENGTH_2	180
Encoding	なし	0
CodedCharSetId	なし	0
Format	MQFMT_NONE	ブランク
Flags	MQCIH_NONE	0
ReturnCode	MQCRC_OK	0
CompCode	MQCC_OK	0
Reason	MQRC_NONE	0
UOWControl	MQCUOWC_ONLY	273
GetWaitInterval	MQCGWI_DEFAULT	-2
LinkType	MQCLT_PROGRAM	1
OutputDataLength	MQCODL_AS_INPUT	-1
FacilityKeepTime	なし	0
ADSDescriptor	MQCADSD_NONE	0
ConversationalTask	MQCCT_NO	0

表 1. MQCIH 内のフィールドの初期値
(続き)

フィールド名	定数の名前	定数の値
TaskEndStatus	MQCTES_NOSYNC	0
Facility	MQCFAC_NONE	NULL
Function	MQCFUNC_NONE	ブランク
AbendCode	なし	ブランク
Authenticator	なし	ブランク
Reserved1	なし	ブランク
ReplyToFormat	MQFMT_NONE	ブランク
RemoteSysId	なし	ブランク
RemoteTransId	なし	ブランク
TransactionId	なし	ブランク
FacilityLike	なし	ブランク
AttentionId	なし	ブランク
StartCode	MQCSC_NONE	ブランク
CancelCode	なし	ブランク
NextTransactionId	なし	ブランク
Reserved2	なし	ブランク
Reserved3	なし	ブランク
CursorPosition	なし	0
ErrorOffset	なし	0
InputItem	なし	0
Reserved4	なし	0

注:

1. シンボル **b** は、単一のブランク文字を表します。
2. C プログラミング言語では、マクロ変数 `MQCIH_DEFAULT` に、この表にリストされている値が含まれます。このマクロ変数を以下のように使用して、構造体のフィールドに初期値を設定します。

```
MQCIH MyCIH = {MQCIH_DEFAULT};
```

AbendCode (MQCHAR4)

このフィールドに戻される値は、`ReturnCode` フィールドの値が `MQCRC_APPLICATION_ABEND` または `MQCRC_BRIDGE_ABEND` の場合にのみ意味を持ちます。その場合、`AbendCode` には CICS ABCODE 値が入ります。

このフィールドは応答フィールドです。このフィールドの長さは、`MQ_ABEND_CODE_LENGTH` によって指定されます。このフィールドの初期値は 4 個のブランク文字です。

ADSDescriptor (MQLONG)

この標識は、SEND 要求および RECEIVE BMS 要求で ADS 記述子を送信するかどうかを指定します。

以下の値が定義されます。

MQCADSD_NONE

ADS 記述子の送受信を行わない。

MQCADSD_SEND

ADS 記述子を送信する。

MQCADSD_RECV

ADS 記述子を受信する。

MQCADSD_MSGFORMAT

ADS 記述子のメッセージ形式を使用する。このオプションは、ADS 記述子の長形式を使用して ADS 記述子を送受信します。この長形式には、4 バイト境界で位置合わせされるフィールドがあります。

ADSDescriptor フィールドは、以下のように設定します。

- ADS 記述子を使用していない場合は、フィールドを MQCADSD_NONE に設定します。
- ADS 記述子を使用しており、各環境における CCSID が同一の場合は、このフィールドを MQCADSD_SEND と MQCADSD_RECV の合計に設定します。
- ADS 記述子を使用しており、各環境における CCSID が異なっている場合には、このフィールドを MQCADSD_SEND、MQCADSD_RECV、および MQCADSD_MSGFORMAT の合計に設定します。

これは、3270 トランザクションにのみ使用される要求フィールドです。このフィールドの初期値は MQCADSD_NONE です。

AttentionId (MQCHAR4)

このフィールドは、トランザクションの開始時の AID キーの初期値です。

値は 1 バイト値で、左寄せされます。

この要求フィールドは、3270 トランザクションにのみ使用されます。このフィールドの長さは MQ_ATTENTION_ID_LENGTH によって指定されます。このフィールドの初期値は 4 個のブランクです。

Authenticator (MQCHAR8)

このフィールドは、パスワードまたはパスチケットを保持します。ユーザー ID 認証が CICS-MQ ブリッジに対してアクティブである場合、Authenticator は、メッセージの送信側を認証するために、MQMD ID コンテキスト内のユーザー ID と一緒に使用されます。

このフィールドは、要求フィールドです。このフィールドの長さは、MQ_AUTHENTICATOR_LENGTH によって指定されます。このフィールドの初期値は 8 個のブランクです。

CancelCode (MQCHAR4)

このフィールドは、トランザクション (通常、より多くのデータを要求する会話型トランザクション) を停止するために使用される異常終了コードを保持します。それ以外の場合、このフィールドはブランクに設定されます。

このフィールドは、3270 トランザクションにのみ使用される要求フィールドです。このフィールドの長さは MQ_CANCEL_CODE_LENGTH によって指定されます。このフィールドの初期値は 4 個のブランクです。

CodedCharSetId (MQLONG)

このフィールドは予約済みです。したがって、値に意味はありません。

このフィールドの初期値は 0 です。

CompCode (MQLONG)

このフィールドに戻される値は、ReturnCode によって異なります。

[136 ページの表 14](#) を参照してください。

このフィールドは応答フィールドです。このフィールドの初期値は MQCC_OK です。

ConversationalTask (MQLONG)

この標識により、タスクは、より多くの情報を求める要求を出すか、タスクを異常終了するか指定することができます。

値は、以下のいずれかでなければなりません。

MQCCT_YES

タスクは会話型です。複数のメッセージを使用してトランザクションのベクトルを提供できる場合は、この値を指定します。

MQCCT_NO

タスクは会話型ではありません。この CICS トランザクションに必要なすべてのベクトルが入力メッセージに提供される場合は、この値を指定します。

この要求フィールドは、3270 トランザクションにのみ使用されます。このフィールドの初期値は MQCCT_NO です。

CursorPosition (MQLONG)

このフィールドは、トランザクション開始時の初期カーソル位置を保持します。その後、会話型トランザクションの場合、カーソル位置は RECEIVE ベクトルで示されます。

この要求フィールドは、3270 トランザクションにのみ使用されます。このフィールドの初期値は 0 です。Version が MQCIH_VERSION_2 より小さい場合、このフィールドは存在しません。

Encoding (MQLONG)

このフィールドは予約済みです。したがって、値に意味はありません。

このフィールドの初期値は 0 です。

ErrorOffset (MQLONG)

このフィールドは、ブリッジ出口で検出された無効なデータの位置を保持します。このフィールドは、メッセージの先頭から無効なデータの位置までのオフセットを示します。

この応答フィールドは、3270 トランザクションにのみ使用されます。このフィールドの初期値は 0 です。Version が MQCIH_VERSION_2 より小さい場合、このフィールドは存在しません。

Facility (MQBYTE8)

このフィールドは、8 バイトのブリッジ機能トークンを保持します。ブリッジ機能トークンにより、疑似会話内の複数のトランザクションが同じブリッジ機能 (仮想 3270 端末) を使用することができます。

疑似会話内の最初のメッセージまたは唯一のメッセージに、値 MQCFAC_NONE を設定します。この値は、このメッセージに新規ブリッジ機能を割り振るよう、CICS に指示します。入力メッセージにゼロ以外の FacilityKeepTime 値が指定された場合、応答メッセージでブリッジ機能トークンが戻されます。その後、疑似会話での後続の入力メッセージは、同じブリッジ機能トークンを使用する必要があります。

以下の特殊値が定義されます。

MQCFAC_NONE

機能トークンの指定なし。

C プログラミング言語の場合、定数 MQCFAC_NONE_ARRAY も定義されます。この定数は MQCFAC_NONE と同じ値ですが、ストリングではなく文字の配列です。

このフィールドは、要求フィールドでも応答フィールドでもあり、3270 トランザクションにのみ使用されます。このフィールドの長さは、MQ_FACILITY_LENGTH によって指定されます。このフィールドの初期値は MQCFAC_NONE です。

FacilityKeepTime (MQLONG)

このフィールドは、ユーザー・トランザクションの終了後にブリッジ機能が保持される時間の長さ (秒単位) を指定します。

予想される疑似会話の期間に対応する値を指定します。疑似非会話型トランザクションの場合、この値はゼロでなければなりません。非会話型トランザクションの場合、値はゼロでなければなりません。

この要求フィールドは、3270 トランザクションにのみ使用されます。このフィールドの初期値は 0 です。

FacilityLike (MQCHAR4)

このフィールドは、ブリッジ機能のモデルとして使用される設置済み端末の名前を指定します。

値としてブランクを指定すると、FacilityLike がブリッジ・トランザクション・プロファイル定義からとられるか、またはデフォルト値が使用されます。

この要求フィールドは、3270 トランザクションにのみ使用されます。このフィールドの長さは MQ_FACILITY_LIKE_LENGTH によって指定されます。このフィールドの初期値は 4 個のブランクです。

Flags (MQLONG)

このフィールドは、メッセージのすべてのフラグを保持します。

このフィールドの値は、以下のいずれかでなければなりません。

MQCIH_NONE

フラグなし。

MQCIH_PASS_EXPIRATION

応答メッセージに、以下のものが含まれます。

- 要求メッセージと同じ有効期限レポート・オプション
- 要求メッセージからの残りの存続時間。ブリッジの処理時間に対する調整は行われません。

この値を省略した場合、存続時間は「無制限」に設定されます。

MQCIH_REPLY_WITHOUT_NULLS

CICS/DPL プログラム要求の応答メッセージの長さは、DPL プログラムが戻す COMMAREA の末尾の末尾ヌル (X'00') を除外するよう調整されます。この値が設定されない場合、ヌルが意味を持つ可能性があり、COMMAREA 全体が戻されます。

MQCIH_SYNC_ON_RETURN

DPL 要求のための CICS リンクでは、SYNCONRETURN オプションを使用します。このオプションにより、CICS は、別の CICS 領域に送られた場合、プログラムが完了すると同期点をとります。ブリッジはどの CICS 領域に要求を送るかを指定しません。この制御は、CICS プログラム定義またはワークロード・balancing機能によって行われます。

このフィールドは、要求フィールドです。このフィールドの初期値は MQCIH_NONE です。

Format (MQCHAR8)

このフィールドは、MQCIH 構造体の後続くデータの IBM MQ 形式名を保持します。

MQPUT 呼び出しまたは MQPUT1 呼び出しでは、アプリケーションは、このフィールドをデータに適した値に設定する必要があります。このフィールドのコーディングの規則は、MQMD の「Format」フィールドの規則と同じです。

この形式名は、ReplyToFormat フィールドの値が MQFMT_NONE の場合、応答メッセージにも使用されます。

- DPL 要求の場合、Format は COMMAREA の形式名でなければなりません。

- 3270 要求の場合、Format は DFHMQDCI または CSCQBDCI でなければならず、ブリッジは、応答メッセージについて、形式を DFHMQDCO または CSCQBDCO に設定します。

これらの形式のデータ変換出口は、それらが実行されるキュー・マネージャーにインストールする必要があります。

要求メッセージによってエラー応答メッセージが生成される場合、エラー応答メッセージの形式名は MQFMT_STRING となります。

このフィールドは、要求フィールドです。このフィールドの長さは、MQ_FORMAT_LENGTH によって指定されます。このフィールドの初期値は MQFMT_NONE です。

Function (MQCHAR4)

このフィールドに戻される値は、ReturnCode によって異なります。

136 ページの表 14 を参照してください。IBM MQ 呼び出し名に Function が含まれている場合には、以下の値が使用可能です。

MQCFUNC_MQCONN
MQCONN 呼び出し

MQCFUNC_MQGET
MQGET 呼び出し

MQCFUNC_MQINQ
MQINQ 呼び出し

MQCFUNC_MQOPEN
MQOPEN 呼び出し

MQCFUNC_MQPUT
MQPUT 呼び出し

MQCFUNC_MQPUT1
MQPUT1 呼び出し

MQCFUNC_NONE
呼び出しなし

C プログラミング言語では、上記のすべてについて、定数 MQCFUNC_*_ARRAY も定義されます。この定数の値は、対応する MQCFUNC_* 定数と同じですが、ストリングではなく文字の配列を使用します。

このフィールドは応答フィールドです。このフィールドの長さは MQ_FUNCTION_LENGTH によって指定されます。このフィールドの初期値は MQCFUNC_NONE です。

GetWaitInterval (MQLONG)

このフィールドは、UOWControl の値が MQCUOWC_FIRST の場合にのみ適用されます。これにより、送信側アプリケーションでは、ブリッジで発行された MQGET 呼び出しが、このメッセージによって開始された作業単位についての 2 番目以降の要求メッセージを待機する、おおよその時間をミリ秒単位で指定できます。この値は、ブリッジで使用するデフォルトの待機間隔を指定変更します。

以下の特殊値を使用できます。

MQCGWI_DEFAULT
デフォルトの待機間隔。この値により、CICS-MQ ブリッジは、ブリッジの開始時に指定された時間間隔で待機します。

MQWI_UNLIMITED
無制限の待機間隔。

このフィールドは、要求フィールドです。このフィールドの初期値は MQCGWI_DEFAULT です。

InputItem (MQLONG)

このフィールドは予約済みです。値は 0 でなければなりません。

Version が MQCIH_VERSION_2 より小さい場合、このフィールドは表示されません。

LinkType (MQLONG)

このフィールドは、ブリッジがリンクを試みるオブジェクトのタイプを指定します。

値は、以下のいずれかでなければなりません。

MQCLT_PROGRAM

DPL プログラム

MQCLT_TRANSACTION

3270 トランザクション

このフィールドは、要求フィールドです。このフィールドの初期値は MQCLT_PROGRAM です。

NextTransactionId (MQCHAR4)

このフィールドは、ユーザー・トランザクション (通常、**EXEC CICS RETURN TRANSID**) によって戻される次のトランザクションの名前を保持します。次のトランザクションが存在しない場合、このフィールドはブランクに設定されます。

この応答フィールドは、3270 トランザクションにのみ使用されます。このフィールドの長さは、MQ_TRANSACTION_ID_LENGTH によって指定されます。このフィールドの初期値は 4 個のブランクです。

OutputDataLength (MQLONG)

このフィールドは、応答メッセージでクライアントに戻されるユーザー・データの長さを保持します。この長さには、8 バイトのプログラム名が含まれます。

要求が COMMAREA でデータをユーザー・プログラムに渡した場合、ユーザー・プログラムは COMMAREA で応答データを戻すことができます。応答データは要求データよりも長くすることができます。その場合には、OutputDataLength で応答データの長さを指定する必要があります。この長さには、8 バイトのプログラム名が含まれます。

要求がデータをチャネルまたはコンテナのどちらでユーザー・プログラムに渡しても、ユーザー・プログラムはすべての応答データをコンテナで戻します。コンテナ自身が、保持するデータの長さを示すため、OutputDataLength は使用されません。

注：メッセージ内のユーザー・データの長さは、MQCIH 構造体を除いたメッセージの長さです。

要求メッセージ内のユーザー・データの長さが OutputDataLength より短い場合は、LINK コマンドの DATALENGTH オプションが使用されます。このオプションにより、LINK は、別の CICS 領域に対して効率的に機能シップすることができます。

以下の特殊値を使用できます。

MQCODL_AS_INPUT

出力長は、入力長と同じです。

この値は、リンクされたプログラムに渡される COMMAREA のサイズを十分なものにするために、応答が要求されない場合でも、必要になることがあります。

この要求フィールドは、DPL プログラムにのみ使用されます。このフィールドの初期値は MQCODL_AS_INPUT です。

Reason (MQLONG)

このフィールドに戻される値は、ReturnCode によって異なります。

[136 ページの表 14](#) を参照してください。

このフィールドは応答フィールドです。このフィールドの初期値は MQRC_NONE です。

RemoteSysId (MQCHAR4)

このフィールドは、要求を処理している CICS システムの CICS システム ID を保持します。このフィールドがブランクの場合、CICS システム 要求は、ブリッジ・モニターと同じ CICS システムで処理されます。使用された SYSID は応答メッセージで戻されます。

3270 疑似会話では、会話内の後続のすべてのメッセージが、初期応答で戻されるリモート SYSID を指定する必要があります。SYSID を指定する場合は、以下の特性を持っている必要があります。

- アクティブである
- IBM MQ 要求キューへのアクセス権限を持っている
- ブリッジ・モニターが稼働している CICS システムから、CICS ISC リンクによりアクセス可能である

RemoteTransId (MQCHAR4)

このフィールドは、オプションの要求フィールドです。このフィールドは、指定した場合、CICS START の RTRANSID 値として使用されます。

このフィールドの長さは、MQ_TRANSACTION_ID_LENGTH によって指定されます。

ReplyToFormat (MQCHAR8)

このフィールドは、現在のメッセージへの応答で送信される応答メッセージの IBM MQ 形式名を保持します。

要求が COMMAREA でデータをユーザー・プログラムに渡した場合、ユーザー・プログラムは COMMAREA で応答データを戻すことができます。応答データは、バイナリー形式でも文字形式でも可能です。ReplyToFormat には、どちらかを指定する WebSphere® MQ 形式名が含まれている必要があります。

要求がデータをチャネルまたはコンテナのどちらでユーザー・プログラムに渡しても、ユーザー・プログラムはすべての応答データをコンテナで戻します。コンテナ自身が、保持するデータの形式を示すため、ReplyToFormat は使用されません。

このフィールドのコーディングの規則は、MQMD の「Format」フィールドの規則と同じです。

この要求フィールドは、DPL プログラムにのみ使用されます。このフィールドの長さは、MQ_FORMAT_LENGTH によって指定されます。このフィールドの初期値は MQFMT_NONE です。

Reserved1 (MQCHAR8)

このフィールドは予約済みです。値は 8 個のブランクでなければなりません。

Reserved2 (MQCHAR8)

このフィールドは予約済みです。値は 8 個のブランクでなければなりません。

Reserved3 (MQCHAR8)

このフィールドは予約済みです。値は 8 個のブランクでなければなりません。

Reserved4 (MQLONG)

このフィールドは予約済みです。値は 0 でなければなりません。

Version が MQCIH_VERSION_2 より小さい場合、このフィールドは表示されません。

ReturnCode (MQLONG)

このフィールドは、ブリッジが実行する処理の結果を示す CICS ブリッジ・モニターからの戻りコードを保持します。

Function、CompCode、Reason、および AbendCode の各フィールドには、追加情報が含まれている場合があります。[136 ページの表 14](#) を参照してください。

値は、以下のいずれかです。

MQCRC_APPLICATION_ABEND

(5, X'005') アプリケーションが異常終了した。

MQCRC_BRIDGE_ABEND

(4, X'004') CICS ブリッジが異常終了した。

MQCRC_BRIDGE_ERROR

(3, X'003') CICS ブリッジがエラーを検出した。

MQCRC_BRIDGE_TIMEOUT

(8, X'008') 指定された時間内に、現行作業単位の 2 番目以降のメッセージを受信しなかった。

MQCRC_CICS_EXEC_ERROR

(1, X'001') EXEC CICS ステートメントでエラーが検出された。

MQCRC_MQ_API_ERROR

(2, X'002') MQ 呼び出しでエラーが検出された。

MQCRC_OK

(0, X'000') エラーなし。

MQCRC_PROGRAM_NOT_AVAILABLE

(7, X'007') プログラムが使用できない。

MQCRC_SECURITY_ERROR

(6, X'006') セキュリティー・エラーが発生した。

MQCRC_TRANSID_NOT_AVAILABLE

(9, X'009') トランザクションが使用できない。

このフィールドは応答フィールドです。このフィールドの初期値は MQCRC_OK です。

StartCode (MQCHAR4)

この標識は、ブリッジが端末トランザクションをエミュレートするのか、START によって開始されたトランザクションをエミュレートするのかを示します。

値は、以下のいずれかでなければなりません。

MQCSC_START

開始

MQCSC_STARTDATA

開始データ

MQCSC_TERMINPUT

端末入力

MQCSC_NONE

なし

C プログラミング言語では、上記のすべてについて、定数 MQCSC_*_ARRAY も定義されます。この定数の値は、対応する MQCSC_* 定数と同じですが、ストリングではなく文字の配列です。

ブリッジからの応答では、このフィールドは、NextTransactionId フィールド内の次のトランザクション ID に適した開始コードに設定されます。応答では、以下の開始コードが使用可能です。

- MQCSC_START
- MQCSC_STARTDATA
- MQCSC_TERMINPUT

このフィールドは、要求フィールドでも応答フィールドでもあります。

このフィールドは、3270 トランザクションにのみ使用されます。このフィールドの長さは MQ_START_CODE_LENGTH によって指定されます。このフィールドの初期値は MQCSC_NONE です。

StrucId (MQCHAR4)

このフィールドは、CICS 情報ヘッダー構造体の ID を保持します。

値は、以下のものでなければなりません。

MQCIH_STRUC_ID

CICS 情報ヘッダー構造体の ID。

C プログラミング言語の場合、定数 MQCIH_STRUC_ID_ARRAY も定義されます。この定数は MQCIH_STRUC_ID と同じ値ですが、ストリングではなく文字の配列です。

このフィールドは、要求フィールドです。このフィールドの初期値は MQCIH_STRUC_ID です。

StrucLength (MQLONG)

このフィールドは、CICS 情報ヘッダー構造体の長さを指定します。

値は、以下のいずれかでなければなりません。

MQCIH_LENGTH_1

バージョン 1 の CICS 情報ヘッダー構造体の長さ。

MQCIH_LENGTH_2

バージョン 2 の CICS 情報ヘッダー構造体の長さ。

以下の定数は、現行バージョンの長さを指定します。

MQCIH_CURRENT_LENGTH

CICS 情報ヘッダー構造体の現行バージョンの長さ。

このフィールドは、要求フィールドです。このフィールドの初期値は MQCIH_LENGTH_2 です。

TaskEndStatus (MQLONG)

このフィールドは、タスク終了時のユーザー・トランザクションの状況を示します。

以下の値のいずれかが戻されます。

MQCTES_NOSYNC

同期していない。

ユーザー・トランザクションはまだ完了しておらず、同期点をとっていません。この場合、MQMD 内の MsgType フィールドは MQMT_REQUEST です。

MQCTES_COMMIT

作業単位をコミットする。

ユーザー・トランザクションはまだ完了していませんが、最初の作業単位の同期点はとりました。この場合、MQMD 内の MsgType フィールドは MQMT_DATAGRAM です。

MQCTES_BACKOUT

作業単位をバックアウトする。

ユーザー・トランザクションはまだ完了していません。現行の作業単位はバックアウトされます。この場合、MQMD 内の MsgType フィールドは MQMT_DATAGRAM です。

MQCTES_ENDTASK

タスクを終了する。

ユーザー・トランザクションは終了(または異常終了)しました。この場合、MQMD 内の MsgType フィールドは MQMT_REPLY です。

このフィールドは、3270 トランザクションにのみ使用される応答フィールドです。このフィールドの初期値は MQCTES_NOSYNC です。

TransactionId (MQCHAR4)

このフィールドは、CICS で実行するユーザー・トランザクションのトランザクション ID、または CICS プログラムの実行で使用するトランザクション・コードを提供します。

LinkType フィールドの値が MQCLT_TRANSACTION の場合、TransactionId は、実行するユーザー・トランザクションのトランザクション ID です。この場合は、非ブランクの値を指定します。

LinkType フィールドの値が MQCLT_PROGRAM の場合、TransactionId は、該当する作業単位内のすべてのプログラムの実行で使用するトランザクション・コードです。値としてブランクを指定した場合は、CICS DPL ブリッジのデフォルト・トランザクション・コード (CKBP) が使用されます。値が非ブランクの場合は、CKBP か CKBC、あるいはその初期プログラムとして DFHMQBP0 または DFHMQBP3 を持つ CICS に対して定義したローカル・トランザクション ID のいずれかを指定する必要があります。TransactionId フィールドは、UOWControl フィールドの値が MQCUOWC_FIRST または MQCUOWC_ONLY の場合にのみ適用されます。

TransactionId フィールドは、要求フィールドです。このフィールドの長さは、MQ_TRANSACTION_ID_LENGTH によって指定されます。このフィールドの初期値は 4 個のブランクです。

UOWControl (MQLONG)

このフィールドは、CICS ブリッジが実行する作業単位処理を制御します。

ブリッジに対して、単一トランザクションの実行を要求することも、1つの作業単位内で1つ以上のプログラムの実行を要求することもできます。このフィールドでは、ある作業単位を CICS ブリッジで開始するか、要求した機能を現行の作業単位内で実行するか、あるいは作業単位をコミットまたはバックアウトすることによって終了させるかを指定します。データ伝送の流れを最適化するために、様々な組み合わせがサポートされます。

値は、以下のいずれかでなければなりません。

MQCUOWC_ONLY

作業単位を開始し、機能を実行した上で、その作業単位をコミットする。

MQCUOWC_CONTINUE

現行の作業単位の追加データ (3270 のみ)。

MQCUOWC_FIRST

作業単位を開始し、機能を実行する。

MQCUOWC_MIDDLE

現行の作業単位内で機能を実行する。

MQCUOWC_LAST

機能を実行した上で、その作業単位をコミットする。

MQCUOWC_COMMIT

作業単位をコミットする (DPL のみ)。

MQCUOWC_BACKOUT

作業単位をバックアウトする (DPL のみ)。

このフィールドは、要求フィールドです。このフィールドの初期値は MQCUOWC_ONLY です。

Version (MQLONG)

このフィールドは、CICS 情報ヘッダー構造体のバージョンを指定します。

値は、以下のいずれかでなければなりません。

MQCIH_VERSION_1

バージョン 1 の CICS 情報ヘッダー構造体。

MQCIH_VERSION_2

バージョン 2 の CICS 情報ヘッダー構造体。

これより新しいバージョンの構造体にのみ存在するフィールドは、該当するフィールドの説明にその旨記載されています。以下の定数は、現行バージョンのバージョン番号を指定します。

MQCIH_CURRENT_VERSION

CICS 情報ヘッダー構造体の現行バージョン。

このフィールドは、要求フィールドです。このフィールドの初期値は MQCIH_VERSION_2 です。

CICS-MQ ブリッジ用の IBM MQ C++ メッセージ・ヘッダー

IBM MQ for z/OS を介して CICS-MQ ブリッジにメッセージを送信する C++ アプリケーションでは、MQCIH データ構造体との対話に ImqCICSBridgeHeader クラスを使用します。

以下の例は、CICS-MQ ブリッジ・ヘッダー情報をメッセージに追加する方法を示しています。

```
ImqQueueManager mgr ;           // The queue manager.
ImqQueue queueIn ;              // Incoming message queue.
ImqQueue queueBridge ;          // CICS-MQ bridge message queue.
ImqMessage msg ;                // Incoming and outgoing message.
ImqCicsBridgeHeader header ;    // CICS-MQ bridge header information.

// Retrieve the message to be forwarded.
queueIn.setConnectionReference( mgr );
queueIn.setName( MY_QUEUE );
queueIn.get( msg );

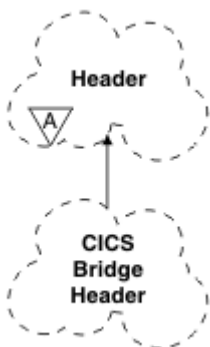
// Set up the CICS-MQ bridge header information.
// The reply-to format is often specified.
// Other attributes can be specified, but all have default values.
header.setReplyToFormat( /* ? */ );

// Insert the CICS-MQ bridge header information. This will vary
// the encoding, character set and format of the message.
// Message data is moved along, past the header.
msg.writeItem( header );

// Send the message to the CICS-MQ bridge queue.
queueBridge.setConnectionReference( mgr );
queueBridge.setName( /* ? */ );
queueBridge.put( msg );
```

ImqCICSBridgeHeader クラス

ImqCICSBridgeHeader クラスは、MQCIH データ構造体の特定の機能をカプセル化します。このクラスのオブジェクトは、IBM MQ for z/OS を介して CICS-MQ ブリッジにメッセージを送信するアプリケーションによって使用されます。



このクラスのオブジェクト属性は、次のとおりです。

図 13. ImqCICSBridgeHeader クラス

ADS 記述子

ADS 記述子を送受信します。MQCADSD_NONE を使用して記述子を設定します。初期値は MQCADSD_NONE です。ほかに、以下の値が使用可能です。

- MQCADSD_NONE
- MQCADSD_SEND
- MQCADSD_RECV
- MQCADSD_MSGFORMAT

アテンション ID

AID キー。このフィールドの長さは MQ_ATTENTION_ID_LENGTH でなければなりません。

authenticator

RACF パスワードまたはパスチケット。初期値にはブランクが含まれており、長さは MQ_AUTHENTICATOR_LENGTH です。

bridge abend code

長さが MQ_ABEND_CODE_LENGTH のブリッジ異常終了コード。初期値は 4 つのブランク文字です。このフィールドに戻される値は、戻りコードによって異なります。

bridge cancel code

ブリッジ異常終了トランザクション・コード。このフィールドは予約済みです。値にブランクが含まれている必要があり、長さは MQ_CANCEL_CODE_LENGTH でなければなりません。

bridge completion code

完了コード。IBM MQ 完了コードまたは CICS EIBRESP 値のどちらかになります。このフィールドの初期値は MQCC_OK です。このフィールドに戻される値は、戻りコードによって異なります。

bridge error offset

ブリッジ・エラーのオフセット。初期値はゼロです。この属性は読み取り専用です。

bridge reason code

理由コード。このフィールドの値は、IBM MQ 理由コードまたは CICS EIBRESP2 値のどちらかになります。このフィールドの初期値は MQRC_NONE です。このフィールドに戻される値は、戻りコードによって異なります。

bridge return code

ブリッジからの戻りコード。初期値は MQCRC_OK です。

conversational task

タスクが会話型であるかどうか。初期値は MQCCT_NO です。ほかに、以下の値が使用可能です。

- MQCCT_YES
- MQCCT_NO

カーソル位置

カーソル位置。初期値はゼロです。

facility keep time

CICS ブリッジ機能解放時間。

facility like

端末でエミュレートされた属性。このフィールドの長さは MQ_FACILITY_LIKE_LENGTH でなければなりません。

facility token

BVT トークン値。このフィールドの長さは MQ_FACILITY_LENGTH でなければなりません。初期値は MQCFAC_NONE です。

function

関数。このフィールドの値は、IBM MQ 呼び出し名または CICS EIBFN 関数のどちらかになります。このフィールドの初期値は MQCFUNC_NONE で、長さは MQ_FUNCTION_LENGTH です。このフィールドに戻される値は、戻りコードによって異なります。

「function」に IBM MQ 呼び出し名が含まれている場合、ほかに以下の値が使用可能です。

- MQCFUNC_MQCONN

- MQCFUNC_MQGET
- MQCFUNC_MQINQ
- MQCFUNC_NONE
- MQCFUNC_MQOPEN
- MQCFUNC_PUT
- MQCFUNC_MQPUT1

get wait interval

ブリッジ・タスクによって発行された MQGET 呼び出しの待機間隔。初期値は MQCGWI_DEFAULT です。このフィールドは、UOW 制御の値が MQCUOWC_FIRST の場合にのみ適用されます。ほかに、以下の値が使用可能です。

- MQCGWI_DEFAULT
- MQWI_UNLIMITED

link type

リンク・タイプ。初期値は MQCLT_PROGRAM です。ほかに、以下の値が使用可能です。

- MQCLT_PROGRAM
- MQCLT_TRANSACTION

next transaction identifier

次に接続するトランザクションの ID。このフィールドの長さは MQ_TRANSACTION_ID_LENGTH でなければなりません。

output data length

COMMAREA データの長さ。初期値は MQCODL_AS_INPUT です。

reply-to format

応答メッセージの形式名。初期値は MQFMT_NONE で、値の長さは MQ_FORMAT_LENGTH です。

start code

トランザクション開始コード。このフィールドの長さは MQ_START_CODE_LENGTH でなければなりません。初期値は MQCSC_NONE です。ほかに、以下の値が使用可能です。

- MQCSC_START
- MQCSC_STARTDATA
- MQCSC_TERMINPUT
- MQCSC_NONE

task end status

タスク終了状況。初期値は MQCTES_NOSYNC です。ほかに、以下の値が使用可能です。

- MQCTES_COMMIT
- MQCTES_BACKOUT
- MQCTES_ENDTASK
- MQCTES_NOSYNC

transaction identifier

接続するトランザクションの ID。初期値にはブランクが含まれている必要があり、値の長さは MQ_TRANSACTION_ID_LENGTH でなければなりません。このフィールドは、UOW 制御の値が MQCUOWC_FIRST または MQCUOWC_ONLY の場合にのみ適用されます。

UOW control

UOW 制御。初期値は MQCUOWC_ONLY です。ほかに、以下の値が使用可能です。

- MQCUOWC_FIRST
- MQCUOWC_MIDDLE
- MQCUOWC_LAST
- MQCUOWC_ONLY

- MQCUOWC_COMMIT
- MQCUOWC_BACKOUT
- MQCUOWC_CONTINUE

version

MQCIH バージョン番号。初期値は MQCIH_VERSION_2 です。これ以外のサポートされる値は MQCIH_VERSION_1 のみです。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができません。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様自身の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Director of Licensing

IBM Corporation

North Castle Drive, MD-NC119 Armonk,

NY 10504-1785

United States of America

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関す

る実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

プログラミング・インターフェース情報

CICS には、プログラミング・インターフェースと見なすことのできる資料と、プログラミング・インターフェースと見なすことのできない資料があります。

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが含まれています。

- [アプリケーションの開発](#)
- [Developing system programs](#)
- [CICS TS セキュリティー](#)
- [外部インターフェースに向けた開発](#)
- [アプリケーション開発のリファレンス](#)
- [リファレンス: システム・プログラミング](#)
- [リファレンス: 接続](#)

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が含まれています。

- [Troubleshooting and support](#)
- [CICS TS 診断リファレンス](#)

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが以下のマニュアルに含まれています。

- [アプリケーション・プログラミング・ガイドおよびアプリケーション・プログラミング・リファレンス](#)
- [Business Transaction Services](#)
- [Customization Guide](#)
- [C++ OO Class Libraries](#)
- [Debugging Tools Interfaces Reference](#)
- [Distributed Transaction Programming Guide](#)
- [External Interfaces Guide](#)
- [Front End Programming Interface Guide](#)

- IMS Database Control Guide
- インストール・ガイド
- セキュリティー・ガイド
- Supplied Transactions
- CICSplex SM Managing Workloads
- CICSplex SM Managing Resource Usage
- CICSplex SM アプリケーション・プログラミング・ガイドおよび CICSplex SM アプリケーション・プログラミング・リファレンス
- Java Applications in CICS

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が以下のマニュアルに含まれています。

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSplex SM Problem Determination Guide

商標

IBM、IBM ロゴおよび ibm.com[®] は、世界の多くの国で登録された International Business Machines Corporation の商標または登録商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Apache、Apache Axis2、Apache Maven、Apache Ivy、Apache Software Foundation (ASF) ロゴ、および ASF feather ロゴは、Apache Software Foundation の商標です。

Gradle および Gradlephant ロゴは、Gradle, Inc. およびその子会社の米国およびその他の国における登録商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

登録商標 Linux は、世界中で商標の所有者である Linux Torvalds の独占的ライセンシーである Linux Foundation のサブライセンスに従って使用されています。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Red Hat[®]、および Hibernate[®] は、Red Hat, Inc. またはその子会社の米国およびその他の国における商標または登録商標です。

Spring Boot は、Pivotal Software, Inc. の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Zowe[™]、Zowe ロゴ、および Open Mainframe Project[™] は、Linux Foundation の商標です。

製品資料に関するご使用条件

これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。

適用範囲

IBM Web サイトの「ご利用条件」に加えて、以下のご使用条件が適用されます。

個人使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商用使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

権利

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

IBM オンラインでのプライバシー・ステートメント

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (ソフトウェア・オフファリング) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オフファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オフファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オフファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

CICSplex SM Web ユーザー・インターフェース (メイン・インターフェース) の場合:

このソフトウェア・オフファリングは、展開される構成に応じて、セッション管理、認証、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの Cookie および持続的な Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSplex SM Web ユーザー・インターフェース (データ・インターフェース) の場合:

このソフトウェア・オフファリングは、展開される構成に応じて、セッション管理、認証、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名またはその他の個人情報を、セッションごとの Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSplex SM Web ユーザー・インターフェース (「Hello World」ページ) の場合:

このソフトウェア・オフファリングは、展開される構成に応じて、個人情報を収集しないセッションごとの Cookie を使用する場合があります。これらの Cookie を無効にすることはできません。

CICS Explorer の場合:

このソフトウェア・オフファリングは、展開される構成に応じて、セッション管理、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの設定および持続的な設定を使用して収集する場合があります。これらの設定を無効にすることはできませんが、ユーザー・パスワードの暗号化形式でのディスクへの保管は、サインオン中にチェック・ボックスにチェック・マークを付けることによるユーザーの明示的な操作によってのみ有効化することができます。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、[『IBM プライバシー・ポリシー』](#) および [『IBM オンラインでのプライバシー・ステートメント』](#) の『クッキー、Web ビーコン、その他のテクノロジー』および「[IBM ソフトウェア製品および Software as a Service のプライバシー・ステートメント](#)」を参照してください。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。
なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アプリケーション・プログラム
 CKQC DISPLAY [46](#)
異常終了
 CICS トランザクションの切断 [122](#)
一時データ・キュー (TDQ)、CKQQ [33](#)
オンライン・リソース定義 (RDO) [17](#)

[カ行]

解決
 作業単位 [126](#)
起動
 動作しないとき [128](#)
キュー・マネージャーの停止 [123](#)
更新
 CSQINP2 [17](#)
コールバック・ルーチン
 サンプル
 概要 [110](#)
 設計 [110](#)
 セットアップ [112](#)

[サ行]

再同期 [123](#), [125](#)
作業
 CICS、手動での作業単位の解決 [126](#)
作業、単位 [126](#)
作業トークン [126](#)
自動インストール、CICS [7](#)
セキュリティ
 端末ユーザー ID [58](#)
 CKSG ユーザー ID [58](#)
接続
 開始する
 EXEC CICS SET MQCONN [38](#)
 開始する場所
 CICS アプリケーション・プログラム [38](#), [39](#)
 PLTPI プログラム [17](#), [22](#)
 から停止
 CICS アプリケーション・プログラム [44](#)
 CICS コマンド行 [43](#)
 停止
 CICS アプリケーション・プログラム [43](#)
 EXEC CICS SET MQCONN [43](#)
接続セキュリティ [75](#)

[タ行]

データ・タイプ、詳細な説明
 構造
 MQCIH [135](#)

トランザクション・トラッキング [12](#)

[ハ行]

非同期メッセージ・コンシューム
 サンプル
 概要 [110](#)
 設計 [110](#)
 セットアップ [112](#)
表示
 CICS 内の作業単位 [126](#)
ブリッジ [12](#)
プログラムの自動インストール、CICS [7](#)
プログラム・リスト・テーブル (PLT) [17](#), [22](#)

[マ行]

未確定作業単位
 CICS からの解決 [125](#)
未確定の作業単位
 CICS [124](#)
モニター
 CICS 接続アクティビティ [48](#)
モニター (monitor) [12](#)

[ヤ行]

ユーザー出口 [114](#)

[ラ行]

リソース・タイプ
 MQseries [121](#)
 WMQ_INIT [121](#)
 WMQCDISC [121](#)
ログ印刷ユーティリティ (CSQ1LOGP)
 開始 RBA の検索 [126](#)

A

AbendCode フィールド [137](#)
ADSDescriptor フィールド [138](#)
AttentionId フィールド [138](#)
Authenticator フィールド [138](#)

C

C++ [147](#)
CancelCode フィールド [138](#)
CEDF [121](#)
CICS
 記憶保護機能 [116](#)
 作業単位 [126](#)
 システム定義 (CSD) データ・セット [17](#)
 自動インストール [7](#)
 未確定作業単位の解決 [125](#)

CICS アダプター
 静的シャットダウン [121](#)
CICS 実行診断機能 [121](#)
CICS プログラムの実行キー [116](#)
CICS-MQ API 交差出口
 起動 [115](#)
 サンプル [117](#)
 CSQCAPX [117](#)
CICS-WebSphere MQ アダプター
 コンポーネント [3](#)
 再同期 [123](#)
 自動再接続 [123](#)
 図 [3](#)
CICS-WebSphere MQ ブリッジ (CICS-WebSphere MQ
 bridge)
 チューニング考慮事項 [27](#)
CICS-WebSphere MQ ブリッジのチューニング [27](#)
CICSplex SM ビュー [35](#)
CKAM トランザクション [4](#)
CKBM セキュリティー [73](#)
CKCN セキュリティー [73](#)
CKDL セキュリティー [73](#)
CKDP セキュリティー [73](#)
CKQC
 DISPLAY コマンド [46](#)
 START コマンド [37](#)
 STARTCKTI コマンド [61](#)
 STOP コマンド [43](#)
 STOPCKTI コマンド [64](#)
CKQC トランザクション
 セキュリティ [73](#)
CKQQ、一時データ・キュー [33](#)
CKRS セキュリティー [73](#)
CKRT セキュリティー [73](#)
CKSD セキュリティー [73](#)
CKSG MCA トランザクション [58](#)
CKSQ セキュリティー [73](#)
CKTI [61](#)
CKTI トランザクション
 開始する [58](#)
 開始の自動化 [63](#)
 セキュリティ [73](#)
 停止 [63](#), [64](#)
 表示 [65](#)
CodedCharSetId フィールド [138](#)
CompCode フィールド [139](#)
ConversationalTask フィールド [139](#)
CREATE MQCONN [34](#)
CREATE MQMONITOR [34](#)
CSD (CICS システム 定義データ・セット) [17](#)
CSQ1LOGP (ログ印刷ユーティリティー)
 開始 RBA の検索 [126](#)
CSQ4CVCN [110](#)
CSQ4CVCT [110](#)
CSQ4CVEV [110](#)
CSQ4CVPT
 構文規則 [113](#)
CSQ4CVRG [110](#)
CSQ4SAMP [7](#), [17](#)
CSQCAPX API 交差出口のサンプル [117](#)
CSQCAPX の API 横断出口プログラムの例 [114](#)
CSQCAT1 [7](#), [17](#)
CSQCKB [7](#), [17](#)
CSQCQCON [39](#)

CSQCSTUB [107](#)
CSQINP2
 更新 [17](#)
CursorPosition フィールド [139](#)

D

DFHMQ [7](#), [17](#)
DFHMQCOD サンプル PLTPI プログラム [17](#), [22](#)
DISCARD MQCONN [34](#)
DPL プログラム [81](#)

E

Encoding フィールド [139](#)
ErrorOffset フィールド [139](#)
EXEC CICS LINK
 COMMAREA オプション [22](#)
 INPUTMSG オプション [33](#)

F

Facility フィールド [139](#)
FacilityKeepTime フィールド [140](#)
FacilityLike フィールド [140](#)
Flags フィールド [140](#)
Format フィールド [140](#)
Function フィールド [141](#)

G

GetWaitInterval フィールド [141](#)
GRPLIST システム 初期設定パラメーター [17](#)

I

ImqCICSBridgeHeader クラス [147](#)
InputItem フィールド [142](#)
INQUIRE MQCONN [34](#)
INQUIREMQMONITOR [34](#)
IRC および CICS アダプター [17](#)

L

LinkType フィールド [142](#)

M

MQ コマンド
 DISPLAY CONN [126](#)
 RESOLVE INDOUBT [126](#)
MQCB
 サンプル
 概要 [110](#)
 設計 [110](#)
 セットアップ [112](#)
MQCFUNC_* の値 [141](#)
MQCGWI_* の値 [141](#)
MQCIH 構造体 [135](#)
MQCIH_* の値 [145](#)
MQCIH_DEFAULT [136](#)
MQCLT_* の値 [142](#)

MQCODL_* の値 [142](#)

MQCONN

セキュリティ [74](#)

MQCONN コマンド [34](#)

MQCRC_* の値 [144](#)

MQCTL

サンプル

概要 [110](#)

設計 [110](#)

セットアップ [112](#)

MQCUOWC_* 値 [146](#)

MQINI [19](#)

MQMONITOR

セキュリティ [74](#), [75](#)

MQMONITOR コマンド [34](#)

N

NextTransactionId フィールド [142](#)

NID (ネットワーク ID) [126](#)

O

OutputDataLength フィールド [142](#)

P

PLTPI (プログラム・リスト・テーブルの初期化後処理)
接続の開始 [17](#), [22](#)

R

RDO (オンライン・リソース定義) [17](#)

Reason フィールド [142](#)

RemoteSysId フィールド [143](#)

RemoteTransId フィールド [143](#)

ReplyToFormat フィールド [143](#)

Reserved1 フィールド [143](#)

Reserved2 フィールド [143](#)

Reserved3 フィールド [143](#)

Reserved4 フィールド [143](#)

RESOLVE INDOUBT コマンド、ロックされたリソースの解放
[126](#)

RESYNCMEMBER [123](#), [125](#)

ReturnCode フィールド [144](#)

S

SET MQCONN [34](#)

SET MQMONITOR [34](#)

SIT (システム 初期設定テーブル)

GRPLIST パラメーター [17](#)

PLTPI パラメーター [17](#)

StartCode フィールド [144](#)

StrucId フィールド [145](#)

StrucLength フィールド [145](#)

T

TaskEndStatus フィールド [145](#)

TransactionId フィールド [146](#)

TYPETERM 定義、UCTRAN [37](#)

U

UCTRAN、TYPETERM 定義上の [37](#)

UOWControl フィールド [146](#)

V

「Version」 フィールド [146](#)

[特殊文字]

CICS DPLブリッジ

アプリケーション・プログラミング [83](#)

作業単位の管理 [92](#)

分散環境でのトランザクション [95](#)

メッセージ構造 [82](#)

MsgId および CorrelId の管理 [92](#)

CICS 3270ブリッジ

アウトバウンド・メッセージ構造 [86](#)

アプリケーション・プログラミング [84](#)

インバウンド・メッセージ構造 [85](#)

MQCIH フィールドの設定 [90](#)

アプリケーション・プログラミング

CICS 3270ブリッジ用 [84](#)

CICS DPLブリッジ用 [83](#)

作業単位

CICS DPLブリッジを使用する場合の管理 [92](#)

メッセージ構造

CICS 3270ブリッジを使用する場合のアウトバウンド
[86](#)

CICS 3270ブリッジを使用する場合のインバウンド [85](#)

CICS DPLブリッジ用 [82](#)

ADS

CICS-MQブリッジ・ベクトル [101](#)

CICS-MQブリッジ・ベクトルで使用される [98](#)

BMS

CICS-MQブリッジを使用する場合 [96](#)

BRMQ

CICS-MQブリッジを使用する場合のアウトバウンド構
造体 [86](#)

CICS-MQブリッジを使用する場合のインバウンド構造
体 [85](#)

CEMT I TASK

CICS-MQブリッジを使用する場合の例 [87](#)

CICS-MQアダプター

CICS-MQ API 交差出口 [114](#)

CICS-MQブリッジ

アウトバウンド BRMQ 構造体 [86](#)

アプリケーション・データ構造 (ADS) [81](#), [101](#)

インバウンド BRMQ 構造体 [85](#)

開始データを使用したトランザクション [84](#)

基本マッピング・サポート (BMS) [96](#)

最適化されたエミュレーションの例 [106](#)

作業単位の管理 [93](#)

同期点を使用するトランザクション [86](#)

トランザクション [84](#)

非最適化エミュレーションの例 [105](#)

分散プログラミング [95](#)

ベクトルの使用 [84](#)

レガシー・アプリケーション [81](#)

3270 レガシー・アプリケーション [104](#)

CEMT I TASK の例 [87](#)

COMMAREA データ [81](#)

- CICS-MQ ブリッジ (続き)
 - DPL プログラムの使用 [82](#)
 - MQCIH フィールドの設定 [89](#)
 - MQMD フィールドの設定 [88](#)
 - MsgId および CorrelId の管理 [93](#)
 - RECEIVE MAP ベクトルの解釈 [101](#)
 - SEND MAP ベクトルの解釈 [98](#)
 - z/OS でのアプリケーション [81](#)
- COMMAREA データ
 - CICS-MQ ブリッジを使用する場合 [82](#)
- CorrelId
 - CICS-MQ ブリッジを使用する場合の管理 [92](#), [93](#)
- MQCIH
 - CICS 3270 ブリッジを使用する場合のフィールドの設定 [90](#)
 - CICS-MQ ブリッジを使用する場合のフィールドの設定 [89](#)
- MQMD
 - CICS-MQ ブリッジを使用する場合のフィールドの設定 [88](#)
- MsgId
 - CICS-MQ ブリッジを使用する場合の管理 [92](#), [93](#)
- RECEIVE MAP ベクトル
 - CICS-MQ ブリッジを使用する場合の解釈 [101](#)
- SEND MAP ベクトル
 - CICS-MQ ブリッジを使用する場合の解釈 [98](#)
- transactions
 - CICS-MQ ブリッジ [84](#)
- アプリケーション
 - CICS-MQ ブリッジを使用する場合 [81](#)
- アプリケーション・データ構造
 - CICS-MQ ブリッジ・ベクトル [101](#)
- エミュレーション
 - CICS-MQ ブリッジを使用する場合の例 [105](#), [106](#)
- 基本マッピング・サポート
 - CICS-MQ ブリッジを使用する場合 [96](#)
- 単位、作業の
 - 未確定の表示 [126](#)
 - CICS からの解決 [125](#)
 - CICS-MQ ブリッジを使用する場合 [93](#)
 - CICS、手動での解決 [126](#)
- 出口プログラム (exit program)
 - CICS-MQ アダプター [114](#)
- 同期点を使用するトランザクション
 - CICS-MQ ブリッジ [86](#)
- 分散環境でのトランザクション
 - CICS DPL ブリッジ [95](#)
- ベクトル
 - CICS-MQ ブリッジでの使用 [84](#)
- レガシー・アプリケーション
 - CICS-MQ ブリッジを使用する場合 [81](#), [104](#)

