

z/OS용 CICS Transaction
Server버전 5 릴리스 6

CICS와 함께 IBM MQ 사용



참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [제품 법적 주의사항](#)에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한 IBM® CICS® Transaction Server for z/OS®, 버전 5 릴리스 6 (제품 번호 5655-Y305655-BTA) 및 모든 후속 릴리스와 수정에 적용됩니다.

© Copyright International Business Machines Corporation 1974, 2020.

목차

이 PDF 정보.....	vii
제 1 장 CICS 및 IBM MQ.....	1
작동 방식: CICS-MQ 어댑터.....	3
경보 모니터(CKAM).....	4
태스크 초기자 또는 트리거 모니터(CKTI).....	4
CICS-MQ 어댑터에 대해 제공된 자원 정의.....	5
작동 방식: CICS-MQ 브릿지.....	6
CICS DPL 프로그램이 CICS-MQ 브릿지에서 실행되는 방식.....	6
CICS 3270 트랜잭션이 CICS-MQ 브릿지에서 작동되는 방법.....	8
작동 방식: CICS와 함께 JMS용 IBM MQ 클래스 사용.....	9
CICS-MQ 트랜잭션 추적 지원.....	10
MQ 지원을 위한 CICS 자원: MQCONN 및 MQMONITOR.....	11
MQMONITOR에 대한 z/OS Workload Manager 상태 서비스의 영향.....	13
제 2 장 IBM MQ에 대한 연결 구성.....	15
CICS-MQ 어댑터 설정.....	15
MQCONN 자원 정의 및 설치.....	16
MQMONITOR 자원 정의 및 설치.....	17
IBM MQ로의 연결을 시작하도록 PLTPI 프로그램 작성.....	19
PLTPI 목록에서 DFHMQCOD 또는 사용자의 고유 프로그램 지정.....	20
CICS-MQ 브릿지 설정.....	20
CICS에 있는 CICS-MQ 브릿지 트랜잭션 및 프로그램의 정의.....	22
CICS-MQ 브릿지에 맞게 MQMONITOR 설정.....	22
여러 CICS-MQ 브릿지 모니터 설정.....	23
CICS-MQ 브릿지 처리량 제어.....	24
제 3 장 CICS-MQ 어댑터 관리.....	25
CICS-MQ 어댑터 제어 패널에 액세스.....	26
명령행의 CKQC 명령.....	27
CICS 애플리케이션 프로그램의 CKQC 명령.....	28
EXEC CICS LINK 인터페이스 메시지.....	29
CICS-MQ 연결 및 모니터를 위한 EXEC CICS 및 CEMT 명령.....	29
CICS-MQ 연결 및 모니터를 위한 CICSplex SM 보기.....	30
CICS-MQ 연결 시작.....	31
CICS-MQ 어댑터 제어 패널에서 CICS-MQ 연결 시작.....	32
CICS 명령행에서 CICS-MQ 연결 시작.....	32
CICS 애플리케이션 프로그램에서 EXEC CICS SET MQCONN 을 사용하여 CICS-MQ 연결 시작.....	33
CICS 애플리케이션 프로그램에서 DFHMQCN에 링크하여 CICS-MQ 연결 시작.....	34
CICS CEMT 트랜잭션을 통해 CICS-MQ 연결 시작.....	35
CICSplex SM 웹 사용자 인터페이스를 통해 CICS-MQ 연결 시작.....	35
CICS-MQ 연결 중지.....	36
CICS-MQ 어댑터 제어판에서 CICS-MQ 연결 중지.....	37
CICS 명령행에서 CICS-MQ 연결 중지.....	37
CICS 애플리케이션 프로그램에서 EXEC CICS SET MQCONN 을 사용하여 CICS-MQ 연결 중지.....	38
CICS 애플리케이션 프로그램에서 DFHMQDSC에 링크하여 CICS-MQ 연결 중지.....	38
CICS CEMT 트랜잭션을 통해 CICS-MQ 연결 중지.....	38
CICSplex SM 웹 사용자 인터페이스를 통해 CICS-MQ 연결 중지.....	39
CICS-MQ 연결 정보 표시.....	39
CICS-MQ 연결 통계 및 설정 표시.....	40

CICS-MQ 연결 및 연결 끊기 시간 표시.....	42
CICS-MQ 연결 통계 및 호출 유형 표시.....	42
CICS-MQ 연결을 사용하는 태스크 표시.....	43
CICS MQ 모니터 시작.....	44
CICS 애플리케이션 프로그램에서 EXEC CICS SET MQMONITOR 를 사용하여 CICS MQ 모니터 시작....	45
CICSplex SM 애플리케이션 프로그램에서 EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START) 를 사용하여 CICS MQ 모니터 시작.....	45
CICS CEMT 트랜잭션을 통해 CICS MQ 모니터 시작.....	46
CICSplex SM 웹 사용자 인터페이스를 통해 CICS MQ 모니터 시작.....	47
CICS MQ 모니터 중지.....	47
CICS 애플리케이션 프로그램에서 EXEC CICS SET MQMONITOR 를 사용하여 CICS MQ 모니터 중지....	48
CICSplex SM 애플리케이션 프로그램에서 EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP) 을 사용하여 CICS MQ 모니터 중지.....	48
CICS CEMT 트랜잭션을 통해 CICS MQ 모니터 중지.....	48
CICSplex SM 웹 사용자 인터페이스를 통해 CICS MQ 모니터 중지.....	49
CICS MQ 모니터에 대한 정보 표시.....	49
CICS-MQ 연결 통계 재설정.....	49
CICS-MQ 연결을 사용 중인 태스크 제거.....	50
CICS-MQ 트리거 모니터.....	50
CKTI의 인스턴스 시작.....	51
CKTI의 인스턴스 중지.....	55
CKTI의 현재 인스턴스 표시.....	57
사용자 작성 MQ 트리거 모니터 및 MQ 메시지 이용자 개발과 사용.....	58
제 4 장 CICS-MQ 브릿지 관리.....	61
CICS-MQ 브릿지 시작.....	61
CICS-MQ 브릿지 중지.....	62
제 5 장 CICS-MQ 어댑터를 위한 보안.....	65
CICS-MQ 어댑터 트랜잭션을 위한 보안 구현	65
CICS-MQ 어댑터 사용자 ID.....	65
MQCONN 및 MQMONITOR 자원을 위한 명령 보안.....	66
MQMONITOR 자원을 위한 대리 사용자 보안.....	67
CICS-MQ 어댑터를 위한 IBM MQ 연결 보안.....	67
제 6 장 CICS-MQ 브릿지를 위한 보안.....	69
제 7 장 CICS-MQ 브릿지를 사용하기 위한 애플리케이션 개발.....	73
CICS-MQ 브릿지의 DPL 메시지 구조	73
예제: CICS-MQ 브릿지를 통한 DPL 프로그램에 대한 요청 메시지.....	75
CICS-MQ 브릿지의 3270 트랜잭션 메시지 구조	75
CICS-MQ 브릿지를 사용하는 3270 트랜잭션의 벡터	76
CICS-MQ 브릿지를 사용하는 3270 요청 메시지(인바운드)의 구조	77
CICS-MQ 브릿지를 사용하는 3270 응답 메시지(아웃바운드)의 구조	77
예: CICS-MQ 브릿지를 통해 CEMT INQUIRE TASK를 발행하기 위한 요청 메시지.....	78
CICS-MQ 브릿지를 위해 MQMD 및 MQCIH 구조에서 설정해야 하는 필드	79
CICS-MQ 브릿지 메시지용 MQMD 필드.....	79
DPL 프로그램 요청 메시지용 MQCIH 필드.....	80
3270 트랜잭션 요청 메시지용 MQCIH 필드.....	81
DPL 프로그램의 MsgId, CorrelId 및 UOWControl 필드	83
3270 트랜잭션에 대한 MsgId, CorrelId 및 UOWControl 필드	84
분산 환경에서 CICS-MQ 브릿지에 대한 데이터 변환.....	86
CICS-MQ 브릿지를 사용하여 BMS(Basic Mapping Support) 애플리케이션 실행.....	87
CICS-MQ 브릿지에 대한 SEND MAP 벡터 해석.....	88
CICS-MQ 브릿지에 대한 RECEIVE MAP 벡터 해석.....	92
ADSDL 및 ADS의 예.....	93

CICS-MQ 브릿지에서 실행되는 3270 트랜잭션의 예.....	94
최적화 없는 정확한 에뮬레이션	95
최적화로 향상된 에뮬레이션.....	96
제 8 장 CICS-MQ 어댑터를 사용하기 위한 애플리케이션 개발.....	97
IBM MQ MQI 호출에 액세스하기 위한 API 스텝 프로그램.....	97
비동기 메시지 이용 및 콜백 루틴.....	98
비동기 메시지 이용을 위한 샘플 프로그램.....	99
CICS-MQ API 교차 엑시트.....	103
CICS-MQ API 교차 엑시트 호출 방법.....	104
CICS-MQ 교차 엑시트 프로그램과의 통신.....	104
사용자 고유의 CICS-MQ API 교차 엑시트 프로그램 작성.....	105
샘플 API 교차 엑시트 프로그램, CSQCAPX.....	106
CICS-MQ API 교차 엑스트 사용.....	107
CICS-MQ API 교차 엑스트 사용 안함.....	108
제 9 장 CICS-MQ 어댑터 문제점 해결.....	111
IBM MQ 대기.....	111
CICS-MQ 연결 종료 시 발생하는 상황.....	111
정상(또는 순서에 따른) 종료.....	112
강제 종료.....	112
큐 관리자를 중지할 때 발생하는 상황.....	113
자동 재연결 및 재동기화.....	113
CICS-MQ 어댑터가 다시 시작될 때 발생하는 상황.....	114
CICS에서 인다우트 작업 단위를 해결하는 방법.....	115
CICS 작업 단위를 수동으로 해결하는 방법.....	116
트리거가 작동하지 않는 경우.....	117
제 10 장 CICS-MQ 브릿지 문제점 해결.....	119
오류가 발생한 경우 CICS-MQ 브릿지가 수행하는 조치.....	120
CICS-MQ 브릿지 디버깅.....	122
제 11 장 MQCIH – CICS-MQ 브릿지 헤더.....	125
초기값 및 언어 선언.....	126
AbendCode (MQCHAR4).....	127
ADSDescriptor (MQLONG).....	127
AttentionId (MQCHAR4).....	128
Authenticator (MQCHAR8).....	128
CancelCode (MQCHAR4).....	128
CodedCharSetId (MQLONG).....	128
CompCode (MQLONG).....	128
ConversationalTask (MQLONG).....	128
CursorPosition (MQLONG).....	129
Encoding (MQLONG).....	129
ErrorOffset (MQLONG).....	129
Facility (MQBYTE8).....	129
FacilityKeepTime (MQLONG).....	129
FacilityLike (MQCHAR4).....	129
Flags (MQLONG).....	130
Format (MQCHAR8).....	130
Function (MQCHAR4).....	130
GetWaitInterval (MQLONG).....	131
InputItem (MQLONG).....	131
LinkType (MQLONG).....	131
NextTransactionId (MQCHAR4).....	131
OutputDataLength (MQLONG).....	132
Reason (MQLONG).....	132

RemoteSysId (MQCHAR4).....	132
RemoteTransId (MQCHAR4).....	132
ReplyToFormat (MQCHAR8).....	132
Reserved1 (MQCHAR8).....	133
Reserved2 (MQCHAR8).....	133
Reserved3 (MQCHAR8).....	133
Reserved4 (MQLONG).....	133
ReturnCode (MQLONG).....	133
StartCode (MQCHAR4).....	133
StrucId (MQCHAR4).....	134
StrucLength (MQLONG).....	134
TaskEndStatus (MQLONG).....	134
TransactionId (MQCHAR4).....	135
UOWControl (MQLONG).....	135
Version (MQLONG).....	136
CICS-MQ 브릿지의 IBM MQ C++ 메시지 헤더.....	136
주의사항.....	141
색인.....	145

이 PDF 정보

이 PDF는 CICS-MQ 어댑터와 CICS-MQ 브릿지의 구성 및 사용과 CICS-MQ 환경 정의 및 유지보수에 대한 소개와 안내 정보를 제공합니다.

사용된 용어 및 표기법에 대한 세부사항은 IBM Knowledge Center에서 [CICS 문서에서 사용하는 규칙과 용어](#)의 내용을 참조하십시오.

이 PDF의 날짜

이 PDF는 2020년 5월 28일에 작성되었습니다.

제 1 장 CICS 및 IBM MQ

CICS는 IBM MQ 메시지를 사용하여 CICS에서 작업을 실행하기 위한 다양한 방법을 제공합니다. Java™ 애플리케이션에서 Java용 IBM MQ 클래스 또는 JMS용 IBM MQ 클래스를 통해 IBM MQ에 액세스할 수 있습니다. 다른 애플리케이션에서 CICS와 함께 제공되는 두 가지 인터페이스 즉, CICS-MQ 어댑터 및 CICS-MQ 브릿지를 통해 IBM MQ에 액세스할 수 있습니다.

z/OS에 있는 IBM MQ의 개념 및 아키텍처에 대한 요약은 IBM MQ 문서에 있는 [IBM MQ 제품 문서의 z/OS용 IBM MQ 개념](#)의 내용을 참조하십시오. CICS와 함께 작동하는 IBM MQ 버전을 확인하려면 [IBM MQ의 시스템 요구사항](#)의 내용을 참조하고 IBM MQ의 관련 릴리스에 대한 z/OS 항목 아래를 살펴보십시오.

CICS가 IBM MQ와 함께 작동하는 방법

CICS는 IBM MQ의 지원을 위해 두 가지 자원을 제공합니다.

- MQCONN - CICS-MQ 연결의 속성을 정의합니다.
- MQMONITOR - IBM MQ 메시지 이용자의 속성을 정의합니다.

자원은 11 페이지의 [『MQ 지원을 위한 CICS 자원: MQCONN 및 MQMONITOR』](#)에서 자세히 설명되어 있습니다.

CICS-MQ 어댑터를 통해 IBM MQ에 액세스

CICS-MQ 어댑터는 CICS와 함께 제공되고 CICS 자원 관리자 인터페이스(RMI)를 사용하여 외부 자원 관리자로 IBM MQ와 통신합니다. 사용자가 MQ 메시지 트리거 메커니즘을 통해 CICS에서 사용자 트랜잭션을 초기화할 수 있으므로 트리거 모니터로 알려져 있기도 합니다. 트리거링에 대해 사용으로 설정된 애플리케이션 큐에 메시지를 입력할 수 있습니다. 메시지가 트리거를 릴리스할 때 IBM MQ는 트리거 메시지를 CICS에 전송하여 지정된 사용자 트랜잭션을 초기화합니다. CICS-MQ 어댑터는 다음과 같은 두 가지 주요 기능을 제공합니다.

- 어댑터를 관리하기 위해 시스템 프로그래머 및 관리자가 사용할 제어 기능 세트
- CICS 애플리케이션에 대한 메시지 큐 인터페이스(MQI) 지원

CICS-MQ 어댑터는 필요에 따라 **EXEC CICS ASSIGN** 및 **EXEC CICS ABEND**와 같은 표준 CICS 명령 레벨 서비스를 사용합니다. 일부 CICS-MQ 어댑터는 메시징 요청을 발행하는 트랜잭션의 제어로 실행됩니다. 그러므로 CICS 서비스의 호출은 트랜잭션으로 발행된 것처럼 보입니다.

CICS 다중 리전 조작 또는 인터시스템 통신(ISC) 환경의 각 CICS 주소 공간에는 큐 관리자 서브시스템의 고유한 첨부가 있습니다. 단일 CICS 주소 공간은 한 번에 하나의 큐 관리자에만 연결할 수 있습니다. 그러나 각 주소 공간은 큐 관리자 서브시스템에 연결할 수 있습니다.

자세한 정보는 3 페이지의 [『작동 방식: CICS-MQ 어댑터』](#)의 내용을 참조하십시오.

CICS-MQ 브릿지를 통해 IBM MQ에 액세스

CICS-MQ 브릿지는 CICS와 함께 제공됩니다. 이를 통해 대상 프로그램의 이름이 포함된 메시지를 IBM MQ 큐에 전송하여 IBM MQ 애플리케이션에서 CICS 시스템의 애플리케이션으로의 직접 액세스가 허용됩니다. MQ 트리거링과는 달리 CICS-MQ 브릿지는 MQ 미인식 CICS 애플리케이션에 대한 직접 액세스를 제공합니다. 브릿지가 내재 MQI(Message Queue Interface) 지원을 사용으로 설정하므로 CICS-MQ 브릿지를 사용하여 CICS에서 프로그램 또는 트랜잭션을 실행하는 경우 CICS 프로그램에는 IBM MQ 호출이 포함되지 않습니다. 그러므로 레거시 CICS 애플리케이션을 다시 작성, 다시 컴파일, 다시 연결하지 않고도 IBM MQ 메시지를 통해 기타 운영 체제에서 제어되도록 레거시 CICS 애플리케이션을 다시 설치할 수 있습니다. IBM MQ 메시지에는 CICS 애플리케이션에 대한 제어 옵션을 제공하기 위한 IBM MQ CICS 정보 헤더(MQCIH 구조)도 포함되어 있습니다.

다음 유형의 CICS 애플리케이션은 CICS-MQ 브릿지와 함께 사용하기에 적합합니다.

- DPL(Distributed Program Link) 프로그램이라고 하는 **EXEC CICS LINK** 명령으로 호출된 CICS 프로그램. 프로그램은 CICS API의 DPL 서브세트를 준수해야 합니다. 즉, CICS 터미널 또는 동기점 기능을 사용하면 안 됩니다. CICS-MQ 브릿지를 사용하여 단일 CICS 프로그램 또는 작업 단위를 구성하는 CICS 프로그램 세트를 실행할 수 있습니다.

- 3270 트랜잭션이라고 하는 3270 터미널에서 실행되도록 디자인된 CICS 트랜잭션. 트랜잭션은 BMS(Basic Mapping Support) 또는 터미널 제어 명령을 사용할 수 있습니다. 대화식 또는 의사 대화의 일부일 수 있습니다. 동기점을 발행할 수 있습니다.

자세한 정보는 [CICS-WebSphere MQ 브릿지 정보](#)의 내용을 참조하십시오.

Java 애플리케이션에서 IBM MQ에 액세스

CICS에서 실행되는 Java 애플리케이션은 다음과 같은 두 가지 인터페이스를 통해 IBM MQ에 액세스할 수 있습니다.

- JMS용 IBM MQ 클래스 사용
- Java용 IBM MQ 클래스 사용

Java 애플리케이션은 다음과 같은 방법으로 IBM MQ에 연결됩니다.

- TCP/IP를 사용하여 IBM MQ MQI 클라이언트로서 MQ 클라이언트 모드로 연결됩니다.
- JNI(Java Native Interface)를 사용하여 IBM MQ에 직접 연결하는 MQ 바인딩 모드로 연결됩니다.

자세한 정보는 [Java 애플리케이션에서 IBM MQ에 액세스](#)의 내용을 참조하십시오. IBM MQ 통신 유형에 대한 자세한 정보는 [IBM MQ 제품 문서의 JMS용 IBM MQ 클래스에 대한 연결 모드](#)의 내용을 참조하십시오.

JMS용 IBM MQ 클래스

JMS(Java Message Service)는 애플리케이션이 신뢰성 있는 비동기 통신을 사용하여 메시지를 전송하고 수신할 수 있는 Java EE(Java Enterprise Edition)로 정의되는 API입니다. IBM MQ, WebSphere® Liberty 임베디드 JMS 메시징 제공자 또는 써드파티 메시징 제공자를 포함하여 다양한 메시징 제공자를 사용할 수 있는 기능을 제공합니다. JMS용 IBM MQ 클래스는 `javax.jms` 패키지에 정의된 인터페이스를 구현하고 JMS API에 두 개의 확장기능 세트도 제공합니다. Java SE(Java™ Platform, Standard Edition) 및 Java EE(Java Platform, Enterprise Edition) 애플리케이션은 JMS용 IBM MQ 클래스를 사용할 수 있습니다.

버전 8.0에서 IBM MQ는 JMS 표준의 JMS 2.0 버전을 지원합니다. 이 구현은 클래식 API의 모든 기능을 제공하지만 더 적은 인터페이스만 필요하며 사용하는 데 간편합니다. 자세한 정보는 IBM MQ 문서의 [JMS 모델](#) 및 [Java.net](#)의 JMS 2.0 스펙을 참조하십시오.

JMS용 IBM MQ 클래스는 다음에서 사용할 수 있습니다.

- 제한사항이 포함된 OSGi JVM 서버입니다.
- 바인딩 모드 또는 클라이언트 모드 전송을 사용하여 JMS 애플리케이션이 큐 관리자에 연결하는 경우의 CICS 표준 모드 Liberty JVM 서버입니다.
- 클라이언트 모드 전송을 사용하여 JMS 애플리케이션이 큐 관리자에 연결하는 경우의 CICS 통합 모드 Liberty JVM 서버입니다.

새 애플리케이션은 Java용 IBM MQ 클래스가 아닌 JMS용 IBM MQ 클래스를 사용해야 합니다.

통신 유형 및 제한사항을 포함한 자세한 정보는 9 페이지의 [『작동 방식: CICS와 함께 JMS용 IBM MQ 클래스 사용』](#)의 내용을 참조하십시오.

Java용 IBM MQ 클래스

Java용 IBM MQ 클래스는 CICS에서 실행되는 Java 애플리케이션에서 사용할 IBM Message Queue Interface(MQI)의 Java 변형을 제공합니다. IBM MQ에 대한 호출 유형은 MQI 호출로 변환되고 기존 CICS-MQ 어댑터에서 정상적으로 처리됩니다. 변환된 요청은 다른 프로그램(예: COBOL 프로그램)의 MQI 요청과 정확히 동일한 방식으로 CICS-MQ 어댑터로 유입됩니다. 따라서 Java 프로그램과 IBM MQ에 액세스하는 기타 프로그램 사이에 운영상의 차이는 없습니다.

Java용 IBM MQ 클래스는 OSGi JVM 서버에서만 사용할 수 있습니다. Java용 IBM MQ 클래스에서 사용하는 기존 애플리케이션이 계속해서 완전히 지원을 받더라도 새 애플리케이션은 JMS용 IBM MQ 클래스를 사용해야 합니다.

자세한 정보는 [OSGi JVM 서버에서 Java용 IBM MQ 클래스 사용](#)의 내용을 참조하십시오.

작동 방식: CICS-MQ 어댑터

CICS 및 CICS-MQ 어댑터는 동일한 주소 공간을 공유합니다. IBM MQ 큐 관리자는 고유한 주소 공간에서 실행됩니다.

3 페이지의 그림 1에서는 CICS, CICS-MQ 어댑터 및 IBM MQ 간의 관계를 보여줍니다.

어댑터의 일부는 IBM MQ 메시지 어댑터와 통신하는 CICS 태스크 관련 사용자 엑시트입니다. CICS 관리 모듈은 종료를 직접 호출합니다. 즉, 애플리케이션 프로그램은 제공된 API 스텝 프로그램(CSQCSTUB)을 통해 종료를 호출합니다. 태스크 관련 사용자 엑시트 및 스텝 프로그램에 대한 자세한 정보는 [태스크 관련 사용자 엑시트 프로그램](#)의 내용을 참조하십시오.

각 CKTI 트랜잭션은 보통 MQGET WAIT 상태로 초기화 큐에 배치된 트리거 메시지에 응답할 준비가 되어 있습니다.

어댑터 관리 인터페이스는 조작 및 제어 기능을 제공합니다(CICS-MQ 어댑터 관리에 설명되어 있음).

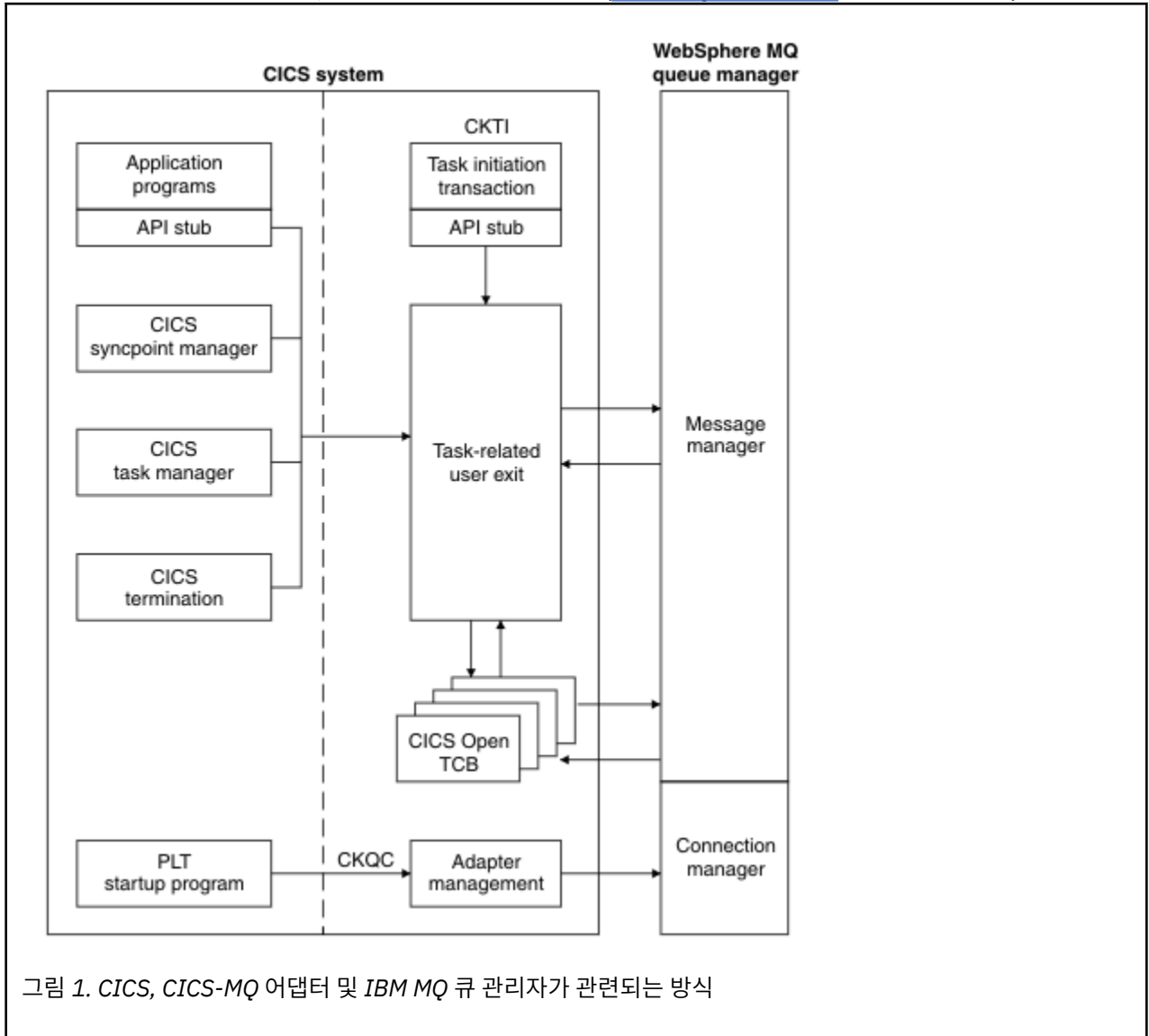


그림 1. CICS, CICS-MQ 어댑터 및 IBM MQ 큐 관리자가 관련되는 방식

CICS-MQ 어댑터 표시 기능은 찾아보기를 위해 출력 데이터를 저장할 각 호출 태스크에 대한 두 가지 기본 임시 스토리지 큐를 사용합니다. 큐의 이름은 `ttttCKRT` 및 `ttttCKDP`입니다. 여기서, `tttt`는 표시 기능이 요청되는 터미널의 ID입니다. 큐에 액세스하지 마십시오.

CICS-MQ 어댑터는 이름 `DFH.genericapplid(8).QMGR`을 사용하여 연결 시작 및 중지 및 같은 처리 중에 CICS ENQ 및 CICS DEQ 호출을 발행합니다. CICS ENQ 또는 DEQ 목적으로 유사한 이름을 사용하지 마십시오.

정보 모니터(CKAM)

정보 모니터 트랜잭션인 CKAM은 두 가지 기능 즉, IBM MQ의 인스턴스에 대한 연결 또는 연결 끊기 요청의 결과로 발생하는 보류 이벤트 처리 및 MQMONITOR에서 초당 발행할 수 있는 최대 MQGET 호출 수 계산이 포함됩니다.

CKAM 및 보류 이벤트

보류 이벤트가 발생하는 경우 정보 모니터는 시스템 콘솔로 전송되는 메시지를 생성합니다.

보류 이벤트의 두 가지 유형은 다음과 같습니다.

연기된 연결

적합한 큐 관리자가 시작되기 전에 CICS가 IBM MQ에 연결하려고 하는 경우 보류 이벤트(연기된 연결이라고 함)가 활성화됩니다. 적합한 큐 관리자는 연결 설정에 따라 IBM MQ 큐 공유 그룹의 이름 지정된 단일 큐 관리자 또는 멤버일 수 있습니다. 적합한 큐 관리자가 사용 가능한 경우 CICS-MQ 어댑터는 연결 요청을 발행하고, 연결되고, 보류 이벤트가 취소됩니다.

다중 연기된 연결이 있을 수 있으며 그 중 하나는 적합한 큐 관리자가 시작될 때 연결됩니다.

연결을 위해 IBM MQ 큐 공유 그룹을 사용 중이고 그룹의 모든 큐 관리자가 사용 불가능한 경우, CICS-MQ 어댑터가 각 큐 관리자에 대한 연결을 초기화하고 이어서 다중 연기된 연결이 발생하게 됩니다. 큐 관리자 중 하나가 시작되면 CICS-MQ 어댑터가 연결합니다. 둘 이상의 큐 관리자가 동일한 LPAR에서 시작되면 CICS-MQ 어댑터는 그 중 하나에 연결됩니다.

종료 알림

성공적으로 IBM MQ에 연결되면 종료 알림을 호출한 보류 이벤트가 작성됩니다. 이 보류 이벤트는 다음 중 하나가 발생하면 만료됩니다.

- CICS가 연결된 큐 관리자가 MODE(QUIESCE)로 정상 종료됩니다. 정보 모니터는 연결 중에 거부 요청을 발행합니다.
- 큐 관리자는 MODE(FORCE)로 종료되거나 비정상적으로 종료됩니다.
- CKQC 트랜잭션에서 연결이 종료됩니다.

CICS에서 처리할 수 있는 최대 보류 이벤트 수는 99입니다. 이 한계에 도달하면 최소 하나의 현재 이벤트가 만료될 때까지 추가 이벤트를 작성할 수 없습니다.

모든 보류 이벤트가 만료될 때 정보 모니터가 자체적으로 중지됩니다. 정보 모니터는 이후에 새 연결 요청을 통해 자동으로 다시 시작됩니다.

CKAM 및 MQMONITOR

CICS 리전에서 z/OS Workload Manager 상태 서비스가 활성 상태인 경우 리전의 상태에 대한 변경사항은 CKAM이 다음과 같이 역할을 수행하는 MQMONITOR에 영향을 미칠 수 있습니다.

- 리전의 z/OS WLM 상태 값이 100 미만인 경우 CKAM은 MQMONITOR에서 초당 발행할 수 있는 최대 MQGET 호출 수를 계산합니다. 이 제한을 계산하는 데 사용되는 공식과 z/OS Workload Manager 상태 서비스에 대한 세부사항은 [z/OS Workload Manager Health 서비스가 MQMONITOR에 미치는 영향의 내용](#)을 참조하십시오.
- z/OS WLM 상태 열기 상태가 OPENING이고 리전의 상태 값이 0보다 큰 경우(예를 들어, CICS 워업 프로세스 중에) CKAM은 AUTOSTART(YES)로 정의된 모든 MQMONITOR를 시작합니다.
- z/OS WLM 상태 열기 상태가 CLOSED이고 리전의 상태 값이 0인 경우(예를 들어, CICS 냉각 프로세스가 종료될 때까지 또는 **SET WLMHEALTH IMMCLOSE**가 발행된 후) CKAM은 모든 MQMONITOR를 중지합니다.

CICS에서 MXT 조건이 발생하면 CKAM은 다음 공식에 표시된 대로 이 조건이 존재하는 동안 MQMONITOR에서 초당 발행할 수 있는 MQGET 호출 수를 계산합니다. 이 제한사항은 CICS가 MXT에 도달할 때 MQMONITOR로 시작되는 태스크의 수를 제한합니다. 조건이 더 이상 존재하지 않으면 CKAM은 제한사항을 제거합니다.

공식: 최대 MQGET 호출 수는 $MXT + 10\%$ 입니다.

MXT 게이팅이 발생하는지 여부를 판별하려면 [트랜잭션 통계 해석](#)의 내용을 참조하십시오.

태스크 초기자 또는 트리거 모니터(CKTI)

CKTI는 MQ 트리거 메시지를 읽을 때 CICS 트랜잭션을 시작합니다. 예를 들어, 메시지는 특정 큐에 배치됩니다.

메시지가 애플리케이션이 메시지 큐에 배치되는 경우 트리거 조건이 충족되면 트리거가 생성됩니다. 그런 다음 큐 관리자는 해당 메시지 큐에 지정된 초기화 큐에 트리거 메시지라고 하는 사용자 정의 데이터가 포함된 메시지를 작성합니다. CICS 환경에서 CKTI의 인스턴스를 설정하여 초기화 큐를 모니터링하고 트리거 메시지가 도달할 때 트리거 메시지를 검색할 수 있습니다. CKTI는 DEFINE PROCESS 명령을 사용하여 지정된 다른 CICS 트랜잭션을 시작하며 일반적으로 애플리케이션 메시지 큐에서 메시지를 읽은 후 처리합니다. 프로세스는 초기화 큐가 아닌 애플리케이션 큐 정의에서 이름이 지정되어야 합니다.

각 CKTI의 사본은 단일 초기화 큐에 서비스를 제공합니다. MQ 초기화 큐를 모니터링하는 둘 이상의 CKTI 모니터링이 제공됩니다.

CKTI 설정 방법

CKTI의 인스턴스는 다음 방법 중 하나로 설정될 수 있습니다.

MQCONN 및 MQMONITOR 자원 사용

CICS 리전에 대한 MQCONN 자원 정의에서 기본 초기화 큐의 이름을 지정할 수 있습니다. MQCONN 자원 정의를 설치하는 경우 CICS는 기본 초기화 큐를 나타내는 예약된 DFHMQINI의 이름으로 MQMONITOR 자원을 동적으로 작성하고 설치합니다. 추가 MQMONITOR 자원을 설치하여 다른 초기화 큐를 모니터링할 수 있습니다.

MQMONITOR 자원을 통해 MQ 큐에 서비스를 제공하는 연관된 트랜잭션이 MQ 큐 관리자에 대한 연결이 설정될 때 자동으로 다시 시작할 수 있습니다. CEMT 또는 CICS API를 사용하여 MQMONITOR를 수동으로 시작하거나 중지할 수 있습니다.

새 MQMONITOR 자원 정의를 작성하기 위해 MQCONN 자원 정의를 변경하고 다시 설치하여 기본 초기화 큐의 이름을 변경할 수 있습니다. 또한 CICS-MQ 연결을 수동으로 시작하는 경우 대체되는 기본 초기화 큐의 이름을 지정할 수도 있습니다.

CKQC 트랜잭션 또는 DFHMQSSQ에 연결하는 사용자 작성 프로그램에서 터미널 사용

CKQC 트랜잭션은 CICS®와 IBM MQ 간의 연결을 초기화하고, 관리하고, 보기 위해 CICS-MQ 어댑터의 제어 기능에 대한 액세스를 제공합니다. CKTI의 인스턴스를 시작하거나 중지하기 위해 **CKQC STARTCKTI** 또는 **CKQC STOPCKTI** 명령을 사용할 수 있습니다. CKQC 트랜잭션 또는 어댑터 태스크 초기화 프로그램인 DFHMQSSQ에 연결하는 CICS 프로그램에서 CKTI를 시작할 수 있습니다.

CKTI의 사본을 시작하거나 중지하려면 이 CKTI가 서비스를 제공할 예정이거나 지금 서비스를 제공 중인 큐의 이름을 제공해야 합니다.

초기화 큐를 지정하지 않고 **CKQC STARTCKTI** 또는 **CKQC STOPCKTI** 명령을 실행하는 경우, 이러한 명령은 MQCONN 자원의 INITQNAME 속성에 지정된 CICS 리전의 기본 초기화 큐를 참조하는 것으로 자동으로 해석되며, 암시적 MQMONITOR이 이미 설치되어 시작된 경우 요청은 실패합니다. 사용자가 MQCONN 자원 정의를 설치하는 경우, CICS는 기본 초기화 큐를 표시하기 위해 암시적 MQMONITOR 자원 정의를 작성하고 설치합니다. 새 암시적 MQMONITOR 자원 정의를 작성하기 위해 MQCONN 자원 정의를 변경하고 다시 설치하여 기본 초기화 큐의 이름을 변경할 수 있습니다. 또한 CICS-MQ 연결을 수동으로 시작하는 경우 대체되는 기본 초기화 큐의 이름을 지정할 수도 있습니다.

CKQC는 MQMONITOR 자원을 제어하지 않습니다.



경고: CKTI 트랜잭션의 인스턴스를 관리하기 위해 CKQC 트랜잭션과 MQMONITOR를 동시에 사용하면, CKQC가 MQMONITOR를 인식하지 않고 MQMONITOR가 CKQC를 사용하여 관리되는 CKTI 트랜잭션을 인식하지 못하기 때문에 MQMONITOR 상태와 통계가 혼동될 수 있습니다.

참고: **CKQC STOPCKTI**가 실행되는 경우 MQMONITOR와 연관된 트랜잭션을 포함하여 초기화 큐를 모니터링하는 모든 트랜잭션이 중지됩니다.

CICS-MQ 어댑터에 대해 제공된 자원 정의

CICS는 DFHLIST의 일부로 DFHMQ 그룹에 있는 CICS-MQ 어댑터에 대한 CSD 정의를 제공합니다. 샘플 애플리케이션 프로그램의 정의는 CSQ4SAMP 그룹의 IBM MQ에서 제공됩니다.

DFHMQ에는 다음 정의가 포함됩니다.

- 제공된 어댑터 프로그램
- 제공된 어댑터 관리 트랜잭션

- 제공된 BMS 맵 세트(어댑터 패널에 필요함)

프로그램 이름의 형식은 DFHMQxxx(및 호환성을 위한 일부 CSQCxxxx 별명)이고 트랜잭션 이름의 형식은 CKxx입니다.

CICS-MQ 어댑터가 IBM MQ 제품과 함께 제공했던 경우에는 IBM MQ에서 CSQCAT1 및 CSQCKB CSD 그룹을 제공했습니다. CSQCAT1 및 CSQCKB 그룹은 CICS TS for z/OS, 버전 5.6에 설치되지 않아야 합니다. 이 그룹을 처리하는 방법에 대한 정보는, [CICS-MQ 어댑터 설정](#)의 내용을 참조하십시오.

샘플 애플리케이션 프로그램의 정의가 포함된 CSQ4SAMP 샘플을 CICS CSD에 설치할 수 있습니다. JCL을 이 샘플에 설치하려면 [CICS-MQ 어댑터 설정](#)의 내용을 참조하십시오. CICS CSD에 자원을 정의하지 않고 CICS 프로그램 자동 설치를 사용할 경우 자동 설치된 정의가 CICS CSD DFHMQ에 제공된 정의에 맵핑되었는지 확인해야 합니다.

작동 방식: CICS-MQ 브릿지

CICS-MQ 브릿지는 CICS 환경에서 실행 중이 아닌 애플리케이션이 IBM MQ 메시지를 사용하여 CICS에서 프로그램 또는 트랜잭션을 실행하고 응답을 얻을 수 있게 합니다.

비CICS 애플리케이션은 z/OS용 IBM MQ를 포함하는 IBM MQ 네트워크에 대한 액세스 권한이 있는 환경에서 CICS-MQ 브릿지를 사용할 수 있습니다. 비CICS 애플리케이션은 다음 CICS 프로그램이 실행되기 전에 응답이 다시 돌아오기를 기다릴 수 있거나(동기 처리) 응답을 기다리지 않고 하나 이상의 CICS 프로그램을 실행하도록 요청할 수 있습니다(비동기 처리).

CICS-MQ 브릿지를 사용하여 CICS에서 프로그램 또는 트랜잭션을 실행할 때 브릿지는 내재된 메시지 큐 인터페이스(MQI) 지원을 사용하므로 CICS 프로그램에 IBM MQ 호출이 포함되지 않습니다. 따라서 레거시 CICS 애플리케이션을 다시 작성, 재컴파일 또는 재링크할 필요없이 IBM MQ 메시지를 통해 다른 플랫폼에서 제어되도록 리엔지니어링할 수 있습니다. IBM MQ 메시지에는 CICS 애플리케이션에 대한 제어 옵션을 제공하기 위한 IBM MQ CICS 정보 헤더(MQCIH 구조)도 포함되어 있습니다.

다음 유형의 CICS 애플리케이션은 CICS-MQ 브릿지와 함께 사용하기에 적합합니다.

- DPL 프로그램으로 알려진 **EXEC CICS LINK** 명령을 사용하여 호출된 CICS 프로그램. 프로그램은 CICS API의 DPL 서브세트를 준수해야 합니다. 즉, CICS 터미널 또는 동기점 기능을 사용해서는 안 됩니다. CICS-MQ 브릿지를 사용하여 단일 CICS 프로그램 또는 작업 단위를 구성하는 CICS 프로그램 세트를 실행할 수 있습니다.
- 3270 트랜잭션으로 알려지도록 3270 터미널에서 실행되도록 디자인된 CICS 트랜잭션. 트랜잭션은 BMS(Basic Mapping Support) 또는 터미널 제어 명령을 사용할 수 있습니다. 대화식 또는 의사 대화의 일부일 수 있습니다. 동기점을 발행할 수 있습니다.

일반적으로 비CICS 애플리케이션이 CICS 트랜잭션에서 제어의 내부 로직 및 플로우와 상호작용해야 하므로 CICS-MQ 브릿지를 통해 3270 트랜잭션을 실행하려면 좀 더 복잡한 애플리케이션 프로그래밍이 필요합니다. CICS 애플리케이션의 비즈니스 로직을 포함하는 DPL 프로그램을 실행하는 것이 바람직합니다. 그러나 일부 CICS 애플리케이션은 표시 로직과 다른 애플리케이션의 비즈니스 로직으로 구성되지 않으므로 CICS-MQ 브릿지를 사용하여 애플리케이션의 유형과 통신할 수 있습니다.

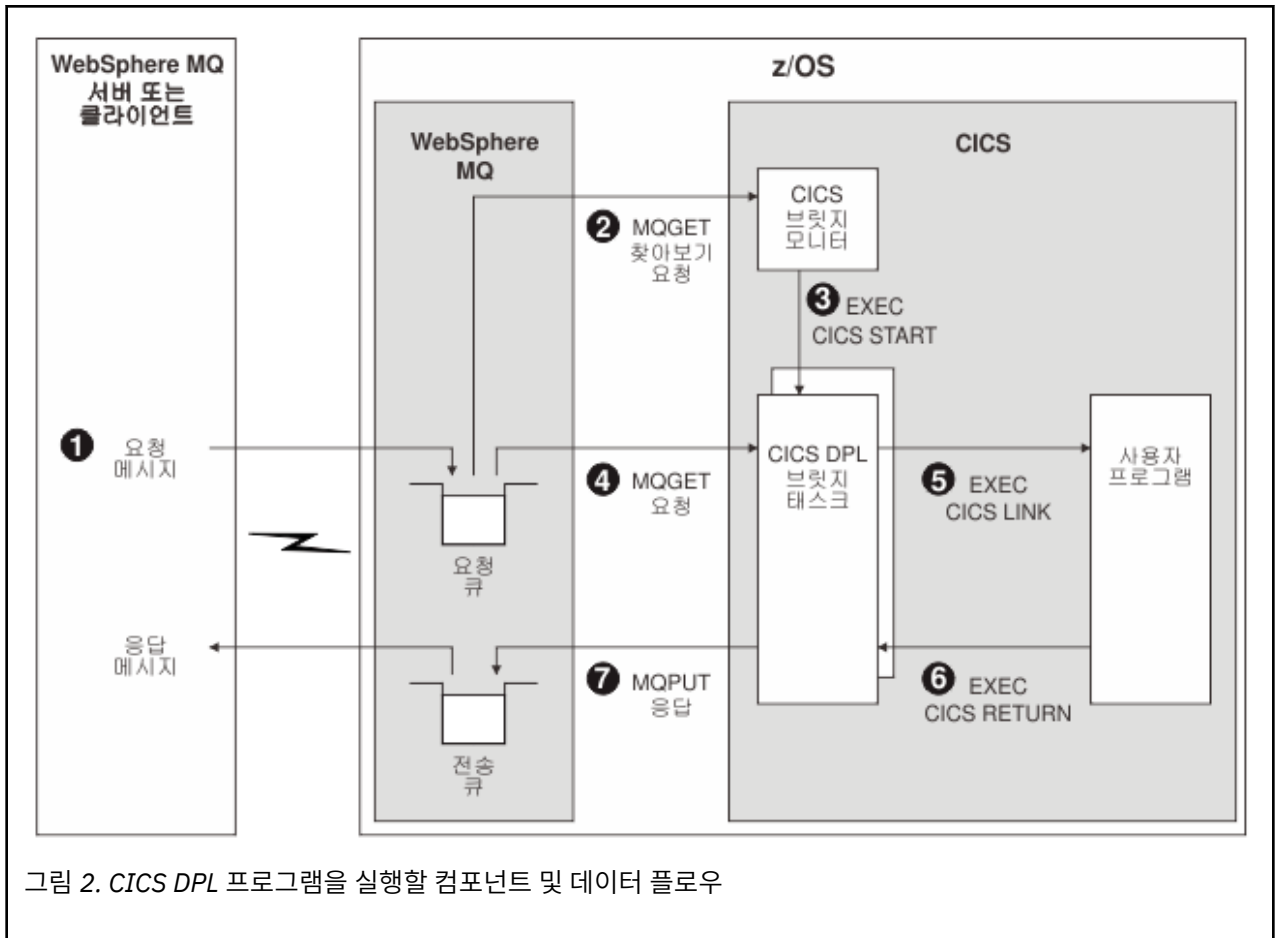
CICS-MQ 브릿지는 표준 CICS 및 IBM MQ 보안 기능을 사용합니다. 요청자의 사용자 ID를 인증, 신뢰 또는 무시하도록 구성할 수 있습니다. IBM MQ 메시지의 MQCIH 구조는 보안 검사에 대한 데이터를 제공합니다.

CICS-MQ 브릿지는 기본 이름이 CKBR인 트랜잭션인 브릿지 모니터를 사용하며 CICS 애플리케이션을 실행하기 위한 새 요청에 대한 IBM MQ 요청 큐를 찾습니다. 요청 큐는 CICS-MQ 어댑터에 연결된 로컬 z/OS 큐 관리자에 있어야 합니다. CICS-MQ 브릿지 태스크는 일반적으로 브릿지 모니터와 같이 CICS 리전에서 실행됩니다. 사용자 프로그램은 필요에 따라 CICS 트랜잭션 라우팅을 사용하여 동일하거나 다른 CICS 리전에 있을 수 있습니다. 특정 CICS 리전에서 요청을 처리해야 하는 경우 MQCIH 헤더에서 CICS 리전의 이름을 지정할 수 있습니다.

CICS DPL 프로그램이 CICS-MQ 브릿지에서 실행되는 방식

IBM MQ 메시지는 CICS DPL 프로그램을 실행하는 데 필요한 데이터를 제공합니다. CICS-MQ 브릿지는 이 데이터에서 COMMAREA 또는 컨테이너를 빌드하고 **EXEC CICS LINK**를 사용하여 프로그램을 실행합니다.

[7 페이지의 그림 2](#)에서는 CICS DPL 프로그램을 실행하기 위해 단일 메시지를 처리하는 데 수행될 조치의 순서를 보여줍니다.



1. CICS 프로그램을 실행하기 위한 요청이 포함된 메시지가 요청 큐에 도달합니다.
2. 계속해서 큐를 찾고 있는 브릿지 모니터 태스크가 메시지를 확인합니다.
3. 브릿지 모니터 태스크는 지정된 트랜잭션 또는 CKBC를 사용하여 CICS DPL 브릿지 태스크를 시작하거나(컨테이너를 사용하도록 선택한 경우) 기본 CKBP 트랜잭션을 시작할 수 있습니다(COMMAREA를 사용하도록 선택한 경우). CKBC는 COMMAREA 대신 채널 및 컨테이너를 사용합니다. 사용된 프로그램은 DFHMQBP3입니다. CKBP는 COMMAREA를 사용하고 사용된 프로그램은 DFHMQBP0입니다. 고유한 지정된 트랜잭션을 사용하는 경우 사용하도록 선택한 관련 프로그램을 선택할 수 있습니다.
4. DFHMQBP0 또는 DFHMQBP3 프로그램 중 하나를 위한 CICS DPL 브릿지 태스크는 요청 큐에서 메시지를 수신합니다.
5. DFHMQBP3 프로그램은 DFHMQBR_CHANNEL 채널의 DFHREQUEST 컨테이너에 메시지 페이로드를 배치하고 지정된 프로그램에 대해 EXEC CICS LINK를 실행합니다. DFHMQBP0 프로그램은 COMMAREA에 메시지 페이로드를 배치하고 지정된 프로그램에 대해 **EXEC CICS LINK**를 실행합니다.
6. DFHMQBP3 프로그램은 애플리케이션이 DFHMQBR_CHANNEL 채널의 DFHRESPONSE 컨테이너에 응답 메시지 페이로드를 배치할 것을 예상합니다. DFHMQBP0 프로그램은 애플리케이션이 요청에서 사용된 COMMAREA에 응답 메시지 페이로드를 배치할 것을 예상합니다.
애플리케이션이 응답 메시지의 페이로드를 제공하지 않으면 이는 여전히 컨테이너를 리턴해야 하며, 이 경우에 이는 비어 있는 컨테이너입니다. 참고로, 응답은 여전히 페이로드 없이 헤더만으로 발송됩니다.
클라이언트가 응답이 발송되지 않음을 확인하려면 요청 메시지 MQMD에서 리플라이-투 큐 이름 (**ReplytoQueue**)을 생략하십시오.
7. DFHMQBP3 프로그램은 DFHRESPONSE 컨테이너의 콘텐츠를 응답 메시지 페이로드에 배치하고 응답 대상 큐에 응답 메시지도 배치합니다. DFHMQBP0 프로그램은 COMMAREA 컨테이너의 콘텐츠를 응답 메시지 페이로드에 배치하고 응답 대상 큐에 응답 메시지도 배치합니다.

사용자 트랜잭션이 종료되고 추가 입력을 요청합니다. 이 플로우가 의사 대화에서 마지막 플로우인 경우 트랜잭션이 종료됩니다. 메시지가 마지막 메시지가 아닌 경우 트랜잭션이 다음 메시지가 수신될 때까지 기다리거나 지정된 제한시간 간격이 만료됩니다.

다중 CICS 프로그램

작업 단위가 단일 사용자 프로그램일 수 있거나 다중 사용자 프로그램일 수 있습니다. 작업 단위를 빌드하기 위해 전송할 수 있는 메시지의 수는 제한되지 않습니다.

이 시나리오에서는 많은 메시지로부터 빌드된 작업 단위가 동일한 방식으로 작동됩니다. 단, 작업 단위에서 마지막 메시지가 아닌 한 브릿지 태스크가 마지막 단계에서 다음 요청 메시지를 기다리는 경우는 제외됩니다.

CICS 3270 트랜잭션이 CICS-MQ 브릿지에서 작동되는 방법

IBM MQ 메시지는 CICS 3270 프로그램을 실행하는 데 필요한 데이터를 제공합니다. CICS 트랜잭션은 실제 3270 터미널이 있는 것처럼 실행되지만, 대신 CICS 트랜잭션과 IBM MQ 애플리케이션 간에 통신하기 위해 하나 이상의 IBM MQ 메시지를 사용합니다.

CICS-MQ 브릿지는 CICS Link3270 메커니즘을 사용하여 CICS 트랜잭션에 액세스합니다. 기존 3270 에뮬레이터와는 달리 CICS-MQ 브릿지는 z/OS Communications Server 플로우를 IBM MQ 메시지로 대체하여 작동되지 않습니다. 대신, 메시지는 벡터라고 하는 많은 파트로 구성되며, 메시지는 각각 **EXEC CICS** 요청에 해당합니다. 그러므로 이 애플리케이션은 에뮬레이터를 통하지 않고 트랜잭션(애플리케이션 데이터 구조 또는 ADS라고 함)에 사용된 데이터를 사용하여 CICS 트랜잭션과 직접 대화합니다.

CICS-MQ 브릿지로 CICS 3270 트랜잭션을 실행하기 위한 컴포넌트 및 데이터 플로우는 다음과 같습니다.

1. CICS 트랜잭션을 실행하기 위한 요청이 포함된 메시지는 요청 큐에 배치됩니다.
2. 계속해서 큐를 찾는 CICS-MQ 브릿지 모니터 태스크는 '시작 작업 단위' 메시지가 대기 중임을 인식합니다 (CorrelId=MQCI_NEW_SESSION).
3. 관련 인증 검사가 수행되고 CICS 3270 브릿지 태스크는 특정 사용자 ID를 사용하여 적절한 권한으로 시작됩니다(브릿지 모니터를 시작하는 데 사용되는 옵션에 따라 달라짐).
4. CICS-MQ 브릿지는 큐에서 메시지를 제거하고 태스크를 변경하여 사용자 트랜잭션을 실행합니다.
5. 메시지의 벡터는 트랜잭션에서 모든 터미널 관련 출력 **EXEC CICS** 요청에 응답하기 위해 데이터를 제공합니다.
6. 터미널 관련 출력 **EXEC CICS** 요청으로 출력 벡터가 빌드됩니다.
7. CICS-MQ 브릿지는 단일 메시지로 모든 출력 벡터를 빌드하고 응답 대상 큐에 이를 배치합니다.
8. CICS 3270 브릿지 태스크가 종료됩니다. 이 플로우가 트랜잭션에서 마지막 플로우인 경우 트랜잭션이 종료됩니다. 메시지가 마지막 메시지가 아닌 경우 트랜잭션이 다음 메시지가 수신될 때까지 기다리거나 지정된 제한시간 간격이 만료됩니다.

다중 CICS 트랜잭션

CICS 애플리케이션은 종종 의사 대화로 함께 연결된 하나 이상의 트랜잭션으로 구성됩니다. 일반적으로 각 트랜잭션은 화면에 데이터를 입력하고 AID 키를 누르는 3270 터미널 사용자가 시작합니다. 애플리케이션의 이 모델은 CICS-MQ 브릿지를 사용하는 비CICS 애플리케이션으로 에뮬레이트될 수 있습니다.

이 모델에서 메시지는 트랜잭션 및 입력 벡터에 대한 정보를 포함하여 첫 번째 트랜잭션을 위해 빌드됩니다. 이 메시지는 큐에 배치됩니다. 응답 메시지는 출력 벡터, 실행될 다음 트랜잭션의 이름 및 토큰(의사 대화를 나타내는 데 사용됨)으로 구성됩니다. IBM MQQ 애플리케이션은 다음 트랜잭션에 설정된 트랜잭션 이름을 사용하여 새 입력 메시지와 이전 메시지에서 리턴된 값으로 설정된 기능 토큰 및 원격 시스템 ID를 빌드합니다. 이 두 번째 트랜잭션을 위한 벡터는 메시지에 추가되고 메시지가 큐에 배치됩니다. 이 프로세스는 애플리케이션이 종료될 때까지 계속됩니다.

동일한 브릿지 세션에서 다중 트랜잭션을 위해 모든 IBM MQ 메시지를 포함할 수 있으며, 이를 통해 모니터링 오버헤드가 줄어들고 성능이 개선됩니다.

CICS 애플리케이션을 작성하기 위한 대체 방법은 대화식 모델입니다. 이 모델에서 원래의 메시지에는 트랜잭션을 실행하기 위한 모든 데이터가 포함되지 않을 수 있습니다. 트랜잭션이 메시지의 벡터로 응답할 수 없는 요청을 발행하는 경우 메시지는 추가 데이터를 요청하는 응답 대상 큐에 배치됩니다. IBM MQ 애플리케이션은 요청을 충족시키도록 벡터를 사용하여 이 메시지를 가져오고 큐에 다시 새 메시지를 배치합니다.

작동 방식: CICS와 함께 JMS용 IBM MQ 클래스 사용

JMS용 IBM MQ 클래스는 CICS에서 실행되는 Java 애플리케이션의 IBM MQ에 대해 선호되는 인터페이스입니다. (Java용 IBM MQ 클래스는 계속해서 지원되지만 최신 애플리케이션은 JMS용 IBM 클래스를 사용해야 합니다.) JMS용 IBM MQ 클래스는 제한사항이 포함된 OSGi JVM 서버 또는 Liberty JVM 서버에서 실행될 수 있습니다.

Java EE 플랫폼에서 JMS용 IBM MQ 클래스는 애플리케이션의 컴포넌트와 IBM MQ 큐 관리자에 다음과 같은 두 가지 통신 유형을 지원합니다.

아웃바운드 통신

JMS API를 직접 사용하면 애플리케이션 컴포넌트는 큐 관리자에 대한 연결을 작성한 후 메시지를 전송하고 수신합니다.

예를 들어, 애플리케이션 컴포넌트는 애플리케이션 클라이언트, 서블릿, JavaServer 페이지(JSP), 엔터프라이즈 Java Bean(EJB) 또는 MDB(Message Driven Bean)일 수 있습니다. 이 통신 유형에서는 애플리케이션 서버 컨테이너가 연결 풀링 및 스레드 관리와 같은 메시징 조작의 지원으로 낮은 레벨 기능만 제공합니다.

인바운드 통신

대상에 도달한 메시지는 메시지를 처리하는 MDB로 전달됩니다.

Java EE 애플리케이션은 MDB를 사용하여 비동기식으로 메시지를 처리합니다. `onMessage()`는 JMS 메시지 리스너의 역할을 하며 메시지 처리 방법을 정의하는 `onMessage()` 메소드로 구현됩니다. MDB는 애플리케이션 서버의 EJB 컨테이너에 배치됩니다. MDB가 구성되는 정확한 방법은 사용 중인 애플리케이션 서버에 따라 달라지지만, 구성 정보는 연결하는 큐 관리자, 큐 관리자에 연결하는 방법, 메시지에 대해 모니터링되는 대상, MDB의 트랜잭션 동작을 지정해야 합니다. 그러면 이 정보가 EJB 컨테이너에서 사용됩니다. MDB의 선택 기준을 충족시키는 메시지가 지정된 대상에서 도달하면 EJB 컨테이너는 JMS용 IBM MQ 클래스를 사용하여 큐 관리자에서 메시지를 검색한 후 `onMessage()` 메소드를 호출하여 메시지를 MDB에 제공합니다.

OSGi JVM 서버에서 JMS용 IBM MQ 클래스

다음 제한사항이 적용됩니다.

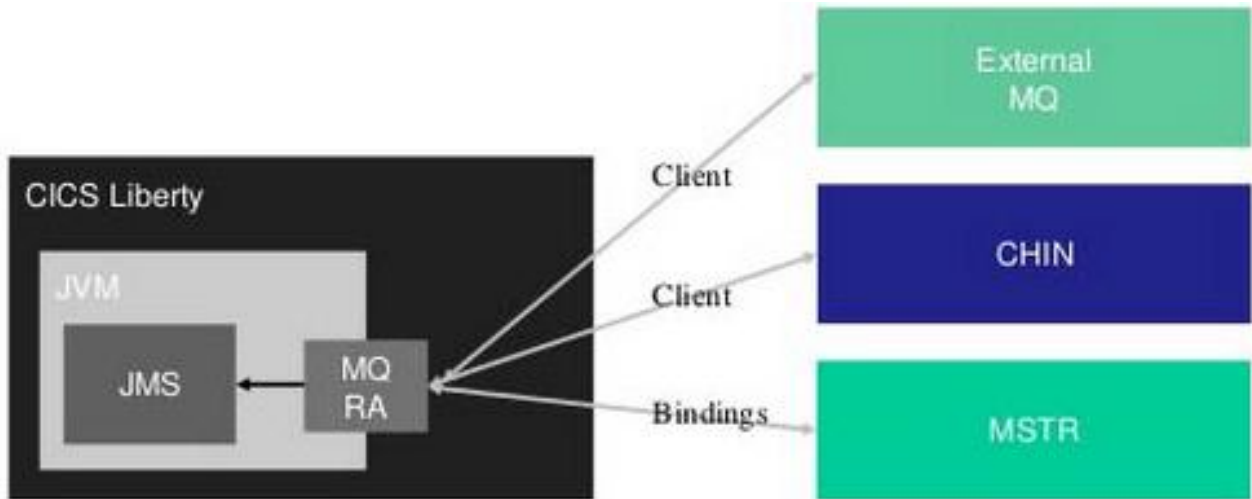
- CICS OSGi JVM 서버에서의 JMS용 IBM MQ 클래스 사용은 CICS 버전 5.2 이상에서만 지원됩니다.
- 클라이언트 모드 연결은 지원되지 않습니다.
- 연결은 IBM MQ 버전 7.1 또는 IBM MQ 버전 8.0 이상에서 관리자를 큐 처리하는 경우에만 지원됩니다. 연결 팩토리의 PROVIDERVERSION 속성은 지정되지 않거나 7 이상의 값이어야 합니다.
- XA 연결 팩토리(예: `com.ibm.mq.jms.MQXAConnectionFactory`) 사용은 지원되지 않습니다.

CICS 표준 모드 Liberty JVM 서버에서의 JMS용 IBM MQ 클래스

JMS 애플리케이션이 바인딩 모드 또는 클라이언트 모드 연결을 사용하여 큐 관리자에 연결하는 경우의 CICS 표준 모드 Liberty JVM 서버입니다.

- IBM MQ 자원 어댑터는 클라이언트 모드에서 IBM MQ의 서비스 중인 버전에 연결할 수 있습니다.
- IBM MQ 자원 어댑터는 동일한 CICS 리전에서 동일한 큐 관리자에 대한 CICS 연결(활성 CICS MQCONN 자원 정의)이 없는 경우 바인딩 모드에서 z/OS용 IBM MQ의 서비스 중인 버전에 연결할 수 있습니다.

Liberty에서 IBM MQ에 연결하려면 버전 9.0.1 이상의 IBM MQ 자원 어댑터가 필요합니다. Liberty에는 IBM MQ 자원 어댑터가 포함되어 있지 않으므로 Fix Central에서 이를 가져와야 합니다([Liberty에서 자원 어댑터 설치 참조](#)).

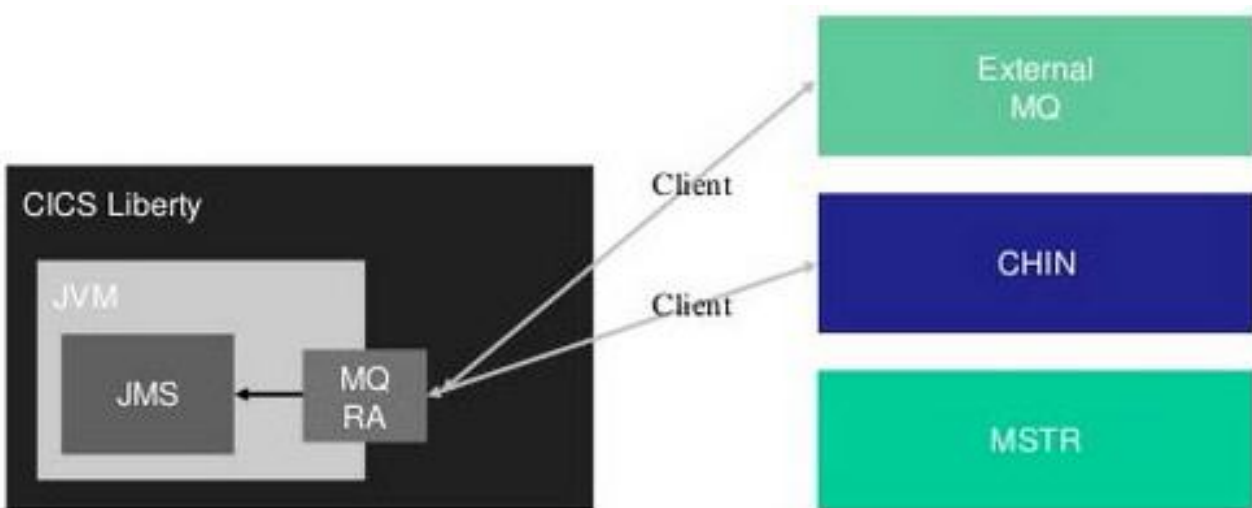


CICS 통합 모드 Liberty JVM 서버의 JMS용 IBM MQ 클래스

클라이언트 모드 연결만 사용하여 JMS 애플리케이션이 큐 관리자에 연결하는 경우의 CICS 통합 모드 Liberty JVM 서버입니다.

- IBM MQ 자원 어댑터는 클라이언트 모드에서 IBM MQ의 서비스 중인 버전에 연결할 수 있습니다.
- 바인딩 모드 연결은 지원되지 않습니다.

Liberty에서 IBM MQ에 연결하려면 버전 9.0.1 이상의 IBM MQ 자원 어댑터가 필요합니다. Liberty에는 IBM MQ 자원 어댑터가 포함되어 있지 않으므로 Fix Central에서 이를 가져와야 합니다([Liberty에서 자원 어댑터 설치 참조](#)).



다음에 수행할 항목

CICS와 함께 JMS용 IBM MQ 클래스를 사용하여 애플리케이션을 개발하는 방법은 CICS 문서에 있는 [OSGI JVM 서버에서 JMS용 IBM MQ 클래스 사용](#) 또는 Liberty JVM 서버에서 [Java용 IBM MQ 클래스 사용](#)의 내용을 참조하십시오. 애플리케이션의 JVM 서버 설정에 대한 정보는 [JMS를 지원하도록 Liberty JVM 서버 구성의 내용](#)을 참조하십시오.

IBM MQ 클래스에 대한 정보는 IBM MQ 문서의 [JMS용 IBM MQ 클래스 사용](#)의 내용을 참조하십시오.

CICS-MQ 트랜잭션 추적 지원

CICS 트랜잭션 추적 지원에는 IBM MQ 트리거와 브릿지 모니터 모두로 초기화되는 태스크가 포함됩니다.

CICS 어댑터 추적 지원에는 IBM MQ 트리거 또는 브릿지 모니터로 초기화되는 태스크가 포함됩니다.

IBM MQ 트리거 또는 브릿지 모니터 태스크는 CICS 리전 내에서 시작하는 태스크의 원본 데이터에 있는 어댑터 필드를 설정하여 CICS 트랜잭션에서 지원됩니다. 다음 어댑터 필드는 계층 구조 순서에 사용됩니다. 어댑터 ID 는 일반 정보를 지정하고 어댑터 데이터 3은 가장 특정한 정보를 제공합니다. 이 방법은 태스크를 식별하고 격리 하기 위한 일정한 방식을 제공합니다.

1. 어댑터 ID 시작

사용 중인 IBM MQ 큐 버전의 제품 ID입니다. 예를 들면, ID=IBM WebSphere MQ for z/OS V7.0.1입니다.

2. 어댑터 데이터 1 시작

IBM MQ 큐 관리자의 이름입니다. 예를 들면, QMGR=RQ38입니다.

3. 어댑터 데이터 2 시작

트리거 모니터가 트리거 메시지를 검색하고 이 태스크를 시작한 초기화 큐의 이름입니다. 예를 들면, INITQ=ALIINIT1입니다.

초기화 큐의 이름이 CICS-MQ Bridge 시나리오에 적용되지 않으므로 어댑터 데이터 2는 기본값으로 설정됩니다. 예를 들면, INITQ=적용되지 않음입니다.

4. 어댑터 데이터 3 시작

시작된 태스크의 결과로 메시지가 배치된 애플리케이션 메시지 큐의 이름입니다. 예를 들면, QNAME=ALITRIG1입니다.

원격 트랜잭션의 경우 개별 START 명령에 설정되는 항목에 관계 없이 미리 트랜잭션이 처음 첨부될 때 미리 트랜잭션에 적용되는 어댑터 데이터는 해당 미러를 사용하는 모든 후속 START 명령에 사용됩니다.

MQ 지원을 위한 CICS 자원: MQCONN 및 MQMONITOR

MQCONN은 CICS와 IBM MQ 간의 연결 속성을 정의하는 CICS 자원입니다. MQMONITOR는 트리거 모니터 트랜잭션 CKTI와 같은 MQ 메시지 이용자의 속성을 정의하는 CICS 자원입니다.

MQCONN 정보

CICS와 IBM MQ 간의 연결을 시작하기 전에 MQCONN 자원을 설치해야 합니다.

CICS 리전에서 한 번에 하나의 MQCONN 자원 정의만 설치할 수 있습니다. CICS가 IBM MQ에 연결되지 않은 경우에만 MQCONN 자원을 설치할 수 있습니다. MQCONN 정의를 변경하고 다시 설치하는 경우 연결을 먼저 중지해야 합니다.

MQCONN 자원은 CICS와 IBM MQ 간의 연결에 대한 다음 설정을 지정합니다.

- CICS가 연결하는 단일 MQ 큐 관리자 또는 MQ 큐 관리자의 큐 공유 그룹의 이름입니다(MQNAME 속성).
- 연결이 작업 단위 미처리로 종료되면 CICS가 채택하는 재동기화 전략입니다(RESYNCMEMBER 속성).
- 연결을 위한 기본 초기화 큐의 이름입니다(INITQNAME 속성).

CICS Explorer®, CEDA, CEMT 또는 SPI 명령을 사용하고 CICSplex® SM도 사용하여 MQCONN 자원을 관리할 수 있습니다.

MQMONITOR 정보

MQMONITOR 자원은 CICS MQ 모니터에 대한 다음 설정을 지정합니다.

- MQ 큐 관리자에 대한 연결이 설정될 때 MQMONITOR가 자동으로 시작할 수 있도록 하는 MQMONITOR의 자동 시작 전략입니다(AUTOSTART 속성).
- 자원의 상태입니다(ENABLESTATUS 속성).
- 해당하는 경우 CICS-MQ 브릿지와 같이 큐를 모니터하는 트랜잭션에 전달할 데이터입니다(MONDATA 속성).
- 모니터할 MQ 큐의 이름입니다(QNAME 속성).
- MQ 큐를 모니터하는 태스크가 사용하는 사용자 ID입니다(MONUSERID 속성).
- MQ 모니터의 상태입니다(MONSTATUS 속성).
- MQ 큐를 모니터하는 태스크의 트랜잭션 ID입니다(TRANSACTION 속성).

- 다른 소스에서 사용 가능한 적절한 사용자 ID가 없는 경우 애플리케이션 트랜잭션을 시작하기 위해 기본적으로 사용할 수 있는 사용자 ID입니다(USERID 속성).

초기화 큐를 포함하여 MQ 큐에서 수신 메시지의 처리를 위해 임의의 수의 MQMONITOR 자원을 설치할 수 있습니다. MQ 큐에서 수신 메시지를 처리하는 MQMONITOR 자원은 두 개 이상일 수 있습니다.

MQMONITOR 자원을 사용하여 MQ 초기화 큐의 수신 메시지를 처리하려면 CKTI로 설정된(또는 비워둔) TRANSACTION 속성과 MQ 초기화 큐의 이름으로 설정된 MQMONITOR 속성을 사용하여 MQMONITOR 자원을 정의해야 합니다.

기본 CKTI 대신에 사용자가 쓴 MQ 모니터 프로그램을 MQMONITOR 자원에 사용할 경우, 사용자가 쓴 프로그램이 애플리케이션 큐에서 직접 메시지를 가져오고 필수 로직을 수행해야 합니다. 프로그램을 코딩할 때에는 반드시 사용자 작성 MQ 트리거 모니터 및 MQ 메시지 이용자 개발과 사용에 설명된 가이드라인을 따르십시오.

CICS Explorer, CEDA, CEMT 또는 SPI 명령을 사용하고 CICSplex SM도 사용하여 MQMONITOR 자원을 관리할 수 있습니다.

보안 검사가 활성화 상태일 때 CICS는 시작할 MQ 모니터 상태를 설정하려는 트랜잭션과 연관된 사용자 ID의 보안 검사를 수행해야 합니다. 사용자 ID가 MONUSERID에 정의된 사용자 ID의 대용이며 MONUSERID와 연관된 트랜잭션을 시작하도록 권한이 부여되었는지 확인해야 합니다. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTPUIUSR 사용자 ID(지정된 경우)입니다.

MQMONITOR에 미치는 영향

여러 측면의 CICS 리전은 MQMONITOR에서 발행할 수 있는 MQGET 호출 수에 영향을 줄 수 있습니다.

활성 z/OS Workload Manager 상태 서비스

CICS 리전에서 z/OS 워크로드 관리자 상태 서비스가 활성화 상태인 경우 MQMONITOR는 리전의 상태에 반응하며 z/OS 워크로드 관리자 상태 서비스 오픈 상태 설정의 영향을 받을 수 있습니다. 세부사항은 [z/OS Workload Manager Health 서비스](#)가 MQMONITOR에 미치는 영향의 내용을 참조하십시오.

MXT 조건 하의 CICS

CICS에서 MXT 조건이 발생하면 제한사항은 이 조건이 존재할 때 MQMONITOR에서 초당 발행할 수 있는 MQGET 호출의 수에 적용됩니다. 자세한 정보는 4 페이지의 『정보 모니터(CKAM)』의 내용을 참조하십시오.

동적으로 설치된 MQMONITOR 자원 DFHMQINI 정보

또한 INITQNAME 속성의 값을 지정하는 MQCONN 자원을 설치할 때 CICS는 기본 시작 큐를 나타내는 예약된 이름 DFHMQINI로 MQMONITOR 자원을 동적으로 작성하고 설치합니다.

DFHMQINI의 속성은 다음과 같습니다.

QNAME

시작 큐 이름을 지정합니다.

TRANSACTION

CKTI(기본값)

MONUSERID

다음과 같이 속성을 가져옵니다.

- PLTPUIUSR 시스템 초기화 매개변수에서(사용 가능한 경우)
- 그 외의 경우 CICS 리전 사용자 ID에서

USERID

값은 CICS 기본 사용자 ID입니다.

DFHMQINI는 MQ 연결이 설정되고 CKTI 트랜잭션과 연관된 사용자 ID를 MONUSERID에서 얻게 되면 자동으로 시작됩니다.

EXEC CICS INQUIRE MQMONITOR 또는 CEMT INQUIRE MQMONITOR 명령을 사용하여 동적으로 작성된 MQMONITOR 자원의 QNAME 속성을 조회할 수 있습니다. 이 QNAME 값을 변경하려면 먼저 MQCONN 자원의

INITQNAME 속성을 변경한 후 MQCONN 자원을 다시 설치해야 합니다. **INITQNAME** 속성에 대한 설정이 포함된 MQCONN 자원을 버리는 경우에는 동적으로 작성된 MQMONITOR 자원과 사용자 정의 MQMONITOR 자원도 함께 버려집니다.



경고: CKTI 트랜잭션의 인스턴스를 관리하기 위해 CKQC 트랜잭션과 MQMONITOR를 동시에 사용하면, CKQC가 MQMONITOR를 인식하지 않고 MQMONITOR가 CKQC를 사용하여 관리되는 CKTI 트랜잭션을 인식하지 못하기 때문에 MQMONITOR 상태와 통계가 혼동될 수 있습니다.

MQMONITOR에 대한 z/OS Workload Manager 상태 서비스의 영향

z/OS Workload Manager 상태 서비스가 CICS 리전에서 활성 상태인 경우 CICS에는 시스템 초기화 완료 후의 워업 프로세스와 종료 전의 냉각 프로세스가 있을 수 있습니다. 시스템 워업 또는 냉각 중에 CICS는 작업의 제어 플로우에 대한 리전의 상태 값을 리전으로 조정합니다. 리전의 상태 변경은 MQMONITOR에 영향을 미칠 수 있습니다. 예를 들어, 속성이 AUTOSTART(YES)인 MQMONITOR가 시작되거나 중지되는 경우입니다. 상태 값이 100 미만인 경우 시작된 모든 MQMONITOR는 단일 MQMONITOR에서 처리할 수 있는 메시지를 제한하는 제한이 적용됩니다.

MQMONITOR 시작으로 미치는 영향

MQMONITOR가 자동으로 시작되도록 설정되는 경우 WLMHEALTH 열기 상태가 OPEN 또는 OPENING이고 리전의 상태 값이 0보다 클 때 시작됩니다.

예를 들어, 시스템 초기화가 완료된 경우 CICS는 리전의 상태 값이 100%에 도달할 때까지 0부터 증가하도록 사전 설정된 간격으로 z/OS WLM Health API(IWM4HLTH) 호출을 시작합니다. 그러므로 첫 번째 간격 후 상태 값은 0보다 커지게 되며 그런 다음 속성이 AUTOSTART(YES)인 MQMONITOR가 시작됩니다. 간격 및 조정 값은 **WLMHEALTH** 시스템 초기화 매개변수에 지정됩니다.

참고: z/OS Workload Manager 상태 서비스가 CICS 리전(즉, **WLMHEALTH**가 OFF로 설정됨)에서 사용 안함으로 설정되는 경우 해당 MQMONITOR는 시스템 초기화 종료 시 제어가 CICS에 제공된 후 즉시 시작됩니다.

MQMONITOR 중지로 미치는 영향

WLMHEALTH 열기 상태가 CLOSED이고 리전의 상태 값이 0인 경우 속성이 AUTOSTART(YES)인 MQMONITOR가 중지됩니다. 예를 들어, **SET WLMHEALTH OPENSTATUS(CLOSE)**가 실행되면 리전의 상태 값이 일정 간격으로 줄어들고 상태 값이 0이 되면 리전의 해당 MQMONITOR가 중지됩니다.

SET WLMHEALTH OPENSTATUS(IMMCLOSE)가 실행되면 속성이 AUTOSTART(YES)인 MQMONITOR가 즉시 중지됩니다.

초당 MQGET 호출 수에 대한 제한

리전의 z/OS WLM 상태 값이 100 미만인 경우 MQMONITOR가 초당 발행할 수 있는 MQGET 호출 수에 대한 제한이 있으므로 트리거 태스크의 수의 제어가 시작됩니다. 제한은 리전에서 시작된 모든 MQMONITOR에 영향을 줍니다. 리전의 상태 값이 100에 도달하면 MQGET 호출에 대한 제한이 제거됩니다. 제한은 다음과 같이 **CKAM**으로 계산됩니다.

초당 최대 MQGET 호출 수는 **MXT** 값을 리전의 z/OS WLM 상태와 곱한 값의 두 배입니다.

WLMHEALTH=(20, 25) 및 MXT=400이 리전에 지정된다고 가정하십시오. 다음 예제는 초당 최대 MQGET 호출량이 전체 시스템 워업 프로세스 중에 각 간격 후 변경되는 방법에 대해 설명합니다.

표 1. 예제		
시간	상태 값(%)	초당 최대 MQGET 호출
0(시스템 초기화 종료)	0	0
20	25	200
40	50	400
60	75	600
80(시스템 워업 종료)	100	제한이 제거됨

마찬가지로 시스템 냉각 프로세스 중에 리전의 상태 값은 일정 간격으로 감소하기 시작하며 MQGET 호출에 대한 제한이 적용됩니다. MQMONITOR가 초당 발행할 수 있는 최대 MQGET 호출의 양은 매 간격마다 점차 감소합니다.

제 2 장 IBM MQ에 대한 연결 구성

MQ 메시지를 사용하여 CICS에 워크로드를 제공하기 위해 CICS에서 두 개의 인터페이스, 즉 CICS-MQ 어댑터와 CICS-MQ 브릿지를 제공합니다. 이 절에서는 CICS 리전에서 어댑터와 브릿지를 사용 가능하게 하는 설정 지침을 제공합니다.

CICS-MQ 어댑터 설정

다음 태스크를 완료하여 CICS-MQ 어댑터를 CICS 리전에서 사용할 수 있게 설정합니다.

시작하기 전에

CICS에서 자원을 정의하는 것에 익숙하지 않은 경우 다음 정보를 참조하십시오.

- CICS 리전 설정 및 CICS 시스템 초기화 매개변수 지정에 대한 정보는 [공유 데이터 세트, CSD 및 SYSIN 설정 참조](#)
- CICS에 대한 자원 정의 및 PLT 항목 코드화에 대한 정보는 [CICS 자원 참조](#)

프로시저

1. IBM MQ 라이브러리 `thlqual.SCSQAUTH`를 CICS 프로시저의 STEPLIB 연결에 포함하십시오. 이 라이브러리를 CICS 라이브러리 뒤에 포함시켜 올바른 코드가 사용되는지 확인하십시오.
`thlqual`은 IBM MQ 라이브러리의 상위 레벨 규정자입니다.
2. 다음 IBM MQ 라이브러리를 CICS 프로시저의 DFHRPL 연결에 포함시키십시오. 이러한 라이브러리를 CICS 라이브러리 뒤에 포함시켜 올바른 코드가 사용되는지 확인하십시오.

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

`thlqual`은 IBM MQ 라이브러리의 상위 레벨 규정자입니다.

CICS-MQ API 교차 엑시트(CSQCAPX)를 사용하는 경우 프로그램에 대한 로드 모듈을 포함하는 라이브러리의 이름도 추가하십시오.

SCSQCICS 라이브러리는 IBM MQ 제공 샘플을 실행하려는 경우에만 필요합니다. 그렇지 않은 경우에 이 라이브러리는 CICS 프로시저에서 제거할 수 있습니다.

3. CICS 리전의 MQCONN 자원을 정의하고 설치하십시오.
MQCONN 자원은 연결에 사용할 기본 IBM MQ 큐 관리자 또는 큐 공유 그룹의 이름을 포함하여 CICS와 IBM MQ 사이의 연결 속성을 지정합니다. 지시사항은 [MQCONN 자원 정의 및 설치](#)의 내용을 참조하십시오.
4. 옵션: CICS 리전의 MQMONITOR 자원을 정의하고 설치하십시오.
MQMONITOR 자원에서는 CICS-MQ 트리거 모니터, CKTI와 같은 MQ 메시지 이용자의 속성을 정의합니다. MQMONITOR 자원을 사용하는 것이 CKTI 인스턴스를 제어하는 데 권장되는 메소드입니다. 다른 MQ 큐를 모니터링하도록 MQMONITOR 자원도 설치할 수 있습니다. 추가 정보는 17 페이지의 [『MQMONITOR 자원 정의 및 설치』](#)의 내용을 참조하십시오.
5. CICS 초기화 중에 CICS-MQ 어댑터에서 자동으로 IBM MQ에 연결하려면 다음 메소드 중 하나를 선택하고 지정하십시오.
 - CICS 시스템 초기화 매개변수 **MQCONN=YES**를 지정하십시오. 이 방법은 자동으로 IBM MQ에 연결하는 데 권장됩니다.
 - 구성에서 PLTPI 목록의 DFHMQCOD 또는 사용자의 고유 프로그램 지정의 지시사항에 따라 CICS PLTPI 목록에 DFHMQCOD 프로그램을 포함시키십시오. PLTPI 목록은 **PLTPI** 시스템 초기화 매개변수에서 지정하는 접미부가 포함된 프로그램 목록 테이블(PLT)입니다.
 - WebSphere MQ로의 연결을 시작하도록 PLTPI 프로그램 작성의 지시사항에 따라 자체 프로그램을 작성하고 구성에서 PLTPI 목록의 DFHMQCOD 또는 사용자의 고유 프로그램 지정의 지시사항에 따라 이 프로그램을 CICS PLTPI 목록에 포함시키십시오.

CICS가 IBM MQ에 대한 연결을 시작하려면 먼저 MQCONN 자원 정의가 설치되어 있어야 합니다. CICS 초기화 시에 자동으로 연결을 시작하는 경우, 초기 또는 콜드 스타트에 대해 MQCONN 자원이 **GRPLIST** 시스템 초기화 매개변수에 의해 이름 지정된 하나 이상의 목록에서 이름 지정된 그룹 중 하나에 존재해야 합니다. CICS의 워밍업 또는 긴급 시작의 경우, MQCONN 자원 정의는 이전 CICS 실행이 끝날 때까지는 설치되어 있어야 합니다.

6. 원격 CICS 시스템에 대한 리전 간 통신(IRC)이 있는 CICS 시스템에서 CICS-MQ 어댑터를 사용하는 경우 어댑터를 시작하기 전에 CICS 시스템 초기화 매개변수 IRCSTRT=YES를 지정하여 IRC 기능이 OPEN인지 확인하십시오.

IRC 액세스 방법이 교차 메모리로 정의되는 경우(즉, ACCESSMETHOD(XM)) IRC 기능은 OPEN이어야 합니다.

7. 큐 관리자 및 CICS에서 사용되는 코드화된 문자 세트 ID(CCSID) 및 UTF-8과 UTF-16 코드 페이지가 z/OS 변환 서비스로 구성되는지 확인하십시오.

CICS 코드 페이지는 **LOCALCCSID** 시스템 초기화 매개변수에서 지정됩니다.

8. 다음과 같이 CICS CSD를 갱신하십시오.

- a) CICS의 이전 릴리스와 CSD를 공유하지 않는 경우 CSD에서 그룹 CSQCAT1 및 CSQCKB와 해당 그룹 또는 해당 그룹 항목의 사본을 제거하십시오. 그룹 CSQCAT1에서 CKQQ TDQUEUE도 삭제해야 합니다.

이제 CKQQ에 대한 정의가 CICS CSD 그룹 DFHDCTG에서 제공됩니다.

- b) 이전 CICS 릴리스와 CSD를 공유하면 CSQCAT1 및 CSQCKB와 이러한 그룹의 사본 또는 해당 콘텐츠는 CICS TS V4.1에 설치되지 않습니다. 그룹 CSQCAT1에서 CKQQ TDQUEUE도 삭제해야 합니다.

이제 CKQQ에 대한 정의가 CICS CSD 그룹 DFHDCTG에서 제공됩니다.

9. 샘플 애플리케이션 프로그램에 대한 정의를 포함하는 CSQ4SAMP 샘플을 CSD에 설치하려는 경우 JCS의 이 단편을 CSD 업그레이드(DFHCSDUP) 작업에 추가하십시오.

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

여기서, *yourlist*는 시스템을 콜드 스타트하는 동안 CICS에서 설치할 그룹 목록이며 CICS의 **GRPLIST** 시스템 초기화 매개변수에서 지정됩니다.

참고: CEDA 트랜잭션을 사용하여 재정의된 어댑터 자원을 활성 CICS 시스템에 설치하는 경우 먼저 어댑터를 종료하고 경보 모니터가 작동을 멈출 때까지 대기하십시오.

10. 데드 레터 큐, 기본 전송 큐 및 CICS-MQ 어댑터 오브젝트의 IBM MQ 정의를 업데이트하십시오.

샘플 CSQ4INYG를 사용할 수 있지만 CICS 리전의 MQINI 자원 정의에 있는 기본 시작 큐 이름과 일치시키기 위해 시작 큐 이름을 변경해야 될 수도 있습니다. 큐 관리자 시작 프로시저의 CSQINP2 DD 연결에서 이 멤버를 사용하거나 CSQUTIL 유틸리티의 COMMAND 함수에 대한 입력으로 이 멤버를 사용하여 필수 DEFINE 명령을 실행할 수 있습니다. IBM MQ를 재시작할 때마다 이 오브젝트를 다시 정의하지 않아도 되기 때문에 CSQUTIL 유틸리티를 사용하는 것이 좋습니다.

MQCONN 자원 정의 및 설치

먼저 MQCONN 자원을 설치해야 CICS에서 IBM MQ에 연결을 시작할 수 있습니다.

이 태스크 정보

MQCONN 자원은 CICS와 IBM MQ 사이의 연결 속성을 지정합니다.

프로시저

1. **CICS 자원 정의 방법**에 나열된 메소드 중 하나를 사용하여 원하는 이름과 그룹으로 MQCONN 자원을 작성하십시오.

CICS 초기화 시 CICS-MQ 연결을 자동으로 시작하려면 CICS **GRPLIST** 시스템 초기화 매개변수를 통해 이름 지정된 목록에 있는 그룹 중 하나에 MQCONN 자원을 두십시오.

2. **MQNAME** 속성을 단일 IBM MQ 큐 관리자 또는 IBM MQ 큐 관리자 큐 공유 그룹의 이름(1-4자)으로 지정하십시오.

큐 공유 그룹을 지정하면 CICS에서 단일 큐 관리자를 대기하지 않고 그룹에 있는 임의 큐 관리자를 사용합니다. 큐 공유 그룹을 사용하면 IBM MQ에 재연결할 때 유연성이 증가하고, CICS 리전과 z/OS 이미지 전체에서 이 CICS 설정 요소를 표준화할 수 있습니다.

3. IBM MQ 큐 관리자의 큐 공유 그룹을 지정한 경우, 연결이 끊기면 **RESYNCMEMBER** 속성을 사용하여 CICS에 적절한 재동기화 전략을 선택하십시오.
 - 기본값인 RESYNCMEMBER(YES)를 통해서는 연결할 미해결 작업 단위가 있는 경우 CICS가 동일한 큐 관리자에 재연결하고 필요하면 대기하도록 지정합니다. 이 전략을 사용하면 인다우트 작업 단위가 해결될 확률이 증가하지만, IBM MQ에 재연결하는 데 걸리는 시간이 늘어날 수 있습니다.
 - RESYNCMEMBER(NO)를 통해서는 CICS에서 같은 큐 관리자에 재연결을 시도하도록 지정합니다. 이 시도에 실패하면 CICS에서 그룹에 있는 적합한 다른 큐 관리자에 연결하고 원본 큐 관리자에 인다우트 작업 단위가 있으면 DFHMQ2064 메시지를 발행합니다. 큐 관리자가 현재 CICS 리전과 동일한 LPAR에서 활성화된 경우 CICS 리전에 연결할 수 있습니다. 이 전략을 사용하면 IBM MQ에 재연결하는 데 걸리는 시간이 줄어들 수 있습니다. 그러나 CICS에서 다른 큐 관리자에 연결하면 인다우트 작업 단위를 자동으로 해결할 수 없으므로 수동으로 해결해야 합니다.
 - RESYNCMEMBER(GROUPRESYNC)를 통해서는 CICS에서 미해결 인다우트 작업 단위와 상관없이 큐-공유 그룹의 멤버에 연결하도록 지정합니다. IBM MQ에서 CICS가 연결하는 큐 관리자를 선택하고 큐 공유 그룹에 있는 적합한 모든 큐 관리자 대신 인다우트 작업 단위를 해결하도록 CICS에 요청합니다. 이 기능은 그룹 복구 단위라고 합니다. 이 옵션은 CICS의 그룹 복구 단위를 지원하는 IBM MQ 릴리스를 실행 중이거나 큐 관리자에서 그룹 복구 단위가 사용으로 설정된 경우에만 사용할 수 있습니다.
4. 연결에 사용할 기본 시동 큐의 이름(1 - 48자)으로 **INITQNAME** 속성을 지정하십시오.
CICS01.INITQ는 CSQ4INYG 오브젝트 샘플로 정의된 시동 큐의 이름입니다.
5. MQCONN 자원을 설치하십시오.

결과

MQCONN 자원 설치 외에도, CICS에서는 기본 시동 큐를 나타내는 예약어, DFHMQINI를 사용하여 MQMONITOR 자원을 동적으로 설치합니다. DFHMQINI에 대한 자세한 정보는 [MQMONITOR 자원](#)의 내용을 참조하십시오.

다음에 수행할 작업

MQCONN 자원을 수정하고 다시 설치하려면 CICS와 IBM MQ 사이의 연결을 중지해야 합니다. CICS가 IBM MQ에 연결되지 않은 경우에만 MQCONN 자원을 설치할 수 있습니다.

동적으로 설치된 MQMONITOR의 **QNAME** 속성을 변경하려면 우선 MQCONN 자원의 **INITQNAME** 속성을 변경한 후에 MQCONN 자원을 다시 설치해야 합니다.

시동 큐 등의 MQ 큐를 모니터링하도록 추가 MQMONITOR 자원을 설치할 수 있습니다. MQ 큐를 모니터링하는 MQMONITOR 자원이 두 개 이상일 수 있습니다.

MQMONITOR DFHMQINI의 기본 설정에 문제가 있으면(예: DFHMQINI로 이주하기가 예상보다 복잡한 것으로 입증됨) DFHMQINI라는 이름의 사용자 정의 MQMONITOR 자원을 설치할 수 있습니다. 그러면 역호환성을 위해 AUTOSTART, STATUS, MONUSERID 및 USERID 속성을 사용자 정의 값으로 설정하는 데 유연성이 있으므로 이 추가 쉬워집니다. TRANSACTION 속성은 CKTI입니다.

MQMONITOR 자원 정의 및 설치

시동 큐 등의 MQ 큐에서 수신 메시지를 처리하기 위해 MQMONITOR 자원을 설치하도록 선택할 수 있습니다. MQ 큐에서 수신 메시지를 처리하는 MQMONITOR 자원은 두 개 이상일 수 있습니다. CICS-MQ 트리거 모니터의 인스턴스인 CKTI를 제어하는 방법으로 MQMONITOR 자원을 사용하는 것이 좋습니다.

시작하기 전에

이전에 MQCONN 자원을 설치한 경우에만 MQMONITOR 자원을 설치할 수 있습니다.

이 태스크 정보

CICS-MQ 트리거 모니터, CKTI를 제어하도록 MQMONITOR 자원을 정의하고 설치할 수 있습니다.

사용자 작성 트리거 모니터 또는 MQ 메시지 이용자를 위한 MQMONITOR 자원도 정의하고 설치할 수 있습니다. 기본 CKTI 대신에 사용자가 쓴 MQ 모니터 프로그램을 MQMONITOR 자원에 사용할 경우, 사용자가 쓴 프로그램이 애플리케이션 큐에서 직접 메시지를 가져오고 필수 로직을 수행해야 합니다. 프로그램을 코딩할 때에는 반드시 사용자 작성 MQ 트리거 모니터 및 MQ 메시지 이용자 개발과 사용에 설명된 가이드라인을 따르십시오.

참고: CICS 제공 MQ 트리거 모니터 프로그램 DFHMQTSK는 CICS-MQ 트리거 모니터 및 태스크 초기자 트랜잭션 CKTI에 사용하도록 예약되어 있습니다. DFHMQTSK를 사용자 트랜잭션으로 호출하려 시도하면 사용자 트랜잭션이 이상 종료 코드 AMQO를 생성하며 이상 종료됩니다.

CICS-MQ 어댑터가 MQ에 연결된 경우에도 언제든지 새 MQMONITOR를 설치할 수 있습니다.

MQMONITOR이 사용 안함으로 설정되고 트랜잭션에서 사용 중이지 않을 때만 기존 MQMONITOR를 (대체하여) 다시 설치할 수 있습니다. **SET MQMONITOR DISABLED STOPPED** 명령을 사용하여 연관된 태스크를 중지하고 자원을 사용 안함으로 설정하십시오.

MQMONITOR이 사용 안함으로 설정되고 연관된 태스크가 없을 때만 버릴 수 있습니다. **SET MQMONITOR DISABLED STOPPED** 명령을 사용하여 연관된 태스크를 중지하고 자원을 사용 안함으로 설정하십시오.

또한 **INITQNAME** 속성의 값을 지정하는 MQCONN 자원을 설치할 때 CICS는 기본 시작 큐를 나타내는 예약된 이름 DFHMQINI로 MQMONITOR 자원을 동적으로 작성하고 설치합니다.

DFHMQINI에 대한 자세한 정보는 [MQMONITOR 자원의 내용](#)을 참조하십시오.

프로시저

1. CICS 자원 정의 방법에 나열된 메소드 중 하나를 사용하여 MQMONITOR 자원을 정의하십시오.
예를 들어 **CREATE MQMONITOR** 명령을 사용하여 MQMONITOR 자원을 정의할 수 있습니다.
[18 페이지의 표 2](#) 목록에서는 MQMONITOR 자원을 작성하는 데 중요한 속성을 보여줍니다.

표 2. 중요한 MQMONITOR 속성			
속성	필수/선택사항	기본값	설명
AUTOSTART	선택사항	예	이 속성은 MQ 모니터의 자동 재시작을 제어합니다. AUTOSTART(YES) MQ 큐 관리자에 대한 연결을 설정할 때 TRANSACTION 속성에 지정된 대로 트랜잭션을 사용하여 자동으로 재시작하십시오. AUTOSTART(NO) MQ 모니터가 자동으로 시작되지 않습니다. MQ 큐 관리자에 대한 연결이 설정된 후 수동으로 MQ 모니터를 시작해야 합니다.
MONUSERID	필수	-	MQ 큐를 모니터링하는 트랜잭션과 연관시킬 사용자 ID를 지정합니다. 이 속성은 보안 검사가 활성 상태(즉, SEC 시스템 초기화 매개변수가 YES로 설정됨)인 경우에만 유효합니다. 보안 검사가 사용 안함으로 설정되면(즉, SEC 가 NO로 설정됨) MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 MQMONITOR 자원의 상태를 시작됨으로 설정한 트랜잭션의 사용자 ID입니다.
QNAME	선택사항	QNAME 이 생략된 경우에는 자원이 설치될 때 값이 기본적으로 &APPLID..INITIATION.QUEUE로 설정됩니다.	모니터링할 MQ 큐의 이름을 지정합니다.

표 2. 중요한 MQMONITOR 속성 (계속)			
속성	필수/선택사항	기본값	설명
STATUS	선택사항	ENABLED	이 속성은 자원을 리전에서 사용 가능하게 만듭니다.
TRANSACTION	선택사항	CKTI	MQ 큐를 모니터링하는 CICS 트랜잭션의 4자로 된 ID를 지정합니다. MQ 시동 큐를 모니터링하기 위해 MQMONITOR를 정의하는 경우, CKTI를 지정하거나 공백으로 두십시오.

CICS에서 MQMONITOR와 연관된 트랜잭션을 자동으로 시작하도록 MQMONITOR 자원에 AUTOSTART(YES) 및 STATUS(ENABLED) 속성이 있는지 확인하십시오.

2. MQMONITOR 자원을 설치하십시오.

MQMONITOR 자원 정의 검사

MQMONITOR를 정의하면 CICS에서 동일한 그룹이나 목록에 있는 다른 자원 정의와의 일관성을 검사합니다.

MQMONITOR 오브젝트의 경우 CICS에서 다음을 검사합니다.

- 동일한 이름의 MQMONITOR가 동일한 목록에 표시되지 않습니다. 동일한 이름이 있으면 경고 메시지가 발행됩니다.
- MQCONN이 그룹 또는 목록에 있습니다. 없으면 경고 메시지가 발행됩니다. MQCONN이 다른 그룹에 있을 수 있으므로 오류가 아닐 수 있습니다. 그러나 MQCONN을 설치해야 MQMONITOR를 설치할 수 있습니다.

IBM MQ로의 연결을 시작하도록 PLTPI 프로그램 작성

샘플 DFHMQPLT를 기반으로, CICS-MQ 연결을 시작하도록 사용자 자신의 PLTPI 프로그램을 쓸 수 있으며, SDFHSAMP 라이브러리에 제공됩니다.

이 태스크 정보

이 샘플은 어셈블러 언어로 작성되지만 사용자 고유 프로그램은 CICS에서 지원하는 모든 언어로 작성할 수 있습니다. CICS PLTPI 프로그램을 작성하는 데 대한 자세한 정보는 [초기화 및 시스템 종료 프로그램 작성의 내용](#)을 참조하십시오.

PLTPI 프로그램은 CICS-MQ 어댑터 프로그램 DFHMQQCN에 링크해야 하며 사용될 연결 값을 지정하는 매개변수 목록을 전달해야 합니다. 매개변수 목록은 [CICS 애플리케이션 프로그램에서 DFHMQQCN에 링크하여 연결 시작](#)에 설명되어 있습니다.

프로시저

- PLTPI 프로그램에서 다음과 같은 LINK 명령을 실행하여 DFHMQQCN에 링크하고 매개변수 목록을 전달해야 합니다.

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

이 예에서 매개변수 목록은 CONNPL이라고 합니다.

이 CICS 시작 단계에서는 터미널이 사용 가능하지 않으므로 COMMAREA 옵션을 사용하여 매개변수 목록을 전달해야 합니다.

결과

MQ 모니터를 사용하는 경우, AUTOSTART(YES) 및 STATUS(ENABLED) 속성으로 MQMONITOR 자원이 설치되어 있으면 MQ 연결이 시작될 때 MQ 모니터가 자동으로 시작됩니다.

PLTPI 목록에서 DFHMQCOD 또는 사용자의 고유 프로그램 지정

DFHMQCOD 또는 고유 프로그램을 사용하여 CICS 초기화 중에 IBM MQ로의 연결을 시작하는 경우 다음 지시사항에 따라 CICS PLTPI 목록에 프로그램을 추가하십시오.

이 태스크 정보

PLTPI 목록은 PLTPI 시스템 초기화 매개변수에서 지정하는 접미부가 포함된 프로그램 목록 테이블(PLT)입니다. CICS 시스템 초기화 매개변수 **MQCONN=YES**를 지정하는 경우 DFHMQCOD 또는 고유 프로그램을 CICS PLTPI 목록에 추가하지 않아도 됩니다. **MQCONN** 매개변수에서는 항상 DFHMQCOD 프로그램을 사용하여 CICS-MQ 연결을 시작하며 다른 이름의 사용자 제공 접속 프로그램을 사용하도록 사용자 정의할 수 없습니다. **MQCONN** 매개변수를 사용하지 않으려는 경우 다음 단계를 계속하십시오.

프로시저

1. 현재 PLTPI 목록에 DFHDELIM 항목이 없으면 하나를 추가하십시오.
DFHMQCOD 또는 동등한 고유 프로그램은 이 CICS 초기화 단계에서 실행해야 하므로, DFHDELIM의 항목 다음에 추가해야 합니다.
2. CICS DFHPLT 매크로를 사용하여 DFHMQCOD 또는 프로그램을 PLTPI 목록에 추가하십시오.
[20 페이지의 그림 3](#)의 내용을 참조하십시오.
3. **PLTPI** 시스템 초기화 매개변수에 PLTPI 목록의 접미부를 올바르게 지정했는지 확인하십시오.
4. **GRPLIST** 시스템 초기화 매개변수로 이름 지정된 하나 이상의 목록에 있는 이름 지정된 그룹 중 하나에 MQCONN 자원이 있는지 확인하십시오.
IBM MQ에 대한 연결을 시작하려면 MQCONN 자원이 필요합니다.

결과

MQ 모니터를 사용하는 경우, AUTOSTART(YES) 및 STATUS(ENABLED) 속성으로 MQMONITOR 자원이 설치되어 있으면 MQ 연결이 시작될 때 MQ 모니터가 자동으로 시작됩니다.

예

이 예에서는 DFHPLT41이라는 CICS PLT에서 DFHDELIM 항목 다음에 DFHMQCOD의 항목을 코딩하는 방법을 보여줍니다.

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

그림 3. CICS-MQ 어댑터와 사용할 샘플 PLT

이 샘플 PLT를 사용하려면 **PLTPI** 시스템 초기화 매개변수에 접미부 41을 지정합니다. PLT 항목을 코딩하는 데 대한 자세한 정보는 [프로그램 목록 테이블\(PLT\)](#)의 내용을 참조하십시오.

CICS-MQ 브릿지 설정

다음 태스크를 완료하여 CICS-MQ 브릿지를 CICS 리전에서 사용할 수 있게 설정합니다.

시작하기 전에

CICS-MQ 브릿지를 사용하려면 IBM MQ 및 CICS 둘 다 동일한 z/OS 이미지에서 실행 중이어야 합니다.

프로시저

CICS에서 [20 페이지의 『1』](#) - [21 페이지의 『4』](#) 단계를 완료하십시오.

1. CICS에서 [15 페이지의 『CICS-MQ 어댑터 설정』](#)의 지시사항을 따라 CICS-MQ 어댑터를 설정하십시오.

2. CICS에서 CICS-MQ 브릿지 구성요소의 자원 정의를 다음과 같이 수정하십시오.

a) 접두부가 CKB인 CICS 임시 저장영역 ID가 복구 불가능으로 정의되었는지 확인하십시오.

CICS-MQ 브릿지에서는 이 임시 저장영역 ID를 사용합니다.

b) CICS-MQ 브릿지를 사용하여 24비트 CICS DPL 프로그램을 실행하려면 CKBP, CKBC 또는 둘 다, 트랜잭션 정의의 TASKDATALOC 속성을 BELOW로 변경하십시오.

CICS DPL 프로그램은 기본적으로 트랜잭션 코드 CKBP를 사용하여 실행됩니다. TASKDATALOC 속성을 변경하지 않고 24비트 프로그램을 실행하는 경우 CICS 이상 종료 AEZC가 발행될 수 있습니다.

c) 다른 트랜잭션 코드를 사용하여 CICS DPL 프로그램을 실행하려는 경우 CKBP, CKBC 또는 둘 다 정의의 사본을 설치하고 트랜잭션 이름을 원하는 대로 변경하십시오.

24비트 프로그램을 실행하려는 경우 TASKDATALOC 속성을 BELOW로 변경하십시오. IBM MQ 요청 메시지의 MQCIH 헤더에 대체 트랜잭션 코드를 지정해야 합니다.

DFHMQ 그룹에 설치된 CICS-MQ 브릿지 구성요소의 전체 목록은 22 페이지의 『CICS에 있는 CICS-MQ 브릿지 트랜잭션 및 프로그램의 정의』의 내용을 참조하십시오.

3. 옵션: 브릿지를 제어하도록 CICS MQ 모니터를 설정하십시오.

MQMONITOR를 사용하면 IBM MQ 관리자에 대한 연결을 설정할 때 자동으로 브릿지를 다시 시작할 수 있습니다. 이 방법은 CICS-MQ 브릿지를 제어하는 데 권장되는 방법 중 하나입니다. 설정 지시사항은 22 페이지의 『CICS-MQ 브릿지에 맞게 MQMONITOR 설정』의 내용을 참조하십시오.

참고: CICS-MQ 브릿지를 제어하는 대체 방법은 브릿지 요청 큐를 시동 큐와 연관시키고, 메시지가 브릿지 요청 큐에 도착하면 CKBR 트랜잭션을 시작하게 할 프로세스를 구현하는 것입니다.

4. 표시 만료 메시지를 DLQ로 라우팅할지 결정하는 ROUTEMEM 브릿지 시작 매개변수를 사용하려면 사용하는 브릿지 큐가 공유인지 아니면 비공유인지에 따라 CICS에 두 선택적 TSMODEL 자원 정의 중 하나를 설치하십시오.

정의는 다음과 같습니다.

a. 비공유 브릿지 큐

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

b. 공유 브릿지 큐

```
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 브릿지 시작 매개변수에 대한 자세한 정보는 CICS-MQ 브릿지 관리의 내용을 참조하십시오.

IBM MQ의 21 페이지의 『5』 - 22 페이지의 『8』 단계를 완료하십시오.

5. IBM MQ에서 요청 메시지의 로컬 큐를 정의하십시오.

IBM MQ 요청 큐는 CICS-MQ 브릿지에 로컬이어야 하고 다른 애플리케이션에서 사용하지 않아야 합니다.

샘플 thlqual.SCSQPROC(CSQ4CKBM)를 사용하여 SYSTEM.CICS.BRIDGE.QUEUE라는 기본 큐를 정의하거나 고유 큐를 정의할 수 있습니다.

고유 로컬 큐를 정의하는 경우 브릿지 모니터 트랜잭션 CKBR을 시작할 때마다 해당 이름을 지정해야 합니다.

고유 로컬 큐를 정의하는 경우 다음 속성을 설정하십시오.

a) 브릿지 모니터와 브릿지 태스크 둘 다에서 큐를 읽을 수 있도록 **SHARE** 속성을 설정하십시오.

b) 복구해야 하는 경우 기본적으로 큐에서 메시지가 지속되도록 **DEFPSIST(YES)** 속성을 설정하십시오.

c) FIFO 순서로 메시지를 처리하려면 **MSGDLVSQ(FIFO)** 속성을 설정하십시오.

그렇지 않으면 우선순위에 따라 메시지가 처리됩니다.

d) **QSGDISP(SHARED)**를 사용하여 요청 큐를 정의한 경우 **INDXTYPE(CORRELID)**를 사용해서도 정의하십시오.

이 설정은 비공유 큐에도 사용하면 좋습니다.

- e) 트리거하여 브릿지 모니터를 시작하려면 **TRIGGER TRIGTYPE(FIRST)PROCESS(procid)** 속성을 설정하십시오. 여기서 *procid*는 **APPLICID(CKBR)**를 지정하는 프로세스입니다.
 - f) 요청 큐의 **BOQNAME** 및 **BOTHRESH**를 지정해 보십시오. 그러면 메시지를 처리할 때 메시지를 데드 레터 큐에 두지 않고, **BOQNAME** 속성으로 지정된 백아웃 큐에 메시지를 넣은 후 **BOTHRESH** 속성으로 지정된 횟수만큼 백아웃합니다.
백아웃 큐를 지정하는 경우 이 큐의 메시지를 처리하도록 프로세스를 제 위치에 두십시오.
 - g) 컨테이너를 사용하는 경우 큐의 **MaxMsgLen** 속성을 사용하여 브릿지 입력 메시지의 크기를 제한해 보십시오. 큐가 수용할 수 있는 최대 메시지 길이는 **MaxMsgLen** 속성을 사용하여 정의합니다.
6. IBM MQ에서 응답을 보유할 큐를 하나 이상 정의하십시오.
응답 큐는 로컬이거나 원격일 수 있습니다.
응답 큐가 원격이면 응답 큐로 전달하기 전에 응답을 보유할 전송 큐를 정의해야 합니다.
 7. IBM MQ에서 데드 레터 큐가 정의되고 이 큐의 메시지를 처리하도록 프로시저가 정의되었는지 확인하십시오.
 8. IBM MQ에서 원격으로 브릿지에 액세스하는 경우 채널 및 전송 큐 정의와 요청 큐의 원격 큐 정의를 작성하십시오.
원격 큐를 사용하는 데 대한 자세한 정보는 [IBM MQ 제품 문서](#)의 내용을 참조하십시오.

CICS에 있는 CICS-MQ 브릿지 트랜잭션 및 프로그램의 정의

브릿지의 정의는 그룹 목록 DFHLIST의 일부인 DFHMQ 그룹에 제공됩니다.

CKBC	채널 및 컨테이너 DPL 브릿지의 Bridge ProgramLink 트랜잭션
DFHMQBP3	채널 및 컨테이너와 사용할 CICS-MQ Bridge DPL 프로그램
CKBP	COMMAREA DPL 브릿지의 Bridge ProgramLink 트랜잭션입니다. 기본적으로 CICS DPL 프로그램은 이 트랜잭션 코드를 사용하여 실행됩니다.
DFHMQBP0	COMMAREA와 사용할 CICS-MQ Bridge DPL 프로그램
CKBR	브릿지 모니터 트랜잭션
DFHMQDCI	데이터 변환 엑시트
DFHMQBRO	브릿지 모니터 프로그램
DFHMQBP1	CICS-MQ 브릿지 DPL 이상 종료 핸들러
DFHMQBRO	CICS-MQ 브릿지 모니터 프로그램
CSQCBP00	DPL 프로그램의 CICS-MQ 브릿지 별명
CSQCBR00	모니터 프로그램의 CICS-MQ 브릿지 별명
CSQCBP53	MQ V531 브릿지, DPL 프로그램의 별명
CSQCBP10	MQ V531 브릿지, DPL 이상 종료 핸들러
CSQCBR53	MQ V531 브릿지, 모니터 프로그램의 별명
DFHMQDCI	CICS-MQ 브릿지 데이터 변환 엑시트
CSQCBDCI	데이터 변환 엑시트의 CICS-MQ 브릿지 별명
DFHCKBR	런타임 처리를 위한 CICS-MQ 브릿지 tsmodel

CICS-MQ 브릿지에 맞게 MQMONITOR 설정

선택적으로 CICS-MQ 브릿지 제어를 위해 MQMONITOR 자원을 정의하고 설치할 수 있습니다. MQMONITOR 자원을 사용하면 IBM MQ 관리자에 대한 연결을 설정할 때 자동으로 브릿지를 다시 시작할 수 있습니다. 이 방법은 CICS-MQ 브릿지를 제어하는 데 권장되는 방법 중 하나입니다.

프로시저

1. 다음과 같은 속성이 있는 MQMONITOR 자원을 정의하십시오.

표 3. CICS-MQ 브릿지를 제어하는 데 사용하는 MQMONITOR 자원 속성			
속성	필수/선택사항	기본값	설명
AUTOSTART	선택사항	예	이 속성은 MQ 모니터의 자동 재시작을 제어합니다. AUTOSTART(YES) IBM MQ 큐 관리자에 대한 연결을 설정할 때 자동으로 브릿지를 다시 시작할 수 있게 합니다. AUTOSTART(NO) MQ 모니터가 자동으로 시작되지 않습니다. IBM MQ 큐 관리자에 대한 연결이 설정된 후 수동으로 MQ 모니터를 시작해야 합니다.
MONDATA	필수	-	CKBR 트랜잭션에 전달될 매개변수를 지정합니다.
MONUSERID	필수	-	CKBR과 연관시킬 사용자 ID를 지정합니다. 이 속성은 보안 검사가 활성 상태(즉, SEC 시스템 초기화 매개변수가 YES로 설정됨)인 경우에만 유효합니다.
STATUS	선택사항	ENABLED	이 속성은 자원을 리전에서 사용 가능하게 만듭니다.
TRANSACTION	필수	CKBR	CICS-MQ 브릿지의 4자로 된 트랜잭션 ID를 지정합니다.

참고: AUTOSTART(YES) 및 STATUS(ENABLED)를 사용하면 MQ 관리자에 대한 연결을 설정할 때 자동으로 브릿지를 시작할 수 있습니다.

2. MQMONITOR 자원을 설치하십시오.

여러 CICS-MQ 브릿지 모니터 설정

CICS-MQ 브릿지를 통해 CICS 애플리케이션에 대한 요청이 많을 것으로 예상되면 추가 CICS 리전에서 추가 브릿지 모니터를 설정해 보십시오. 추가 브릿지 모니터도 사용하여 더 세분화된 보안을 제공할 수 있습니다.

이 태스크 정보

브릿지 모니터의 기본 트랜잭션 ID는 CKBR이지만, 이 ID를 변경하거나 추가 트랜잭션 ID를 정의할 수 있습니다.

여러 브릿지 모니터를 설정할 때 각 브릿지 모니터에서 단독으로 사용하도록 개별 요청 큐를 작성할 수 있습니다. 이 접근 방식의 이점은 각 브릿지 모니터에 서로 다른 서비스 특성을 제공할 수 있다는 것입니다. 그러나 애플리케이션이 CICS-MQ 브릿지를 통해 요청할 때 사용할 적절한 요청 큐의 이름을 알아야 한다는 단점이 있습니다. 따라서 여러 브릿지 모니터에서 동일한 요청 큐를 공유하는 것을 선호할 수도 있습니다.

여러 브릿지 모니터와 공유 요청 큐를 사용하려면 다음 요구사항을 따르십시오.

프로시저

- 브릿지 모니터는 WebSphere 버전 6 이상에서 큐 관리자와 연관되어야 합니다. 브릿지 모니터가 WebSphere MQ 버전 5.3.1 이하 시스템에 연결된 공유 큐에서 여러 브릿지 모니터를 실행하려고 하지 마십시오. 그러면 예기치 못한 결과가 발생할 수 있습니다.

- 브릿지 모니터가 실행 중인 각 CICS 리전에는 고유한 SYSID가 있어야 합니다.
- 3270 의사 대화의 모든 트랜잭션은 의사 대화에 포함된 모든 후속 메시지의 첫 번째 트랜잭션이 리턴한 원격 SYSID를 지정해야 합니다. 원격 SYSID는 CICS 트랜잭션 라우팅 기능을 사용하여 트랜잭션을 다른 CICS 리전으로 보내는 경우에도 필요합니다.
- 브릿지 모니터가 실행 중인 모든 CICS 리전 사이에 CICS ISC 링크를 정의해야 합니다.
- 사용자 유효성 검증에 passticket을 사용하는 경우 브릿지 모니터가 시작될 때 큐의 모든 브릿지 모니터에서 PASSTKTA 매개변수를 사용하여 동일한 APPLID를 지정해야 합니다.

다음에 수행할 작업

여러 CICS 브릿지 모니터로 문제점을 판별하려면 모든 CICS 리전의 모든 로그를 확인하여 브릿지에서 생성한 오류 메시지를 찾아야 할 수도 있습니다. 각 큐 관리자에서 DISPLAY QSTATUS(queue name) TYPE(HANDLE) 명령을 사용하여 CICS 리전에 큐가 열려 있는지 표시할 수 있습니다.

CICS-MQ 브릿지 처리량 제어

고유 클래스에 브릿지 프로그램 트랜잭션, CKBP를 두고 요구사항에 맞게 CLASSMAXTASK를 설정하여 브릿지의 처리량을 제어할 수 있습니다.

브릿지 모니터 CKBR을 통해 검색한 요청 메시지는 CKBR에서 시작한 다음 CKBP에서 이 메시지를 받을 수 있는 시간을 허용하기 위해 MarkBint(Mark Browse Interval)로 지정된 기간 동안 표시되고 숨깁니다. CKBP가 MarkBint 내에 메시지를 받지 못하면 다시 처리할 수 있습니다. CKBP가 특정 이유로 큐에서 메시지를 받지 못하면 재처리가 무기한 계속됩니다. 이는 이 환경에서 브릿지의 기본 조치입니다.

브릿지 시작 데이터에 ROUTEMEM=Y 매개변수를 지정하여 기본 조치를 변경할 수 있습니다. 이 매개변수를 사용하는 경우 표시가 만료되고 재처리 가능하게 되면 메시지가 데드 레터 큐(DLQ)로 라우팅됩니다. ROUTEMEM 브릿지 시작 매개변수에 대한 자세한 정보는 [CICS-MQ 브릿지 관리](#)의 내용을 참조하십시오.

제 3 장 CICS-MQ 어댑터 관리

CICS-MQ 어댑터의 제어 기능을 사용하여 CICS와 IBM MQ 간 연결을 시작하고 관리하고 볼 수 있습니다. CICS-MQ 모니터를 사용하여 CICS-MQ 트리거 모니터 CKTI를 제어할 수 있습니다.

시작하기 전에

CICS-MQ 어댑터를 작동하려면 CICS 리전을 적절하게 구성하고 MQCONN 자원 정의를 설치해야 합니다.

또한 **INITQNAME** 속성의 값을 지정하는 MQCONN 자원을 설치할 때 CICS는 기본 시작 큐를 나타내는 예약된 이름 DFHMQINI로 MQMONITOR 자원을 동적으로 작성하고 설치합니다.

MQ 연결이 설정될 때 DFHMQINI가 자동으로 시작됩니다.

선택적으로 추가 MQMONITOR 자원을 설치하여 IBM MQ 메시지 이용자의 속성을 정의할 수 있습니다(예: 트리거 모니터 트랜잭션 CKTI).

자세한 지시사항은 [CICS-MQ 어댑터 설정의 내용](#)을 참조하십시오.

이 태스크 정보

다음 방법 중 하나를 사용하여 CICS-MQ 어댑터를 관리할 수 있습니다.

CICS-MQ 어댑터의 제어 기능 사용

CKQC 트랜잭션, **EXEC CICS** SPI 및 CEMT 명령, CICSplex SM 또는 CICS Explorer를 사용하여 제어 기능에 액세스할 수 있습니다.

- CEMT 트랜잭션을 사용하여 CICS 환경에서 명령을 실행하거나, 애플리케이션 프로그램에서 **EXEC CICS** 시스템 프로그래밍 인터페이스(SPI) 명령을 사용할 수 있습니다.
- CICS-MQ 어댑터 제어판, 명령행 또는 애플리케이션 프로그램을 통해 CKQC 트랜잭션을 사용할 수 있습니다.
- CICS 리전이 CICSplex SM에서 관리되는 경우, 여러 방법(예: CICSplex SM 웹 사용자 인터페이스, 애플리케이션 프로그램의 EXEC CPSM API(Application Programming Interface) 사용 또는 CMCI 프로그래밍 인터페이스 사용)으로 연결을 제어할 수 있습니다. 이러한 방법은 효율적으로 CICS 시스템 프로그래밍 인터페이스(SPI) 명령을 사용하여 사용 가능한 CICSplex SM 자원 테이블을 업데이트합니다.
- CICS Explorer에는 CICS-MQ 연결 및 MQ 모니터에 대한 SM 조작과 SM 관리 보기가 포함되며, 이 보기는 **EXEC CICS** SPI 명령과 동일한 기능을 제공합니다. CICS 탐색기를 사용하여 CICS 자원에 대한 작업을 수행하는 방법에 대한 정보는 CICS Explorer에서 제공하는 도움말 정보를 참조하십시오.

CICS-MQ 모니터를 사용하여 CICS-MQ 트리거 모니터 CKTI의 인스턴스 제어

MQMONITOR 자원은 CKTI를 제어하는 권장 방법입니다. 이점은 다음과 같습니다.

- CKTI의 여러 인스턴스를 설정하여 MQ 시작 큐를 모니터링할 수 있습니다.
- MQ 큐 관리자에 대한 연결이 설정되면 CKTI를 자동으로 시작할 수 있습니다.
- MQMONITOR를 사용하면 CKTI 또는 메시지 이용 트랜잭션이 사전 설정된 사용자 ID로 실행될 수 있습니다.



경고: CKQC를 사용하여 CKTI의 인스턴스를 시작하거나 중지할 수 있습니다. 하지만 리전이 MQ 모니터를 설치했고 CKTI의 여러 인스턴스가 실행 중인 경우 CKQC를 사용하여 CKTI를 중지하면 예측 불가능한 결과가 나타날 수 있습니다.

운영자 조치 요약

다음 표에는 CICS-MQ 연결에 대해 수행할 수 있는 운영자 조치 및 **EXEC CICS** 및 CEMT 명령, CKQC 트랜잭션, CICS Explorer 또는 CICSplex SM을 사용하여 이러한 조치를 수행할 수 있는지 여부가 요약되어 있습니다.

표 4. CICS-MQ 연결에 대한 운영자 조치			
운영자 조치	EXEC CICS, CEMT	CKQC	CICS Explorer 또는 CICSplex SM
CICS-MQ 연결 시작	예, SET MQCONN 사용. 하지만 기본 시작 큐 이름은 지정할 수 없음	예	예
CICS-MQ 연결 중지	예, SET MQCONN 사용	예	예
연결 상태 및 설정 표시	예, INQUIRE MQCONN 사용	예	예
연결 및 연결 끊기 시간 표시	예, CICS 통계 명령 사용	아니오	예
호출 유형을 포함한 자세한 연결 통계 표시 및 재설정	예, CICS 통계 명령 사용 (모든 통계 재설정)	예(CICS-MQ 연결 통계만 재설정)	아니오
CICS-MQ 연결을 사용 중인 태스크 표시	예, 태스크 수만 표시. INQUIRE MQCONN 사용	예, 태스크 전체 목록	아니오
CICS-MQ를 사용 중인 개별 태스크 제거	예, SET TASK FORCEPURGE 사용	아니오	예
CICS-MQ API 교차 엑시트 사용 또는 사용 안함	아니오	예	아니오
CKTI의 인스턴스 시작 (CICS-MQ 트리거 모니터 또는 태스크 초기자)	아니오	예, 하지만 MQMONITOR를 사용하여 CKTI를 제어하는 것이 권장됨	아니오
CKTI의 인스턴스 중지	아니오	예, 하지만 MQMONITOR를 사용하여 CKTI를 제어하는 것이 권장됨	아니오
MQ 모니터 (MQMONITOR 자원) 시작	예	아니오	예
MQ 모니터 (MQMONITOR 자원) 중지	예	아니오	예
호출 유형을 포함한 자세한 MQMONITOR 통계 표시 및 재설정	예	아니오	예 CICSplex SM은 MQMONITOR 레코드의 CICS 통계를 리턴합니다. CICSRRGN 자원 테이블을 사용하여 통계를 재설정할 수 있습니다.

CICS-MQ 어댑터 제어 패널에 액세스

CICS 트랜잭션 CKQC를 사용하여 어댑터 제어판에 액세스합니다.

이 태스크 정보

큐 관리자를 시작하지 않고 어댑터 제어판에 액세스할 수 있습니다. 또한 연결을 시작할 수 있지만 큐 관리자가 시작될 때까지 활성이 아닙니다.

프로시저

1. CKQC를 입력하고 Enter를 누르십시오.
CICS-MQ 어댑터 초기 제어판이 표시됩니다.
2. 메뉴 표시줄에서 TAB 키를 사용하여 세 가지 옵션 **연결**, **CKTI** 및 **태스크** 간에 이동하십시오. Enter를 눌러 선택하십시오.
3. 선택사항의 번호를 입력한 다음 확인하려면 Enter를 누르고 취소하려면 기능 키 F12를 누르는 방법으로 메뉴 중 하나에서 조치를 선택하십시오.
4. 표시된 패널 또는 보조 창에서 필요에 따라 필드에 새 값을 입력하십시오.
5. 기능 키 F1을 눌러 패널 또는 창에 도움말을 가져오십시오.

명령행의 CKQC 명령

명령행에서 직접 CKQC 트랜잭션을 사용하고 명령 매개변수를 지정하여 CICS-MQ 어댑터 제어판을 건너뛸 수 있습니다.

예를 들어, CICS 명령을 통해, 기본 연결 값을 사용하여 또는 다음 명령을 입력하여 CICS-MQ 연결을 시작할 수 있습니다.

```
CKQC START
```

z/OS 명령을 사용하여 콘솔에서 CKQC 트랜잭션 명령을 실행할 수도 있습니다. 이러한 명령은 다음 양식을 사용합니다.

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

CKQC 트랜잭션 명령은 다음과 같습니다.

CKQC START

IBM MQ에 대한 연결을 시작합니다. 이 명령 및 예제의 구문은 [32 페이지의 『CICS 명령행에서 CICS-MQ 연결 시작』](#)의 내용을 참조하십시오.

CKQC STOP

IBM MQ에 대한 연결을 중지합니다. 이 명령 및 예제의 구문은 [37 페이지의 『CICS 명령행에서 CICS-MQ 연결 중지』](#)의 내용을 참조하십시오.

CKQC DISPLAY

CICS 메시지의 양식으로 IBM MQ에 대한 연결의 중요한 정보를 리턴합니다. 이 명령에는 옵션이 없습니다. 예를 들어, [40 페이지의 『CICS-MQ 연결 통계 및 설정 표시』](#)의 내용을 참조하십시오.

CKQC STARTCKTI

CICS-MQ 트리거 모니터 CKTI의 인스턴스를 시작합니다. 이 명령 및 예제의 구문은 [54 페이지의 『CICS 명령행에서 CKTI 시작』](#)의 내용을 참조하십시오.

CKQC STOPCKTI

CKTI의 인스턴스를 중지합니다. 이 명령 및 예제의 구문은 [56 페이지의 『터미널에서 CKTI의 인스턴스 중지』](#)의 내용을 참조하십시오.

CKQC MODIFY

CICS-MQ 연결 통계를 재설정하고 CICS-MQ 교차 엑시트를 사용 또는 사용 안함으로 설정합니다. 이 명령의 예제는 [CICS-WebSphere MQ API 교차 엑시트 사용, CICS-WebSphere MQ API 교차 엑시트 사용 안함 및 49 페이지의 『CICS-MQ 연결 통계 재설정』](#)의 내용을 참조하십시오. 명령의 전체 구문은 다음과 같습니다.

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

여기서,

Y|N

다음 중 하나를 지정하십시오.

Y

연결 통계를 재설정합니다.

N

연결 통계를 재설정하지 않습니다.

이 매개변수는 필수입니다.

EID

다음 중 하나를 지정하십시오.

E

CICS-MQ API 교차 엑시트를 사용으로 설정합니다.

D

CICS-MQ API 교차 엑시트를 사용 안함으로 설정합니다.

CICS 애플리케이션 프로그램의 CKQC 명령

EXEC CICS LINK 명령을 사용하여 CICS 애플리케이션 프로그램에서 대부분의 CICS-MQ 어댑터 제어 기능에 액세스할 수 있습니다.

연결 시작과 같은 조치를 수행하려면 애플리케이션 프로그램이 적절한 CICS-MQ 어댑터 프로그램에 링크되고 CKQC 명령을 실행해야 합니다. 28 페이지의 표 5에는 이러한 프로그램 및 해당 기능이 나열됩니다.

표 5. CICS-MQ 어댑터 프로그램			
CICS-MQ 어댑터 프로그램	호환성을 위해 허용된 대체 이름	기능	CKQC 명령
DFHMQQCN	CSQCCON	IBM MQ에 대한 연결 시작	CKQC START
DFHMQDSC	CSQCDSC	연결 중지	CKQC STOP
DFHMQRS	CSQCRST	연결 수정(통계 재설정 또는 API 교차 엑시트 사용)	CKQC MODIFY
DFHMQSSQ	CSQCSSQ	CKTI의 인스턴스 시작 또는 중지	CKQC STARTCKTI 또는 CKQC STOPCKTI
DFHMQDSL	CSQCDSP	연결 상태 표시	CKQC DISPLAY

애플리케이션 프로그램의 명령 구분

애플리케이션 프로그램에서 실행되는 몇 가지 명령은 후행 공백으로 채워 명령 길이가 10자가 되도록 설정해야 합니다.

인수가 명령 뒤에 오면 추가 공백 문자를 구분 기호로 추가해야 합니다. 이 제한사항과 각 명령에 필요한 후행 공백의 수에 의해 영향을 받는 명령은 다음과 같습니다.

명령	후행 공백의 수(구분 기호는 포함하지 않음)
START	5
MODIFY	4
STARTCKTI	1
STOPCKTI	2

다른 모든 명령에서는 채우기가 선택사항입니다.

이 예제는 어댑터 명령을 채우는 방법을 보여줍니다. MODIFY 명령은 4개의 후행 공백 및 구분 기호로 사용되는 다른 공백으로 채워야 합니다. MODIFY의 M에서 시작할 때 인수 Y는 12번째 문자입니다.

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

CICS 애플리케이션 프로그램의 매개변수 전달

다음 규칙을 사용하여 CICS와 IBM MQ 간에 매개변수를 전달하는 방법을 판별하십시오.

- CICS 트랜잭션은 접속된 터미널에서 실행되고 있어야 합니다. 실행되지 않으면 모든 IBM MQ 명령이 무시됩니다.
- 접속된 터미널의 CICS 애플리케이션 프로그램이 IBM MQ에 연결되어 있는 경우, **EXEC CICS LINK**와 함께 INPUTMSG 옵션을 사용하여 매개변수를 전달해야 합니다(PLTPI 시간의 경우는 제외).
- PLTPI 시간에 IBM MQ에 연결하는 경우 COMMAREA 옵션을 사용하여 매개변수를 전달해야 합니다. INPUTMSG 옵션을 사용하는 경우 명령이 무시됩니다. 하지만 어댑터 STOP 명령 CKQC STOP 및 CKQC STOP FORCE는 INPUTMSG 옵션 또는 COMMAREA 옵션 중에 어느 옵션을 사용하는지에 관계없이 PLTPI 시간에 실행될 수 없습니다.

EXEC CICS LINK 인터페이스 메시지

EXEC CICS LINK를 사용하여 애플리케이션 프로그램에서 어댑터 오퍼레이션 함수 START 및 STOP을 호출하면 결과 메시지가 시스템 콘솔과 트랜지언트 데이터 큐(TDQ) CKQQ에 기록됩니다.

애플리케이션 프로그램이 LINK에서 리턴되면 큐가 비워질 때까지 **EXEC CICS READQ TD QUEUE(CKQQ)**를 반복하여 메시지를 다시 읽을 수 있습니다. 다음 제한사항이 적용됩니다.

- TDQ 큐 이름은 CKQQ이며 변경될 수 없습니다. CKQQ에 대한 정의가 CICS CSD 그룹 DFHDCTG에 제공됩니다.
- 큐는 기록되기 전에 지워지지 않습니다.
- TDQ에 둘 이상의 애플리케이션이 기록하는 경우 메시지는 직렬화되지 않습니다. 호출 프로그램은 자체 직렬화해야 합니다.
- 동일한 메시지 세트도 시스템 콘솔에 표시됩니다.

CICS-MQ 연결 및 모니터를 위한 EXEC CICS 및 CEMT 명령

EXEC CICS 시스템 프로그래밍 인터페이스(SPI) 명령 및 CEMT 명령을 사용하여 MQCONN 및 MQMONITOR 자원 정의를 관리하고, CICS-MQ 연결 및 MQ 모니터를 시작하고 중지하며, 연결 및 MQ 모니터에 대한 정보와 통계를 표시하고, 연결을 사용 중인 태스크를 제거할 수 있습니다.

CICS 애플리케이션 프로그램에서 **EXEC CICS SPI** 명령을 실행할 수 있습니다. 애플리케이션 프로그램을 사용하지 않고 CICS CEMT 트랜잭션을 통해 직접 CEMT 명령을 실행할 수 있습니다.

다음 표에는 **EXEC CICS SPI** 및 CEMT 명령과 해당 기능이 표시됩니다.

표 6. CICS-MQ 연결 및 모니터에 대한 SPI 및 CEMT 명령		
EXEC CICS 명령	CEMT 명령	기능
CREATE MQCONN	사용할 수 없음	MQCONN 자원 정의 설정
DISCARD MQCONN	CEMT DISCARD MQCONN	MQCONN 자원 정의 버리기 참고: INITQNAME 속성에 대한 설정을 포함하는 MQCONN 자원을 버리면 동적으로 작성된 MQMONITOR 자원 및 모든 사용자 정의 MQMONITOR 자원도 삭제됩니다.
INQUIRE MQCONN	CEMT INQUIRE MQCONN	MQCONN 자원 정의 속성, 연결 상태 및 연결을 사용 중인 태스크의 수 조회

표 6. CICS-MQ 연결 및 모니터에 대한 SPI 및 CEMT 명령 (계속)

EXEC CICS 명령	CEMT 명령	기능
SET MQCONN	CEMT SET MQCONN	연결 시작 및 중지, 기본 큐 관리자 또는 큐 공유 그룹 이름 및 재동기화 전략 변경
CREATE MQMONITOR	사용할 수 없음	MQMONITOR 자원 정의 설정
DISCARD MQMONITOR	CEMT DISCARD MQMONITOR	MQMONITOR 자원 정의 버리기
INQUIRE MQMONITOR	CEMT INQUIRE MQMONITOR	MQMONITOR 자원 정의 속성 및 모니터의 현재 상태 조회(예: 0이 아닌 태스크 ID를 사용하는 MONSTATUS(STARTED))
SET MQMONITOR	CEMT SET MQMONITOR	MQMONITOR 시작 및 중지
INQUIRE SYSTEM	CEMT INQUIRE SYSTEM	설치된 MQCONN 자원 정의의 이름 찾기
SET TASK FORCEPURGE	CEMT SET TASK FORCEPURGE	CICS-MQ 연결을 사용하는 태스크 제거
EXTRACT STATISTICS MQCONN	사용할 수 없음	CICS-MQ 연결에 대한 통계를 온라인으로 보기
SET STATISTICS	CEMT SET STATISTICS	CICS-MQ 연결에 대한 통계를 포함한 모든 통계 재설정

CICS-MQ 연결 및 모니터를 위한 CICSplex SM 보기

CICSplex SM 보기를 사용하여 MQCONN 및 MQMONITOR 자원 정의를 작성하고 수정하며, CICS-MQ 연결을 시작하고 중지하고, 연결 및 모니터에 대한 정보와 통계를 표시할 수 있습니다.

다음 보기 세트는 CICS-MQ 연결 및 모니터를 위한 CICSplex SM에서 사용 가능합니다.

CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 연결

MQCON 보기 세트의 보기에는 연결 상태 및 연결과 연결 끊기 시간을 포함한 연결에 대한 정보가 표시됩니다. 기본 보기에 연결을 시작하고 중지하는 데 사용할 수 있는 조치 단추가 있으며 연결에 대한 설정을 변경할 수 있습니다.

CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 시작 큐

MQINI 보기 세트의 보기에는 자원 서명이 포함된 CICS 시스템에 대한 설치된 MQINI 자원 정의의 속성이 표시됩니다. 이 자원 정의에 대한 어떤 조치도 수행할 수 없습니다.

CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 연결 통계

MQCONN 보기 세트의 보기에는 연결에 대한 정보 및 통계가 표시됩니다. MQCONN 보기 세트는 CICS-MQ 연결을 위해 CICS 자원 정의를 도입하기 전에 존재했던 보기 세트이므로, 이러한 보기에는 연결 시작 및 중지를 위한 조치 단추가 없으며 이러한 보기에서 연결에 대한 설정을 변경할 수 없습니다.

CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 모니터

MQMON 보기 세트의 보기는 CICS 리전에 정의된 MQ 모니터에 대한 상태 정보와 통계를 표시합니다.

관리 > 기본 CICS 자원 관리 > 자원 정의 > WebSphere MQ 연결 정의

MQCONDEF 보기 세트의 보기는 MQCONN 자원 정의의 속성을 표시하며, 이 속성은 MQCONDEF BAS에 의해 CICSplex SM에 표시됩니다. 이러한 보기를 사용하여 MQCONN 자원 정의를 작성하거나 변경하거나 버리거나 설치할 수 있습니다.

관리 > 기본 CICS 자원 관리 > 자원 그룹의 자원 정의 > 자원 그룹의 MQCONN 자원

MQCINGRP 보기 세트는 자원 그룹(RESGROUP)의 MQCONN 자원 정의(MQCONDEF)의 멤버십에 대한 정보를 표시합니다.

MQMINGRP 보기 세트는 자원 그룹(RESGROUP)의 MQMONITOR 자원 정의(MQMONDEF)의 멤버십에 대한 정보를 표시합니다.

CICS-MQ 연결 시작

CICS-MQ 연결이 CICS 초기화 시 시작되지 않으면 여러 방법으로 연결을 시작할 수 있습니다.

시작하기 전에

CICS-MQ 연결을 시작하려면 먼저 CICS 리전을 적절하게 구성하고 MQCONN 자원 정의를 설치해야 합니다. [CICS-MQ 어댑터 설정](#)에서 이를 수행하는 방법을 설명합니다.

이 태스크 정보

나중에 CICS 초기화 시 시작하도록 CICS-MQ 연결을 설정하려면 CICS 시스템 초기화 매개변수 **MQCONN=YES**를 지정하거나 CICS PLTPI 목록에 DFHMQCOD 프로그램 또는 고유 해당 프로그램을 포함하십시오. 이 작업을 수행하기 위한 지시사항은 [CICS-MQ 어댑터 설정](#)의 내용을 참조하십시오.

프로시저

다음 방법 중 하나를 사용하여 CICS-MQ 연결을 시작할 수 있습니다.

- [CICS-MQ 어댑터 제어판 사용](#)

CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어판에서 CICS-MQ 연결을 시작할 수 있습니다.

- [CICS 명령행에서 CKQC START 실행](#)

명령행에서 **CKQC START** 명령을 실행하여 CICS-MQ 연결을 시작할 수 있습니다.

- [EXEC CICS SET MQCONN을 실행하는 CICS 애플리케이션 프로그램 사용](#)

CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 시작할 수 있습니다.

- [DFHMQOCN에 링크된 CICS 애플리케이션 프로그램 사용](#)

CICS 애플리케이션 프로그램에서 CICS-MQ 어댑터 연결 프로그램 DFHMQOCN(또는 프로그램의 이전 이름인 CSQCQCON)에 링크하여 IBM MQ에 대한 연결을 시작할 수 있습니다. 이 연결 시작 방법이 호환성을 위해 유지보수되지만 **EXEC CICS SET MQCONN** 명령 사용은 CICS 애플리케이션 프로그램에서 연결을 시작하는 선호되는 방법입니다.

- [CICS CEMT 트랜잭션 사용](#)

CICS CEMT 트랜잭션에서 **SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 시작할 수 있습니다.

- [CICSplex SM 웹 사용자 인터페이스 사용](#)

CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ 연결** 보기에서 IBM MQ에 대한 연결을 시작할 수 있습니다.

- [CICS Explorer 사용](#)

CICS Explorer에서 CICS-MQ 연결을 시작할 수 있습니다. CICS Explorer는 CEMT 트랜잭션 및 CICSplex SM 웹 사용자 인터페이스에 해당하는 기능을 제공합니다.

결과

CICS 및 MQ 간의 연결이 설정될 때, MQCONN 자원을 CONNECTED로 설정하는 태스크와 연관된 사용자 ID에 연관 트랜잭션을 시작할 충분한 권한이 있으면 CICS는 AUTOSTART(YES) 및 STATUS(ENABLED) 속성으로 설치된 MQ 모니터를 시작합니다.

CICS-MQ 어댑터 제어 패널에서 CICS-MQ 연결 시작

CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어판에서 CICS-MQ 연결을 시작할 수 있습니다.

프로시저

1. CICS-MQ 어댑터 초기 제어판의 메뉴 표시줄에서 **연결**을 선택하십시오.
2. 메뉴에서 **시작** 조치를 선택하십시오.
3. **연결 시작** 보조 매개변수 창에 표시되는 연결 값은 CICS 리전에 대한 설치된 MQCONN 및 MQINI 자원 정의에서 가져옵니다. 큐 관리자 이름은 단일 큐 관리자 또는 큐 공유 그룹일 수 있습니다. 변경하려는 경우 이러한 설정을 겹쳐쓰십시오.
설정을 변경하는 경우 설치된 MQCONN 및 MQINI 자원 정의의 해당 속성(MQNAME 및 INITQNAME) 값은 지정된 설정으로 대체됩니다. 원래 설정으로 되돌리려면 자원 정의를 재설치하십시오.
4. Enter를 눌러 확인하십시오.

결과

연결 시작 시도의 성공 또는 실패를 표시하는 메시지는 어댑터 메시지 패널 CKQCM1에 표시됩니다.

예

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> <div>Start a Connection</div> <div>Type parameters. Then press Enter.</div> <div> 1. Queue Manager Name (SN) . . . QMGR 2. Initiation Queue Name (IQ) CICS.INITIATION.QUEUE1 </div> <div>F1=Help F12=Cancel</div> </div>

F1=Help F3=Exit

그림 4. 연결 시작

CICS 명령행에서 CICS-MQ 연결 시작

명령행에서 **CKQC START** 명령을 실행하여 CICS-MQ 연결을 시작할 수 있습니다.

시작하기 전에

기본적으로 CICS는 키워드와 매개변수 둘 다에 대해 소문자 입력이 대문자 입력과 대칭되게 합니다. 따라서 기본적으로 이러한 명령은 다음과 같습니다.

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

소문자 IBM MQ 큐 이름을 사용하려는 경우 다음을 수행하십시오.

1. 어댑터 제어 기능을 시작하는 터미널의 TYPETERM 정의에 UCTRAN(TRANID)을 지정하십시오.
2. 모든 “CKxx” 트랜잭션에서 사용되는 트랜잭션 프로파일에 UCTRAN(NO)를 지정하십시오.

그 다음 어댑터는 모든 소문자 인수(큐 이름 제외)를 대문자 인수 로 변환합니다.

프로시저

- 큐 관리자 이름 및 기본 시작 큐 이름에 대한 기본 연결 값을 사용하여 연결을 시작하려면 매개변수 없이 명령 **CKQC START**를 실행하십시오.
트랜잭션은 CICS 리전에 대한 설치된 **MQCONN** 및 **MQMONITOR** 자원 정의에서 기본 연결 값을 사용합니다.
- 명시적으로 정의한 연결 값을 사용하여 연결을 시작하려면 다음 명령을 입력하십시오.

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

여기서,

Y|N

다음 중 하나를 지정하십시오.

Y

기본 연결 값을 사용합니다. 즉, 공백 인수가 있는 경우 설치된 **MQCONN** 및 **MQMONITOR** 자원 정의에서 가져온 기본값을 대체합니다.

N

기본 연결 값을 사용하지 않습니다.

<subsystem ID>

CICS가 연결되는 IBM MQ 큐 관리자의 이름입니다. 큐 관리자 이름은 단일 큐 관리자 또는 큐 공유 그룹일 수 있습니다.

<filler>

호환성을 위해 더 이상 사용하지 않는 3자의 추적 번호를 대신하는 3자의 필터입니다. 매개변수가 위치와 관련되어 있으므로 필터는 필수입니다.

<initiation queue name>

연결을 위한 기본 시작 큐의 이름입니다.

매개변수는 위치와 관련됩니다. 기본값을 대체하려는 경우 모든 필드를 최대 길이까지 입력해야 합니다.

이러한 방식으로 연결 값을 지정하면 설치된 **MQCONN** 자원 정의의 **INITQNAME** 값 및 설치된 **MQMONITOR** 자원 정의의 **QNAME** 값이 지정된 설정으로 대체됩니다. 원래 설정으로 되돌리려면 자원 정의를 재설치하십시오.

CICS 애플리케이션 프로그램에서 EXEC CICS SET MQCONN을 사용하여 CICS-MQ 연결 시작

CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 시작할 수 있습니다.

이 태스크 정보

SET MQCONN CONNECTED 명령은 **MQCONN** 속성 **INITQNAME**에 올바른 데이터가 있을 때 동적으로 작성된 **MQMONITOR** 자원 **DFHMQINI**를 포함해, 속성 **AUTOSTART(YES)** 및 **STATUS(ENABLED)**가 있는 설치된 모든 **MQMONITOR** 자원을 시작합니다.

프로시저

- 기본 연결 값을 사용하여 CICS 애플리케이션 프로그램에서 연결을 시작하려면 다음 명령을 실행하십시오.

```
EXEC CICS SET MQCONN CONNECTED
```

이 명령은 큐 관리자 이름에 대한 기본 연결 값을 사용하여 연결을 시작합니다. 단일 큐 관리자의 이름 또는 큐 공유 그룹의 이름일 수 있는 이 값은 CICS 리전에 대한 설치된 **MQCONN** 자원 정의에서 가져옵니다.

- CICS 애플리케이션 프로그램에서 연결을 시작할 때 고유 연결 값을 지정하려면 다음 예제와 같은 명령을 실행하십시오.

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

여기서 **qqqq**는 IBM MQ 큐 관리자 또는 큐 공유 그룹의 1 - 4자 이름입니다.

연결이 유실되면 RESYNC 또는 NORESYNC를 지정하여 재동기화 작동을 선택하십시오. 재동기화에 대한 자세한 정보는 [자동 재연결 및 재동기화](#)의 내용을 참조하십시오.

결과

요청된 큐 관리자가 활성화인 경우 CICS 및 IBM MQ가 연결되면 제어가 리턴됩니다. 요청된 큐 관리자가 활성화가 아니면 CICS가 RESP2=8인 NORMAL 응답을 리턴합니다. 이 응답은 CICS-MQ 어댑터가 연결 중 상태이고 요청된 큐 관리자가 활성화 되자마자 IBM MQ에 연결된다고 표시합니다.

다음에 수행할 작업

MQNAME에 대한 고유 값을 지정하는 경우, 설치된 MQCONN 자원 정의의 MQNAME 속성에 지정한 큐 관리자 이름 또는 큐 공유 그룹이 이 명령에 지정한 이름으로 대체됩니다. 원래 큐 관리자 또는 큐 공유 그룹으로 되돌리려면 MQNAME을 다시 설정하십시오.

CICS 애플리케이션 프로그램에서 DFHMQQCN에 링크하여 CICS-MQ 연결 시작

CICS 애플리케이션 프로그램에서 CICS-MQ 어댑터 연결 프로그램 DFHMQQCN(또는 프로그램의 이전 이름인 CSQCQCON)에 링크하여 IBM MQ에 대한 연결을 시작할 수 있습니다. 이 연결 시작 방법이 호환성을 위해 유지 보수되지만 **EXEC CICS SET MQCONN** 명령 사용은 CICS 애플리케이션 프로그램에서 연결을 시작하는 선호되는 방법입니다.

이 태스크 정보

CICS 애플리케이션 프로그램은 C, COBOL, PL/I 또는 어셈블러 언어로 기록될 수 있습니다.

프로시저

1. 프로그램은 사용할 연결 값을 지정하는 매개변수 목록을 전달해야 합니다.

매개변수 목록은 다음과 같습니다.

CKQC

4자의 트랜잭션 ID입니다. 'CKQC'여야 합니다.

DISPMODE

1바이트 필드입니다. 공백을 포함해야 합니다.

CONNREQ

10자의 필드입니다. 'START'를 포함해야 합니다.

DELIM1

1바이트 구분 기호 필드입니다. 공백을 포함해야 합니다.

MQDEF

이 연결이 CICS 리전에 대한 설치된 MQCONN 및 MQMONITOR 자원 정의에서 기본 연결 값을 사용할지 여부를 지정하는 1자의 필드입니다. CICS TS 4.1 전에 이 필드는 INITP로 알려져 있었습니다. 가능한 값은 다음과 같습니다.

Y

CICS 리전에 대한 설치된 MQCONN 및 MQMONITOR 자원 정의에서 기본 시작 큐 및 큐 관리자 이름의 값을 사용합니다.

N

기본값을 사용하지 않습니다. 'N'을 지정하면 CONNSSN 필드에 큐 관리자 이름을 제공해야 하고 CONNIQ 필드에 기본 시작 큐 이름을 선택적으로 지정할 수 있습니다. 이러한 값을 제공하면, 설치된 MQCONN 자원 정의의 INITQNAME 값 및 설치된 MQMONITOR 자원 정의의 QNAME 값이 지정된 설정으로 대체됩니다. 원래 설정으로 되돌리려면 자원 정의를 재설치하십시오.

..

'Y'에 해당합니다.

DELIM2

1바이트 구분 기호 필드입니다. 공백을 포함해야 합니다.

CONNSSN

대상 IBM MQ 큐 관리자의 z/OS 서브시스템 이름을 지정하는 데 사용되는 4자 필드입니다. 단일 큐 관리자 또는 큐 공유 그룹을 지정할 수 있습니다.

DELIM3

5바이트 구분 기호 필드입니다. 공백을 포함해야 합니다. 이전 릴리스는 추적점 대체를 지원했습니다. 이는 이제 무시됩니다.

CONNIQ

기본 시작 큐의 이름을 지정하는 48자 필드입니다.

2. 프로그램은 이와 같은 명령을 실행하여 어댑터 연결 프로그램 DFHMQQCN에 링크해야 합니다.

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

이 예제에서 매개변수 목록의 이름은 CONNPL입니다.

결과

DFHMQQCN의 출력 메시지가 시스템 콘솔에 표시됩니다.

CICS CEMT 트랜잭션을 통해 CICS-MQ 연결 시작

CICS CEMT 트랜잭션에서 **SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 시작할 수 있습니다.

이 태스크 정보

SET MQCONN CONNECTED 명령은 MQCONN 속성 INITQNAME에 올바른 데이터가 있을 때 동적으로 작성된 MQMONITOR 자원 DFHMQINI를 포함해, 속성 AUTOSTART(YES) 및 STATUS(ENABLED)가 있는 설치된 모든 MQMONITOR 자원을 시작합니다.

CEMT 트랜잭션을 시작하고 사용하는 방법에 대한 세부사항은 [CEMT - 마스터 터미널](#)의 내용을 참조하십시오.

프로시저

1. CICS 명령행에 명령 CEMT SET MQCONN을 입력하십시오.
연결의 상태 및 기본 연결 값이 표시됩니다. CICS 리전에 대한 설치된 MQCONN 자원 정의에서 값을 가져옵니다.
2. 옵션: 연결에 대한 IBM MQ 큐 관리자 또는 큐 공유 그룹의 이름을 변경하려는 경우 **Mqname** 필드의 값을 다른 이름으로 겹쳐쓰십시오.
이 값을 변경할 때, 설치된 MQCONN 자원 정의의 MQNAME 속성에 지정한 큐 관리자 이름 또는 큐 공유 그룹이 이 명령에 지정한 이름으로 대체됩니다. 원래 큐 관리자 또는 큐 공유 그룹으로 되돌리려면 값을 다시 변경하십시오.
3. 옵션: 재동기화 동작을 변경하려는 경우 **Resyncmember** 필드를 Resync, Noresync 또는 Groupresync로 겹쳐쓰십시오.
재동기화에 대한 자세한 정보는 [자동 재연결 및 재동기화](#)의 내용을 참조하십시오.
4. 연결을 시작하려면 값 Notconnected를 값 Connected로 겹쳐쓰고 Enter를 누르십시오.
CICS가 지정된 새 설정을 사용하여 연결을 시작합니다.

CICSplex SM 웹 사용자 인터페이스를 통해 CICS-MQ 연결 시작

CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ 연결** 보기에서 IBM MQ에 대한 연결을 시작할 수 있습니다.

이 태스크 정보

CICS Explorer에서 MQ 연결(MQCON) 보기는 WUI의 이 보기에 해당하는 기능을 제공합니다.

프로시저

1. CICSplex SM WUI 기본 메뉴에서 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 연결**을 선택하십시오.
이 보기 세트의 정보에 대한 전체 설명 및 목록은 [WebSphere MQ 연결 - MQCON](#)의 내용을 참조하십시오.
2. 연결을 시작할 CICS 시스템의 이름을 클릭하십시오.

CICSplex SM은 **WebSphere MQ 연결 - 일반 정보** 보기를 표시합니다.

보기는 연결의 현재 상태 및 기본 연결 값을 표시하고, 이 값은 CICS 시스템에 대한 설치된 MQCONN 자원 정의에서 가져옵니다.

3. 옵션: 연결을 위한 IBM MQ 큐 관리자 또는 큐 공유 그룹의 이름을 변경하려면 **MQ 큐 관리자 또는 QSG 이름** 필드에 다른 이름을 입력하십시오.
이 값을 변경할 때, 설치된 MQCONN 자원 정의의 MQNAME 속성에 지정한 큐 관리자 이름 또는 큐 공유 그룹이 이 명령에 지정한 이름으로 대체됩니다. 원래 큐 관리자 또는 큐 공유 그룹으로 되돌리려면 값을 다시 변경하십시오.
4. 옵션: 재동기화 동작을 변경하려는 경우, **재동기화 멤버** 필드에서 대체 옵션을 선택하십시오.
재동기화에 대한 자세한 정보는 [자동 재연결 및 재동기화의 내용](#)을 참조하십시오.
5. 연결에 대한 설정을 변경한 경우 **변경사항 적용** 단추를 클릭하십시오.
6. 연결을 시작하려면 **CICS-MQ 연결 시작** 단추를 클릭하십시오.
CICS 시스템은 지정된 설정을 사용하여 연결을 시작합니다.

CICS-MQ 연결 중지

여러 방법으로 CICS-MQ 연결을 중지할 수 있습니다. 연결을 중지하면 단계적 종료 또는 강제 종료 중에서 수행할 종료를 선택할 수 있습니다.

이 태스크 정보

단계적 종료를 사용하면 각 CICS 트랜잭션이 인터페이스가 닫히기 전에 종료될 수 있습니다. 강제 종료는 큐 관리자에 연결된 CICS 트랜잭션을 비정상적으로 종료합니다. CICS-MQ 어댑터가 단계적 종료 및 강제 종료에 대해 수행하는 조치에 대한 설명은 [CICS-WebSphere MQ 연결이 종료되면 발생하는 사항](#)의 내용을 참조하십시오.

CICS 리전에 대한 변경된 MQCONN 자원 정의를 설치하기 전에 연결을 중지해야 합니다.

강제 종료 다음에 연결을 다시 시작하는 경우 인다우트 작업 단위가 있을 수 있습니다. 인다우트 작업 단위에 대한 정보는 [자동 재연결 및 재동기화의 내용](#)을 참조하십시오. 필요한 경우 인다우트 작업 단위를 수동으로 해결하는 데 대한 지시사항은 [CICS-WebSphere MQ 어댑터가 다시 시작하면 발생하는 사항](#)의 내용을 참조하십시오.

프로시저

다음 방법 중 하나를 사용하여 CICS-MQ 연결을 중지할 수 있습니다.

- [CICS-MQ 어댑터 제어판 사용](#)
CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어판을 사용하여 CICS-MQ 연결을 중지할 수 있습니다.
- [CICS 명령행에서 CKQC STOP 실행](#)
명령행에서 **CKQC STOP** 명령을 실행하여 CICS-MQ 연결을 중지할 수 있습니다.
- [EXEC CICS SET MQCONN을 실행하는 CICS 애플리케이션 프로그램 사용](#)
CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 중지할 수 있습니다.
- [DFHMQDSC에 링크된 CICS 애플리케이션 프로그램 사용](#)
CICS 애플리케이션 프로그램에서 CICS-MQ 어댑터 프로그램 DFHMQDSC(또는 프로그램의 이전 이름인 CSQCDSC)에 링크하여 IBM MQ에 대한 연결을 중지할 수 있습니다. 이 연결 중지 방법이 호환성을 위해 유지 보수되지만 **EXEC CICS SET MQCONN** 명령 사용은 CICS 애플리케이션 프로그램에서 연결을 중지하는 선호되는 방법입니다.
- [CICS CEMT 트랜잭션 사용](#)
CICS CEMT 트랜잭션에서 **SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 중지할 수 있습니다.
- [CICSplex SM 웹 사용자 인터페이스 사용](#)
CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ 연결** 보기에서 IBM MQ에 대한 연결을 중지할 수 있습니다.

- CICS Explorer 사용

CICS Explorer에서 CICS-MQ 연결을 중지할 수 있습니다. CICS Explorer는 CEMT 트랜잭션 및 CICSplex SM 웹 사용자 인터페이스에 해당하는 기능을 제공합니다.

결과

MQ 모니터를 사용하고 있는 경우 MQ 연결이 중지되면 MQ 모니터가 자동으로 중지됩니다.

CICS-MQ 어댑터 제어판에서 CICS-MQ 연결 중지

CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어판을 사용하여 CICS-MQ 연결을 중지할 수 있습니다.

프로시저

1. CICS-MQ 어댑터 초기 제어판의 메뉴 표시줄에서 **연결**을 선택하십시오.
2. 메뉴에서 **중지** 조치를 선택하십시오.
3. **연결 중지** 보조 매개변수 창을 사용하여 필요한 시스템 종료 유형을 선택하십시오(단계적 또는 강제).

결과

연결 중지와 연관된 메시지는 시스템 콘솔에 표시됩니다.

예

Connection	CKTI	Task
Select an action.	CS Adapter Control -- Initial panel	
2 1. Start...		sing Tab key. Then press Enter
2. Stop...		
3. Modify...		
4. Display		
F1=Help F12=Cancel		
		<div> <div>Stop Connection</div> <div>Select stop type. Then press Enter</div> <div> 1 1. Quiesce 2. Force </div> <div>F1=Help F12=Cancel</div> </div>

F1=Help F3=Exit

그림 5. CKQC 초기 패널에서 연결 중지

CICS 명령행에서 CICS-MQ 연결 중지

명령행에서 **CKQC STOP** 명령을 실행하여 CICS-MQ 연결을 중지할 수 있습니다.

프로시저

- 단계적 종료를 시작하려면 명령 CKQC STOP을 실행하십시오.
연결은 마지막 태스크가 작업을 완료한 후에만 종료됩니다.
- 강제 종료를 시작하려면 명령 CKQC STOP FORCE를 실행하십시오.
인플라이트 태스크의 상태에 관계없이 연결이 즉시 종료됩니다.

CICS 애플리케이션 프로그램에서 EXEC CICS SET MQCONN을 사용하여 CICS-MQ 연결 중지

CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 중지할 수 있습니다.

프로시저

- 연결이 중지될 때까지 제어가 애플리케이션으로 리턴되지 않는 단계적 종료를 수행하려면 다음 명령을 실행하십시오.

```
EXEC CICS SET MQCONN NOTCONNECTED
```

요청을 동기식으로 설정하는 **BUSY(WAIT)** 옵션은 기본값입니다.

- 연결이 중지되기 전에 제어가 애플리케이션으로 리턴되는 단계적 종료를 수행하려면(비동기 요청) 다음 명령을 실행하십시오.

```
EXEC CICS SET MQCONN NOTCONNECTED NOWAIT
```

- 강제 종료를 수행하려면 다음 명령을 실행하십시오.

```
EXEC CICS SET MQCONN NOTCONNECTED FORCE
```

강제 종료를 사용하면 연결이 중지될 때까지 제어가 애플리케이션으로 리턴되지 않습니다.

CICS 애플리케이션 프로그램에서 DFHMQDSC에 링크하여 CICS-MQ 연결 중지

CICS 애플리케이션 프로그램에서 CICS-MQ 어댑터 프로그램 DFHMQDSC(또는 프로그램의 이전 이름인 CSQCDSC)에 링크하여 IBM MQ에 대한 연결을 중지할 수 있습니다. 이 연결 중지 방법이 호환성을 위해 유지보수되지만 **EXEC CICS SET MQCONN** 명령 사용은 CICS 애플리케이션 프로그램에서 연결을 중지하는 선호되는 방법입니다.

이 태스크 정보

EXEC CICS LINK 명령을 실행하여 DFHMQDSC에 링크하는 경우, 프로그램에 터미널 연관 태스크가 필요합니다.

프로시저

- 단계적 종료를 사용하여 CICS 애플리케이션 프로그램에서 연결을 중지하려면 다음 명령을 실행하십시오.

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

QUIESCE 매개변수는 기본값입니다.

- 강제 종료를 사용하여 CICS 애플리케이션 프로그램에서 연결을 중지하려면 다음 명령을 실행하십시오.

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

결과

DFHMQDSC의 출력 메시지가 시스템 콘솔에 표시됩니다.

CICS CEMT 트랜잭션을 통해 CICS-MQ 연결 중지

CICS CEMT 트랜잭션에서 **SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 중지할 수 있습니다.

이 태스크 정보

CEMT 트랜잭션을 시작하고 사용하는 방법에 대한 세부사항은 [CEMT - 마스터 터미널](#)을 참조하십시오.

프로시저

1. CICS 명령행에 명령 CEMT SET MQCONN을 입력하십시오.
연결의 현재 상태 및 기본 연결 값을 나열하는 표시를 확보하며, 이 값은 CICS 리전에 대한 설치된 MQCONN 자원 정의에서 가져옵니다.
2. 연결을 중지하려면 다음과 같이 값 Connected를 겹쳐써서 시스템 종료의 유형을 선택하십시오.
 - a) 단계적 종료를 원하는 경우 Notconnected를 입력하십시오.
 - b) 강제 종료를 원하는 경우 Forcenotcon을 입력하십시오.
3. Enter를 누르십시오.
CICS가 연결을 중지합니다.

CICSplex SM 웹 사용자 인터페이스를 통해 CICS-MQ 연결 중지

CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ 연결** 보기에서 IBM MQ에 대한 연결을 중지할 수 있습니다.

이 태스크 정보

CICS Explorer에서 MQ 연결(MQCON) 보기는 WUI의 이 보기에 해당하는 기능을 제공합니다.

프로시저

1. CICSplex SM WUI 기본 메뉴에서 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 연결**을 선택하십시오.
이 보기 세트의 정보에 대한 전체 설명 및 목록은 [WebSphere MQ 연결 - MQCON](#)의 내용을 참조하십시오.
2. 연결을 중지할 CICS 시스템의 이름을 클릭하십시오.
CICSplex SM은 **WebSphere MQ 연결 - 일반 정보** 보기를 표시합니다.
3. 연결을 중지하려면 **CICS-MQ 연결 중지** 단추를 클릭하십시오.
CICSplex SM은 **CICS-MQ 연결 중지** 보조 패널을 표시합니다.
4. 다음과 같이 **사용 중인 값** 상자를 사용하여 원하는 연결 끊기 유형을 선택하십시오.
 - 연결이 중지될 때까지 CICSplex SM이 일시정지되는 단계적 종료를 수행하려면(동기 요청) **WAIT**를 선택하십시오.
 - CICSplex SM이 시스템 종료 요청을 발행한 다음 연결 중지 여부에 관계없이 작업을 계속하는 단계적 종료를 수행하려면(비동기 요청) **NOWAIT**를 선택하십시오.
 - 강제 종료를 수행하려면 **FORCE**를 선택하십시오.
5. '**연결 끊기**' 수행 여부에서 예 단추를 클릭하십시오.
선택된 CICS 시스템이 연결을 중지합니다.

CICS-MQ 연결 정보 표시

현재 작업 중인 환경 및 필요한 정보에 따라 CICS-MQ 연결에 대한 자세한 통계 또는 기본 정보를 얻을 수 있습니다.

CEMT 또는 **EXEC CICS INQUIRE** 명령, CICS-MQ 어댑터 제어판, CKQC DISPLAY 명령 또는 CICSplex SM을 사용하여 연결 정보를 표시할 수 있습니다. 각 방법은 이 환경에 대해 선택된 일련의 정보를 생성합니다. 다음 표에는 각 방법을 사용하여 사용 가능한 정보 항목이 표시됩니다.

표 7. CICS-MQ 연결 정보 표시				
정보	INQUIRE MQCONN(CEMT를 사용하거나 애플리케이션에서)	CICS-MQ 어댑터 제어판	CKQC DISPLAY(명령행을 통해 또는 애플리케이션에서)	CICSplex SM 보기
연결 상태(예: 연결됨)	예	예	예	예
연결에 지정된 큐 관리자 또는 큐 공유 그룹의 이름	예	예	예	예
실제로 연결된 큐 관리자의 이름	예	예	예	예
기본 시작 큐의 이름	INQUIRE MQMONITOR 사용	예	아니오	예
연결을 사용하는 인플라이트 태스크의 수	예	예	예	예
연결을 사용하는 트리거 모니터 (CKTI) 태스크의 수	예	예	아니오	예
연결을 사용하는 태스크의 세부사항(예: 트랜잭션 ID 및 태스크 번호)	아니오	예	아니오	아니오
CICS-MQ API 교차 엑시트의 상태	아니오	예	아니오	아니오
재동기화 동작	예	아니오	아니오	예
IBM MQ 릴리스 번호	예	아니오	아니오	예
MQCONN 자원 정의에 대한 자원 서명	예	아니오	아니오	예
설치된 MQCONN 자원 정의의 수	INQUIRE SYSTEM 사용	아니오	아니오	예
연결 및 연결 끊기 시간	EXTRACT STATISTICS 사용(EXEC CICS에만 해당)	아니오	아니오	예
MQI 호출의 수와 같은 자세한 통계	EXTRACT STATISTICS 사용(EXEC CICS에만 해당)	예	아니오	예

CICS-MQ 연결 통계 및 설정 표시

CICS-MQ 연결의 상태 및 연결에 대한 큐 관리자 또는 큐 공유 그룹의 이름과 같은 기본 정보를 찾기 위해 CEMT 또는 **EXEC CICS INQUIRE** 명령, CICS-MQ 어댑터 제어판, **CKQC DISPLAY** 명령 또는 CICSplex SM을 사용할 수 있습니다.

이 태스크 정보

필요한 정보 및 현재 환경에 따라 CICS-MQ 연결에 대한 기본 정보를 확보하기 위해 가장 적합한 방법을 선택하십시오. 각각의 방법으로 확보할 수 있는 정보 목록은 [39 페이지의 『CICS-MQ 연결 정보 표시』](#)의 내용을 참조하십시오.

프로시저

- CICS 명령행에서 연결 상태 및 설정을 표시하려면 다음 명령 중 하나를 입력하십시오.

- CEMT INQUIRE MQCONN**

이 명령은 CEMT 트랜잭션을 시작하고 화면에 연결 정보를 표시합니다. 기본 시작 큐의 이름을 찾으려면 같은 방법으로 **CEMT INQUIRE MQMONITOR** 명령을 사용하고, 설치된 MQCONN 자원 정의의 이름을 찾으려면 **CEMT INQUIRE SYSTEM** 명령을 사용하십시오.

- CKQC DISPLAY**

이 명령은 연결에 대한 가장 필요한 정보만 생성합니다. 이 명령을 실행하면 CICS가 다음과 같이 메시지 DFHM00453I를 연결 정보와 함께 트랜지언트 데이터 큐 CKQQ에 씁니다.

```
DFHM00453I 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 애플리케이션 프로그램에서 연결 상태와 설정을 확보하려면 다음 방법 중 하나를 선택하십시오.
- EXEC CICS INQUIRE MQCONN** 명령을 실행하십시오. 이 명령은 정보를 애플리케이션 프로그램으로 리턴합니다. 기본 시작 큐의 이름을 찾으려면 같은 방법으로 **INQUIRE MQMONITOR** 명령을 사용하고, 설치된 MQCONN 자원 정의의 이름을 찾으려면 **INQUIRE SYSTEM** 명령을 사용하십시오.
- 다음 예제와 같이 **EXEC CICS LINK** 명령을 실행하여 CICS-MQ 어댑터 프로그램 DFHM0DSL(또는 호환성을 위해 허용된 CSQCDSPL)에 링크하고 **CKQC DISPLAY** 명령을 실행하십시오.

```
EXEC CICS LINK PROGRAM('DFHM0DSL') INPUTMSG('CKQC DISPLAY')
```

프로그램이 PLT 시간에 실행되는 경우에만 INPUTMSG 대신 COMMAREA 옵션을 사용할 수 있습니다. CICS는 명령행에서 **CKQC DISPLAY** 명령을 실행할 때와 같이 메시지 DFHM00453I를 트랜지언트 데이터 큐 CKQQ에 씁니다.

- CICS-MQ 어댑터 제어판에서 연결 상태 및 설정을 보려면 초기 패널의 메뉴 표시줄에서 **연결**을 선택한 다음 메뉴에서 **표시** 조치를 선택하십시오.
여기에 표시된 표시 연결 패널에는 연결 정보가 포함됩니다. 이 패널의 특정 필드에 대한 설명을 보려면 기능 키 F1을 누르고, 패널을 새로 고치려면 Enter를 누르십시오.

```
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 SM의 연결 정보를 표시하려면 기본 메뉴에서 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 연결**을 선택하십시오.
이 보기 세트의 정보에 대한 설명 및 목록은 [WebSphere MQ 연결 - MQCON](#)의 내용을 참조하십시오.

기본 시작 큐의 이름을 보려면 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 시작 큐를 선택하십시오.**

CICS-MQ 연결 및 연결 끊기 시간 표시

CICSplex SM 또는 CICS 통계를 사용하여 CICS-MQ 연결에 대한 연결 및 연결 끊기 시간을 표시할 수 있습니다. CKQC 트랜잭션은 연결 및 연결 끊기 시간을 제공하지 않습니다.

프로시저

1. CICS에서 연결 및 연결 끊기 시간을 표시하려면 CICS 애플리케이션 프로그램에서 명령 **EXEC CICS EXTRACT STATISTICS MQCONN**을 실행하십시오.

CICS는 통계를 애플리케이션으로 리턴합니다.

2. CICSplex SM의 연결 및 연결 끊기 시간을 표시하려면 기본 메뉴에서 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 연결**을 선택하십시오.

이 보기 세트의 정보에 대한 설명 및 목록은 [WebSphere MQ 연결 - MQCON](#)의 내용을 참조하십시오.

CICS-MQ 연결 통계 및 호출 유형 표시

연결 및 인다우트 작업 단위의 수와 상태를 사용하여 작성된 각 MQI 호출 유형의 수와 같은 CICS-MQ 연결에 대한 자세한 통계를 표시하기 위해 CICS 통계, CICS-MQ 어댑터 제어판 또는 CICSplex SM을 사용합니다.

프로시저

1. CICS에서 연결 통계를 표시하려면 CICS 애플리케이션 프로그램에서 명령 **EXEC CICS EXTRACT STATISTICS MQCONN**을 실행하십시오.

CICS는 통계를 애플리케이션으로 리턴합니다.

이러한 통계의 목록 및 설명은 [CICS-MQ 연결 통계](#)의 내용을 참조하십시오.

2. CICS-MQ 어댑터 제어판에 연결 통계를 표시하려면 초기 패널의 메뉴 표시줄에서 **연결**을 선택한 다음 메뉴에서 **표시** 조치를 선택하십시오.

여기에 표시된 표시 연결 패널에는 연결 통계가 포함됩니다.

```

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
  
```

인플라이트 태스크, CKTI의 인스턴스 및 API 호출에 대해 표시된 통계는 통계가 마지막으로 재설정된 이후 현재 연결의 총 수입입니다. 표제 APIs and flows analysis, Syncpoint 및 Recovery 아래에 표시된 통계는 어댑터에서 생성된 통계입니다. 이 패널의 특정 필드에 대한 설명을 보려면 기능 키 F1을 누르고, 패널을 새로 고치려면 Enter를 누르십시오.

3. CICSplex SM의 연결 통계를 표시하려면 기본 메뉴에서 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 연결 통계**를 선택하십시오.

이 보기 세트의 정보에 대한 설명 및 목록은 [WebSphere MQ 연결 통계 - MQCONN](#)의 내용을 참조하십시오.

CICS-MQ 연결을 사용하는 태스크 표시

CICS-MQ 어댑터 제어판을 사용하여 트랜잭션 ID 및 태스크 번호를 포함해 MQI 호출을 사용 중인 CICS 태스크에 대한 자세한 정보를 표시할 수 있습니다.

이 태스크 정보

연결을 사용하는 태스크의 수만 보려는 경우 [CICS-MQ 연결 상태 및 설정 표시](#)에 설명된 방법 중 하나를 사용할 수 있습니다. 또한 **INQUIRE MQCONN** 명령 및 CICS-MQ 어댑터 **연결 표시** 패널은 연결을 사용하는 트리거 모니터(CKTI) 태스크의 수를 지정합니다.

CICS-MQ 어댑터 **태스크 표시** 패널은 다음과 같이 연결을 사용하는 각 CICS 태스크에 대한 세부사항을 제공합니다.

- 트랜잭션 ID(이름)
- 사용자 ID
- CICS 태스크 번호
- 태스크 상태
- 스레드 상태
- 이 태스크에서 발행된 API 호출의 총 수
- 이 태스크에 대한 자원 보안 검사가 활성화인지 여부
- 이 태스크가 현재 CICS-MQ API 교차 엑시트에 있는지 여부
- 이 태스크에서 실행되는 최신 API 호출
- IBM MQ에서 사용되는 스레드 ID

44 페이지의 [그림 6](#)에는 이 정보의 레이아웃이 표시됩니다. 각 속성에 대한 자세한 정보는 PF1을 눌러 액세스되는 CKQC 도움말 패널을 참조하십시오.

프로시저

CICS-MQ 어댑터 제어판을 사용하여 태스크 정보를 표시하려면 다음을 수행하십시오.

- a) CKQC를 입력하고 Enter를 눌러 CICS-MQ 어댑터 제어판에 액세스하십시오.
- b) 메뉴 표시줄에서 **태스크**를 선택하십시오.
- c) 현재 활성화된 모든 태스크에 대한 정보를 확보하려면 메뉴에서 **모든 태스크 나열** 조치를 선택하십시오.
- d) 표시할 첫 번째 태스크의 시작 번호를 지정하려면 메뉴에서 **태스크에서 나열** 조치를 선택하십시오.

예

CKQCM3

Display Task panel

Read task status information. Then press F12 to cancel.

Tasks1 to3 of3

Tran Id	User Id	Task Num	Task Status	Thread Status	Total APIs	Res Sec	API Exit	Last MQ call	Thread 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=HelpF7=BackwardF8=ForwardF12=CancelEnter=Refresh

그림 6. CICS-MQ 어댑터 표시 태스크 패널

CICS MQ 모니터 시작

올바르게 설정되면 MQ 연결이 설정될 때 CICS MQ 모니터가 자동으로 시작될 수 있습니다. 또한 여러 방법으로 MQ 모니터를 수동으로 시작할 수 있습니다.

시작하기 전에

- MQMONITOR 자원은 CICS 리전에 설치하여 사용해야 합니다. 세부사항은 [CICS-MQ 어댑터 설정의 내용](#)을 참조하십시오.
- 보안 검사가 활성화된 경우(즉, **SEC** 시스템 초기화 매개변수가 YES로 설정된 경우), MQ 모니터 상태를 시작됨으로 설정하려는 트랜잭션과 연관된 사용자 ID가 **MONUSERID**에 정의된 사용자 ID의 대리이고 **MONUSERID**와 연관된 트랜잭션을 시작할 수 있는 권한이 부여되었는지 확인하십시오. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTIUSR 사용자 ID(지정된 경우)입니다.
- z/OS Workload Manager(WLM) 상태 서비스가 활성화된 경우(WLMHEALTH 참조), CICS 리전의 z/OS WLM **HEALTH** 값을 0에서 100%까지 증분할 때마다 AUTOSTART(YES)로 정의된 중지된 모든 MQ 모니터를 시작하려고 시도합니다. 중지된 MQ 모니터가 시작될 때 제어하려면 이 MQMONITOR 자원에 AUTOSTART(NO)를 설정해야 합니다. 자세한 정보는 [z/OS Workload Manager Health 서비스가 MQMONITOR에 미치는 영향 및 경보 모니터\(CKAM\)의 내용](#)을 참조하십시오.

프로시저

- MQ 연결이 설정될 때 CICS MQ 모니터를 자동으로 시작하려면 속성 AUTOSTART(YES) 및 STATUS(ENABLED)로 MQMONITOR 자원을 정의해야 합니다.
CICS 및 MQ 간의 연결이 설정될 때, MQCONN 자원을 CONNECTED로 설정하는 태스크와 연관된 사용자 ID에 연관 트랜잭션을 시작할 충분한 권한이 있으면 CICS는 AUTOSTART(YES) 및 STATUS(ENABLED) 속성으로 설치된 MQ 모니터를 시작합니다.
- CICS-MQ 모니터를 수동으로 시작하기 위해 다음 방법 중 하나를 사용할 수 있습니다.
 - **EXEC CICS SET MQMONITOR**를 실행하는 CICS 애플리케이션 프로그램 사용
CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQMONITOR** 명령을 실행하여 MQ 모니터를 시작할 수 있습니다.

- **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START)**을 실행하는 CICSplex SM 애플리케이션 프로그램 사용
CICSplex SM 애플리케이션 프로그램에서 **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START)** 명령을 실행하여 MQ 모니터를 시작할 수 있습니다.
- CICS CEMT 트랜잭션 사용
CICS CEMT 트랜잭션에서 **SET MQMONITOR** 명령을 실행하여 MQ 모니터를 시작할 수 있습니다.
- CICSplex SM 웹 사용자 인터페이스 사용
CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ 모니터** 보기에서 MQ 모니터를 시작할 수 있습니다.
- CICS Explorer 사용
CICS Explorer에서 CICS-MQ 모니터를 시작할 수 있습니다. CICS Explorer는 CEMT 트랜잭션 및 CICSplex SM 웹 사용자 인터페이스에 해당하는 기능을 제공합니다.
- MQMONITOR 자원에 정의된 시작 큐에 대한 CKTI의 인스턴스를 시작하기 위해 CKQC 트랜잭션 및 STARTCKTI 명령을 사용하지 마십시오. CKQC는 MQMONITOR를 인식하지 않으며 MQMONITOR는 CKQC에서 관리되는 CKTI 트랜잭션을 인식하지 않습니다. CKQC STARTCKTI 명령 사용을 MQMONITOR를 통해 관리되지 않는 CKTI 트랜잭션으로 제한하십시오. MQMONITOR의 경우 위에 설명된 MONITOR SPI 명령을 사용하십시오.

CICS 애플리케이션 프로그램에서 EXEC CICS SET MQMONITOR를 사용하여 CICS MQ 모니터 시작

CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQMONITOR** 명령을 실행하여 MQ 모니터를 시작할 수 있습니다.

시작하기 전에

- MQMONITOR 자원은 CICS 리전에 설치하여 사용해야 합니다. 세부사항은 [CICS-MQ 어댑터 설정의 내용](#)을 참조하십시오.
- 보안 검사가 활성화인 경우(즉, **SEC** 시스템 초기화 매개변수가 YES로 설정된 경우), MQ 모니터 상태를 시작됨으로 설정하려는 트랜잭션과 연관된 사용자 ID가 **MONUSERID**에 정의된 사용자 ID의 대리이고 **MONUSERID**와 연관된 트랜잭션을 시작할 수 있는 권한이 부여되었는지 확인하십시오. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTIUSR 사용자 ID(지정된 경우)입니다.
- z/OS Workload Manager(WLM) 상태 서비스가 활성화인 경우(WLMHEALTH 참조), CICS 리전의 z/OS WLM **HEALTH** 값을 0에서 100%까지 증분할 때마다 AUTOSTART(YES)로 정의된 중지된 모든 MQ 모니터를 시작하려고 시도합니다. 중지된 MQ 모니터가 시작될 때 제어하려면 이 MQMONITOR 자원에 AUTOSTART(NO)를 설정해야 합니다. 자세한 정보는 [z/OS Workload Manager Health 서비스가 MQMONITOR에 미치는 영향 및 경보 모니터\(CKAM\)의 내용](#)을 참조하십시오.

프로시저

MQ 모니터를 시작하려면 CICS 애플리케이션 프로그램에서 다음 명령을 실행하십시오.

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

여기서 *name*은 시작할 MQMONITOR 자원 정의의 이름입니다.

CICSplex SM 애플리케이션 프로그램에서 EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START)를 사용하여 CICS MQ 모니터 시작

CICSplex SM 애플리케이션 프로그램에서 **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(START)** 명령을 실행하여 MQ 모니터를 시작할 수 있습니다.

시작하기 전에

- MQMONITOR 자원은 CICS 리전에 설치하여 사용해야 합니다. 세부사항은 [CICS-MQ 어댑터 설정의 내용](#)을 참조하십시오.
- 보안 검사가 활성화된 경우(즉, **SEC** 시스템 초기화 매개변수가 YES로 설정된 경우), MQ 모니터 상태를 시작됨으로 설정하려는 트랜잭션과 연관된 사용자 ID가 **MONUSERID**에 정의된 사용자 ID의 대리이고 **MONUSERID**와 연관된 트랜잭션을 시작할 수 있는 권한이 부여되었는지 확인하십시오. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTPUIUSR 사용자 ID(지정된 경우)입니다.
- z/OS Workload Manager(WLM) 상태 서비스가 활성화된 경우([WLMHEALTH](#) 참조), CICS 리전의 z/OS WLM **HEALTH** 값이 0에서 100%까지 증분할 때마다 AUTOSTART(YES)로 정의된 중지된 모든 MQ 모니터를 시작하려고 시도합니다. 중지된 MQ 모니터가 시작될 때 제어하려면 이 MQMONITOR 자원에 AUTOSTART(NO)를 설정해야 합니다. 자세한 정보는 [z/OS Workload Manager Health 서비스가 MQMONITOR에 미치는 영향 및 경보 모니터\(CKAM\)](#)의 내용을 참조하십시오.

프로시저

MQ 모니터를 시작하려면 CICSplex SM 애플리케이션 프로그램에서 다음 명령을 실행하십시오.

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

여기서 **MQMON**은 시작할 MQMONITOR 자원 정의의 이름입니다.

CICS CEMT 트랜잭션을 통해 CICS MQ 모니터 시작

CICS CEMT 트랜잭션에서 **SET MQMONITOR** 명령을 실행하여 MQ 모니터를 시작할 수 있습니다.

시작하기 전에

- MQMONITOR 자원은 CICS 리전에 설치하여 사용해야 합니다. 세부사항은 [CICS-MQ 어댑터 설정의 내용](#)을 참조하십시오.
- 보안 검사가 활성화된 경우(즉, **SEC** 시스템 초기화 매개변수가 YES로 설정된 경우), MQ 모니터 상태를 시작됨으로 설정하려는 트랜잭션과 연관된 사용자 ID가 **MONUSERID**에 정의된 사용자 ID의 대리이고 **MONUSERID**와 연관된 트랜잭션을 시작할 수 있는 권한이 부여되었는지 확인하십시오. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTPUIUSR 사용자 ID(지정된 경우)입니다.
- z/OS Workload Manager(WLM) 상태 서비스가 활성화된 경우([WLMHEALTH](#) 참조), CICS 리전의 z/OS WLM **HEALTH** 값이 0에서 100%까지 증분할 때마다 AUTOSTART(YES)로 정의된 중지된 모든 MQ 모니터를 시작하려고 시도합니다. 중지된 MQ 모니터가 시작될 때 제어하려면 이 MQMONITOR 자원에 AUTOSTART(NO)를 설정해야 합니다. 자세한 정보는 [z/OS Workload Manager Health 서비스가 MQMONITOR에 미치는 영향 및 경보 모니터\(CKAM\)](#)의 내용을 참조하십시오.

이 태스크 정보

CEMT 트랜잭션을 시작하고 사용하는 방법에 대한 세부사항은 [CEMT - 마스터 터미널의 내용](#)을 참조하십시오.

프로시저

1. CICS 명령행에 다음 명령을 입력하십시오.

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

여기서 **name**은 시작할 MQMONITOR 자원의 이름입니다.

지정된 MQMONITOR 자원 정의의 AUTOSTART, ENABLESTATUS 및 MONSTATUS 값이 표시됩니다.

2. MQMONITOR 자원 정의를 사용할 수 없는 경우 ENABLESTATUS 필드의 값 DISABLED를 값 ENABLED로 겹쳐써서 CICS 리전에 사용할 자원을 사용으로 설정하십시오.
3. MONSTATUS 필드의 값 STOPPED를 값 STARTED로 겹쳐써서 MQ 모니터를 시작하십시오.

4. AUTOSTART를 사용할 수 없는 경우, AUTOSTART 필드의 값 NOAUTOSTART를 값 AUTOSTART로 겹쳐써서 MQMONITOR에 대한 AUTOSTART를 사용으로 설정할 수 있습니다.

CICSplex SM 웹 사용자 인터페이스를 통해 CICS MQ 모니터 시작

CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ** 모니터 보기에서 MQ 모니터를 시작할 수 있습니다.

시작하기 전에

- MQMONITOR 자원은 CICS 리전에 설치하여 사용해야 합니다. 세부사항은 [CICS-MQ 어댑터 설정의 내용](#)을 참조하십시오.
- 보안 검사가 활성화된 경우(즉, **SEC** 시스템 초기화 매개변수가 YES로 설정된 경우), MQ 모니터 상태를 시작됨으로 설정하려는 트랜잭션과 연관된 사용자 ID가 **MONUSERID**에 정의된 사용자 ID의 대리이고 **MONUSERID**와 연관된 트랜잭션을 시작할 수 있는 권한이 부여되었는지 확인하십시오. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTIUSR 사용자 ID(지정된 경우)입니다.
- z/OS Workload Manager(WLM) 상태 서비스가 활성화된 경우(WLMHEALTH 참조), CICS 리전의 z/OS WLM **HEALTH** 값을 0에서 100%까지 증분할 때마다 AUTOSTART(YES)로 정의된 중지된 모든 MQ 모니터를 시작하려고 시도합니다. 중지된 MQ 모니터가 시작될 때 제어하려면 이 MQMONITOR 자원에 AUTOSTART(NO)를 설정해야 합니다. 자세한 정보는 [z/OS Workload Manager Health 서비스가 MQMONITOR에 미치는 영향 및 경보 모니터\(CKAM\)의 내용](#)을 참조하십시오.

이 태스크 정보

CICS Explorer에서 MQ 모니터(MQMON) 보기는 WUI의 이 보기에 해당하는 기능을 제공합니다.

프로시저

1. CICSplex SM WUI 기본 메뉴에서 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 모니터**를 선택하십시오.
이 보기 세트의 정보에 대한 전체 설명 및 목록은 [WebSphere MQ 모니터 - MQMON](#)의 내용을 참조하십시오.
2. 시작할 WebSphere MQ 모니터에 대한 **레코드** 상자를 선택하십시오.
3. 시작 단추를 클릭하십시오.

CICS MQ 모니터 중지

MQ 연결이 중지될 때 MQ 모니터가 자동으로 중지됩니다. 또한 여러 방법으로 MQ 모니터를 수동으로 중지할 수 있습니다.

z/OS Workload Manager(WLM) 상태 서비스가 활성화된 경우(WLMHEALTH 참조), CICS 리전의 z/OS WLM **HEALTH** 값을 0에서 100%까지 증분할 때마다 AUTOSTART(YES)로 정의된 중지된 모든 MQ 모니터를 시작하려고 시도합니다. 따라서 이 기간 동안 특정 MQ 모니터를 중지된 상태로 유지하려면 이러한 MQMONITOR 자원에 대한 AUTOSTART(NO)를 설정해야 합니다. 자세한 정보는 [z/OS Workload Manager Health 서비스가 MQMONITOR에 미치는 영향 및 경보 모니터\(CKAM\)의 내용](#)을 참조하십시오.

프로시저

다음 방법 중 하나를 사용하여 MQ 모니터를 수동으로 중지할 수 있습니다.

- **EXEC CICS SET MQMONITOR**를 실행하는 CICS 애플리케이션 프로그램 사용
CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQMONITOR** 명령을 실행하여 MQ 모니터를 중지할 수 있습니다.
- **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP)**을 실행하는 CICSplex SM 애플리케이션 프로그램 사용
CICSplex SM 애플리케이션 프로그램에서 **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP)** 명령을 실행하여 MQ 모니터를 중지할 수 있습니다.

- [CICS CEMT 트랜잭션 사용](#)

CICS CEMT 트랜잭션에서 **SET MQMONITOR** 명령을 실행하여 MQ 모니터를 중지할 수 있습니다.

- [CICSplex SM 웹 사용자 인터페이스 사용](#)

CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ 모니터** 보기에서 MQ 모니터를 중지할 수 있습니다.

- CICS Explorer 사용

CICS Explorer에서 CICS-MQ 모니터를 중지할 수 있습니다. CICS Explorer는 CEMT 트랜잭션 및 CICSplex SM 웹 사용자 인터페이스에 해당하는 기능을 제공합니다.

CICS 애플리케이션 프로그램에서 EXEC CICS SET MQMONITOR를 사용하여 CICS MQ 모니터 중지

CICS 애플리케이션 프로그램에서 **EXEC CICS SET MQMONITOR** 명령을 실행하여 MQ 모니터를 중지할 수 있습니다.

프로시저

MQ 모니터를 중지하려면 CICS 애플리케이션 프로그램에서 다음 명령을 실행하십시오.

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

여기서 *name*은 중지할 MQMONITOR 자원 정의의 이름입니다.

CICSplex SM 애플리케이션 프로그램에서 EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP)을 사용하여 CICS MQ 모니터 중지

CICSplex SM 애플리케이션 프로그램에서 **EXEC CPSM PERFORM SET OBJECT(MQMON) ACTION(STOP)** 명령을 실행하여 MQ 모니터를 중지할 수 있습니다.

이 태스크 정보

MQMONITOR 자원은 CICS 리전에 설치하여 사용해야 합니다. 세부사항은 [CICS-MQ 어댑터 설정의 내용](#)을 참조하십시오.

프로시저

MQ 모니터를 중지하려면 CICSplex SM 애플리케이션 프로그램에서 다음 명령을 실행하십시오.

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

여기서 *MQMON*은 중지할 MQMONITOR 자원 정의의 이름입니다.

CICS CEMT 트랜잭션을 통해 CICS MQ 모니터 중지

CICS CEMT 트랜잭션에서 **SET MQMONITOR** 명령을 실행하여 MQ 모니터를 중지할 수 있습니다.

이 태스크 정보

CEMT 트랜잭션을 시작하고 사용하는 방법에 대한 세부사항은 [CEMT - 마스터 터미널의 내용](#)을 참조하십시오.

프로시저

1. CICS 명령행에 다음 명령을 입력하십시오.

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

여기서 *name*은 중지할 MQMONITOR 자원의 이름입니다.

- 지정된 MQMONITOR 자원 정의의 AUTOSTART, ENABLESTATUS 및 MONSTATUS 값이 표시됩니다.
2. MONSTATUS 필드의 값 STARTED를 값 STOPPED로 겹쳐써서 MQ 모니터를 중지하십시오.

CICSplex SM 웹 사용자 인터페이스를 통해 CICS MQ 모니터 중지

CICSplex SM 웹 사용자 인터페이스(WUI)의 **WebSphere MQ 모니터** 보기에서 MQ 모니터를 중지할 수 있습니다.

이 태스크 정보

CICS Explorer에서 MQ 모니터(MQMON) 보기는 WUI의 이 보기에 해당하는 기능을 제공합니다.

프로시저

1. CICSplex SM WUI 기본 메뉴에서 **CICS 조작 > DB2, DBCTL 및 WebSphere MQ 조작 보기 > WebSphere MQ 모니터**를 선택하십시오.
이 보기 세트의 정보에 대한 전체 설명 및 목록은 [WebSphere MQ 모니터 - MQMON](#)의 내용을 참조하십시오.
2. 중지할 WebSphere MQ 모니터에 대한 **레코드** 상자를 선택하십시오.
3. 중지 단추를 클릭하십시오.

CICS MQ 모니터에 대한 정보 표시

CEMT INQUIRE MQMONITOR 또는 **EXEC CICS INQUIRE MQMONITOR** 명령을 사용하거나 CICSplex SM을 사용하여 MQ 모니터에 대한 정보를 표시할 수 있습니다.

이 태스크 정보

INQUIRE MQMONITOR 명령을 사용하여 CICS 리전에서 설치된 모든 MQMONITOR 자원에 대해 조회할 수 있습니다. 설치된 MQMONITOR 자원에 대한 다음 정보를 찾을 수 있습니다.

- MQMONITOR 자원이 사용으로 아니면 사용 안함으로 설정되는지에 대한 정보.
- 자동 재시작에서 MQMONTIOR가 사용되는지 여부.
- MQMONTIOR가 시작되는지 아니면 중지되는지에 대한 정보 및 시작되는 경우 현재 MQ 큐를 모니터링하는 ID. **CEMT INQUIRE TASK**를 사용하여 이 태스크에 대한 세부사항을 확인하십시오.
- MQ 큐를 모니터링하는 태스크로 전달할 데이터(MONDATA 속성). 예를 들어, MQ 브릿지 모니터 트랜잭션 CKBR로 전달되는 매개변수입니다.

참고:

MONDATA 데이터는 모니터링 태스크를 통해 표시 및 검색할 때 다음 18바이트가 앞에 추가됩니다.

바이트 1: <(왼쪽 꺾쇠 괄호)

바이트 2 - 9: *MQMONITOR resource name*

바이트 10 - 17: *USERID*

바이트 18: >(오른쪽 꺾쇠 괄호)

바이트 19 - 218에는 사용자가 입력한 MONDATA가 포함됩니다.

- MQ 큐를 모니터링하는 태스크에서 사용되는 트랜잭션의 이름.
- MQ 모니터에서 모니터링되고 있는 MQ 큐의 이름.

CICS-MQ 연결 통계 재설정

CKQC 트랜잭션을 사용하여 CICS-MQ 통계만 재설정하거나, 표준 CICS 통계 재설정 명령을 사용하여 CICS-MQ 통계를 포함한 모든 통계를 재설정할 수 있습니다. CICS-MQ 연결 통계도 통계 간격의 마지막에 재설정됩니다.

프로시저

- CICS 명령행에서 CICS-MQ 연결 통계만 재설정하려면 명령 **CKQC MODIFY Y**를 실행하십시오. 옵션 Y는 통계를 재설정합니다.

- CICS 애플리케이션 프로그램을 사용하여 CICS-MQ 연결 통계만 재설정하려면 **EXEC CICS LINK** 명령을 실행하여 어댑터 재설정 프로그램 DFHMQRS(또는 호환성을 위해 보유된 CSQCRST)에 링크하고 옵션 Y와 함께 **CKQC MODIFY** 명령을 실행하십시오.
다음 예제는 이를 수행하는 방법을 보여줍니다.

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

MODIFY 명령은 4개의 후행 공백 및 구별자로 사용되는 다른 공백으로 채워야 합니다(28 페이지의 『CICS 애플리케이션 프로그램의 CKQC 명령』 참조).

- CICS-MQ 어댑터 제어판을 사용하여 CICS-MQ 연결 통계만 재설정하려면 다음을 수행하십시오.
 - a) CKQC를 입력하고 Enter를 눌러 CICS-MQ 어댑터 제어판에 액세스하십시오.
 - b) 메뉴 표시줄에서 **Connection**을 선택하십시오.
 - c) 메뉴에서 **Modify** 조치를 선택하십시오.
 - d) **Modification Options** 보조 매개변수 창에서 **Reset statistics**를 선택하고 Enter를 누르십시오.
- CICS-MQ 연결 통계를 포함해 모든 CICS 통계를 재설정하려면 표준 CICS 통계 재설정 방법 중 하나를 다음과 같이 선택하십시오.
 - CEMT 트랜잭션을 사용하여 RESETNOW 옵션과 함께 **CEMT SET STATISTICS** 명령을 실행하십시오.
 - CICS 애플리케이션 프로그램에서 RESETNOW 옵션과 함께 **EXEC CICS SET STATISTICS** 명령을 실행하거나 **EXEC CICS PERFORM STATISTICS RECORD ALL RESETNOW** 명령을 실행하십시오.

CICS-MQ 연결을 사용 중인 태스크 제거

CICS CEMT 트랜잭션을 사용하여 CICS-MQ 어댑터를 사용하는 사용자 태스크를 제거할 수 있습니다.

이 태스크 정보

어댑터에서 대기하는 태스크가 **CEMT SET TASK FORCEPURGE** 명령에만 응답합니다. **CEMT SET TASK PURGE** 명령이 무시됩니다. **CEMT SET TASK** 명령의 전체 구문은 [CEMT SET TASK](#)의 내용을 참조하십시오.

프로시저

1. CICS-MQ 어댑터 제어판을 사용하여 해당 태스크 번호를 포함해 CICS-MQ 연결을 사용하는 태스크의 세부사항을 표시하십시오. 제거할 태스크의 번호를 기록하십시오.
[43 페이지의 『CICS-MQ 연결을 사용하는 태스크 표시』](#)에는 이를 수행하는 방법이 설명되어 있습니다.
2. 강제 제거할 각 태스크에 대해 CICS 명령행에 명령 CEMT SET TASK(*number*) FORCEPURGE를 입력하십시오. 여기서 *number*는 태스크의 태스크 번호입니다.

결과

CICS-MQ 어댑터가 FORCEPURGE 명령을 처리하는 방법은 태스크의 대기 상태 유형에 따라 다릅니다.

- 태스크가 메시지가 도달하기를 기다리는 경우(예: 애플리케이션이 MQGET WAIT 호출을 발행한 경우), 태스크가 코드 AEXY를 사용하여 즉시 중지됩니다.
- 그렇지 않으면 어댑터가 요청이 완료되기를 기다린 다음 태스크 종료가 적절한지 여부를 확인합니다.
 - 태스크가 위험 상태인 경우 어댑터는 태스크가 계속되도록 하고 제거 시도를 무시하여 데이터 및 시스템 무결성을 유지합니다. 메시지 DFHMQ0415I가 표시됩니다. 예를 들어, 태스크가 2단계 커밋 시퀀스 중 단계 2를 완료하는 과정에 있을 때 태스크가 위험 상태입니다.
 - 태스크가 위험 상태가 아닌 경우, 어댑터는 코드 AEXY를 사용하여 이 태스크를 종료합니다. 메시지 DFHMQ0414I가 표시됩니다.

CICS-MQ 트리거 모니터

CKTI는 연관된 큐의 트리거 조건이 충족될 때 트랜잭션을 시작하기 위해 CICS 환경에서 사용되는 CICS 제공 트리거 모니터(또는 태스크 초기자) 트랜잭션입니다. CICS-MQ 트리거 모니터 CKTI는 MQ 트리거 모니터에서 초

기화된 트랜잭션을 추적하는 기능을 제공합니다. 이 기능의 이점은 작업을 시각화하고 추적하는 기능을 관리자
에게 제공하여 가치를 더합니다.

큐 관리자에게 트리거 모니터는 큐를 제공하는 다른 애플리케이션과 유사합니다. 하지만 트리거 모니터는 시작
큐를 제공합니다.

트리거 모니터는 대개 지속적으로 실행되는 프로그램입니다. 트리거 메시지가 시작 큐에 도달하면 트리거 모니
터가 이 메시지를 검색합니다. 이 트리거 모니터는 메시지의 정보를 사용하여 애플리케이션 큐에서 메시지를 처
리할 애플리케이션을 시작하도록 명령을 실행합니다.

트리거 모니터는 프로그램이 올바른 애플리케이션 큐에서 올바른 조치를 수행할 수 있도록 시작 중인 프로그램
에 충분한 정보를 전달해야 합니다.

CKTI

각 시작 큐에 대해 CKTI의 여러 인스턴스를 시작할 수 있습니다. CKTI는 트리거 메시지의 MQTM 구조를 **EXEC CICS START TRANSID**로 시작되는 프로그램에 전달합니다. 시작된 프로그램은 **EXEC CICS RETRIEVE** 명
령을 사용하여 이 정보를 확보합니다. MQTM 구조에 대한 자세한 정보는 [MQTM 트리거 메시지](#)의 내용을 참조하
십시오.

프로그램은 RTRANSID 옵션과 함께 **EXEC CICS RETRIEVE** 명령을 사용하여 프로그램이 시작된 방식을 판별
할 수 있습니다. 리턴된 값이 CKTI인 경우 프로그램이 CICS-MQ 트리거 모니터에서 시작된 것입니다.

CKTI 사용 방법에 대한 예제는 z/OS용 IBM MQ에 제공되는 신용 조회 샘플 애플리케이션에서 모듈 CSQ4CVB2
에 제공된 소스 코드를 참조하십시오. 이 샘플에 대한 자세한 정보는 [신용 검사 샘플](#)의 내용을 참조하십시오.

MQMONITOR를 사용하여 CKTI 관리

MQMONITOR 자원 사용은 CKTI의 인스턴스를 제어하는 권장 방법입니다. 이점은 다음과 같습니다.

- CKTI의 여러 인스턴스를 설정하여 MQ 시작 큐를 모니터할 수 있습니다.
- MQ 큐 관리자에 대한 연결이 설정되면 CKTI를 자동으로 시작할 수 있습니다.
- MQMONITOR를 사용하면 CKTI 또는 메시지 이용 트랜잭션이 사전 설정된 사용자 ID로 실행될 수 있습니다.



경고: CKTI 트랜잭션의 인스턴스를 관리하기 위해 CKQC 트랜잭션과 MQMONITOR를 동시에 사용하면,
CKQC가 MQMONITOR를 인식하지 않고 MQMONITOR가 CKQC를 사용하여 관리되는 CKTI 트랜잭션을
인식하지 못하기 때문에 MQMONITOR 상태와 통계가 혼동될 수 있습니다.

중요사항: CICS 제공 트리거 모니터를 사용하는 대신 고유 MQ 트리거 모니터를 사용하여 CICS 환경에서 MQ 큐
를 제공하려는 경우, 사용자 작성 MQ 트리거 모니터 프로그램을 디자인하고 구현할 때 [58 페이지의 『사용자
작성 MQ 트리거 모니터 및 MQ 메시지 이용자 개발과 사용』](#)의 지시사항을 따라야 합니다.

CKTI의 인스턴스 시작

여러 방법으로 CKTI(CICS-MQ 트리거 모니터 또는 태스크 초기자)의 인스턴스를 시작할 수 있습니다.
MQMONITOR 자원 사용은 CKTI의 인스턴스를 제어하는 권장 방법입니다.

이 태스크 정보

CKTI의 사용자 ID

CKTI의 인스턴스가 MQMONITOR를 통해 시작되면 CKTI 트랜잭션과 연관된 사용자 ID를 MQMONITOR 자
원의 **MONUSERID** 속성에서 얻습니다.

CKTI가 터미널에서 CKQC 트랜잭션을 통해 또는 DFHMQSSQ(호환성을 위해 보유된 CSQCSSQ)에 링크되는
사용자 작성 프로그램을 통해서 시작되는 경우, CKTI가 사용하는 사용자 ID는 CKTI를 시작한 터미널의 사용
자 ID와 동일합니다.

CKTI가 다른 CICS 트랜잭션(예: 사용자 작성 CICS 애플리케이션)을 시작하는 경우, CKTI의 사용자 ID가 이
러한 애플리케이션으로 전파됩니다. 예를 들어, CKTI가 사용자 ID CIC1으로 실행 중이고 송신자 MCA 트랜
잭션 CKSG를 시작해야 하는 트리거 이벤트가 발생하면, CKSG 트랜잭션도 사용자 ID CIC1으로 실행됩니다.
따라서 사용자 ID CIC1에는 필수 전송 큐에 대한 액세스 권한이 있어야 합니다.

프로시저

다음 방법 중 하나를 사용하여 CKTI를 시작할 수 있습니다.

- CICS MQ 모니터 사용

올바르게 설정된 경우 MQMONITOR 자원을 사용하면 CKTI 트랜잭션이 MQ 큐 관리자에 대한 연결이 설정될 때 자동으로 시작될 수 있습니다. CKTI를 제어하는 MQMONITOR가 자동으로 시작되지 않으면 여러 방법으로 MQ 모니터를 시작할 수 있습니다.

- CICS-MQ 어댑터 제어판 사용

CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어판에서 CKTI 트랜잭션의 인스턴스를 시작할 수 있습니다.

- CICS 명령행에서 **CKQC STARTCKTI** 실행

CKQC STARTCKTI 명령을 실행하여 기본 시작 큐 또는 지정된 시작 큐를 제공하는 CKTI의 인스턴스를 시작할 수 있습니다.

- CICS 애플리케이션 프로그램 사용

CICS 애플리케이션 프로그램에서 CKTI의 인스턴스를 시작하려면 애플리케이션 프로그램이 어댑터 태스크 초기화 프로그램 DFHMQSSQ(또는 호환성을 위해 보유된 CSQCSSQ)에 링크되어야 합니다. MQMONITOR를 사용하여 CKTI를 제어하는 경우, EXEC CICS SET MQMONITOR (name) STARTED를 실행하는 애플리케이션 프로그램에서 MQMONITOR를 시작할 수 있습니다.

- 자동화 제품 사용

특정 사용자 ID로 트리거 모니터의 시작을 자동화하기 위해 자동화 제품(예: NetView)을 사용할 수 있습니다. 이를 사용하여 CICS 콘솔에 사인온하고 **STARTCKTI** 명령을 실행할 수 있습니다.

CICS MQ 모니터를 사용하여 CKTI 시작

올바르게 설정된 경우 MQMONITOR 자원을 사용하면 CKTI 트랜잭션이 MQ 큐 관리자에 대한 연결이 설정될 때 자동으로 시작될 수 있습니다. CKTI를 제어하는 MQMONITOR가 자동으로 시작되지 않으면 여러 방법으로 MQ 모니터를 시작할 수 있습니다.

시작하기 전에

- MQMONITOR를 사용하여 CICS 리전에서 CKTI의 인스턴스를 제어하려면 이와 같은 MQMONITOR 자원을 정의하고 설치해야 합니다. 52 페이지의 표 8에는 CKTI를 제어하는 MQMONITOR에 지정해야 하는 몇 가지 중요한 속성이 나열됩니다.

속성 AUTOSTART(YES) 및 STATUS(ENABLED)는 MQ 연결이 설정될 때 자동으로 다시 시작할 MQMONITOR에 정의해야 합니다. 자동 재시작을 사용 안함으로 설정하고 MQ 모니터를 수동으로 시작하려면, AUTOSTART(NO)를 사용하거나 AUTOSTART 속성 없이 MQMONITOR 자원을 정의하십시오.

표 8. 시작 큐 모니터링을 위한 MQMONITOR 자원 속성			
속성	필수/선택사항	기본값	설명
AUTOSTART	선택사항	예	<p>이 속성은 MQ 모니터의 자동 재시작을 제어합니다.</p> <p>AUTOSTART(YES) MQ 큐 관리자에 대한 연결이 설정될 때 CKTI가 자동으로 재시작될 수 있게 합니다.</p> <p>AUTOSTART(NO) MQ 모니터가 자동으로 시작되지 않습니다. MQ 큐 관리자에 대한 연결이 설정된 후 수동으로 MQ 모니터를 시작해야 합니다.</p>

표 8. 시작 큐 모니터링을 위한 MQMONITOR 자원 속성 (계속)			
속성	필수/선택사항	기본값	설명
MONUSERID	필수	-	CKTI와 연관시킬 사용자 ID를 지정하십시오. 이 속성은 보안 검사가 활성화 상태(즉, SEC 시스템 초기화 매개변수가 YES로 설정됨)인 경우에만 유효합니다.
QNAME	선택사항	생략되면 기본값은 &APPLID..INITIATION.QUEUE입니다. 여기서 &APPLID.는 CICS 리전의 applid입니다.	모니터할 MQ 시작 큐의 이름을 지정하십시오.
STATUS	선택사항	ENABLED	이 속성은 자원을 리전에서 사용 가능하게 만듭니다.
TRANSACTION	선택사항	CKTI	CICS 제공 트리거 모니터(또는 태스크 초기자) 트랜잭션의 4자 ID를 지정합니다. 참고: 고유 MQ 트리거 모니터를 사용하는 경우 이 트랜잭션의 이름을 지정하십시오. CICS 환경의 고유 MQ 트리거 모니터 사용에 대한 고려사항은 50 페이지의 『CICS-MQ 트리거 모니터』의 내용을 참조하십시오.

- 보안 검사가 활성화인 경우(즉, **SEC** 시스템 초기화 매개변수가 YES로 설정된 경우), MQ 모니터 상태를 시작됨으로 설정하려는 트랜잭션과 연관된 사용자 ID가 **MONUSERID**에 정의된 사용자 ID의 대리이고 **MONUSERID**와 연관된 트랜잭션을 시작할 수 있는 권한이 부여되었는지 확인하십시오. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTPIUSR 사용자 ID(지정된 경우)입니다.

프로시저

- CKTI를 제어하는 MQ 모니터를 자동으로 시작하려면 MQMONITOR 자원에 속성 AUTOSTART(YES) 및 STATUS(ENABLED)가 있는지 확인하십시오.
CICS 및 MQ 간의 연결이 설정될 때, MQCONN 자원을 CONNECTED로 설정하는 태스크와 연관된 사용자 ID에 연관 트랜잭션을 시작할 충분한 권한이 있으면 CICS는 AUTOSTART(YES) 및 STATUS(ENABLED) 속성으로 설치된 MQ 모니터를 시작합니다. 그러면 CKTI의 인스턴스가 실행됩니다.
- CKTI를 제어하는 MQ 모니터를 수동으로 시작하기 위해 44 페이지의 『CICS MQ 모니터 시작』에 설명된 방법 중 하나를 사용할 수 있습니다.

CICS-MQ어댑터 제어판에서 CKTI 시작

CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어판에서 CKTI 트랜잭션의 인스턴스를 시작할 수 있습니다.

프로시저

1. CICS-MQ 어댑터 초기 제어판의 메뉴 표시줄에서 **CKTI**를 선택하십시오.
2. 메뉴에서 **시작** 조치를 선택하십시오.
3. **태스크 초기자 시작** 보조 창에서 **시작 큐 이름** 필드를 사용하여 이 CKTI 인스턴스가 제공할 시작 큐의 이름을 지정하십시오.
이 필드를 공백으로 두면, 정의된 경우 기본 시작 큐가 사용됩니다.

예

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

그림 7. CKTI의 인스턴스 시작

CICS 명령행에서 CKTI 시작

CKQC STARTCKTI 명령을 실행하여 기본 시작 큐 또는 지정된 시작 큐를 제공하는 CKTI의 인스턴스를 시작할 수 있습니다.

이 태스크 정보

시작 큐를 지정하지 않고 **CKQC STARTCKTI** 명령을 실행하는 경우, 이 명령은 CICS 리전의 기본 시작 큐 (regionAPPLID.initiation.queue)를 나타내는 것으로 자동 해석됩니다.

프로시저

- 정의된 경우 기본 시작 큐를 제공하도록 다음 명령을 사용하여 CKTI의 인스턴스를 시작하십시오.

```
CKQC STARTCKTI
```

- 지정된 시작 큐 CICS01.INITIATION.QUEUE2를 제공하도록 다음 명령을 사용하여 CKTI의 인스턴스를 시작하십시오.

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

CICS 애플리케이션 프로그램에서 CKTI 시작

CICS 애플리케이션 프로그램에서 CKTI의 인스턴스를 시작하려면 애플리케이션 프로그램이 어댑터 태스크 초기화 프로그램 DFHMQSSQ(또는 호환성을 위해 보유된 CSQCSSQ)에 링크되어야 합니다. MQMONITOR를 사용하여 CKTI를 제어하는 경우, EXEC CICS SET MQMONITOR (name) STARTED를 실행하는 애플리케이션 프로그램에서 MQMONITOR를 시작할 수 있습니다.

이 태스크 정보

DFHMQSSQ에 링크된 애플리케이션 프로그램 사용

DFHMQSSQ에 대해 **EXEC CICS LINK**를 수행하는 경우 프로그램에는 터미널 연관 태스크가 필요합니다.

STARTCKTI 명령의 경우 10자를 채워야 합니다. [28 페이지의 『애플리케이션 프로그램의 명령 구문』](#)의 내용을 참조하십시오.

EXEC CICS SET MQMONITOR (name) STARTED를 실행하는 애플리케이션 프로그램 사용

다음 조건이 충족되는 경우 애플리케이션 프로그램이 CKTI를 제어하는 MQMONITOR를 시작합니다.

- 현재 tranid는 TRANID 속성의 값과 다릅니다.

- MQMONITOR는 아직 STARTED 상태가 아닙니다.
- 현재 트랜잭션과 연관된 사용자 ID는 MQMONITOR와 연관된 트랜잭션의 상태를 시작됨으로 설정할 수 있는 권한을 부여받습니다.

프로시저

- DFHMQSSQ에 링크하고 기본 시작 큐를 사용하는 CKTI를 시작하려면 다음과 같은 명령을 실행하십시오.

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

- DFHMQSSQ에 링크하고 시작 큐 CICS01.INITIATION.QUEUE2를 사용하는 CKTI를 시작하려면 다음과 같은 명령을 실행하십시오.

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

- CKTI를 제어하는 MQMONITOR를 시작하려면 애플리케이션 프로그램에서 다음 명령을 실행하십시오.

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

여기서 *name*은 CKTI를 제어하는 MQMONITOR의 이름입니다.

결과

DFHMQSSQ의 출력 메시지가 시스템 콘솔에 표시됩니다.

자동화 제품을 통해 CKTI 자동 시작

특정 사용자 ID로 트리거 모니터의 시작을 자동화하기 위해 자동화 제품(예: NetView)을 사용할 수 있습니다. 이를 사용하여 CICS 콘솔에 사인온하고 **STARTCKTI** 명령을 실행할 수 있습니다.

CKQC STARTCKTI 명령을 포함하는 순차 터미널 입력을 사용하여 CRLP 터미널을 에뮬레이트하도록 정의된 사전 설정된 보안 순차 터미널도 사용할 수 있습니다.

하지만 어댑터 변경 모니터가 CICS를 IBM MQ에 다시 연결하면(예: 큐 관리자가 다시 시작된 후) 초기 IBM MQ 연결에 지정된 CKTI만 다시 시작됩니다. 추가 CKTI 시작을 자동화해야 합니다.

CKTI의 인스턴스 중지

여러 방법으로 CKTI의 인스턴스를 중지할 수 있습니다. MQMONITOR 자원 사용은 CKTI의 인스턴스를 제어하는 권장 방법입니다.

이 태스크 정보

MQ 모니터를 사용하여 CKTI를 제어하는 경우 MQ 연결이 중지되면 MQ 모니터가 자동으로 중지됩니다.

리전에서 MQ 모니터를 사용하여 CKTI를 제어하는 경우, CKTI의 인스턴스가 여러 개 실행 중일 때 CKQC를 사용하여 CKTI의 인스턴스를 중지하면 예측 불가능한 결과가 발생할 수 있습니다.

프로시저

- 다음 방법을 사용하여 CKTI의 인스턴스를 중지할 수 있습니다.

- [CKTI를 제어하는 CICS MQ 모니터 중지](#)

MQ 모니터를 사용하여 CKTI의 인스턴스를 제어하는 경우 여러 방법으로 MQ 모니터를 중지할 수 있습니다.

- [CICS-MQ 어댑터 제어판 사용](#)

CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어 패널에서 CKTI 트랜잭션의 인스턴스를 중지할 수 있습니다.

- [터미널에서 CKQC STOPCKTI 명령 실행](#)

터미널에서 **CKQC STOPCKTI** 명령을 실행하여 CKTI의 인스턴스를 중지할 수 있습니다.

- 애플리케이션 프로그램 사용

어댑터 태스크 초기자 프로그램 DFHMQSSQ(호환성을 위해 보유한 CSQCSSQ)에 링크하여 CKTI의 인스턴스를 중지할 수 있습니다.

CICS-MQ 어댑터 제어판에서 CKTI의 인스턴스 중지

CICS-MQ 어댑터 제어 트랜잭션 CKQC를 실행한 다음 CICS-MQ 어댑터 제어 패널에서 CKTI 트랜잭션의 인스턴스를 중지할 수 있습니다.

프로시저

1. CICS-MQ 어댑터 초기 제어판의 메뉴 표시줄에서 **CKTI**를 선택하십시오.
2. 메뉴에서 **중지** 조치를 선택하십시오.
3. **태스크 초기자 중지** 보조 창을 사용하여 CKTI의 이 인스턴스에서 제공되는 시작 큐의 이름을 지정하십시오. 이름을 공백으로 두면, 정의된 경우 기본 시작 큐가 사용됩니다.

예

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

그림 8. 태스크 초기자 CKTI의 인스턴스 중지

터미널에서 CKTI의 인스턴스 중지

터미널에서 **CKQC STOPCKTI** 명령을 실행하여 CKTI의 인스턴스를 중지할 수 있습니다.

이 태스크 정보

시작 큐를 지정하지 않고 **CKQC STOPCKTI** 명령을 실행하는 경우, 이 명령은 CICS 리전의 기본 시작 큐 (regionAPPLID.initiation.queue)를 나타내는 것으로 자동 해석됩니다.

프로시저

- 이 명령을 사용하여 기본 시작 큐를 제공하는 CKTI의 인스턴스를 중지하십시오.

```
CKQC STOPCKTI
```

- 이 명령을 사용하여 지정된 시작 큐를 제공하는 CKTI의 인스턴스를 중지하십시오.

```
CKQC STOPCKTI queue_name
```

여기서 *queue_name*은 시작 큐의 이름입니다.

참고: MQMONITOR 자원을 사용하는 경우, 이 명령은 지정된 MQ 큐를 모니터하는 모든 MQMONITOR를 중지합니다.

예를 들어, CICS01.INITIATION.QUEUE2를 제공하는 CKTI의 인스턴스를 중지하려면 다음 명령을 실행하십시오.

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

애플리케이션 프로그램에서 CKTI의 인스턴스 중지

어댑터 태스크 초기자 프로그램 DFHMQSSQ(호환성을 위해 보유된 CSQCSSQ)에 링크하여 CKTI의 인스턴스를 중지할 수 있습니다.

이 태스크 정보

이러한 예제는 CICS 프로그램에서 CKTI의 인스턴스를 중지하기 위한 대체 LINK 명령을 보여줍니다. STOPCKTI 명령의 경우 10자를 채워야 합니다. [28 페이지의 『애플리케이션 프로그램의 명령 구문』](#)의 내용을 참조하십시오.

프로시저

- 이 명령은 기본 시작 큐를 제공하는 CKTI를 중지합니다.

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

- 이 명령은 지정된 시작 큐를 제공하는 CKTI를 중지합니다.

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

CKTI의 현재 인스턴스 표시

CICS-MQ 어댑터 제어판을 사용하여 CKTI의 현재 인스턴스에 대한 세부사항을 표시할 수 있습니다. 이에 해당하는 기능은 CICS 명령행이나 CICS 애플리케이션 프로그램을 통해 사용할 수 없습니다.

프로시저

- CICS-MQ 어댑터 초기 제어판의 메뉴 표시줄에서 **CKTI**를 선택하십시오.
- 폴다운 메뉴에서 **표시** 조치를 선택하십시오.
[58 페이지의 그림 9](#)에는 CKTI의 각 인스턴스에 제공된 세부사항이 표시됩니다.
 - CICS 태스크 번호
 - 태스크 상태
 - 스레드 상태
 - 발행된 API 호출의 수
 - 발행된 최신 API 호출
 - 제공되는 시작 큐의 이름
- 기능 키 F1을 눌러 패널의 각 필드에 대한 도움말 정보를 표시하십시오.

예

```
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
```

그림 9. CKQC 표시 CKTI 패널

사용자 작성 MQ 트리거 모니터 및 MQ 메시지 이용자 개발과 사용

CICS 제공 트리거 모니터 CKTI를 사용하는 대신 고유 MQ 트리거 모니터를 작성하고 이 모니터를 사용하여 CICS 환경에 MQ 큐를 제공할 수 있습니다. 애플리케이션 입력 큐에서 메시지를 직접 가져와서 필수 로직을 수행하는 MQ 메시지 이용자를 직접 작성할 수도 있습니다. 이 주제는 사용자 작성 MQ 트리거 모니터와 MQ 메시지 이용자의 개발 및 사용에 관한 중요 지시사항을 제공합니다.

사용자 작성 MQ 모니터 또는 MQ 메시지 이용자 프로그램의 책임

MQMONITOR를 사용하여 MQ 큐를 처리하는 경우, MQMONITOR가 시작되면 **EXEC CICS START** 명령이 FROM 데이터로서 <MQMONITOR_resource_nameUSERID>를 접두부로 추가한 **MONDATA**와 함께 트랜잭션을 시작하므로 사용자 작성 프로그램이 MQ에 대한 호출을 발행하기 전에 **MONDATA** 속성에 대한 설명과 일치하는 이 데이터를 구조에서 검색해야 합니다.

그러면 MQ 큐를 열기 전에 프로그램이 검색된 데이터에 지정된 MQ 모니터의 상태를 시작됨으로 설정해야 합니다.

CICS로 돌아가기 전에 프로그램은 MQ 모니터의 상태를 중지됨으로 설정해야 합니다.

MQ 메시지 이용자가 원격 리전으로 작업을 라우팅하는 데 사용되는 경우의 보안 고려사항

MQ 메시지 이용자를 사용하여 원격 리전으로 작업을 라우팅하는 경우, 상호시스템 통신 정의가 원격 리전의 트랜잭션과 연관될 사용자 ID를 판별하는 데 미칠 수 있는 보안 구성의 영향을 고려해야 합니다. 세부사항은 [START](#)에서 **START** 명령으로 시작되는 라우팅된 트랜잭션에 대한 정보를 참조하십시오.

프로시저

사용자 작성 MQ 모니터 또는 MQ 메시지 이용자 프로그램은 아래에 표시된 순서로 다음 단계를 수행해야 합니다.

1. **EXEC CICS ASSIGN STARTCODE** 명령을 사용하여 트랜잭션이 데이터로 시작되었는지 확인하십시오.
2. **EXEC CICS RETRIEVE** 명령을 실행하여 트랜잭션이 시작되었을 때 구조로 전달되었던 FROM 데이터를 검색하십시오.

다음 단계에서 이 데이터의 2 - 9바이트로 전달된 MQMONITOR 이름을 사용하십시오.

3. EXEC CICS SET MQMONITOR(MQMONITOR_name) STARTED를 실행하여 검색된 데이터에 지정된 MQ 모니터의 상태를 시작됨으로 설정하십시오.
4. MQ 큐를 여십시오.
5. 입력 큐의 메시지를 받고 필수 애플리케이션 로직을 수행하십시오.
6. 어떤 이유로든 프로그램을 종료할 때 EXEC CICS SET MQMONITOR(MQMONITOR_name) STOPPED를 실행하여 MQ 모니터의 상태를 중지됨으로 설정하십시오. MQMONITOR가 시작되었을 때 수집된 CICS 통계를 기록하십시오.
7. CICS로 돌아가십시오.



경고:

- 사용자 작성 MQ 트리거 모니터 또는 MQ 메시지 이용자 프로그램이 이전 프로시저에 설명된 순서로 단계를 수행하지 못하면, 예상치 못한 결과가 발생할 수 있습니다. 예를 들어, 레코딩된 통계를 예측할 수 없으며 MQMONITOR의 MONSTATUS 값이 예측 불가능 상태가 됩니다.
- CICS 제공 MQ 트리거 모니터 프로그램 DFHMQTSK는 CICS-MQ 트리거 모니터 및 태스크 초기자 트랜잭션 CKTI에 사용하도록 예약되어 있습니다. DFHMQTSK를 사용자 트랜잭션으로 호출하려 시도하면 사용자 트랜잭션이 이상 종료 코드 AMQO를 생성하며 이상 종료됩니다.

참고: 사용자 작성 MQ 트리거 모니터 또는 MQ 메시지 이용자 프로그램이 **EXEC CICS START** 요청을 발행하는 경우 및 **EXEC CICS START** 요청을 제한하는 정책이 배치되고 동일한 리전에서 활성화인 경우, 정책이 사용자 작성 프로그램을 이상 종료할 수 있습니다. 이 상황에서 **EXEC CICS START** 요청을 제한하는 정책을 활성화하는 것은 적절하지 않습니다. 정책 규칙에 대한 자세한 정보는 [정책 태스크 규칙](#)의 내용을 참조하십시오.

자세히 보기

- [CICS Developer Center: MQMONITOR를 사용하여 CICS-MQ 트리거 모니터 및 MQ 메시지 이용자의 관리 간소화](#)

이 블로그는 MQMONITOR를 사용하고 참조를 위해 코드 예제를 제공하는 MQ 메시지 이용자 프로그램을 개발하는 방법을 보여줍니다.

제 4 장 CICS-MQ 브릿지 관리

브릿지를 시작하기 위해 CICS-MQ 브릿지 모니터 트랜잭션을 실행합니다. 트랜잭션의 기본 이름은 CKBR이지만 고유 대체 트랜잭션을 정의할 수 있습니다.

이 태스크 정보

CICS-MQ 모니터를 사용하여 CICS-MQ 브릿지를 제어하는 것이 좋습니다. MQMONITOR 자원을 사용하면 IBM MQ 관리자에 대한 연결이 설정될 때 자동으로 브릿지를 다시 시작할 수 있습니다. 설정 지시사항은 [CICS-MQ 브릿지용 MQMONITOR 자원 설정의 내용](#)을 참조하십시오.

CKBR을 실행할 때 다음 선택적 매개변수를 지정할 수 있습니다.

Q=qqq

qqq는 CICS-MQ 브릿지에 대한 IBM MQ 요청 큐의 이름입니다. 고유 요청 큐를 정의한 경우 여기에 지정해야 합니다. IBM MQ의 오브젝트 이름이 대소문자를 구분합니다. 큐 이름을 지정하지 않으면 CKBR은 기본 요청 큐 SYSTEM.CICS.BRIDGE.QUEUE를 사용합니다.

AUTH=LOCAL/IDENTIFY/VERIFY_UOW/VERIFY_ALL

사용할 인증 레벨을 지정합니다. 기본값은 LOCAL입니다. LOCAL을 사용하면, 브릿지 태스크에서 실행되는 CICS 프로그램이 브릿지 모니터가 시작된 사용자 ID로 시작되며, 이는 브릿지 모니터를 시작하는 방법 선택에 영향을 줄 수 있습니다. 다른 레벨의 인증을 사용하면 CICS 프로그램이 요청 메시지에서 추출된 사용자 ID로 실행됩니다. 인증 레벨에 대한 자세한 정보는 [CICS-WebSphere MQ 브릿지의 보안의 내용](#)을 참조하십시오.

WAIT=nnn

nnn은 브릿지 태스크가 많은 사용자 프로그램을 실행하는 작업 단위를 처리할 때 제한시간이 초과되기 전에 브릿지 태스크가 후속 요청을 기다릴 시간(초)입니다. 이 값은 0 - 999 범위에 있어야 합니다. 이 매개변수가 지정되지 않으면 기본값은 -1 값을 가진 MQWI_UNLIMITED입니다. 대기 시간을 지정하지 않는 경우 브릿지가 CICS 또는 큐 관리자의 종료를 막을 수 있으므로 대기 시간을 지정하는 것이 좋습니다.

MSG=CSMT/LOG/BOTH

CICS-MQ 브릿지에서 생성된 메시지가 CICS 작업 로그로 보내는지, CICS 마스터 터미널로 보내는지 아니면 둘 다로 보내는지를 판별합니다. 기본값은 BOTH입니다.

PASSTKTA=applid

applid는 PassTicket 유효성 검증에 사용할 애플리케이션 ID를 지정합니다. 기본값은 CICS 리전 애플리케이션 ID입니다. PassTickets에 대한 정보는 [IBM MQ 제품 문서의 z/OS에 대한 보안 설정의 내용](#)을 참조하십시오.

ROUTEMEM=Y/N

표시 만료 메시지가 DLQ로 라우팅되는지 여부를 판별합니다. 매개변수를 지정하지 않으면 기본값은 N(라우팅되지 않음)입니다.

SMFMQGET=number

CICS-MQ 브릿지가 시작될 때 시스템 초기화 매개변수 **MNSYSNC=NO**가 지정되고 성능 클래스 모니터링이 활성화된 경우, 브릿지는 지정된 MQGET 요청 수가 브릿지에서 발행될 때마다 SMF 유형 110 하위 유형 1 모니터링 레코드를 씁니다. 올바른 수는 1 - 99 999 범위에 있습니다.

CICS-MQ 브릿지 시작

CICS MQ 모니터를 사용하여 브릿지를 제어하고 있고 MQMONITOR 자원이 올바르게 구성된 경우, MQ 큐 관리자에 대한 연결이 설정되면 브릿지가 자동으로 시작됩니다. MQMONITOR 자원 사용은 브릿지를 제어하는 권장 방법입니다. CKBR 트랜잭션을 사용하여 브릿지를 수동으로 시작할 수도 있습니다.

시작하기 전에

- MQ 모니터를 사용하여 브릿지를 제어하려면 MQ 브릿지를 제어하기 위해 MQMONITOR 자원을 정의하고 설치해야 합니다. 자세한 지시사항은 [CICS-MQ 브릿지용 MQMONITOR 자원 설정의 내용](#)을 참조하십시오.

- 또한 보안 검사가 활성화된 경우(즉, **SEC** 시스템 초기화 매개변수가 YES로 설정된 경우), MQ 모니터 상태를 시작됨으로 설정하려는 트랜잭션과 연관된 사용자 ID가 **MONUSERID**에 정의된 사용자 ID의 대리이고 **MONUSERID**와 연관된 트랜잭션을 시작할 수 있는 권한이 부여되었는지 확인하십시오. CICS Explorer와 같은 CICSplex SM API 인터페이스를 통해 MQ 모니터 상태를 설정하는 경우 MQ 모니터 트랜잭션과 연관시킬 사용자 ID는 리전 사용자 ID 또는 PLTIUSR 사용자 ID(지정된 경우)입니다.

프로시저

- MQ 큐 관리자에 대한 연결이 설정될 때 CICS-MQ 브릿지를 자동으로 시작하려면 MQ 모니터를 사용하여 브릿지를 제어하고 이 MQMONITOR 자원에 속성 AUTOSTART(YES) 및 STATUS(ENABLED)가 있는지 확인해야 합니다.
CICS 및 MQ 간의 연결이 설정될 때, MQCONN 자원을 CONNECTED로 설정하는 태스크와 연관된 사용자 ID에 연관 트랜잭션을 시작할 충분한 권한이 있으면 CICS는 AUTOSTART(YES) 및 STATUS(ENABLED) 속성으로 설치된 MQ 모니터를 시작합니다.
- MQ 모니터를 사용하여 브릿지를 제어하는 경우 여러 방법으로 MQ 모니터를 수동으로 중지할 수 있습니다. 44 페이지의 『CICS MQ 모니터 시작』의 지시사항을 따르십시오.
- CKBR 트랜잭션을 사용하여 브릿지를 수동으로 시작하려는 경우, 다음 방법 중 하나로 CKBR 태스크를 시작하십시오.
 - 다음과 같이 터미널(3270 또는 기타)에서 하나의 행을 입력하십시오.

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

예:

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

입력한 다음에 다른 작업에서 사용 가능하도록 터미널이 잠금 해제됩니다.

- 데이터로서 매개변수와 함께 CKBR 트랜잭션을 위한 **EXEC CICS START** 명령을 실행하십시오.
CICS PLTPI 처리의 일부로 실행되어 이 명령을 실행하고 브릿지 모니터 트랜잭션이 실행되는 사용자 ID를 지정하는 프로그램이 있을 수 있습니다.
- COMMAREA의 데이터로서 매개변수와 함께 **EXEC CICS LINK**를 프로그램 DFHMQBR0(CSQCBR00라고도 함)에 대해 실행하십시오.
DFHMQBR0는 장기 실행 태스크이며 브릿지가 중지된 경우에만 이 프로그램이 리턴됩니다.
- 브릿지 요청 큐에서 TRIGGER TRIGTYPE(FIRST)를 사용하여 USERDATA의 임의 매개변수와 함께 APPLICID(CKBR)를 지정하여 프로세스를 시작하십시오.
USERDATA에서 **Q=qqq** 매개변수를 지정할 수 없으므로 자체 요청 큐를 정의한 경우 이 방법이 적합하지 않습니다.

다음에 수행할 작업

요청 큐를 공유하는 여러 브릿지 모니터를 실행하는 경우, 메시지를 브릿지 요청 큐에 두어 모니터 중 하나를 시작할 수 있습니다. 하지만 개인용 로컬 큐의 경우 하나의 트리거 메시지만 생성되므로 브릿지 모니터가 하나의 CICS 리전에서만 시작됩니다. 따라서 추가 브릿지 모니터를 시작하려면 CICS 시작 PLT 처리에서 프로그램을 사용하여 필수 매개변수로 트랜잭션을 시작하거나 자동화 제품을 사용하여 트랜잭션을 시작하는 등의 대체 방법을 고려해야 합니다. 요청 큐 속성을 변경하여 브릿지를 종료하는 경우 브릿지가 종료되면 GET(ENABLED)을 재설정하십시오.

CICS-MQ 브릿지 중지

GET(DISABLED)를 설정하여 요청 큐의 속성을 변경하거나 CICS를 종료하거나 큐 관리자를 종료하여 CICS-MQ 브릿지를 종료할 수 있습니다. CICS MQ 모니터를 사용하여 브릿지를 제어하는 경우, MQ 모니터를 중지하여 브릿지를 종료할 수 있습니다.

이 태스크 정보

MQ 모니터를 사용하여 브릿지를 제어하는 경우, MQ 큐 관리자에 대한 연결이 중지되면 브릿지가 자동으로 중지됩니다.

프로시저

- MQ 모니터를 사용하여 브릿지를 제어하는 경우 여러 방법으로 MQ 모니터를 수동으로 중지할 수 있습니다. [47 페이지의 『CICS MQ 모니터 중지』](#)의 지시사항을 따르십시오.

제 5 장 CICS-MQ 어댑터를 위한 보안

CICS-MQ 어댑터는 IBM MQ 보안에 사용하기 위한 정보를 IBM MQ에 제공합니다.

제공되는 정보는 다음과 같습니다.

- CICS 자원 레벨 보안이 이 트랜잭션에 대해 활성화되어 있는지 여부. 자세한 정보는 [자원 정의 보안](#)의 내용을 참조하십시오.
- 사용자 ID.
 - 사용자가 서명하지 않은 터미널 태스크의 경우 사용자 ID는 터미널과 연관된 CICS 사용자 ID이며 다음 중 하나입니다.
 - CICS **DFLTUSER** 시스템 초기화 매개변수에 지정된 기본 CICS 사용자 ID.
 - 터미널 정의에 지정된 사전 설정된 보안 사용자 ID.
 - 비터미널 태스크의 경우 어댑터는 사용자 도메인에 대한 호출로 사용자 ID를 획득합니다.

CICS-MQ 어댑터 트랜잭션을 위한 보안 구현

사용자가 CICS-MQ 어댑터를 관리할 수 있도록 하려면 사용자에게 적절한 CICS 트랜잭션에 대한 권한을 부여해야 합니다.

필요한 경우 어댑터의 특정 기능에 대한 액세스를 제한할 수 있습니다. 예를 들어, 사용자에게 어댑터의 현재 상태를 표시하는 것 외의 다른 권한을 주지 않으려면 CKQC, CKBM, CKRT 및 CKDP에 대한 액세스 권한만 부여하십시오.

RESSEC(NO) 및 CMDSEC(NO)를 사용하여 CICS에 해당 트랜잭션을 정의하십시오. 자세한 정보는 [자원 정의 보안](#) 및 [CICS 명령 보안](#)의 내용을 참조하십시오.

트랜잭션	기능
CKAM	경보 모니터
CKBM	어댑터 기능 제어
CKCN	연결
CKDL	라인 모드 표시
CKDP	전체 화면 표시
CKQC	어댑터 기능 제어
CKRS	통계
CKRT	어댑터 기능 제어
CKSD	연결 끊기
CKSQ	CKTI START/STOP
CKTI	트리거 모니터

관리자 뿐만 아니라, IBM MQ에 연결하는 사용자 ID, **PLTPIUSR** 시스템 초기화 매개변수에 설정된 사용자 ID 및 [CICSplex SM MAS 에이전트 사용자 ID](#)도 CKTI와 CKAM 트랜잭션을 실행할 권한이 있어야 합니다.

CICS-MQ 어댑터 사용자 ID

CICS-MQ 어댑터와 연관된 사용자 ID는 IBM MQ에 액세스하는 호출 트랜잭션과 연관된 사용자 ID입니다.

IBM MQ 자원을 위한 사용자 ID 검사

CICS-MQ 자원(MQCONN 또는 MQMONITOR)을 사용하여 IBM MQ에 액세스하는 경우 IBM MQ에서 사용되는 사용자 ID는 MQI 명령을 실행하는 트랜잭션의 사용자 ID입니다.

- PLTPI 프로그램의 경우 이는 **PLTPIUSR** 시스템 초기화 매개변수입니다.
- PLTSD 프로그램의 경우 이는 시스템 종료 트랜잭션과 연관된 사용자 ID입니다.
- 다른 프로그램의 경우 이는 실행 트랜잭션과 연관된 사용자 ID입니다.

CICS-MQ 트리거 모니터의 사용자 ID

보안 검사가 활성화된 경우 항상 MQMONITOR를 사용하여 CKTI 인스턴스를 시작하도록 권장합니다. MQMONITOR를 사용하는 경우에는 CKQC를 사용하지 마십시오. MQMONITOR를 사용하면 시작 방법에 관계없이 MQMONITOR의 **MONUSERID** 속성인 단일 사용자 ID가 항상 CKTI 인스턴스에 사용됩니다.

MQMONITOR가 없는 CICS-MQ 트리거 모니터의 사용자 ID

MQMONITOR 없이 CKTI를 시작하는 것이 지원되지만 권장하지는 않습니다.

MQMONITOR를 사용하지 않고 CKTI의 인스턴스를 시작하는 경우 CKTI 트랜잭션과 연관된 사용자 ID는 CKTI를 시작하는 트랜잭션의 사용자 ID입니다.

- PLTPI 프로그램의 경우 이는 **PLTPIUSR** 시스템 초기화 매개변수입니다.
- CKQC 사용자의 경우 이는 트랜잭션을 실행하는 사인온된 사용자 ID입니다.
- 순차 터미널을 사용하여 CKQC를 실행하는 경우 이는 순차 터미널을 실행하는 데 사용되는 사용자 ID입니다. 이 사용자 ID는 CICS 기본 사용자 ID의 기본값이 아닌 사전 설정된 사용자 ID여야 합니다. [DFHTCT TYPE=TERMINAL](#) 매크로의 내용을 참조하십시오.

MQCONN 및 MQMONITOR 자원을 위한 명령 보안

CICS 명령 보안을 사용하여 사용자가 MQCONN과 MQMONITOR 자원 정의에 대해 SPI 명령을 실행할 수 있는지 여부를 제어합니다. 예를 들면, 이를 사용하여 CICS 리전의 MQCONN 자원 정의에 대해 CREATE와 DISCARD 명령을 실행하도록 허용되는 사용자를 제어할 수 있습니다.

트랜잭션에 대해 명령 보안이 사용되는 경우 외부 보안 관리자는 트랜잭션과 연관된 사용자 ID가 MQCONN 또는 MQMONITOR 자원에 적절하게 명령을 사용할 권한이 있는지 확인합니다. MQCONN과 MQMONITOR 자원에 대해서는 자원 보안을 사용할 수 없습니다.

CICS 명령 보안은 **EXEC CICS CREATE MQCONN, DISCARD MQCONN, SET MQCONN, INQUIRE MQCONN, CREATE MQMONITOR, DISCARD MQMONITOR, SET MQMONITOR** 및 **INQUIRE MQMONITOR** 명령을 다룹니다. 명령 보안에 대한 설명과 CICS 리전에 대한 명령 보안을 설정하기 위한 지시사항은 [CICS 명령 보안의 내용](#)을 참조하십시오. 각 명령에 필요한 권한 레벨의 목록은 [자원 및 명령 검사 상호 참조](#)의 내용을 참조하십시오.

명령 보안이 활성화된 경우 **EXEC CICS SET MQCONN** 명령을 실행하여 IBM MQ에 대한 연결을 시작하는 트랜잭션을 실행하는 사용자 ID는 다음 권한을 가지고 있어야 합니다.

1. **EXEC CICS SET MQCONN** 명령을 사용할 수 있는 권한. 이 권한이 없으면 연결 시작은 NOTAUTH 응답과 RESP2 100과 함께 실패합니다.
2. **EXEC CICS EXTRACT EXIT** 명령을 사용할 수 있는 권한. 이 권한이 없으면 연결 시작은 INVREQ 응답과 RESP2 9와 함께 실패합니다. 이 경우 CICS는 DFHXS1111 및 DFHMQ0302 메시지를 발행합니다.

또한 MQMONITOR를 사용 중인 경우 MQMONITOR를 실행하는 사용자 ID(MQMONITOR 정의의 **MONUSERID** 매개변수에 지정됨)는 명령 보안에 대한 권한이 필요합니다. 이는 CICS-MQ 트리거 모니터, CICS-MQ 브릿지 또는 사용자 작성 MQMONITOR 프로그램을 제어하는 데 사용되는 MQMONITOR에 적용됩니다. MONUSERID는 다음 권한이 있어야 합니다.

1. MQMONITOR 상태를 STARTED 또는 STOPPED로 설정하기 위해 **EXEC CICS SET MQMONITOR** 명령을 사용할 수 있는 권한. 이 권한이 없으면 MQMONITOR 태스크가 실패하고 CICS-MQ 트리거 모니터의 경우 CICS가 DFHMQ0125 메시지를 발행합니다.

2. CICS-MQ 트리거 모니터의 경우 **TRANSID** 옵션이 트리거 메시지에 지정된 트랜잭션으로 설정된 **EXEC CICS START** 명령을 사용할 수 있는 권한. 이 권한이 없으면 CICS는 DFHMQ0102 메시지를 발행하고 트리거 메시지를 데드-레터 큐로 보냅니다.

MQMONITOR 자원을 위한 대리 사용자 보안

CICS 대리 사용자 보안을 사용하여 CICS-MQ 트리거 모니터를 시작할 수 있는 트랜잭션을 제어합니다. 이는 트리거 모니터 인스턴스가 MQMONITOR에 의해 시작된 경우에만 적용됩니다.

보안 검사가 활성화되고 시스템 초기화 매개변수로 **XUSER=YES**가 지정된 경우, CICS®는 USERID 옵션과 함께 **EXEC CICS START** 명령을 사용하여 트랜잭션이 시작될 때 대리 사용자 검사를 수행합니다. CICS-MQ 트리거 모니터가 MQMONITOR를 사용하여 시작된 경우 MQMONITOR 정의의 정보를 사용하여 사용자 트랜잭션을 시작합니다. 트리거 모니터는 MQMONITOR의 **USERID** 속성에서 얻은 값으로 지정된 USERID 옵션과 함께 **EXEC CICS START** 명령을 실행합니다.

CICS에서는 **START** 요청을 발행하는 트랜잭션과 연관된 사용자 ID가 시작 트랜잭션과 연관된 사용자 ID의 대리 사용자여야 합니다. MQMONITOR로 시작된 경우 CICS-MQ 트리거 모니터는 항상 MQMONITOR의 **MONUSERID** 속성에 지정된 사용자 ID로 실행됩니다. 따라서 CKTI가 트리거 메시지에 지정된 사용자 트랜잭션을 시작할 수 있도록 MONUSERID는 시작 사용자 태스크와 연관된 사용자 ID의 대리 사용자여야 합니다. 다른 소스에서 사용 가능한 적절한 사용자 ID가 없는 경우 기본적으로 MQMONITOR 정의에 지정된 USERID를 사용하여 사용자 트랜잭션을 시작하므로 MONUSERID는 USERID의 대리 사용자이기도 해야 합니다. 대리 보안 검사가 실패하면 CICS는 DFHMQ0102 메시지를 발행하고 트리거 메시지를 데드-레터 큐로 보냅니다.

CICS-MQ 어댑터를 위한 IBM MQ 연결 보안

IBM MQ는 애플리케이션 프로그램이 MQCONN 또는 MQCONNX 요청을 실행하여 큐 관리자에 연결하려고 시도하거나 채널 이니시에이터 또는 CICS-MQ 어댑터가 연결 요청을 실행할 때 연결 보안 검사를 실행합니다.

큐 관리자 레벨 보안을 사용 중인 경우 특정 큐 관리자에 대한 연결 보안 검사를 끌 수 있지만 그렇게 하면 모든 사용자가 해당 큐 관리자에 연결할 수 있습니다.

개별 CICS 터미널 사용자 ID가 아닌 CICS 주소 공간 사용자 ID만 연결 보안 검사에 사용됩니다.

큐 관리자 또는 큐 공유 그룹 레벨에서 IBM MQ 연결 보안 검사를 켜거나 끌 수 있습니다.

제 6 장 CICS-MQ 브릿지를 위한 보안

CICS-MQ 브릿지를 시작할 때 인증 레벨을 지정할 수 있습니다. 요청된 경우, 브릿지 모니터는 요청 메시지에서 이름 지정된 CICS 프로그램을 실행하기 전에 IBM MQ 요청 메시지에서 추출된 사용자 ID와 비밀번호를 검사합니다.

CICS-MQ 브릿지 모니터 트랜잭션(예: CKBR 또는 사용자의 트랜잭션 이름)을 실행할 때 **AUTH** 매개변수를 지정하여 다음 인증 레벨 중 하나를 선택할 수 있습니다.

LOCAL

이 레벨은 기본값입니다. 브릿지 모니터가 CICS 기본 사용자 ID로 브릿지 태스크를 시작합니다. 브릿지 태스크가 실행하는 CICS 사용자 프로그램은 이 사용자 ID와 연관된 권한을 가집니다. 메시지의 사용자 ID나 비밀번호가 무시되므로 IBM MQ 요청 메시지에서 상위 권한을 요청할 수 없습니다. 브릿지 태스크가 보호 자원에 액세스하려고 하는 CICS 프로그램을 실행하면, CICS 프로그램이 실패할 수 있습니다.

IDENTIFY

요청 메시지의 메시지 디스크립터(MQMD)에서 사용자 ID를 지정하면 브릿지 모니터가 해당 사용자 ID로 브릿지 태스크를 시작합니다. 브릿지 태스크가 실행하는 CICS 사용자 프로그램은 해당 사용자 ID와 연관된 권한을 가집니다. 사용자 ID는 신뢰할 수 있는 것으로 처리됩니다. 즉, 브릿지 모니터는 비밀번호 또는 PassTicket 정보를 사용하여 ID를 인증하지 않습니다. MQMD에서 사용자 ID를 지정하지 않으면 브릿지 모니터는 LOCAL 옵션과 동일한 방식으로 CICS 기본 사용자 ID로 브릿지 태스크를 시작합니다.

VERIFY_UOW

다음 모든 조건에 해당되는 경우 브릿지 모니터가 비밀번호 또는 PassTicket을 사용하여 사용자 ID를 인증합니다.

- 요청 메시지의 메시지 디스크립터(MQMD)에서 사용자 ID를 지정합니다.
- 요청 메시지에 IBM MQ CICS 정보 헤더(MQCIH)가 포함됩니다.
- MQCIH의 인증자 필드에 비밀번호 또는 PassTicket이 포함됩니다.

인증에 성공하면 브릿지 모니터가 해당 사용자 ID로 브릿지 태스크를 시작합니다. 인증에 실패하면 브릿지 모니터는 MQCRC_SECURITY_ERROR 리턴 코드와 함께 요청이 실패합니다.

이전에 나열된 조건 중 하나라도 충족되지 않으면 브릿지 모니터는 LOCAL 옵션과 동일한 방식으로 CICS 기본 사용자 ID로 브릿지 태스크를 시작합니다. 작업 단위에서 첫 번째 요청 메시지만 검사됩니다. 브릿지는 동일한 작업 단위의 일부인 후속 메시지에서 사용자 ID와 비밀번호 또는 PassTicket 정보를 무시합니다.

VERIFY_ALL

이 레벨은 브릿지 태스크가 동일한 작업 단위에 있는 모든 요청 메시지에서 사용자 ID가 동일한지 검사하는 것을 제외하고는 VERIFY_UOW와 동일하며, 메시지에 포함된 비밀번호 또는 PassTicket을 사용하여 각 요청 메시지에 대한 사용자 ID를 다시 인증합니다.

다른 애플리케이션에 다른 레벨의 인증이 필요한 경우 다른 트랜잭션 ID가 있는 다중 브릿지 모니터를 사용하십시오. CICS 대리 보안을 사용하여 브릿지 모니터 트랜잭션 및 사용자 ID가 시작할 수 있는 트랜잭션과 사용자 ID의 조합을 제한할 수 있습니다.

69 페이지의 표 9에서는 브릿지 모니터를 시작하는 사용자 ID를 표시합니다. 사용자 ID는 브릿지 모니터 트랜잭션(일반적으로 CKBR)을 실행하는 데 사용하는 방법에 따라 다릅니다.

표 9. CICS-MQ 브릿지 모니터 보안		
브릿지 모니터 시작 방법	사인온한 터미널에서	브릿지 모니터를 위한 사용자 ID
터미널 또는 프로그램의 EXEC CICS LINK 에서	예	사인온한 사용자 ID
터미널 또는 프로그램의 EXEC CICS LINK 에서	아니오	CICS 기본 사용자 ID
EXEC CICS START (사용자 ID 사용)	-	START의 사용자 ID
EXEC CICS START (사용자 ID 없음)	-	START의 사용자 ID

표 9. CICS-MQ 브릿지 모니터 보안 (계속)		
브릿지 모니터 시작 방법	사인온한 터미널에서	브릿지 모니터를 위한 사용자 ID
CICS-MQ 트리거 모니터 CKTI	-	START의 사용자 ID
CICS MQ 모니터(MQMONITOR)	-	<ul style="list-style-type: none"> CICS 리전에 대해 보안 검사가 활성화된 경우 (즉, SEC 시스템 초기화 매개변수가 YES로 설정됨) MQMONITOR 자원의 MONUSERID 속성 CICS 리전에 대해 보안 검사가 사용되지 않는 경우(즉, SEC가 NO로 설정됨) MQMONITOR 자원을 시작한 사용자 ID

요청 메시지의 사용자 ID 및 비밀번호

IDENTIFY, VERIFY_UOW 또는 VERIFY_ALL 인증 옵션을 사용하는 경우 브릿지 태스크 및 실행되는 CICS 프로그램은 요청 메시지의 메시지 디스크립터(MQMD)에 지정된 사용자 ID로 시작됩니다. 또한 VERIFY_UOW 및 VERIFY_ALL 옵션을 사용하면 브릿지 모니터가 요청 메시지의 IBM MQ CICS 정보 헤더(MQCIH)에 지정된 비밀번호를 검사합니다.

이러한 레벨의 인증을 사용하려면 IBM MQ 애플리케이션이 MQMD에서 사용자 ID를 제공해야 하고, 비밀번호가 포함된 MQCIH를 제공해야 합니다. RACF®에 이러한 사용자 ID를 정의해야 합니다. 사용되는 사용자 ID를 제어하려면 IBM MQ 애플리케이션이 MQOO_SET_IDENTITY_CONTEXT가 포함된 열기 옵션을 사용하여 브릿지 모니터에 대한 요청 큐를 열어야 합니다. 또한 애플리케이션은 메시지 놓기 옵션에 MQPMO_SET_IDENTITY_CONTEXT 값도 포함해야 합니다.

브릿지 모니터가 요청 메시지의 사용자 ID와 비밀번호에서 문제를 발견한 경우 다음과 같이 작동합니다.

- IDENTIFY 인증 레벨의 경우 메시지에 취소된 사용자 ID가 있으면 AICO 이상 종료가 발생할 수 있습니다. 오류 응답이 리턴 코드 MQCRC_BRIDGE_ERROR 및 이유 MQFB_CICS_BRIDGE_FAILURE를 표시합니다.
- VERIFY_UOW 또는 VERIFY_ALL 인증 레벨의 경우 사용자 ID나 비밀번호가 올바르지 않으면 요청이 리턴 코드 MQCRC_SECURITY_ERROR와 함께 실패합니다.
- 요청 메시지에서 사용자 ID나 비밀번호를 생략하면 브릿지 모니터는 다른 인증 옵션 중 하나로 브릿지 모니터를 시작했어도 LOCAL 인증 레벨을 사용하여 브릿지 태스크를 실행합니다. 이러한 상황에서 브릿지 태스크에 의해 시작된 CICS 프로그램은 브릿지 모니터를 시작한 사용자 ID로 실행됩니다.

메시지에서 비밀번호가 플로우되는 것을 방지하기 위해 비밀번호 대신 PassTicket을 사용할 수 있습니다.

- IBM MQ 애플리케이션은 요청 메시지의 MQCIH에 PassTicket을 제공해야 합니다.
- PassTicket을 생성하려면 애플리케이션 ID는 필수입니다. 기본 애플리케이션 ID는 CICS APPLID입니다. CICS-MQ 브릿지 모니터 트랜잭션(예: CKBR 또는 사용자의 트랜잭션 이름)을 시작할 때 **PASSTKTA** 매개변수를 사용하여 대체 애플리케이션 ID를 지정할 수 있습니다.
- 동일한 요청 큐를 위해 여러 개의 브릿지 모니터를 사용하는 경우 **PASSTKTA** 매개변수를 사용하여 각 브릿지 모니터에 동일한 애플리케이션 ID를 지정해야 합니다.

CICS 서비스가 APPLID를 지정할 수 있도록 허용하지 않으므로, PassTicket 유효성 검증은 **EXEC CICS VERIFY**가 아니라 IBM MQ 서비스를 사용하여 수행됩니다. PassTickets에 대한 자세한 정보는 [보안 사인온에 대한 PassTickets 생성 및 사용 및 IBM MQ 제품 문서의 z/OS에 대한 보안 설정의 내용을 참조하십시오.](#)

권한

CICS-MQ 브릿지에서 사용하는 사용자 ID에 다음 권한을 제공해야 합니다. 사용자 ID는 브릿지 모니터 트랜잭션을 시작하는 사용자 ID(69 페이지의 표 9에 나열됨) 및 IBM MQ 애플리케이션이 요청 메시지에서 지정한 사용자 ID를 포함합니다.

- 브릿지 모니터 트랜잭션을 시작하는 사용자 ID는 CICS DPL 프로그램을 위해 CKBP과 CKBC 트랜잭션을 시작할 권한과 IBM MQ 애플리케이션이 요청 메시지의 MQCIH 구조에 있는 TransactionId 필드에 지정한 대체 트랜잭션을 시작할 권한을 가지고 있어야 합니다.
- IBM MQ 애플리케이션이 요청 메시지에서 사용자 ID를 지정하는 경우 브릿지 모니터 트랜잭션을 시작하는 사용자 ID는 요청 메시지에 사용되는 모든 사용자 ID의 대리로 RACF에 정의되어야 합니다. 대리 사용자는 다른 사용자의 비밀번호 없이 다른 사용자 대신 작업을 시작할 권한이 있는 사용자입니다. 대리 사용자 보안에 대한 자세한 정보는 [대리 사용자 보안](#)의 내용을 참조하십시오.
- 브릿지 모니터 및 모든 브릿지 태스크를 위한 사용자 ID는 요청 큐에서 메시지를 가져올 권한이 필요합니다.
- 브릿지 태스크를 위한 사용자 ID는 리플라이-투 큐에 메시지를 넣을 권한이 필요합니다.
- 오류 응답이 수신되도록 하려면 브릿지 모니터 트랜잭션을 시작하는 사용자 ID가 모든 리플라이-투 큐에 메시지를 넣을 권한을 가지고 있어야 합니다.
- 브릿지 태스크를 위한 사용자 ID는 데드-레터 큐에 메시지를 넣을 권한을 가지고 있어야 합니다.
- 오류 발생 시 브릿지가 중지되지 않도록 하려면 브릿지 모니터 트랜잭션을 시작하는 사용자 ID는 데드-레터 큐에 메시지를 넣을 권한이 필요합니다.
- 브릿지 모니터 및 모든 브릿지 태스크를 위한 사용자 ID는 백아웃 리큐 큐(정의된 경우)에 메시지를 넣을 권한을 가지고 있어야 합니다.

제 7 장 CICS-MQ 브릿지를 사용하기 위한 애플리케이션 개발

비CICS 애플리케이션은 CICS-MQ 브릿지를 통해 IBM MQ 메시지를 송수신하여 CICS 프로그램 또는 트랜잭션과 통신할 수 있습니다. CICS 애플리케이션에 필요한 데이터는 요청 메시지에 포함되며 CICS-MQ 브릿지는 응답 메시지를 사용하여 CICS 애플리케이션이 제공한 데이터를 리턴합니다.

다음 유형의 CICS 애플리케이션은 CICS-MQ 브릿지와 함께 사용하기에 적합합니다.

- DPL 프로그램으로 알려진 **EXEC CICS LINK** 명령을 사용하여 호출된 CICS 프로그램. 프로그램은 CICS API의 DPL 서브세트를 준수해야 합니다. 즉, CICS 터미널 또는 동기점 기능을 사용해서는 안 됩니다. CICS-MQ 브릿지를 사용하여 단일 CICS 프로그램 또는 작업 단위를 구성하는 CICS 프로그램 세트를 실행할 수 있습니다.
- 3270 트랜잭션으로 알려지도록 3270 터미널에서 실행되도록 디자인된 CICS 트랜잭션. 트랜잭션은 BMS(Basic Mapping Support) 또는 터미널 제어 명령을 사용할 수 있습니다. 대화식 또는 의사 대화의 일부일 수 있습니다. 동기점을 발행할 수 있습니다.

일반적으로 비CICS 애플리케이션이 CICS 트랜잭션에서 제어의 내부 로직 및 플로우와 상호작용해야 하므로 CICS-MQ 브릿지를 통해 3270 트랜잭션을 실행하려면 좀 더 복잡한 애플리케이션 프로그래밍이 필요합니다. CICS 애플리케이션의 비즈니스 로직을 포함하는 DPL 프로그램을 실행하는 것이 바람직합니다. 그러나 일부 CICS 애플리케이션은 표시 로직과 다른 애플리케이션의 비즈니스 로직으로 구성되지 않으므로 CICS-MQ 브릿지를 사용하여 애플리케이션의 유형과 통신할 수 있습니다.

비CICS 애플리케이션은 구조화된 IBM MQ 메시지를 CICS-MQ 브릿지에 대한 요청 큐에 전송하여 CICS 애플리케이션을 시작합니다. CICS 애플리케이션에 필요한 모든 데이터는 요청 메시지에 포함될 수 있습니다.

- DPL 프로그램의 경우 필요한 데이터는 CICS 애플리케이션에서 사용하는 CICS COMMAREA(COMMunication AREA) 데이터입니다. CICS 애플리케이션에 COMMAREA 데이터가 필요하지 않은 경우 메시지 데이터는 DPL 프로그램의 이름으로만 구성됩니다.
- 3270 트랜잭션의 경우 필요한 데이터는 CICS 애플리케이션에서 사용하는 애플리케이션 데이터 구조(ADS)를 설명하는 벡터를 포함합니다.

요청 메시지에는 일반적으로 CICS 애플리케이션에 대한 제어 옵션을 제공하는 IBM MQ CICS 정보 헤더(MQCIH 구조)도 포함됩니다. 기본 트랜잭션 코드를 사용하여 단일 DPL 프로그램을 실행 중이고 프로그램에 권한이 필요하지 않은 경우 MQCIH가 필요하지 않습니다. 다른 모든 경우 및 모든 3270 트랜잭션에는 MQCIH가 필요합니다.

CICS-MQ 브릿지는 출력 CICS 애플리케이션에서 데이터를 비CICS 애플리케이션으로 반송하는 방식으로 생성된 데이터를 응답 큐로 전송되는 IBM MQ 메시지에서 가져옵니다.

- DPL 프로그램의 경우 반송된 데이터는 CICS 애플리케이션에 의해 출력으로 생성된 COMMAREA 데이터입니다.
- 3270 트랜잭션의 경우 반송된 데이터는 CICS 애플리케이션이 출력으로 생성한 애플리케이션 데이터 구조(ADS)를 설명하는 벡터를 포함합니다. 비CICS 애플리케이션은 이러한 벡터를 해석하고 응답해야 합니다.

단일 또는 다중 DPL 프로그램 또는 3270 트랜잭션을 실행하기 위해 CICS-MQ 브릿지가 IBM MQ 메시지를 처리하는 방법에 대한 설명은 [CICS-WebSphere MQ 브릿지에서 CICS DPL 프로그램이 실행되는 방식](#) 및 [CICS-WebSphere MQ 브릿지에서 CICS 3270 트랜잭션이 실행되는 방식](#)의 내용을 참조하십시오.

3270 트랜잭션을 실행하려는 경우 CICS 트랜잭션의 로직을 분석하는 데 도움이 되는 CICS SupportPac CA1E CICS 3270 Bridge Passthrough의 사본을 얻는 것이 도움이 될 수 있습니다. SupportPac을 사용하면 인바운드 및 아웃바운드 데이터 플로우를 표시 및 캡처하고 메시지 구성 방법과 MQCIH 및 벡터의 필드에 삽입할 값을 평가할 수 있습니다.

CICS-MQ 브릿지의 DPL 메시지 구조

이 예제는 비CICS 애플리케이션에서 요청 메시지에 CICS-MQ 브릿지를 통해 CICS DPL 프로그램을 실행하는 데 사용할 수 있는 다양한 구조를 보여줍니다.

가장 간단한 경우, 메시지 데이터는 실행할 DPL 프로그램의 이름으로만 구성됩니다. DPL 프로그램(사용자 프로그램)이 시작될 때 데이터를 사용할 수 있게 하려면 COMMAREA 데이터 또는 DFHREQUEST 컨테이너 데이터로 이를 수행하십시오.

작업 단위로 둘 이상의 DPL 프로그램을 실행하려는 경우, 특정 트랜잭션 코드(기본 CKBP 겹쳐쓰기)를 선호하는 경우, DPL 프로그램을 실행하거나 채널 및 컨테이너를 활용하려는 경우 특정 레벨의 권한이 필요한 경우(CKBC 또는 동등한 트랜잭션을 사용하여 프로그램 DFHMQB3을 실행), MQCIH에 정보를 제공해야 합니다. MQCIH는 사용자가 전송한 DPL 프로그램 이름 및 DPL 프로그램 데이터 앞에 와야 합니다.

- 비CICS 애플리케이션이 기본 처리 옵션을 사용하여 단일 CICS DPL 프로그램을 실행하고 DPL 프로그램 데이터를 전송하거나 수신하지 않을 때 이 구조를 사용하십시오.

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

ProgName 에 의해 지정된 프로그램은 CICS에 의해 DPL 프로그램으로 호출됩니다.

- 비CICS 애플리케이션이 기본 처리 옵션을 사용하여 단일 DPL 프로그램을 실행하고 COMMAREA 데이터를 송수신할 때 이 구조를 사용하십시오.

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

- 비CICS 애플리케이션이 작업 단위로 하나 이상의 DPL 프로그램을 실행하거나 프로그램을 실행하기 위해 특정 권한이 필요하지만 DPL 프로그램 데이터를 전송하거나 수신하지 않는 경우 이 구조를 사용하십시오.

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

- 비CICS 애플리케이션이 작업 단위로 하나 이상의 DPL 프로그램을 실행하거나 프로그램을 실행하기 위한 특정 권한이 필요하고 COMMAREA 데이터를 송수신하고 DFHREQUEST 컨테이너 데이터를 보내고 DFHRESPONSE 컨테이너 데이터를 수신할 때 이 구조를 사용하십시오.

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

참고: CICS-MQ 브릿지에 COMMAREA 데이터가 없고 프로그램 이름만 보내는 경우 프로그램 이름은 8자여야 합니다. 이 이름은 오른쪽에 공백으로 패딩된 이름이 아니어야 합니다. CICS-MQ 브릿지일 경우 COMMAREA 음의 길이 오류가 보고됩니다. COMMAREA 데이터 또는 DFHREQUEST 컨테이너 데이터를 보낼 때 프로그램 이름을 공백으로 오른쪽에 채워서 총 8자를 제공해야 합니다.

선택적으로 형식 이름이 MQH로 시작하고 표준 링크 필드를 포함하는 추가 헤더가 MQCIH 헤더보다 우선할 수 있습니다. CICS-MQ 브릿지는 헤더의 데이터를 사용하지 않으므로 이러한 헤더는 출력 메시지에서 수정되지 않은 상태로 리턴됩니다.

응답 메시지 구조

DPL 프로그램을 실행하는 CICS-MQ 브릿지 태스크가 비정상적으로 종료되면 실패 이전의 인바운드 메시지에 MQCIH가 포함되어 있는지 여부에 관계없이 다음 구조로 응답 큐에 메시지를 리턴합니다.

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

DFHMQ* message 는 오류 유형을 표시하는 오류 메시지를 표시합니다. MQCIH.Format 필드의 값은 MQFMT_STRING으로 설정되므로 최종 대상이 다른 CCSID 및 인코딩을 사용하는 경우 메시지가 올바르게 변환될 수 있습니다. 또한 MQCIH에는 문제점을 진단하는 데 사용할 수 있는 다른 필드가 포함됩니다.

예제: CICS-MQ 브릿지를 통한 DPL 프로그램에 대한 요청 메시지

이 C 언어 코드 단편은 메시지 버퍼를 구성하고 IBM MQ CICS 정보 헤더(MQCIH)를 포함하여 CICS-MQ 브릿지를 통한 COMMAREA 데이터를 사용하여 CICS DPL 프로그램을 호출하는 방법을 보여줍니다.

호출되는 DPL 프로그램은 DPL 서브세트 규칙을 준수해야 합니다. 자세한 정보는 [CICS DPL에 대한 애플리케이션 프로그래밍](#)의 내용을 참조하십시오.

```
/* #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 브릿지의 3270 트랜잭션 메시지 구조

이 절의 예제는 비CICS 애플리케이션이 CICS-MQ 브릿지를 통해 CICS 3270 트랜잭션을 실행할 때 요청 및 응답 메시지의 구조를 보여줍니다. CICS-MQ 브릿지 벡터는 트랜잭션에서 EXEC CICS 명령을 표시하도록 메시지에 사용됩니다.

CICS-MQ 브릿지는 터미널 사용자 및 발행 애플레이트와 EXEC CICS 명령 수신을 포함하여 CICS 트랜잭션을 시작하는 다양한 방법을 애플레이트할 수 있습니다. IBM MQ를 사용하지 않으면 다음과 같은 여러 가지 방법으로 CICS 트랜잭션을 시작할 수 있습니다.

- 터미널 사용자는 트랜잭션 이름을 입력한 다음 데이터를 선택적으로 입력할 수 있습니다. 트랜잭션은 시작될 때 EXEC CICS RECEIVE 명령을 실행하여 해당 ID 뒤에 오는 데이터도 획득할 수 있습니다.
- 터미널에 있는 선행 트랜잭션은 EXEC CICS RETURN TRANSID(*transid*)로 중지됩니다. 터미널은 3270 데이터 스트림을 전송하고 새 트랜잭션을 시작합니다. 이러한 방식으로 시작된 트랜잭션은 BMS(Basic Mapping Support) 맵핑을 사용하는지 또는 터미널 제어를 사용하는지에 따라 EXEC CICS RECEIVE MAP 또는 EXEC CICS RECEIVE 명령을 발행하여 3270 데이터 스트림의 데이터를 얻습니다.
- 애플리케이션은 EXEC CICS START 명령을 발행합니다. 시작된 트랜잭션은 START 명령에 지정된 데이터를 검색하기 위해 EXEC CICS RETRIEVE 명령을 발행합니다.

터미널에서 시작된 트랜잭션은 다음에 터미널 사용자와의 대화 또는 의사 대화에서 EXEC CICS CONVERSE, EXEC CICS SEND MAP 및 EXEC CICS RECEIVE MAP과 같은 명령을 발행할 수 있습니다.

CICS-MQ 브릿지는 이러한 CICS 트랜잭션을 시작하는 방법을 에뮬레이트할 수 있습니다. 또한 트랜잭션에서 데이터 화면을 보내고 받는 터미널 사용자를 에뮬레이션할 수 있습니다. 이러한 에뮬레이션은 에뮬레이션되는 EXEC CICS 명령을 표시하며 필요한 데이터를 제공하는 CICS-MQ 브릿지 벡터를 사용하여 수행됩니다. CICS 트랜잭션에 필요한 데이터는 인바운드 메시지와 함께 제공되며 CICS-MQ 브릿지 애플리케이션에 필요한 데이터는 아웃바운드 메시지와 함께 제공됩니다.

CICS-MQ 브릿지를 사용하는 3270 트랜잭션의 벡터

CICS-MQ 브릿지는 최소 기능 BMS를 포함하여 CICS 터미널 API의 모든 기능을 에뮬레이트합니다. 3270 트랜잭션을 실행하기 위해 CICS-MQ 브릿지를 사용하는 경우, 벡터를 사용하여 요청 및 응답 메시지에서 **EXEC CICS** 명령을 표시합니다.

브릿지 메시지에서 벡터는 벡터 디스크립터로 알려진 숫자 문자열로 식별됩니다. 각 벡터 디스크립터는 이를 표시하는 EXEC CICS 명령의 CICS EIBFN 값입니다. 예를 들면, 0402는 **EXEC CICS RECEIVE**의 EIBFN 값이며 RECEIVE 벡터의 벡터 디스크립터이기도 합니다. 벡터는 문자 I 및 O에 의해 추가로 규정되어 인바운드(CICS-MQ 브릿지로) 또는 아웃바운드(CICS-MQ 브릿지에서) 여부를 표시합니다. 메시지에는 여러 개의 연결된 벡터가 포함될 수 있습니다.

3270 트랜잭션을 실행하기 위해 CICS-MQ 브릿지로 전송되는 인바운드 메시지에는 데이터 없이 트랜잭션을 시작하지 않는 한 MQCIH 뒤에 벡터 구조가 포함되어야 합니다. 인바운드 메시지는 CICS 트랜잭션의 향후 RECEIVE MAP 요청을 예상하여 여러 개의 RECEIVE MAP 벡터를 포함할 수 있습니다. 애플리케이션은 메시지를 구성할 수 있도록 트랜잭션의 제어 플로우를 알아야 합니다.

아웃바운드 메시지에는 요청 벡터 또는 응답 벡터가 포함될 수 있습니다. 이러한 설명이 요청 큐 또는 응답 큐로 이동한다는 의미는 아닙니다. 모든 아웃바운드 메시지는 응답 큐로 이동합니다. CICS 트랜잭션은 요청 벡터를 사용하여 가상 터미널 역할을 하는 비CICS 애플리케이션에서 데이터를 요청합니다. CICS 트랜잭션은 데이터가 다시 돌아올 것으로 예상하지 않을 때 응답 벡터를 사용합니다. 인바운드 벡터에 대해서는 그러한 구분이 없습니다.

아웃바운드 메시지에는 여러 개의 벡터가 포함될 수 있습니다(예: 트랜잭션에 의해 실행되는 연속적인 **EXEC CICS SEND MAP** 명령의 결과). CICS 트랜잭션은 아웃바운드 메시지에 단일 벡터 또는 다중 벡터가 포함되는지 여부를 제어하지 않습니다. 트랜잭션이 요청 벡터를 생성하는 명령을 발행하면 요청 벡터는 항상 메시지의 마지막 벡터입니다.

다음 벡터가 사용 가능합니다. 각각이 나타내는 CICS 명령을 가져오려면 EXEC CICS를 벡터 이름의 접두부로 지정하십시오.

인바운드 벡터:

- RECEIVE
- RECEIVE MAP
- CONVERSE
- RETRIEVE

아웃바운드 응답 벡터(다음 인바운드 메시지에 추가 데이터가 필요하지 않음):

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

아웃바운드 요청 벡터(다음 인바운드 메시지에 추가 데이터가 필요함):

- RECEIVE
- RECEIVE MAP
- CONVERSE

이러한 벡터 각각은 가변 길이 데이터가 뒤에 오는 아키텍처 구조입니다. 모든 벡터에는 공통 헤더가 있지만 구조는 다릅니다. 구조에 대한 정보는 [Link3270 메시지 형식](#)의 내용을 참조하십시오.

벡터 구조 정의는 C 언어 헤더 파일 `dfhbrmqh.h` 및 COBOL 카피북 `DFHBRMQ0`에서 사용 가능합니다. 브릿지를 사용하는 애플리케이션에도 포함시키십시오. 이러한 멤버는 z/OS의 CICS Transaction Server에만 제공됩니다. 다른 플랫폼에 애플리케이션을 작성하려는 경우, 해당 환경으로 복사하십시오.

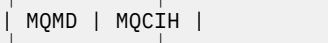
RETRIEVE 벡터

CICS 3270 트랜잭션이 해당 시작 데이터를 검색하기 위해 **EXEC CICS RETRIEVE ... QUEUE(data-area)** 명령을 발행하는 경우, 비CICS 애플리케이션은 RETRIEVE 벡터 구조(C에서 `brmq_retrieve`로 정의됨)를 사용하여 CICS-MQ 브릿지에 메시지를 보내야 합니다. 구조에는 길이가 8바이트인 문자 필드가 포함되며, 여기서 비CICS 애플리케이션은 검색할 데이터가 포함된 임시 저장영역 큐의 이름을 지정해야 합니다. RETRIEVE 벡터를 포함하는 메시지는 항상 대화 또는 의사 대화를 표시하는 교환에서 첫 번째입니다.

CICS-MQ 브릿지를 사용하는 3270 요청 메시지(인바운드)의 구조

이 예제는 비CICS 애플리케이션이 CICS-MQ 브릿지를 사용하여 인바운드 메시지라고 하는 3270 트랜잭션으로 전송된 요청 메시지의 가능한 구조를 표시합니다.

- 비CICS 애플리케이션이 데이터 없이 CICS 트랜잭션을 호출할 때 이 구조를 사용하십시오.



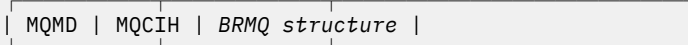
MQCIH.TransactionId 필드를 시작할 트랜잭션 이름으로 설정하십시오. MQCIH의 다른 필드를 애플리케이션에 적합한 값으로 설정하십시오.

- 비CICS 애플리케이션이 데이터를 사용할 수 있을 것으로 예상되는 EXEC CICS 명령을 실행하는 트랜잭션을 실행할 때 이 구조를 사용하십시오.



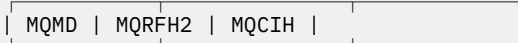
BRMQ structure 인바운드 벡터 구조 RECEIVE, RECEIVE MAP, CONVERSE 또는 RETRIEVE 중 하나를 표시합니다. BRMQ 구조 자체는 헤더와 벡터로 구성되며 이러한 벡터는 데이터를 포함할 수 있습니다.

- 길이가 0인 요청 메시지에 이 구조를 사용하십시오.



예를 들어, 인바운드 RECEIVE MAP 벡터는 사용자가 PF 키만 누른 조치를 표시할 수 있습니다. 이 경우 BRMQ 구조의 필드는 사용된 AID 키를 지정하지만 BRMQ 구조 뒤에는 데이터가 없습니다.

- 선택적으로 형식 이름이 MQH로 시작하고 표준 링크 필드를 포함하는 추가 헤더가 MQCIH 헤더보다 우선할 수 있습니다. CICS-MQ 브릿지는 헤더의 데이터를 사용하지 않으므로 이러한 헤더는 출력 메시지에서 수정되지 않은 상태로 리턴됩니다. MQCIH 앞에 헤더가 있는 요청 메시지에 이 구조를 사용하십시오.



CICS-MQ 브릿지를 사용하는 3270 응답 메시지(아웃바운드)의 구조

이 예제는 3270 트랜잭션 대신 CICS-MQ 브릿지에서 아웃바운드 메시지로 알려진 비CICS 애플리케이션으로 전송되는 응답 메시지의 가능한 구조를 보여줍니다.

CICS-MQ 브릿지의 아웃바운드 메시지는 오류 발생 여부에 따라 세 가지 구조 중 하나를 갖습니다. 이러한 각 예에는 단일 벡터만 표시되지만 오류가 발생한 경우를 제외하고 메시지에는 여러 개의 연결된 벡터가 포함될 수 있습니다.

- 이 구조는 CICS-MQ 브릿지 처리가 정상적으로 종료되고 오류가 발견되지 않고 비CICS 애플리케이션으로 데이터가 리턴되지 않을 때 사용됩니다.

```
| MQMD | MQCIH | BRMQ structure |
```

CICS 애플리케이션 이상 종료도 CICS-MQ 브릿지에 의해 정상적으로 완료된 것으로 간주됩니다. CICS 애플리케이션이 발행한 이상 종료 코드는 MQCIH에 제공됩니다.

- 이 구조는 CICS-MQ 브릿지 처리가 정상적으로 종료되고 오류가 발견되지 않고 비CICS 애플리케이션으로 데이터가 리턴될 때 사용됩니다.

```
| MQMD | MQCIH | BRMQ structure | data |
```

BRMQ structure 아키텍처 아웃바운드 응답 또는 요청 벡터 구조를 표시합니다.

- 이 구조는 CICS-MQ 브릿지 모니터가 오류를 감지한 후 CICS-MQ 처리가 비정상적으로 종료될 때 사용됩니다.

```
| MQMD | MQCIH | DFHMq* message |
```

DFHMq message* 는 오류 유형을 표시하는 오류 메시지를 표시합니다. 필드 MQCIH.Format의 값은 MQFMT_STRING으로 설정되므로 최종 대상이 다른 CCSID 및 인코딩을 사용하는 경우 메시지가 올바르게 변환될 수 있습니다. 또한 MQCIH에는 문제점을 진단하는 데 사용할 수 있는 다른 필드가 포함됩니다.

명시적 동기점 또는 롤백 요청을 실행하는 CICS 트랜잭션의 경우 CICS-MQ 브릿지는 응답 큐에 추가 메시지를 생성하여 MQCIH의 TaskEndStatus 필드에 동기점의 결과를 표시합니다. 이 추가 메시지는 MQMD MsgType의 MQMT_DATAGRAM과 함께 전송됩니다. 애플리케이션의 입력이 예상되지 않으며 추가 메시지 다음에 일반 태스크 종료 메시지가 옵니다.

예: CICS-MQ 브릿지를 통해 CEMT INQUIRE TASK를 발행하기 위한 요청 메시지

이 예는 애플리케이션이 CICS 터미널에서 해당 ID 및 일부 명령행 인수를 입력하여 시작되는 3270 트랜잭션(이 경우 CEMT)을 시작하도록 CICS-MQ 브릿지를 사용하는 방법을 보여줍니다.

CEMT 태스크가 시작되면 해당 ID 뒤에 오는 명령어 입력행 인수를 수신하도록 EXEC CICS RECEIVE 명령을 발행합니다. 명령행 호출을 에뮬레이트하는 비-CICS 애플리케이션은 벡터 구조에 적합한 값을 포함하는 RECEIVE 벡터로 CEMT를 시작하고 명령행 값도 포함해야 합니다.

요청 메시지 구성

다음 C 언어 코드 단편에서는 요청 메시지를 구성할 수 있는 방법을 보여줍니다. C 언어 헤더 파일 dfhbrmqh.h 는 CICS SAMPLIB 라이브러리에 있습니다.

```
/* #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 */
:
```

응답 큐로 리턴되는 아웃바운드 메시지에는 터미널 제어 형식의 데이터가 포함된 SEND 응답 벡터가 포함됩니다. 애플리케이션은 수신하는 데이터를 분석하기 위해 이 형식을 해석해야 합니다.

메시지 버퍼용 저장영역 할당

```
/* 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 설정

```
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 설정

```
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 설정

```
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) ;
```

메시지 빌드

```
/* 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 브릿지를 위해 MQMD 및 MQCIH 구조에서 설정해야 하는 필드

비CICS 애플리케이션은 CICS-MQ 브릿지에 대한 요청 메시지에서 MQCIH 및 MQMD 구조에 많은 필드를 설정해야 합니다.

CICS-MQ 브릿지 메시지에 MQMD 필드

CICS-MQ 브릿지의 오퍼레이션에 영향을 미칠 수 있는 MQMD의 필드는 애플리케이션 프로그램에서 초기화되어야 합니다.

MQMD.CorrelId

요청 큐에 대한 MQPUT 명령의 경우 작업 단위(UOW)에서 첫 번째 또는 유일한 메시지에서 값을 MQCI_NEW_SESSION으로 설정하십시오. 작업 단위(UOW)의 후속 메시지에서 첫 번째 메시지를 요청 큐에 넣을 때 메시지 디스크립터에 IBM MQ가 설정한 MQMD.MsgId로 값을 설정하십시오.

응답 큐에 대한 MQGET 명령의 경우 가장 최근 메시지를 요청 큐에 넣을 때 메시지 디스크립터에 IBM MQ가 설정한 MQMD.MsgId의 값을 사용하거나 MQCI_NONE을 지정하십시오. 83 페이지의 『DPL 프로그램의 MsgId, CorrelId 및 UOWControl 필드』 및 84 페이지의 『3270 트랜잭션에 대한 MsgId, CorrelId 및 UOWControl 필드』의 내용을 참조하십시오.

MQMD.Expiry

애플리케이션이 응답을 기다리는 시간을 기준으로 메시지 만료 시간을 설정합니다. 남은 만료 시간을 응답 메시지에 전파하도록 MQCIH 플래그를 설정하십시오.

MQMD.Format

DPL 프로그램의 경우, 브릿지 요청 큐에 보내는 메시지에 MQCIH를 포함하면 값을 MQCICS로 설정합니다. 그렇지 않으면 다음 데이터 형식으로 설정하십시오.

3270 트랜잭션의 경우, 값은 MQFMT_CICS이어야 합니다.

MQMD.MsgId

요청 큐에 대한 MQPUT 명령의 경우 MsgId를 작업 단위(UOW)의 고유 값 또는 MQMI_NONE으로 설정하십시오.

응답 큐에 대한 MQGET 명령의 경우 첫 번째 메시지를 요청 큐에 넣을 때 메시지 디스크립터에 IBM MQ가 설정한 MQMD.MsgId의 값을 사용하십시오. 83 페이지의 『DPL 프로그램의 MsgId, CorrelId 및 UOWControl 필드』 및 84 페이지의 『3270 트랜잭션에 대한 MsgId, CorrelId 및 UOWControl 필드』의 내용도 참조하십시오.

MQMD.Persistence

요청 큐에 넣은 메시지는 지속적이거나 비지속적일 수 있습니다. 응답 메시지에는 요청 메시지와 같은 지속성이 있습니다.

MQMD.ReplyToQ

브릿지가 응답 메시지를 전송할 큐 이름으로 값을 설정합니다.

MQMD.UserIdentifier

이 필드는 브릿지 모니터가 IDENTIFY, VERIFY_UOW 또는 VERIFY_ALL 권한 레벨로 실행 중인 경우에만 사용됩니다. 이 중 하나를 사용하는 경우 CICS DPL 프로그램에 대한 액세스에 확인되는 사용자 ID로 값을 설정하십시오. 이 필드를 VERIFY_* 옵션 중 하나와 함께 사용하는 경우 MQCIH.Authenticator 필드도 초기화해야 합니다. 사용자 ID와 연관된 비밀번호 또는 패스티켓 값으로 설정하십시오.

참고: IDENTIFY, VERIFY_UOW 또는 VERIFY_ALL의 권한 부여 레벨을 사용하는 경우 브릿지 요청 큐를 열 때 값 MQOO_SET_IDENTITY_CONTEXT를 열기 옵션에 추가하고 메시지를 큐에 전송할 때 값 MQPMO_SET_IDENTITY_CONTEXT를 메시지 넣기 옵션에 추가해야 합니다.

DPL 프로그램 요청 메시지용 MQCIH 필드

CICS-MQ 브릿지를 사용하여 CICS DPL 프로그램을 실행하려면 요청 메시지를 전송할 때, 애플리케이션 프로그램에서 이러한 주요한 입력 필드를 초기화해야 합니다.

MQCIH.Authenticator

VERIFY_UOW 또는 VERIFY_ALL의 권한 레벨을 사용 중인 경우에만 이 필드가 적용됩니다.

MQMD.UserIdentifier 필드에서 사용자 ID와 연관될 비밀번호 또는 패스티켓으로 값을 설정하십시오. 또한 이 값은 사용자가 DPL 프로그램에 링크할 권한이 있는지 여부를 판별하기 위해 외부 보안 관리자가 사용됩니다. 패스티켓을 사용하는 경우 패스티켓 생성에 사용된 applid는 브릿지 모니터를 시작할 때 사용되는 PASSTKA 키워드 값과 동일해야 합니다.

MQCIH.Flags

- 남은 만료 시간을 응답 메시지에 전달하려면 MQCIH_PASS_EXPIRATION으로 설정하십시오.
- 응답 메시지에서 후행 널 문자('00'X)를 제거하려면 MQCIH_REPLY_WITHOUT_nulls로 설정하십시오.
- EXEC CICS LINK 명령에 SYNCONRETURN 옵션을 지정하려면 MQCIH_SYNC_ON_RETURN으로 설정하십시오.

값을 함께 추가하여 결합할 수 있습니다.

MQCIH.Format

MQCIH 구조를 따르는 데이터의 형식을 지정합니다. 데이터가 문자 데이터인 경우, MQFMT_STRING을 사용하십시오. 변환이 필요하지 않는 경우 MQFMT_NONE을 사용하십시오.

MQCIH.GetWaitInterval

이 필드를 기본값으로 설정하면 작업 단위의 메시지에 대한 브릿지 태스크 GET WAIT 간격은 브릿지 모니터가 시작될 때 WAIT 매개변수에 지정된 값입니다. WAIT 매개변수도 기본값으로 설정하면 GET WAIT 간격은 무제한입니다.

MQCIH.LinkType

CICS DPL 브릿지를 사용 중인 경우 MQCLT_PROGRAM을 지정합니다.

MQCIH.OutputDataLength

이 필드는 CICS DPL 브릿지에만 적용되고, 프로그램에 의해 리턴되는 데이터의 길이를 설정합니다. MQCIH.OutputDataLength는 채널 및 컨테이너 인터페이스를 사용하는 DPL 요청에 대해서는 무시됩니다. 이러한 요청의 경우 출력(즉, 응답) 길이는 DFHRESPONSE 컨테이너의 크기입니다.

MQCIH.RemoteSysId

특정 CICS 시스템에서 요청을 처리해야 하는 경우가 아니면 이 필드는 공백으로 두십시오.

MQCIH.ReplyToFormat

애플리케이션과 브릿지가 동일한 CCSID 및 인코딩 환경에서 실행 중인 경우 이를 MQFMT_NONE(기본값)으로 설정하십시오. 그렇지 않으면, 값을 리턴된 COMMAREA 데이터의 형식으로 설정하십시오. MQCIH.replytoformat은 채널 및 컨테이너 인터페이스를 사용하는 DPL 요청에 대해 무시됩니다. 이러한 요청에 대한 reply-to 형식은 DFHRESPONSE 컨테이너의 콘텐츠를 기반으로 설정됩니다. 콘텐츠가 문자 데이터인 경우 reply-to 형식은 MQFMT_STRING입니다. 콘텐츠가 2진 데이터인 경우 reply-to 형식은 MQFMT_NONE입니다.

MQCIH.TransactionId

CICS DPL 브릿지 프로그램을 실행할 트랜잭션 코드를 지정합니다. 제공된 트랜잭션 중 하나를 지정할 수 있습니다.

- CKBP, COMMAREA를 사용하여 프로그램 DFHMQBP0 실행
- CKBC, 채널과 컨테이너를 사용하여 DFHMQBP3 프로그램 실행
- DFHMQBP0 또는 DFHMQBP3을 실행하기 위한 사용자 고유의 트랜잭션 코드

필드를 비워두면(4개의 공백) 제공된 트랜잭션 코드인 CKBP가 기본값입니다.

MQCIH.UOWControl

이 필드는 브릿지에 의해 수행된 작업 단위(UOW)를 제어합니다. 자세한 정보는 83 페이지의 『DPL 프로그램의 MsgId, CorrelId 및 UOWControl 필드』의 내용을 참조하십시오.

3270 트랜잭션 요청 메시지에 MQCIH 필드

CICS-MQ 브릿지를 사용하여 3270 트랜잭션을 실행하려면 요청 메시지를 전송할 때, 애플리케이션 프로그램에서 이러한 주요한 입력 필드를 초기화해야 합니다.

MQCIH.ADSDescriptor

이 필드는 BMS SEND MAP 및 RECEIVE MAP 호출을 사용하는 필드에 적용됩니다. 이 경우 CICS-MQ 브릿지 요청 메시지를 보내는 애플리케이션이 워크스테이션에 있는 경우 이 값을 MQCADSD_SEND + MQCADSD_RECV + MQCADSD_MSGFORMAT로 이 설정합니다. 이 설정은 요청 및 응답 메시지의 벡터가 워크스테이션과 메인프레임의 다른 CCSID 및 인코딩 스킴 간에 올바르게 변환되도록 합니다.

MQCIH.AttentionId

이 필드를 트랜잭션이 예상하는 AID 키를 표시하는 값으로 설정합니다(있는 경우). 그렇지 않으면 CICS 트랜잭션에 ENTER AID 키로 표시되는 기본값인 4개의 공백을 승인하십시오.

인바운드 RECEIVE, RECEIVE MAP 및 CONVERSE 벡터에도 AID 값을 지정할 수 있는 필드가 있습니다. MQCIH의 값은 애플리케이션이 시작될 때 EIBAID가 설정되는 값입니다. 트랜잭션을 시작하는 데 사용되는 기능 키를 표시합니다. 인바운드 벡터의 값은 데이터가 입력될 때 사용된 값입니다. 예를 들면, 이는 EXEC CICS RECEIVE MAP 명령어 뒤의 EIBAID 값입니다.

참고:

1. 대화식 트랜잭션의 경우 초기 MQCIH 값과 벡터의 값은 별도의 값을 갖습니다.
 2. 비-CICS 애플리케이션이 요청 벡터에 대한 응답으로 메시지를 보내는 경우 MQCIH의 값은 무시됩니다.
 3. 의사 대화식 트랜잭션의 경우, MQCIH와 첫 번째 벡터에 동일한 값을 입력하십시오.
- 이 필드의 첫 번째 바이트는 CICS 카피북 DFHAID의 값으로 설정됩니다.

MQCIH.Authenticator

VERIFY_UOW 또는 VERIFY_ALL의 권한 레벨을 사용 중인 경우에만 이 필드가 적용됩니다. MQMD.UserIdentifer 필드에서 사용자 ID와 연관될 비밀번호 또는 패스티켓으로 값을 설정하십시오. 또한 이 값은 사용자가 3270 트랜잭션을 시작할 권한이 있는지 여부를 판별하기 위해 외부 보안 관리자가 사용됩니다. 패스티켓을 사용하는 경우 패스티켓 생성에 사용된 applid는 브릿지 모니터를 시작할 때 사용되는 PASSTKTA 키워드 값과 동일해야 합니다.

MQCIH.ConversationalTask

이 CICS 트랜잭션에 필요한 모든 입력 벡터가 입력 메시지에 제공되면 값 MQCCT_NO를 지정합니다. 다수의 메시지를 사용하여 트랜잭션에 입력 벡터를 제공할 수 있는 경우 MQCCT_YES를 지정하십시오.

MQCIH.Facility

의사 대화의 첫 번째 메시지에 값 MQCFAC_NONE을 지정하고 MQCIH.FacilityKeepTime 필드를 0이 아닌 값으로 설정합니다. CICS-MQ 브릿지는 첫 번째 메시지에서 기능 토큰을 리턴합니다. 의사 대화의 모든 후속 인바운드 메시지에서 이 값을 사용해야 합니다.

MQCIH.FacilityKeepTime

의사 대화에서 둘 이상의 메시지를 보내는 경우 CICS-MQ 브릿지가 기능 토큰을 리턴하도록 첫 번째 메시지에서 이 필드를 0이 아닌 값(초)으로 설정합니다. 의사 대화의 연속 트랜잭션은 이와 같은 방식으로 설정된 후에 동일한 기능 토큰을 사용해야 하므로 의사 대화 기간 동안 연관 터미널 영역(예: TCTUA)이 유지됩니다.

MQCIH.FacilityLike

기본값인 4개의 공백을 사용하거나 설치된 터미널의 이름을 지정합니다. CICS 터미널에서 CICS 명령 CEMT I TASK 또는 CEMT I TERM을 입력하여 설치된 터미널의 이름을 찾을 수 있습니다.

MQCIH.Flags

남은 만료 시간을 응답 메시지에 전달하려면 값을 MQCIH_PASS_EXPIRATION으로 설정합니다.

MQCIH.Format

값을 DFHMQDCI(또는 CSCQBDCI)로 설정합니다. 이 값은 MQCIH 다음의 모든 데이터가 브릿지에 인바운드되어 변환되어야 함을 CICS-MQ 브릿지에 알립니다. 브릿지는 아웃바운드 메시지에서 MQCIH.Format의 값을 설정하며 이는 DFHMQDCO(또는 CSCQBDCO)에 대한 응답 큐으로 리턴됩니다.

MQCIH.GetWaitInterval

이 필드를 기본값으로 설정하면 작업 단위 내의 메시지에 대한 브릿지 태스크 GET WAIT 간격은 CICS-MQ 브릿지 모니터가 시작될 때 WAIT 매개변수에 지정된 값입니다. WAIT 매개변수도 기본값으로 설정하면 GET WAIT 간격은 무제한입니다.

MQCIH.LinkType

3270 트랜잭션의 MQCLT_TRANSACTION을 지정합니다.

MQCIH.RemoteSysid

특정 CICS 시스템에서 요청을 처리해야 하는 경우가 아니면 의사 대화의 첫 번째 메시지에서 이 필드를 공백으로 설정합니다. 의사 대화의 후속 메시지에서 이 필드를 첫 번째 응답 메시지에서 리턴된 값으로 설정합니다.

참고: 이전 버전의 CICS-MQ 브릿지에서는 RemoteSysId 필드가 사용되지 않았습니다. 그러나 다중 CICS-MQ 브릿지 모니터에 대한 기능을 사용할 수 있도록 대화를 패스스루 하는 것이 중요합니다. CICS 프로그래밍의 일반적인 스타일은 의사 대화식입니다. 즉, 서로 링크되어 완전한 애플리케이션을 구성하는 일련의 독립적인 트랜잭션입니다. CICS-MQ 브릿지를 사용할 때 의사 대화의 트랜잭션 사이의 링크는 시퀀스의 첫 번째 트랜잭션이 리턴한 기능 토큰 및 RemoteSysId 값을 대화의 후속 메시지로 전달하여 유지됩니다.

MQCIH.StartCode

3270 트랜잭션을 시작할 때 MQCSC_NONE의 기본값에서 이 필드의 값을 변경해야 합니다. 사용하는 값은 트랜잭션의 특성에 따라 다릅니다. 트랜잭션이 데이터 없이 EXEC CICS START 명령으로 시작되고 EXEC CICS RETRIEVE 명령을 발행하지 않으면 MQCSC_START 값을 사용하십시오. 트랜잭션이 데이터가 있는 EXEC CICS START 명령으로 시작되고 EXEC CICS RETRIEVE 명령을 발행하면 MQCSC_STARTDATA 값을 사용하십시오. 그렇지 않으면, MQCSC_TERMINPUT의 값을 사용하십시오.

MQCIH.TransactionId

이 값은 브릿지 태스크에 의해 실행될 3270 트랜잭션의 트랜잭션 ID입니다. 첫 번째 메시지는 시작될 첫 번째 트랜잭션을 지정해야 합니다. 후속 메시지의 이 필드를 이전 응답 메시지에서 리턴된 MQCIH.NextTransactionId 값으로 설정하십시오.

MQCIH.UOWControl

이 값은 CICS-MQ 브릿지에 의해 수행된 작업 단위(UOW) 처리를 제어합니다. 84 페이지의 『3270 트랜잭션에 대한 MsgId, CorrelId 및 UOWControl 필드』도 참조하십시오.

DPL 프로그램의 MsgId, CorrelId 및 UOWControl 필드

비-CICS 애플리케이션이 단일 DPL 프로그램을 실행하기 위해 CICS-MQ 브릿지를 사용 중인 경우, MQCUOWC_ONLY에 MQCIH.UOWControl 필드의 값을 설정하십시오. 그러나 애플리케이션이 다수의 메시지를 전송하고 수신하는 경우, 작업 단위를 올바르게 처리하기 위한 대체 옵션을 설정해야 합니다.

작업 단위 내에서 여러 개의 사용자 프로그램을 실행하려면 다음과 같이 MQCIH.UOWControl 필드 값을 설정하십시오.

- 첫 번째 요청의 MQCUOWC_FIRST
- 중간 요청의 MQCUOWC_MIDDLE
- 마지막 요청의 MQCUOWC_LAST

애플리케이션은 응답 메시지를 받기 전에 작업 단위로 여러 개의 요청 메시지를 보낼 수 있습니다. 첫 번째 메시지 후 언제든지 MQCIH.UOWControl 필드를 MQCUOWC_COMMIT 또는 MQCUOWC_BACKOUT으로 설정한 메시지를 전송하여 작업 단위를 중지할 수 있습니다.

첫 번째 메시지는 MQMD.CorrelId = MQCI_NEW_SESSION을 지정해야 하고 후속 메시지는 MQMD.CorrelId 필드를 첫 번째 메시지의 메시지 ID로 설정해야 합니다. CICS-MQ 브릿지를 사용하여 둘 이상의 사용자 DPL 프로그램을 실행 중인 경우, 요청 메시지의 MsgId 필드는 큐 관리자에 의해 설정되며(여기에 표시된 예에서 M1로) 다음에 CorrelId 필드에 복사됩니다.

다음 다이어그램은 CICS-MQ 브릿지를 사용하여 실행하는 일반 CICS DPL 프로그램에 대한 MQMD 및 MQCIH의 키 필드에서 사용하고 예상되는 값을 요약합니다.

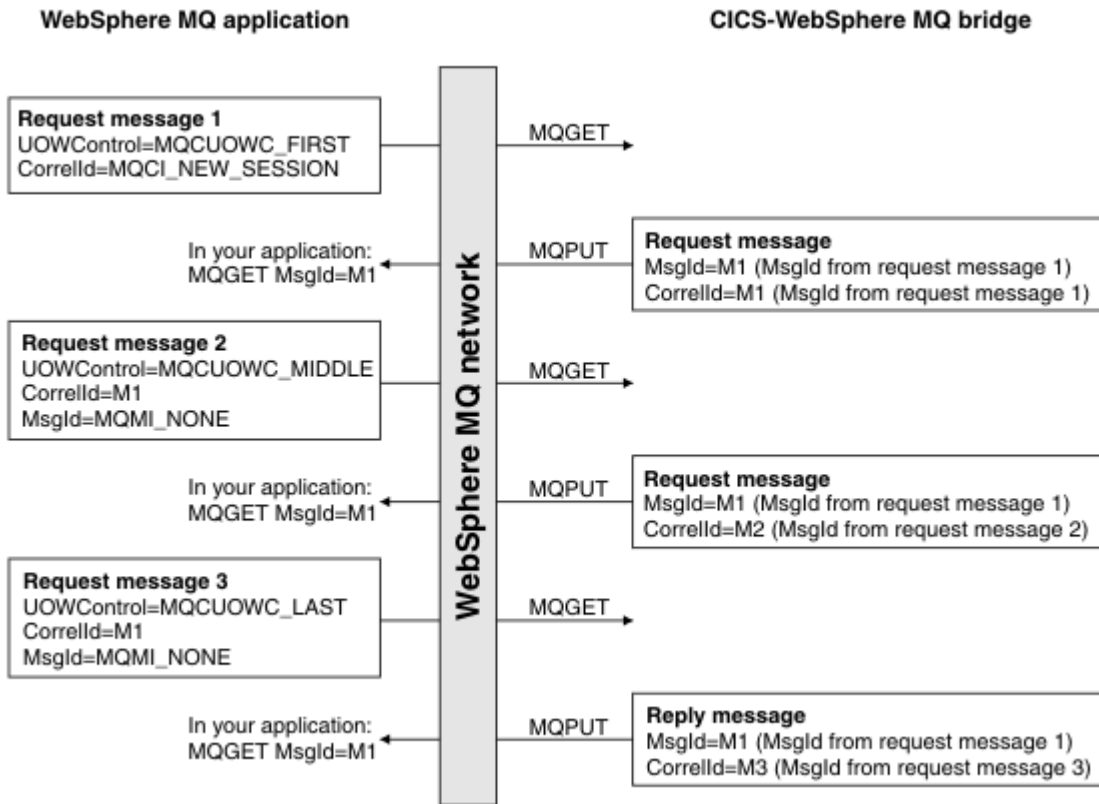


그림 10. 브릿지 관점에서 본 작업 단위(UOW)에서 CICS 사용자 프로그램의 키 필드 설정

3270 트랜잭션에 대한 MsgId, CorrelId 및 UOWControl 필드

일반적으로 일련의 CICS 3270 트랜잭션은 완전한 애플리케이션을 형성하기 위해 의사 대화에서 함께 링크됩니다. 비CICS 애플리케이션이 CICS-MQ 브릿지를 사용하여 3270 트랜잭션을 실행하는 경우 MQCIH.UOWControl 필드를 사용하여 트랜잭션이 작업 단위(UOW)로 그룹화되는 방법을 제어할 수 있습니다.

참고: MQCIH.UOWControl 및 MQMD.Correlid 필드를 설정하는 것뿐만 아니라, MQCIH.Facility 필드 및 MQCIH.RemoteSysId 필드를 사용하여 시퀀스의 첫 번째 트랜잭션이 리턴한 기능 토큰 및 RemoteSysId 값을 대화의 후속 메시지로 전달하는 것이 중요합니다. 자세한 정보는 81 페이지의 『3270 트랜잭션 요청 메시지의 MQCIH 필드』의 내용을 참조하십시오.

다중 3270 트랜잭션을 단일 CICS-MQ 브릿지 세션으로 그룹화하려면 다음과 같이 MQCIH.UOWControl 필드의 값을 설정하십시오.

- 브릿지 세션의 첫 번째 메시지에 MQCUOWC_FIRST를 사용하십시오.
- MQCUOWC_MIDDLE이 트랜잭션에 추가 데이터를 제공하거나 새 트랜잭션을 시작하든 브릿지 세션의 후속 메시지에서 MQCUOWC_MIDDLE을 사용하십시오.
- MQCUOWC_LAST를 사용하여 제안된 세션 종료를 표시하십시오. 85 페이지의 『세션 종료』의 내용을 참조하십시오.

다음과 같이 MQMD.Correlid 필드의 값을 설정하십시오.

- 브릿지 세션의 첫 번째 메시지에 MQCI_NEW_SESSION을 사용하십시오.
- 브릿지 세션에 대한 모든 후속 메시지에서 트랜잭션의 첫 번째 메시지에 대해 생성된 메시지 ID를 사용하십시오.

다음과 같이 MQCIH.Facility 필드의 값을 설정하십시오.

- 브릿지 세션의 첫 번째 메시지에 MQCFAC_NONE을 지정하고 MQCIH.FacilityKeepTime 필드를 0이 아닌 값으로 설정하십시오.
- 브릿지 세션에 대한 모든 후속 메시지에서 첫 번째 응답 메시지에 리턴된 기능 토큰을 사용하십시오.

다음과 같이 MQCIH.RemoteSysid 필드의 값을 설정하십시오.

- 브릿지 세션의 첫 번째 메시지를 위한 공백을 사용하고 요청이 특정 CICS 시스템에서 처리되도록 요청하지 않으면, 그런 경우에 CICS 시스템을 지정하십시오.
- 브릿지 세션에 대한 모든 후속 메시지에서 첫 번째 응답 메시지에 리턴된 값을 사용하십시오.

세션 종료

제안된 세션 종료를 표시하기 위해 MQCIH.UOWControl 필드의 값을 MQCUOW_LAST로 설정하면 애플리케이션은 응답 메시지의 메시지 유형을 확인하여 추가 조치를 판별해야 합니다.

- Msgtype 필드에 표시된 응답 메시지 유형이 MQMT_REPLY인 경우, CICS 트랜잭션이 더 이상 데이터 요청없이 종료되고 세션이 종료됨을 의미합니다.
- 응답 메시지 유형이 MQMT_REQUEST인 경우, CICS 트랜잭션이 더 많은 데이터를 요청함을 의미하며 데이터를 제공하고 MQCIH.UOWControl 필드가 MQCUOWC_LAST 또는 MQCUOWC_CONTINUE로 설정된 다음 메시지를 보낼 수 있습니다.
- 응답 메시지 유형이 MQMT_REQUEST이지만 세션을 계속하지 않으려면 MQCIH.UOWControl 필드가 MQCUOWC_COMMIT로 설정된 메시지를 보내 세션을 종료하십시오.

실행 중인 3270 트랜잭션을 종료해야 하는 경우 MQCIH.CancelCode 필드를 4자 이상 종료 코드로 설정하십시오.

트랜잭션 사이의 세션을 종료하려면 MQCIH.UOWControl 필드를 MQCUOWC_COMMIT로 설정하십시오.

다음 다이어그램은 일반적인 대화식 3270 트랜잭션과 상호작용할 때 MQMD 및 MQCIH의 키 필드에서 사용하고 예상하는 값을 요약합니다.

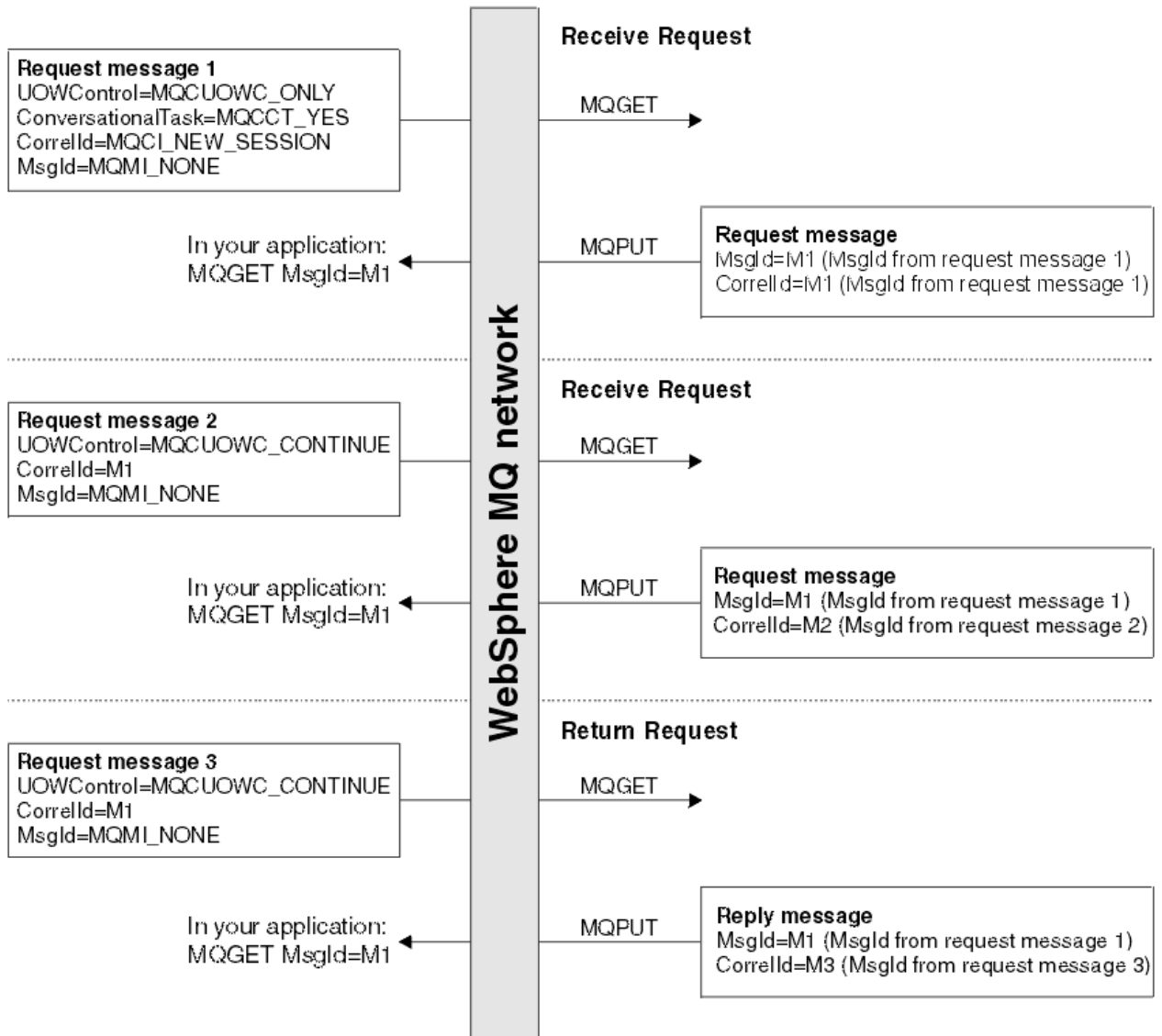


그림 11. 키 필드 설정: IBM MQ - 브릿지의 관점에서 볼 때 대화식 3270 트랜잭션

분산 환경에서 CICS-MQ 브릿지에 대한 데이터 변환

클라이언트 애플리케이션이 워크스테이션에 있으면 CICS DPL 프로그램 및 3270 트랜잭션은 CICS-MQ 브릿지를 통해 구동될 수 있습니다. 분산 환경을 위한 프로그래밍을 하는 경우 고려해야 할 주요 사항은 워크스테이션과 z/OS의 다른 인코딩 스킴과 CCSID 값 간의 데이터 변환입니다.

변환은 MQCIH 구조와 메시지에 제공된 데이터 또는 벡터에 대해 서로 다른 두 가지 루틴으로 수행됩니다. 메시지에 MQCIH가 있는 경우 MQCIH 직전 헤더의 형식 필드에 MQFMT_CICS를 지정해야 합니다. MQCIH 전에 헤더가 없는 경우 MQMD.Format 필드에서 MQFMT_CICS를 지정해야 합니다. 데이터 및 벡터 변환에는 좀 더 많은 고려사항이 필요합니다.

COMMAREA 또는 컨테이너 데이터를 수신하거나 리턴하지 않는 DPL 프로그램을 구동하거나 COMMAREA 또는 컨테이너 데이터가 순수하게 문자 데이터인 경우 MQFMT_STRING을 형식 값으로 지정하여 데이터 변환을 수행할 수 있습니다. COMMAREA 또는 컨테이너 데이터가 순수 2진인 경우 데이터 변환이 필요하지 않으므로 MQFMT_NONE을 형식 값으로 지정하십시오. COMMAREA 또는 컨테이너 데이터가 문자와 2진 데이터의 혼합인 경우 사용자 고유의 변환 루틴을 작성하고 적절한 형식 값을 지정해야 합니다. 변환 루틴 작성에 대한 정보는 [IBM MQ 제품 문서](#)의 내용을 참조하십시오.

메시지에 MQCIH가 포함된 경우, MQCIH.Format에 형식 값을 지정하십시오. 메시지에 MQCIH는 포함되어 있지 않지만 다른 헤더가 포함된 경우 마지막 헤더에 형식 값을 지정하십시오. 메시지에 헤더가 포함되어 있지 않으면 MQMD에서 형식 값을 지정하십시오.

COMMAREA에서 응답 데이터를 생성하는 DPL 프로그램을 구동하고 reply-to 큐를 제공한 경우 MQCIH.ReplyToFormat의 요청 메시지에서 이 응답 데이터의 형식 값을 지정해야 합니다. 컨테이너에서 응답 데이터를 생성하는 DPL 프로그램을 구동하는 경우 컨테이너는 자체적으로 문자 또는 2진 데이터로 식별되므로 MQCIH.ReplyToFormat을 지정할 필요가 없습니다.

3270 트랜잭션을 구동하는 경우 SEND MAP 및 RECEIVE MAP 벡터를 변환하려면 다음을 수행하십시오.

- DFHMSD 매크로에 DSECT=ADSDL을 지정하여 맵을 어셈블해야 합니다. 원래 맵핑 세트 정의가 없는 경우 CICS DFHBMSUP 유틸리티를 사용하여 맵을 다시 작성하십시오.
- 필드 MQCIH.ADSDDescriptor에서 MQCADSD_SEND+MQCADSD_MSGFORMAT의 값을 지정하십시오. ADSD 또는 ADSDL을 사용하여 RECEIVE MAP ADS를 빌드하는 경우 이 필드의 값 MQCADSD_REC를 추가하십시오.
- 모든 인바운드 메시지에서 필드 MQCIH.Format의 DFHMQDCI 또는 CSCQBDCI의 값을 지정하십시오. 입력 형식이 CSQCBDCI이면 출력 형식은 CSQCBDCO입니다. 입력 형식이 DFHMQDCI이면 출력 형식은 DFHMQDCO입니다.
- z/OS와 워크스테이션 사이의 채널에 CONVERT=YES가 지정되어 있고 데이터 변환 프로그램 DFHMQDCI가 채널 초기자에 설치되어 있는지 확인하십시오.

SEND MAP(아웃바운드) 및 RECEIVE MAP(인바운드 및 아웃바운드) 이외의 벡터 워크스테이션과 메인프레임 형식 간 변환은 지원되지 않습니다.

3270 트랜잭션의 응답 메시지가 비z/OS 시스템으로 전송될 때 송신자 채널은 CONVERT(YES) 옵션을 지정해야 합니다.

CICS-MQ 브릿지를 사용하여 BMS(Basic Mapping Support) 애플리케이션 실행

CICS 3270 트랜잭션이 BMS 맵을 사용하는 경우, 비CICS 애플리케이션은 맵 어셈블리 중에 작성된 카피북을 사용하여 벡터의 데이터를 해석하는 데 도움을 될 수 있습니다. 카피북에 액세스할 수 있는 권한이 없는 경우, 애플리케이션은 애플리케이션 데이터 구조(ADS) 디스크립터를 사용하여 데이터를 간접적으로 분석할 수 있습니다.

CICS BMS(Basic Mapping Support)는 CICS 애플리케이션이 여러 가지 다른 터미널 유형을 지원할 수 있는 방법을 제공합니다. 애플리케이션이 EXEC CICS SEND MAP 명령을 발행하면 BMS는 터미널 특정 제어 데이터를 애플리케이션 데이터와 병합하여 터미널에 표시할 수 있는 3270 데이터 스트림을 생성합니다. 애플리케이션이 EXEC CICS RECEIVE MAP 명령을 발행하면 인바운드 3270 데이터 스트림에서 애플리케이션 데이터가 추출되어 애플리케이션으로 리턴됩니다.

CICS 애플리케이션의 BMS 맵은 표시장치에 필요한 필드의 특성을 정의하는 BMS 매크로 세트를 조합하여 작성됩니다. 맵 어셈블리의 출력 중 하나는 표시장치 필드를 ADS에 맵핑하는 카피북입니다. CICS 애플리케이션은 데이터 정의에 카피북을 포함하여 맵의 필드를 기호로 처리할 수 있도록 해야 합니다. SEND MAP의 애플리케이션 데이터와 RECEIVE MAP에 의해 예상되는 애플리케이션 데이터는 카피북의 ADS에 직접 맵핑됩니다.

트랜잭션이 CICS 브릿지에서 실행될 때 EXEC CICS SEND MAP 및 EXEC CICS RECEIVE MAP 명령은 아웃바운드 메시지에서 SEND MAP 및 RECEIVE MAP 벡터를 생성합니다. 3270 데이터 스트림 대신 이러한 벡터에는 CICS 애플리케이션이 맵의 필드를 주소 지정하는 데 사용하는 것과 동일한 ADS가 포함됩니다.

ADS의 형식은 각 맵에 대해 고유합니다. 맵 생성의 일부로 작성된 카피북에 의해 설명됩니다. 이 카피북 없이 데이터를 해석하는 것은 어렵습니다. 일반적으로 IBM MQ 브릿지를 사용하는 비CICS 애플리케이션은 BMS 카피북을 포함하므로 RECEIVE MAP 데이터를 작성하고 SEND MAP 데이터를 해석할 수 있습니다. 그러나 특정 BMS 카피북 없이 애플리케이션을 작성할 수 있습니다. 데이터의 형식은 ADS 디스크립터(ADSD)로 알려진 구조에 의해 설명됩니다. ADSD는 SEND MAP 벡터의 마지막에 추가되고 벡터에서 ADS의 형식을 설명합니다. ADSD 콘텐츠에는 ADS의 필드 이름, 위치 및 길이가 포함됩니다. ADSD는 RECEIVE MAP 요청에 전송될 수도 있습니다. 대화식 애플리케이션에서 이 정보를 사용하여 비-CICS 애플리케이션에 CICS 애플리케이션에서 요청된 ADS의 구조를 알릴 수 있습니다. 비-CICS 애플리케이션은 이 ADS로 RECEIVE MAP 벡터를 빌드하고 새 요청을 전송할 수 있습니다.

애플리케이션 프로그래머는 ADS, ADSD 또는 ADSDL(긴 양식의 애플리케이션 데이터 구조 디스크립터)을 사용하여 메시지의 벡터 데이터를 해석할지 여부를 선택할 수 있습니다. ADS를 직접 해석하려면 CICS-MQ 브릿지 애

플리케이션에 맵 어셈블리의 카피북을 포함시키십시오. 포함시키려고 하지만 카피북 또는 맵에 액세스할 수 없는 경우 CICS 유틸리티 DFHBMSP를 사용하여 맵을 다시 작성하십시오.

ADSD 또는 ADSDL을 통해 ADS를 간접적으로 해석하도록 예를 들어 모든 맵을 처리할 일반 애플리케이션을 작성하는 경우 브릿지 애플리케이션에 카피북을 포함시킬 필요가 없습니다. 대신, 필요에 따라 아웃바운드 SEND MAP 및 RECEIVE MAP 요청 벡터에 ADSD 또는 ADSDL을 포함시키도록 브릿지에 제어 정보를 보냅니다.

애플리케이션이 분산 환경에서 실행되어야 하는 경우 아웃바운드 SEND MAP 벡터에 ADSDL을 포함시키십시오. 그러면 IBM MQ는 아웃바운드 메시지에서 데이터를 변환시킬 수 있습니다.

인바운드 메시지에서 필드 MQCIH.ADSDDescriptor에 적절한 값을 설정하여 다음 옵션 중 하나를 지정할 수 있습니다.

- SEND MAP 벡터에 ADSD(애플리케이션 데이터 구조 디스크립터의 짧은 양식)를 포함시키려면 다음을 설정하십시오.

```
MQCIH.ADSDDescriptor = MQCADSD_SEND
```

이 설정을 사용하면 SEND MAP 벡터에 포함된 ADS(애플리케이션 데이터 구조)의 짧은 양식도 얻을 수 있습니다.

- RECEIVE MAP 벡터로 ADSD를 포함하려면 다음과 같이 설정하십시오.

```
MQCIH.ADSDDescriptor = MQCADSD_RECV
```

ADS는 아웃바운드 RECEIVE MAP 요청 벡터에 존재하지 않습니다.

- ADSDL을 SEND MAP 또는 RECEIVE MAP 벡터에 포함하려면 다음을 설정하십시오.

```
MQCIH.ADSDDescriptor = MQCADSD_MSGFORMAT
```

이 설정을 사용하면 SEND MAP 벡터에 포함된 ADS의 긴 양식도 얻을 수 있습니다.

- ADS 디스크립터를 SEND MAP 또는 RECEIVE MAP 벡터에 포함하지 않으려면 다음을 설정하십시오.

```
MQCIH.ADSDDescriptor = MQCADSD_NONE
```

이 설정이 기본값입니다. 이 설정을 사용하면 SEND MAP 벡터에 포함된 ADS의 짧은 양식을 얻을 수 있습니다.

SEND MAP 및 RECEIVE MAP 벡터 모두에 긴 양식의 애플리케이션 데이터 구조 디스크립터를 포함시키기 위해 MQCADSD_* 값을 함께 추가할 수 있습니다.

```
MQCIH.ADSDDescriptor = MQCADSD_SEND + MQCADSD_RECV + MQCADSD_MSGFORMAT
```

이 경우 SEND MAP 벡터도 ADS를 long 양식으로 포함시킵니다.

CICS-MQ 브릿지에 대한 SEND MAP 벡터 해석

아웃바운드 SEND MAP 벡터는 애플리케이션 데이터 구조(ADS) 및 애플리케이션 데이터 구조 디스크립터를 짧은 양식(ADSD) 또는 긴 양식(ADSDL)으로 포함할 수 있습니다.

SEND MAP 벡터를 해석하려면(메시지에 ADS 및 ADSD 또는 ADSDL이 포함되어 있다고 가정) 다음을 수행하십시오.

1. 브릿지 응답 큐에서 SEND MAP 벡터가 포함된 메시지를 메시지 버퍼로 가져오십시오.
2. 메시지 버퍼에서 아웃바운드 SEND MAP 벡터의 시작을 찾으십시오. 이 벡터는 MQCIH에 추가되므로 메시지 버퍼 시작부터 MQCIH 길이와 동일한 오프셋에 있습니다. 다음 코드 단편을 모델로 사용할 수 있습니다.

```
/* #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 */
:
```



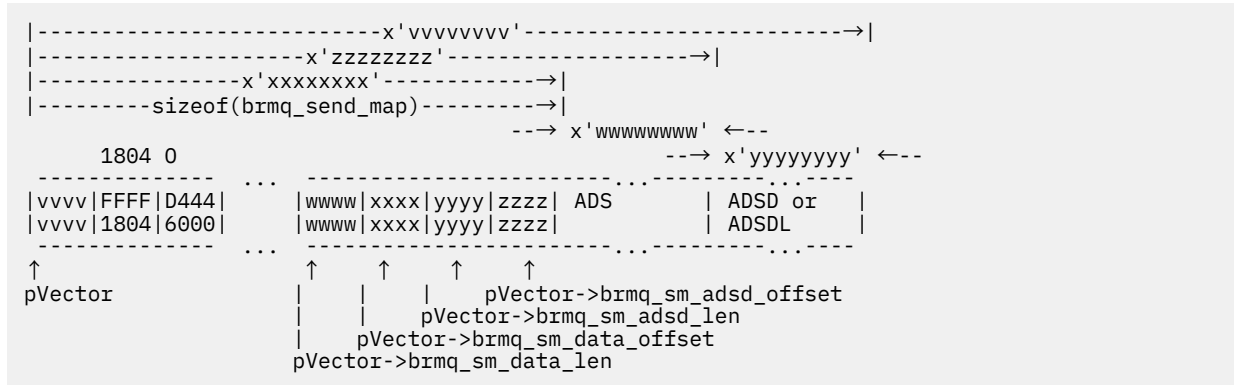
```

/* Set the vector pointer to the start of the vector          */
pVector = MsgBuffer + ((MQCIH *) MsgBuffer)->StrucLength ;

```

- SEND MAP 벡터에서 애플리케이션 데이터 구조(ADS) 및 애플리케이션 데이터 구조 디스크립터(ADSD 또는 ADSDL)의 시작 주소를 식별하십시오.

다음 다이어그램은 이전 코드 단편에서와 같이 brmq_send_map 벡터의 시작을 처리하기 위해 pVector라는 포인터를 설정했다고 가정한 경우 아웃바운드 SEND MAP 벡터의 구조를 보여줍니다.



다이어그램의 값은 다음과 같이 표시됩니다.

```

ABCD
1234

```

16진 설정을 사용하는 ISPF 편집기에서 보는 것과 같이 16진 값을 표시합니다. 이 값은 16진 값 X'A1B2C3D4'와 동일합니다.

필드 pVector-> brmq_sm_data_offset 및 pVector-> brmq_sm_data_len은 ADS의 오프셋과 길이를 제공하며 필드 pVector-> brmq_sm_adsd_offset 및 pVector-> brmq_sm_adsd_len은 ADSD 또는 ADSDL의 오프셋과 길이를 제공합니다.

필드 brmq_sm_adsd_offset 및 brmq_sm_adsd_len은 ADSD 또는 ADSDL이 메시지에 포함되지 않으면 둘 다 0으로 설정됩니다.

- ADSD 또는 ADSDL에서 필드를 식별하십시오.

ADSD 및 ADSDL은 둘 다 헤더 파일 dfhbrarh.h에 정의된 구조에 매핑되며 CICS 라이브러리 <hlq>.SDFHC370에 분배됩니다. 필드가 어떻게 배치되어 있는지 구조 정의를 검사할 수 있습니다. ADSD의 필드는 [Link3270 ADS 디스크립터](#)에도 설명되어 있습니다.

워크스테이션에서 브릿지 애플리케이션을 컴파일하려면 dfhbrarh.h 파일을 해당 환경으로 복사하십시오.

ADSD 및 ADSDL 모두 두 가지 유형의 구조로 표시됩니다. 첫 번째 구조는 디스크립터로 ADSD 또는 ADSDL 시작 시 한 번만 발생합니다. 이러한 유형은 다음과 같이 정의됩니다.

ads_descriptor

ADSD의 디스크립터(짧은 양식)

ads_long_descriptor

ADSDL의 디스크립터(긴 양식)

두 번째 구조는 필드 디스크립터이며 맵의 각 필드마다 한 번씩 반복됩니다. 이러한 유형은 다음과 같이 정의됩니다.

ads_field_descriptor

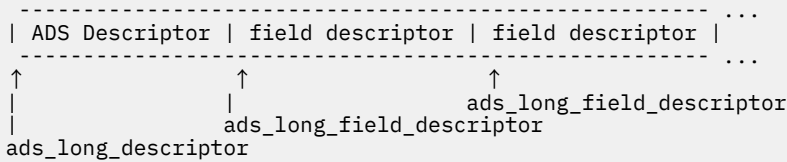
ADSD의 필드 디스크립터(짧은 양식)

ads_long_field_descriptor

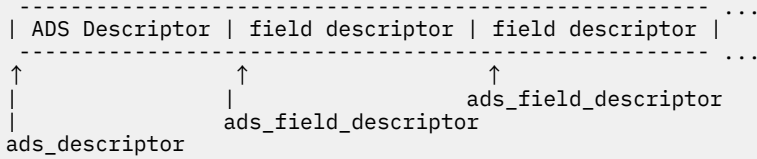
ADSDL의 필드 디스크립터(긴 양식)

이 구조는 ADSDL 및 ADSD에 대해 다이어그램 형식으로 표시될 수 있습니다.

ADSDL:



ADSD:



디스크립터에서 필드 adsd_field_count 및 adsdl_field_count는 ADSD 및 ADSDL에서 필드 디스크립터의 수를 식별합니다.

다음 코드 단편을 사용하여 ADSD 또는 ADSDL 구조의 시작에 대한 포인터를 설정하고 순차적으로 필드 디스크립터를 처리할 수 있습니다. pVector는 이미 brmq_send_map 벡터의 시작을 처리하고 인바운드 메시지의 CIH를 포함하는 mqcih라는 MQCIH 구조를 가지고 있다고 가정합니다.

```

/* #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 mqcih.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. ADS에서 필드를 식별하십시오.

ADS은 사용자의 맵을 어셈블할 때 생성되는 구조에 맵핑됩니다. 맵 세트 정의 매크로에 DSECT=ADSDL의 keyword=parameter 값을 포함시키면 ADS의 긴 양식을 얻게 됩니다. 맵 어셈블리의 출력은 입력 구조와 출력 구조라는 두 가지 구조의 유니온입니다. 이 예는 이러한 유니온의 파트를 표시합니다(첫 번째 필드 정의만 각 구조에 대해 표시되며 주석은 맵 어셈블리에 따라 추가됨).

```
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                     */
```

입력 구조에 7 요소 배열의 확장된 속성 값이 포함되고 출력 구조가 개별적으로 이름 지정된 필드를 제공한다는 점을 제외하면 두 구조는 기능적으로 동일합니다.

다음 코드 단편을 사용하여 ADS 시작에 대한 포인터를 설정할 수 있습니다. DSECT 예에 표시된 구조 이름이 설명에 사용됩니다. 첫 번째는 인바운드 데이터를 처리하고 두 번째는 아웃바운드 데이터를 처리하는 두 개의 포인터가 설정됩니다. pVector는 이미 brmq_send_map 벡터의 시작을 처리했다고 가정합니다.

```
/* #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->adSDL_length ) ;
:

```

CICS-MQ 브릿지에 대한 RECEIVE MAP 벡터 해석

RECEIVE MAP 요청은 다음 입력 메시지에 RECEIVE MAP을 제공하기 위한 클라이언트의 요청입니다.

SEND MAP 벡터와 달리 아웃바운드 RECEIVE MAP 요청 벡터에는 ADS가 포함되지 않습니다.

MQCADSD_RECV가 MQCIH.ADSDescriptor에 지정된 경우 다음 인바운드 RECEIVE MAP 벡터에 필요한 ADS 데이터를 설명하는 ADSD 또는 ADSDL을 포함합니다. RECEIVE MAP 벡터 구조는 SEND MAP 벡터의 구조와 다릅니다. 가장 큰 차이점은 ADS의 오프셋과 길이를 제공하는 필드가 없다는 점입니다.

RECEIVE MAP 벡터를 해석하려면(메시지에 ADSD 또는 ADSDL이 포함되어 있다고 가정) 다음을 수행하십시오.

1. 브릿지 응답 큐에서 RECEIVE MAP 요청 벡터가 포함된 메시지를 메시지 버퍼로 가져오십시오.
2. 메시지 버퍼에서 아웃바운드 RECEIVE MAP 벡터의 시작을 찾으십시오. 이 벡터는 MQCIH에 추가되므로 메시지 버퍼 시작부터 CIH 길이와 동일한 오프셋에 있습니다. 이 코드 단편을 모델로 사용할 수 있습니다.

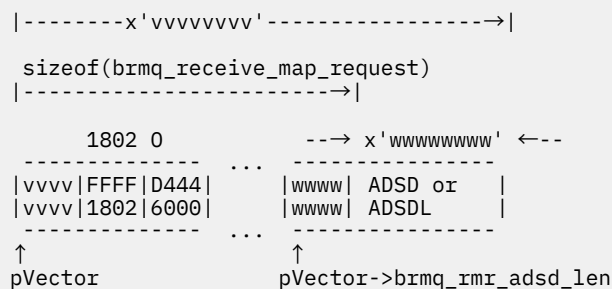
```

/* #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 벡터에서 시작 주소 ADSD 또는 ADSDL을 식별하십시오.

다음 다이어그램은 아웃바운드 RECEIVE MAP 요청 벡터의 구조를 표시합니다. 이 다이어그램에서는 이전 코드 단편에서와 같이 brmq_receive_map_request 벡터의 시작을 처리하기 위해 pVector라는 포인터를 설정했다고 가정합니다.



다이어그램의 값은 다음과 같이 표시됩니다.

```

ABCD
1234

```

16진 설정을 사용하는 ISPF 편집기에서 보는 것과 같이 16진 값을 표시합니다. 이 값은 16진 값 X'A1B2C3D4'와 동일합니다.

필드 pVector->brmq_rmr_adsd_len는 ADSD 또는 ADSDL의 길이를 나타냅니다. ADSDL이 brmq_receive_map_request 벡터에 직접 추가되므로 오프셋이 제공되지 않습니다.

4. ADSD 또는 ADSDL에서 필드를 식별하십시오.

이를 수행하려면 [88 페이지의 『CICS-MQ 브릿지에 대한 SEND MAP 벡터 해석』](#)에 설명된 SEND MAP 벡터에 대해 일반적으로 진행하십시오. 그러나 다음 코드 단편을 사용하여 ADSD 또는 ADSDL의 시작에 대한 포인터를 설정하십시오.

```

:
if (mqcih.ADSDDescriptor && MQCADSD_MSGFORMAT)
{
    pADSDL_D = pVector + sizeof(brmq_receive_map_request) ;
    :
}

else /* Short form */
{
    pADSD_D = pVector + sizeof(brmq_receive_map_request) ;
    :
}
:

```

ADSD 또는 ADSDL은 RECEIVE MAP 벡터에서 SEND MAP 벡터와 동일한 구조를 갖습니다. 따라서 시작 주소를 식별한 후 SEND MAP 벡터에 대해 설명한 대로 진행할 수 있습니다.

ADSDL 및 ADS의 예

ADSDL 및 ADS의 파트를 보여주는 예가 설명되어 있습니다.

다이어그램의 값은 다음과 같이 표시됩니다.

ABCD
1234

16진 설정을 사용하는 ISPF 편집기에서 보는 것과 같이 16진 값을 표시합니다. 이 값은 16진 값 X'A1B2C3D4'와 동일합니다.

이 다이어그램은 ADSDL의 시작을 보여줍니다(아이캐처는 ADSL을 표시하지만).

```

...%ADSL.....±....CHO          L .....┐.....&$...TRANID
000BCCED0000000100040000CDD44444444D40000F000F000100054000EDCDCC44...
005814230001000B001900033860000000003000000F000E008000A00039159400...
↑              ↑                |   |           ↑
|             adsdl_field_count |   |       adsdl_first_field
adsdl_length                    |   |       adsdl_map_columns
                               |   |       adsdl_map_lines

```

이 예에서 이름 지정된 필드는 다음 값을 표시합니다.

adsdl_length

이 ADSDL의 길이는 0x05B8바이트입니다.

adsdl_field_count

ADS에는 0x1B(27)라는 이름의 필드가 있습니다.

adsdl_map_lines

맵에는 0x18(24)개의 행이 있습니다.

adsdl_map_columns

맵에는 0x50(80)개의 열이 있습니다.

adsdl_first_field

ADSDL에서 첫 번째 필드 설명의 시작입니다.

다음 다이어그램은 ADSDL 첫 번째 필드 디스크립터와 다음 필드 디스크립터의 파트를 표시합니다.

```
TRANID .....L ..TERMID  
EDCDCC444444444444444444444400000000000000D400ECDCC444444444...  
3915940000000000000000000000000060000000C00043000359494000000000...  
↑          |      ↑   |     ↑       adsdl_next_field  
            |      |    |             adsdl_field_data_len  
            |      |    |             adsdl_field_offset  
adsdl_field_name        adsdl field name len
```


그런 다음 사용자는 다시 표시된 맵에서 필드를 업데이트하고 RECEIVE MAP - SEND MAP 주기를 다시 시작할 수 있습니다. 로직은 다음과 같이 설명될 수 있습니다(EC는 EXEC CICS를 표시함).

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>		
:		

CICS-MQ 브릿지를 사용하여 이 트랜잭션을 실행하면 실제 터미널은 사용자의 비-CICS 애플리케이션으로 대체됩니다. 3270 트랜잭션의 로직은 변경되지 않으며 수신하는 애플리케이션 데이터는 동일하지만 플로우되는 데이터와 전송 방법은 다릅니다. 3270 데이터 스트림 대신 MQCIH 구조, 벡터 구조 및 선택적으로 애플리케이션 데이터 구조에 대한 표시를 포함하는 IBM MQ 메시지가 사용됩니다.

메시지에 이러한 오브젝트를 포함시키는 것은 메시지가 흐르는 방향(브릿지로 인바운드 또는 브릿지에서 아웃바운드), 교환에서 메시지 순서 및 MQCIH의 필드에 적절한 값을 설정하여 요청된 애플리케이션 데이터 구조 디스크립터가 요청되었는지 여부에 따라 다릅니다.

95 페이지의 『최적화 없는 정확한 에뮬레이션』에는 이전 스킴이 정확하게 에뮬레이트될 때 발생하는 플로우가 표시되어 있습니다. 96 페이지의 『최적화로 향상된 에뮬레이션』에 표시된 대로 인바운드 메시지에 둘 이상의 벡터를 포함시켜 플로우를 최적화할 수 있습니다.

이 예제에서 MQCIH.ADSDescriptor가 다음과 같이 설정되었다고 가정하십시오.

MQCADSD_SEND + MQCADSD_RECV + MQCADSD_MSGFORMAT

결과적으로, 긴 양식의 애플리케이션 데이터 구조 디스크립터는 메시지 교환 중에 아웃바운드 및 인바운드 애플리케이션 데이터 구조 모두에 추가됩니다.

명확하게 하기 위해 메시징 세부사항은 여기서 생략됩니다. CICS-MQ 브릿지에서 사용된 큐잉 모델에 대한 설명은 [IBM MQ 제품 문서](#)의 내용을 참조하십시오.

최적화 없는 정확한 에뮬레이션

이 예제는 CICS-MQ 브릿지를 사용하여 예제 3270 트랜잭션의 원래 논리를 정확하게 에뮬레이트할 때의 데이터 플로우를 표시합니다.

CICS-MQ bridge application		3270 Transaction
MQPUT to Bridge RequestQ	--MQCIH----->	<Initial start>
	(Note 1)	<business logic>
MQGET from Bridge ReplyQ	<--MQCIH+bmq_send_map+ADS+ADSDL-----	EC SEND MAP FROM(ads)
	(Note 2)	EC RETURN TRANSID(BAAA)
MQPUT to Bridge RequestQ	--MQCIH----->	<Start>
	(Note 3)	
MQGET from Bridge ReplyQ	<--MQCIH+bmq_receive_map_request+ADSDL-	EC RECEIVE MAP INTO(ads)
	(Note 4)	
Updates fields		waits
MQPUT to Bridge RequestQ	----MQCIH+bmq_receive_map+ADS----->	↓
	(Note 5)	
		<business logic>
MQGET from Bridge	<--MQCIH+bmq_send_map+ADS+ADSDL-----	EC SEND MAP FROM(ads)
	(Note 6)	EC RETURN

ReplyQ

TRANSID(BAAA)

:

참고:

1. 애플리케이션의 초기 플로우에는 MQCIH만 포함됩니다. MQCIH에는 시작될 트랜잭션을 지정하는 제어 정보가 포함되어 있습니다.
2. 3270 트랜잭션의 리턴 플로우에는 모든 후속 플로우에 사용되는 기능 토큰 및 오류가 발생한 경우의 진단 정보가 포함된 MQCIH가 포함됩니다. 또한 맵 자체와 관련된 제어 정보 및 맵을 표시하는 데이터가 포함된 SEND MAP 벡터 구조도 포함됩니다. 초기화 애플리케이션이 요청한 경우 애플리케이션 데이터 구조 디스크립터도 포함됩니다.
3. 브릿지 애플리케이션은 트랜잭션을 한 번 더 시작하기 위한 제어 정보가 포함된 MQCIH만 포함하는 메시지를 다시 보냅니다.
4. 3270 트랜잭션은 레거시 환경에서와 같이 EXEC CICS RECEIVE MAP 명령을 발행합니다. 그러나 브릿지 환경에서는 맵 데이터를 즉시 사용할 수 없습니다. 호출은 아웃바운드 RECEIVE MAP 요청 벡터를 포함하는 메시지로 변환됩니다. 애플리케이션 업무 데이터 구조 디스크립터도 메시지에 포함됩니다. 이 예에서 브릿지 애플리케이션이 메시지를 전환하는 동안 트랜잭션은 대기합니다. 여기에서 모델은 레거시 환경의 모델과 약간 다릅니다. 그러나 브릿지 아키텍처에서는 메시지에 둘 이상의 벡터를 포함할 수 있으므로 단일 인바운드 메시지로 많은 요청을 충족할 수 있습니다.
5. 애플리케이션 데이터 구조에서 필드를 업데이트한 후 브릿지 애플리케이션은 아웃바운드 요청을 충족시키기 위해 인바운드 RECEIVE MAP 응답 벡터를 보냅니다.
6. 3270 트랜잭션은 EXEC CICS SEND MAP을 발행하여 SEND MAP 벡터로 변환하고 주기가 반복됩니다.

패스루 도구(CICS SupportPac CA1E CICS 3270 Bridge Passthrough)를 사용하여 이러한 트랜잭션을 실행할 때 캡처된 플로우를 검사하면 사용 가능한 온라인 문서의 도움말을 사용하여 해당 구조를 식별할 수 있습니다.

최적화로 향상된 에뮬레이션

이 예제는 브릿지를 사용하는 CICS-MQ 예제 3270 트랜잭션에 대한 최적화된 데이터 플로우를 표시합니다.

CICS-MQ Bridge Application		3270 Transaction
MQPUT to Bridge RequestQ	--MQCIH----->	<Initial start> <business logic>
MQGET from Bridge ReplyQ	<---MQCIH+brmq_send_map+ADS+ADSDL-----	EC SEND MAP FROM(ads) EC RETURN TRANSID(BAAA)
MQPUT to Bridge RequestQ	----MQCIH+brmq_receive_map+ADS----->	<Start> EC RECEIVE MAP INTO(ads) <business logic>
MQGET from Bridge ReplyQ	<---MQCIH+brmq_send_map+ADS+ADSDL-----	EC SEND MAP FROM(ads) EC RETURN TRANSID(BAAA) :

이 순서를 95 페이지의 『최적화 없는 정확한 에뮬레이션』에 표시된 최적화되지 않은 플로우와 비교하면 인바운드 LEFER MAP 벡터가 이미 요구사항을 예상했고 인바운드 맵을 트랜잭션에서 이미 사용할 수 있으므로 CICS 트랜잭션이 RECEIVE MAP을 전송할 필요가 없음을 확인할 수 있습니다.

제 8 장 CICS-MQ 어댑터를 사용하기 위한 애플리케이션 개발

CICS-MQ 어댑터는 CICS 애플리케이션 프로그램에서 사용하도록 IBM MQ MQI(Message Queue Interface)를 구현합니다. 이 어댑터는 CICS-MQ API 교차 엑시트도 지원합니다. 이 엑시트를 사용하여 모니터링, 테스트, 유지보수 또는 보안 용도로 MQI 호출이 실행될 때 인터셉트할 수 있습니다.

트랜잭션 무결성을 위해 어댑터는 CICS Recovery Manager의 제어 하에 동기점 지정을 지원하므로 필요에 따라 작업 단위(UOW)를 커밋하거나 백아웃할 수 있습니다. 어댑터는 또한 보안 서버(이전 RACF)와 같은 적절한 보안 관리 제품과 함께 사용될 때 IBM MQ 자원의 보안 검사를 지원합니다. 어댑터는 큐 관리자 종료 후 자동 재연결 및 재시작 후 자동 자원 재동기화로 고가용성을 제공합니다. 또한 큐 관리자 종료와 같은 스케줄되지 않은 이벤트에 응답하는 경보 모니터 기능도 제공합니다.

IBM MQ MQI 호출에 액세스하기 위한 API 스텝 프로그램

IBM MQ에 액세스하려면 동적 호출을 사용하고 있지 않은 경우 CICS에서 실행되는 애플리케이션 프로그램은 제공된 API 스텝 프로그램, CSQCSTUB로 링크 편집되어야 합니다. CSQCSTUB은 애플리케이션에 모든 MQI 호출에 대한 액세스를 제공합니다.

동적으로 CICS 스텝을 호출하는 데 대한 정보는 [IBM MQ 제품 문서의 애플리케이션 개발](#)의 내용을 참조하십시오.

CSQCSTUB는 SDFHLOAD 로드 라이브러리에서 모듈 DFHMQSTB로서, SDFHAUTH 로드 라이브러리에서 DFHMQSTX로서 제공됩니다. 두 모듈 모두 다음 별명으로 정의되어 있습니다.

CSQCBFMH
CSQCCB
CSQCCLOS
CSQCCONN
CSQCCONX
CSQCCTL
CSQCCTMH
CSQCDISC
CSQCDTMH
CSQCDTMP
CSQCINQ
CSQCIQMP
CSQCMHBF
CSQCOPEN
CSQCGET
CSQCPUT
CSQCPUT1
CSQCSET
CSQCSTAT
CSQCSTMP
CSQCSTUB
CSQCSUB
CSQCSUBR

CSQBFMH, CSQCCB, CSQCCTL, CSQCCTMH, CSQCDTMH, CSQCDTMP, CSQCIQMP, CSQCMHBF, CSQCSTAT, CSQCSTMP, CSQCSUB 및 CSQCSUBR의 별명은 CICS가 버전 7 이상의 IBM MQ에 연결되는 경우에만 사용됩니다.

기존 CICS-MQ 애플리케이션은 CICS 제공 CICS-MQ 어댑터를 사용하여 변경없이 실행할 수 있습니다. 다시 컴파일하거나 링크 편집할 필요가 없습니다. 새 애플리케이션 또는 변경된 애플리케이션의 경우 기존 링크 편집 프

로시저를 사용할 수 있으며 애플리케이션이 IBM MQ 버전 7에 추가된 새 API 호출을 사용하지 않는 한 CICS 또는 IBM MQ와 함께 제공된 스텝을 사용할 수 있습니다. 버전 7 API 호출은 IBM MQ와 함께 제공된 스텝이 아닌 CICS와 함께 제공된 스텝을 사용할 때 CICS에서만 지원됩니다. 새 API 호출은 MQBUFMH, MQCB, MQCTL, MQCRTMH, MQDLTMH, MQDLTMP, MQINQMP, MQMHBUF, MQSETMP, MQSTAT, MQSUB 및 MQSUBRQ입니다.

비동기 메시지 이용 및 콜백 루틴

IBM MQ 버전 7 이상에서 사용할 수 있는 비동기 메시지 이용은 MQCB 및 MQCTL MQI 호출을 사용합니다. CICS 애플리케이션 프로그램에서 이러한 기능을 사용하는 경우 CICS는 처리의 일부 측면을 수행하므로 애플리케이션을 코딩할 때 IBM MQ 프로그래밍 문서를 사용하는 것뿐만 아니라 여기에 제공된 정보를 사용해야 합니다.

IBM MQ 비동기 메시지 이용을 사용하려면 큐 및 주제를 포함하여 다중 메시지 대상에 대한 콜백 루틴으로 알려진 프로그램을 등록할 수 있습니다. 적합한 메시지가 대상으로 전송되면 콜백 루틴으로 전달됩니다. 큐 관리자 정의와 같은 조건을 알리도록 이벤트 핸들러를 설정할 수도 있습니다.

CICS 명령 레벨 애플리케이션 프로그램을 콜백 루틴으로 등록하면 CICS는 IBM MQ에 의해 전달된 메시지를 획득하고 EXEC CICS LINK를 사용하여 콜백 루틴에 링크합니다. CICS는 메시지 데이터를 채널의 컨테이너 세트에 있는 콜백 루틴으로 전달합니다.

콜백루틴으로 사용하는 프로그램은 다음 요구사항을 충족해야 합니다.

- 콜백 루틴은 CICS가 지원하는 모든 프로그래밍 언어일 수 있지만 콜백 루틴에 Java를 사용하면 Java 환경과 비Java 환경 간에 과도한 TCB 전환이 발생하여 성능에 영향을 줍니다.
 - RENT 옵션 및 옵션 AMODE(31), RMODE(ANY)를 사용하여 콜백 루틴을 컴파일하고 링크 편집하십시오.
 - TCB 전환량을 최소화하려면 콜백 루틴을 스레드 안전 표준으로 코딩하고 속성 CONCURRENCY(THREADSAFE) 및 API(CICSAPI)를 사용하여 CICS에 정의하십시오. 콜백 루틴에 스토리지 보호가 활성화된 경우 과도한 TCB 전환이 발생하므로 API(OPENAPI)를 사용하면 성능에 영향을 미칩니다.
- CONCURRENCY(REQUIRED) 및 API(CICSAPI)로 정의된 프로그램을 사용하면 스레드 안전 표준으로 코딩된 프로그램을 개방형 TCB에서 프로그램 시작부터 실행할 수 있습니다. L8 개방형 TCB는 실행 키에 상관없이 사용됩니다.
- 콜백 루틴을 원격 또는 동적이 아닌 로컬 프로그램으로 정의하십시오. CICS가 컨테이너에서 전달하는 데이터 구조에는 리전의 로컬 스토리지에 대한 포인터가 포함되므로 원격 및 동적 프로그램은 콜백 루틴으로 지원되지 않습니다. 원격으로 정의된 콜백 루틴을 등록하려는 시도는 이유 코드 MQRC_MODULE_NOT_FOUND로 실패합니다.

콜백루틴을 위한 CICS 컨테이너

CICS가 콜백 루틴으로 전달하는 컨테이너는 MQ_ASYNC_CONSUME라는 채널을 사용합니다. 채널 이름과 컨테이너 이름은 모두 16자이므로 여기에 제공된 컨테이너 이름은 공백으로 패딩됩니다. 컨테이너는 다음과 같습니다.

표 10. 콜백 루틴에 전달된 메시지 데이터 컨테이너	
컨테이너 이름	컨텐츠
MsgDesc	MQMD2 - 메시지 디스크립터 버전 2
GetMsgOpts	MQGMO - 메시지 옵션 가져오기
버퍼	메세지, 자르기 및 변환과 같은 MQGMO에 지정된 옵션은 MQGET 호출에서와 동일한 방식으로 메시지에 영향을 줍니다.
컨텍스트	MQCBC - 콜백 컨텍스트 버전 2

IBM MQ 제품 문서는 MQMD, MQGMO 및 MQCBC 데이터 구조를 설명합니다. [IBM MQ 제품 문서](#)의 내용을 참조하십시오.

콜백 루틴이 메시지 데이터 컨테이너에서 데이터를 검색할 때 데이터 구조의 크기를 나중에 변경할 수 있도록 GET CONTAINER INTO 명령 대신 GET CONTAINER SET 명령을 사용하십시오. FLENGTH 키워드를 코딩하고

값이 0인지 테스트하여 빈 컨테이너를 확인하십시오. 예를 들면, 콜백 루틴의 이벤트 호출의 경우 MsgDesc, GetMsgOpts 및 버퍼 컨테이너가 비어 있습니다.

콜백 루틴에서 동기점, 이상 종료 및 정지 처리

콜백 루틴이 CICS-MQ 어댑터와 올바르게 상호작용하는지 확인하려면 다음 요구사항이 충족되는지 확인하십시오.

1. EXEC CICS SYNCPOINT 및 EXEC CICS SYNCPOINT ROLLBACK 명령을 사용하여 메시지에 대해 시작될 때 메시지 사용자 콜백 루틴에서만 사용하십시오. 이벤트 핸들러에서 또는 콜백 루틴을 호출하여 등록, 등록 취소, 시작 또는 중지할 때 이러한 명령을 사용하지 마십시오.
2. 콜백 루틴에서 EXEC CICS ABEND 명령을 사용하는 경우 이상 종료 처리되고 제어가 호출자로 리턴되므로 MQCTL 명령이 이유 코드 MQRC_CALLBACK_ERROR(2452)로 실패합니다. 하나의 콜백 루틴에서 이상 종료는 MQCTL 요청을 발행한 CICS 태스크에서 관리 중인 모든 콜백 루틴의 호출을 중지합니다.
3. 이벤트 핸들러 콜백 루틴에서 큐 관리자 또는 연결이 정지될 때 MQCTL STOP 명령을 발행하여 CICS-MQ 인터페이스가 정지되도록 하십시오. 이 명령을 발행하지 않으면 활성 이벤트 핸들러로 인해 CICS-MQ 인터페이스를 정지할 수 없으며 CICS-MQ 인터페이스를 강제로 닫아야 합니다.
4. MQCB를 사용하여 콜백 루틴을 등록하는 경우:
 - a. CallBackDesc에서(MQCB) 매개변수의 옵션 필드에서 MQCBDO_FAIL_IF QUIESCING을 설정하십시오. CICS는 처리 작업의 일부로 MQCB 명령을 발행하여 새 작업 도달을 중지해야 합니다.
 - b. GetMsgOpts(MQGMO) 매개변수에서 콜백 루틴이 올바르게 정지되도록 옵션 필드에서 MQGMO_FAIL_IF QUIESCING을 설정하십시오. 이러한 정지 옵션을 올바르게 설정하지 않으면 활성 이벤트 핸들러로 인해 CICS-MQ 인터페이스를 정지할 수 없으며 CICS-MQ 인터페이스를 강제로 닫아야 합니다.
 - c. CallBackDesc(MQCB) 매개변수에서 CallbackArea 필드를 사용하여 저장영역을 콜백 루틴으로 전달하려면 프로그램의 동적 저장영역이 아닌 EXEC CICS GETMAIN을 사용하여 얻은 저장영역 주소로 설정하십시오. 콜백 루틴은 콜백 컨텍스트 구조 MQCBC의 CallbackArea 필드를 사용하여 이 저장영역에 액세스합니다. 또한 프로그램이 종료되면 저장영역 또는 고정된 추가 저장 영역을 해제하지 말고 CICS가 태스크 종료 시 저장 영역을 비우십시오. 이를 통해 CICS가 태스크 처리 종료 시 등록 취소를 위해 콜백 루틴을 호출하는 경우 저장영역을 사용할 수 있습니다.
5. 콜백 루틴의 호출을 제어하기 위해 MQCTL을 사용하는 경우 ControlOpts(MQCTLO) 매개변수에서 모든 콜백 루틴에 의해 공유될 저장 영역을 전달하기 위해 ConnectionArea 필드를 사용하려면 프로그램의 동적 메모리 대신에 EXEC CICS GETMAIN를 사용하여 확보한 저장영역의 주소로 설정하십시오. 콜백 루틴은 콜백 컨텍스트 구조 MQCBC의 ConnectionArea 필드를 사용하여 이 저장영역에 액세스합니다. 또한 프로그램이 종료되면 저장영역 또는 고정된 추가 저장 영역을 해제하지 말고 CICS가 태스크 종료 시 저장 영역을 비우십시오. 이를 통해 CICS가 태스크 처리 종료 시 등록 취소를 위해 콜백 루틴을 호출하는 경우 저장영역을 사용할 수 있습니다.

비동기 메시지 이용을 위한 샘플 프로그램

COBOL의 이 샘플 프로그램 세트는 사용자에게 IBM MQ 비동기 메시지 이용과 콜백 루틴이 CICS 환경에서 작동하는 방법을 보여줍니다. IBM MQ 비동기 메시지 이용을 사용하려면 버전 7 이상의 IBM MQ가 있어야 합니다. 샘플 프로그램이 IBM MQ와 함께 제공됩니다.

이벤트 핸들러이고 두 개의 메시지 이용자인 세 개의 콜백 루틴을 등록하려면 샘플 등록 클라이언트 프로그램을 실행하고 비동기 메시지 이용을 시작하십시오. 그런 다음 샘플 메시징 클라이언트 프로그램을 사용하여 두 메시지 이용자가 비동기식으로 이용할 수 있도록 CICS 터미널에서 큐 및 IBM MQ의 주제로 메시지를 보낼 수 있습니다. 이러한 메시지를 사용하여 메시지 이용자 중 하나에 콜백 루틴의 등록 취소, 재개 또는 일시중단 또는 비동기 메시지 이용 중지를 지시할 수 있습니다.

표 11. 비동기 메시지 이용을 위한 샘플 COBOL 프로그램		
샘플 프로그램	기능	조치
CSQ4CVRG	등록 클라이언트	이벤트 핸들러와 메시지 이용자를 등록합니다. CICS 트랜잭션 MVRG에서 실행됩니다.

표 11. 비동기 메시지 이용을 위한 샘플 COBOL 프로그램 (계속)		
샘플 프로그램	기능	조치
CSQ4CVPT	메시징 클라이언트	메시지를 주제로 공개하거나 제어 메시지를 큐로 보냅니다. CICS 트랜잭션 MVMP에서 실행됩니다.
CSQ4CVCN	기본 메시지의 메시지 이용자	주제 뉴스/미디어/동영상에 공개되는 IBM MQ 메시지를 이용합니다.
CSQ4CVCT	제어 메시지의 메시지 이용자	SAMPLE.CONTROL.QUEUE 큐에서 사용자의 제어 메시지를 이용하고, 비동기 메시지 이용을 중지하거나 콜백 루틴을 등록 취소하거나, 재개하거나 일시중단하도록 적절한 MQCB 또는 MQCTL 명령을 발행합니다.
CSQ4CVEV	이벤트 핸들러	큐 관리자 중지 또는 정지와 같은 전체 콜백 환경에 영향을 미치는 조건이 발생할 때 IBM MQ의 알림을 받습니다.

비동기 메시지 이용을 위한 샘플 디자인

샘플 COBOL 프로그램에는 CICS 트랜잭션에서 실행되는 등록 클라이언트, 메세징 클라이언트 및 CICS-MQ에서 메시지 또는 이벤트를 이용하는 세 개의 콜백 루틴이 포함되어 있습니다. 각 샘플 프로그램은 해당 작동을 관찰할 수 있도록 적절한 지점에서 메시지를 표시하도록 COBOL DISPLAY문을 사용합니다.

샘플 프로그램의 메시지는 CEEMSG 데이터 세트인 트랜지언트 데이터 큐 CESE로 전송됩니다. 이벤트 핸들러 및 메시지 이용자가 발행한 메시지는 프로그램이 구동된 CICS-MQ 호출 유형(예: START 또는 REGISTER)을 표시합니다.

등록 클라이언트 프로그램 CSQ4CVRG

등록 클라이언트는 CICS 트랜잭션 MVRG 아래 CICS 터미널에서 시작되지만 입력을 사용하지 않습니다. 등록 클라이언트는 IBM MQ를 사용하여 이벤트 핸들러 CSQ4CVEV, 메시지 이용자 CSQ4CVCN 및 메시지 이용자 CSQ4CVCT를 콜백 루틴으로서 등록합니다. 또한 두 개의 메시지 이용자와 연관된 오브젝트 핸들과 함께 세 개의 등록된 콜백 루틴의 이름을 포함하는 데이터 구조를 CSQ4CVCT에 전달합니다.

등록 클라이언트가 콜백 루틴을 등록하면 비동기 메시지 이용을 시작하도록 MQCTL START_WAIT 명령을 발행합니다. 그런 다음 제어가 리턴될 때까지 일시중단됩니다.

콜백 루틴 중 하나가 MQCTL STOP 명령을 발행하여 비동기 메시지 이용을 중지하면 제어가 등록 클라이언트로 리턴됩니다.

메세징 클라이언트 프로그램 CSQ4CVPT

메세징 클라이언트는 CICS 트랜잭션 MVMP 아래 CICS 터미널에서 시작되고 명령행 입력을 사용합니다. 메세징 클라이언트에는 두 개의 기능이 있습니다.

- CSQ4CVCN에서 이용할 주제 News/Media/Movies에 기본 텍스트 메시지를 공개합니다.
- CSQ4CVCT에서 이용할 큐 SAMPLE.CONTROL.QUEUE에 제어 메시지를 넣습니다. 제어 메시지는 CSQ4CVCT에서 샘플 프로그램 세트의 작동을 변경하는 명령을 발행하게 하며 샘플 프로그램으로 표시되는 메시지를 통해 결과를 관찰할 수 있습니다.

102 페이지의 『CSQ4CVPT, 샘플 메세징 클라이언트』에서는 메시지를 작성하기 위해 CSQ4CVPT를 사용하는 방법을 설명합니다.

기본 메시지의 경우, 메시지 이용자 프로그램 CSQ4CVCN

CSQ4CVCN은 메세징 클라이언트를 사용하여 주제 뉴스/미디어/동영상 아래에 공개한 기본 IBM MQ 메시지를 이용하는 메시지 이용자입니다.

CSQ4CVCN이 IBM MQ 호출 유형 MSG_REMOVED로 호출될 때 인바운드 메시지를 검색하고 CICS 작업 로그에 에코합니다.

제어 메시지의 경우, 메시지 이용자 프로그램 CSQ4CVCT

CSQ4CVCT는 큐 SAMPLE.CONTROL.QUEUE에 넣는 IBM MQ 메시지를 이용하는 메시지 이용자입니다. CSQ4CVCT가 IBM MQ 호출 유형 MSG_REMOVED로 호출될 때 인바운드 메시지를 검색하고 등록 클라이언트가 전달한 데이터 구조도 검색합니다.

메세징 클라이언트를 사용하여 CSQ4CVCT가 다음과 같은 조치를 수행하도록 지시하기 위해 제어 메시지를 작성할 수 있습니다. CSQ4CVCT는 적절한 MQCB 또는 MQCTL 명령을 발행하여 요청한 조치를 수행합니다.

- 지정된 콜백 루틴의 등록 취소
- 지정된 콜백 루틴의 재개
- 지정된 콜백 루틴의 일시중단
- MQCTL STOP 명령을 발행하고 등록 클라이언트에 대한 제어를 리턴하여 비동기 메시지 이용 중지

CSQ4CVCT가 자체를 일시중단하거나 등록 취소하도록 지시할 수 있지만 그렇게 하면 더 이상 샘플 프로그램의 작동을 제어할 수 없습니다. 모든 콜백 루틴을 일시중단하거나 등록 취소하면 등록 클라이언트에 제어가 리턴되고 태스크가 종료됩니다.

이벤트 핸들러 프로그램 CSQ4CVEV

이벤트 핸들러로서, CSQ4CVEV는 큐 관리자 중지 또는 정지와 같은 전체 콜백 환경에 영향을 미치는 조건이 발생할 때 IBM MQ 알림을 받습니다.

이유 코드 CONNECTION_QUIESCING이 수신되면 CSQ4CVEV는 MQCTL 중지 명령을 발행하여 비동기 메시지 이용을 중지하고 등록 클라이언트에 제어를 리턴합니다.

비동기 메시지 이용에 대한 샘플 설정

IBM MQ 비동기 메시지 이용과 콜백 루틴을 시연하기 위한 샘플 프로그램을 설정합니다.

이 태스크 정보

샘플 프로그램 CSQ4CVRG, CSQ4CVPT, CSQ4CVCN, CSQ4CVCT 및 CSQ4CVEV가 IBM MQ와 함께 제공됩니다. IBM MQ 버전 7.0.1의 경우, 샘플을 얻으려면 APAR PM06722를 위한 PTF를 적용하십시오. 샘플 소스는 IBM MQ 라이브러리 SCSQCOBS에 있으며 로드 모듈은 IBM MQ 라이브러리 SCSQCICS에 있습니다.

샘플 프로그램에 대한 CICS 자원 정의는 IBM MQ 그룹 CSQ4SAMP에서 제공됩니다. CSQ4SAMP에는 CICS 트랜잭션 MVRG 및 MVMP에 대한 자원 정의도 포함되어 있으며, 이는 각각 CSQ4CVRG(등록 클라이언트) 및 CSQ4CVPT(메시징 클라이언트) 프로그램에 사용됩니다.

프로시저

1. CICS 프로시저의 IBM MQ 라이브러리 *thlqual*.SCSQCICS를 DFHRPL 연결에 포함시키고 상위 레벨 규정자 *thlqual*이 IBM MQ 버전 7.0.1 이상인지 확인하십시오.
이 라이브러리를 CICS 라이브러리 뒤에 포함시켜 올바른 코드가 사용되는지 확인하십시오.
2. CICS에서 다음과 같이 샘플 프로그램 및 트랜잭션 MVRG와 MVMP를 위해 IBM MQ에 의해 제공된 자원 정의를 설치하십시오.
 - a) CICS 리전에서 기존 CSQ4SAMP 그룹을 삭제하십시오.
 - b) IBM MQ 버전 7.0.1에 대한 PTF를 적용하는 경우, PTF와 함께 제공된 라이브러리 SCSQPROC에서 멤버 CSQ4S100의 버전을 사용하십시오.
이 멤버에는 새 자원 정의가 포함됩니다.
 - c) DFHCSDUP를 사용하여 CSQ4SAMP 그룹에 채우는 데이터 세트 *thlqual*.SCSQPROC (CSQ4S100)로 CSD를 업데이트하십시오.
 - d) CSQ4SAMP 그룹을 설치하십시오.
CEDA 트랜잭션을 사용하여 CICS-MQ 어댑터 자원을 활성화 CICS 시스템에 설치하는 경우, 먼저 어댑터를 종료하고 경보 모니터가 작동을 멈출 때까지 대기하십시오.
3. 메시지 이용자 CSQ4CVCT가 사용하는 큐 SAMPLE.CONTROL.QUEUE를 CICS 리전과 연관된 IBM MQ 큐 관리자 또는 큐 공유 그룹에 정의하십시오.

CICS 리전의 MQCONN 자원 정의는 IBM MQ 큐 관리자 또는 큐 공유 그룹을 이름 지정합니다. 큐 이름은 샘플 프로그램에서 코드화됩니다.

4. 옵션: IBM MQ 큐 관리자 또는 큐 공유 그룹에 주제 News/Media/Movies를 정의하십시오.
주제를 정의하지 않은 경우 IBM MQ는 기본 관리 오브젝트 아래 런타임에서 작성합니다.
큐 이름과 같이 주제는 샘플 프로그램에서 코드화됩니다.
5. CICS 트랜잭션 MVRG 아래에서 샘플 등록 클라이언트 프로그램 CSQ4CVRG를 시작하십시오.
등록 클라이언트는 IBM MQ를 사용하여 이벤트 핸들러 CSQ4CVEV, 메시지 이용자 CSQ4CVCN 및 메시지 이용자 CSQ4CVCT를 콜백 루틴으로서 등록합니다.

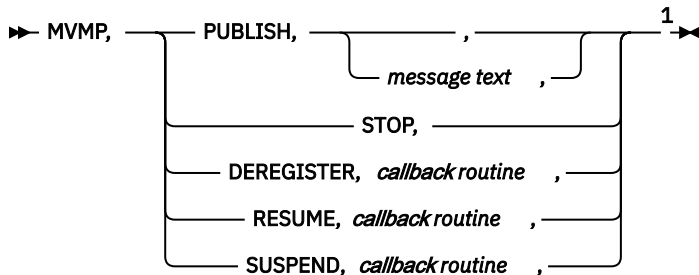
다음에 수행할 작업

이제 샘플 메시징 클라이언트 프로그램 CSQ4CVPT를 사용하여 기본 메시지를 공개하고 제어 메시지를 작성하고 효과를 관찰할 수 있습니다. 102 페이지의 『CSQ4CVPT, 샘플 메시징 클라이언트』에서는 CSQ4CVPT를 사용하는 방법에 대해 설명합니다.

CSQ4CVPT, 샘플 메시징 클라이언트

CICS 트랜잭션 MVMP 아래 CICS 터미널에서 샘플 메시징 클라이언트 프로그램 CSQ4CVPT를 사용하십시오.

메시징 클라이언트 명령



참고:

¹ 키워드 및 콜백 루틴의 이름은 대소문자를 구분하지 않습니다. 입력 필드는 위치이며 쉼표로 구분되고 종료됩니다.

매개변수

MVMP

CSQ4CVPT을 위한 CICS 트랜잭션.

PUBLISH , { , | message text , }

지정된 메시지 텍스트를 주제 뉴스/미디어/동영상 아래 CICS-MQ 보유 메시지로 공개합니다. 메시지는 메시지 이용자 프로그램 CSQ4CVCN에 의해 이용됩니다. 메시지 텍스트를 생략하고 닫는 쉼표를 지정하면 기본 메시지 텍스트가 공개됩니다.

DEREGISTER , callback routine ,

CICS-MQ 메시지를 SAMPLE.CONTROL.QUEUE 큐에 넣어 메시지 이용자 프로그램 CSQ4CVCT가 지정된 콜백 루틴을 등록 취소하도록 지시합니다.

- CSQ4CVCT를 등록 취소하면, 더 이상 샘플 프로그램의 작동을 제어할 수 없습니다.
- 모든 콜백 루틴을 등록 취소하면 제어가 등록 클라이언트로 리턴됩니다.

RESUME , callback routine ,

CICS-MQ 메시지를 SAMPLE.CONTROL.QUEUE 큐에 넣어 메시지 이용자 프로그램 CSQ4CVCT가 지정된 콜백 루틴을 재개하도록 지시합니다.

SUSPEND , callback routine ,

CICS-MQ 메시지를 SAMPLE.CONTROL.QUEUE 큐에 넣어 메시지 이용자 프로그램 CSQ4CVCT가 지정된 콜백 루틴을 일시중단하도록 지시합니다.

- CSQ4CVCT를 일시중단하면, 더 이상 샘플 프로그램의 작동을 제어할 수 없습니다.
- 모든 콜백 루틴을 일시중단하면 제어가 등록 클라이언트로 리턴됩니다.

STOP ,

CICS-MQ 메시지를 SAMPLE.CONTROL.QUEUE 큐에 넣고 MQCTL STOP 명령을 발행하여 메시지 이용자 프로그램 CSQ4CVCT가 비동기 메시지 이용을 중지하도록 지시합니다. 이 조치는 제어를 등록 클라이언트 CSQ4CVRG로 리턴합니다.

CSQ4CVCN이 사용할 기본 메시지 텍스트를 공개하려면 다음 명령을 사용하십시오.

```
MVMP,PUBLISH,,
```

CSQ4CVCN이 사용할 수 있도록 메시지 텍스트 A short message를 공개하려면 다음 명령을 사용하십시오.

```
MVMP,publish,A short message,
```

비동기 메시지 이용을 중지하려면 다음 명령을 사용하십시오.

```
MVMP,STOP,
```

이벤트 핸들러 CSQ4CVEV를 등록 취소하려면 다음 명령을 사용하십시오.

```
MVMP,DEREGISTER,CSQ4CVEV,
```

메시지 이용자 CSQ4CVCN을 재개하려면 다음 명령을 사용하십시오.

```
MVMP, resume,csq4cvcn,
```

이벤트 핸들러 CSQ4CVEV를 일시중단하려면 다음 명령을 사용하십시오.

```
MVMP,SUSPEND,CSQ4CVEV,
```

CICS-MQ API 교차 엑시트

CICS는 CICS-MQ 어댑터에 사용할 API 교차 엑시트를 제공합니다. 이 엑시트를 사용하여 모니터링, 테스트, 유지보수 또는 보안 용도로 IBM MQ 호출이 실행될 때 인터셉트할 수 있습니다. 이 엑시트는 CICS 주소 공간에서 실행됩니다.

이 절에는 제품에 민감한 프로그래밍 인터페이스 정보가 포함되어 있습니다.

다음과 같은 용도로 CICS-MQ API 교차 엑시트를 사용할 수 있습니다.

- 각 MQI 호출 전후에 각 메시지의 콘텐츠를 검사하여 추가 보안 검사를 수행합니다.
- 메시지에 제공된 큐 이름을 다른 큐 이름으로 대체합니다.
- 호출을 취소하고 리턴 코드 0을 발행하여 성공적인 호출을 시뮬레이션하거나 다른 값을 호출하여 호출이 수행되지 않았음을 표시합니다.
- 애플리케이션에서 MQI 호출의 사용을 모니터링합니다.
- 통계를 수집합니다.
- 특정 호출에서 입력 매개변수를 수정합니다.
- 특정 호출의 결과를 수정합니다.

CICS-MQ API 교차 엑시트를 사용하면 IBM MQ 성능이 저하됩니다. 사용을 신중하게 계획하십시오.

CICS-MQ API 엑시트 프로그램을 CSQCAPX하고 합니다. CICS는 샘플 엑시트 프로그램 및 그룹 DFHMQ에서 CSQCAPX에 대한 프로그램 정의를 제공합니다. 엑시트를 사용하려면 이 프로그램은 DFHRPL 연결에 있어야 하며 CICS 시스템 정의 파일(CSD)에 정의되어 있어야 하고 사용으로 설정되어야 합니다. 제공된 샘플 대신 사용자 고유의 엑시트 프로그램을 작성할 수 있으며 프로그램 CSQCAPX도 호출해야 합니다.

CSQCAPX가 로드되면 CICS-MQ 어댑터 제어판, CKQC 또는 콘솔에 확인 메시지가 작성됩니다. 로드할 수 없으면 진단 메시지가 표시되지만 그렇지 않으면 애플리케이션 프로그램이 정상적으로 실행됩니다.

중요사항: 제공된 CSQCAPX 정의는 매개변수 CONCURRENCY(THREADSAFE)를 지정합니다. 사용자 고유의 엑시트 프로그램을 작성하는 경우 정의할 때 CONCURRENCY(THREADSAFE)를 지정하고 엑시트에 스레드 안전 CICS 명령만 사용하십시오. 또한 이 설정을 지정하고 엑시트 프로그램이 호출하는 모든 프로그램에 스레드 안전

CICS 명령만 사용하십시오. 이전 CICS 릴리스로 작성된 API 교차 엑시트를 검사하여 해당 로직이 스레드 안전한 지 확인하십시오.

CICS-MQ API 교차 엑시트 호출 방법

CICS-MQ API 교차 엑시트를 사용하면 z/OS용 IBM MQ의 성능이 저하되므로 사용을 신중하게 계획하십시오. 사용으로 설정된 경우 CICS-MQ API 교차 엑시트는 다음 환경에서 호출됩니다.

- CSQCAPX는 CICS-MQ 어댑터를 사용하는 모든 애플리케이션에 의해 호출됩니다.
- CSQCAPX는 다음 MQI 호출 중 하나가 작성될 때마다 호출됩니다.

- MQCB
- MQCTL
- MQCLOSE
- MQGET
- MQINQ
- MQOPEN
- MQPUT
- MQPUT1
- MQSET
- MQSTAT
- MQSUB
- MQSUBRQ

CSQCAPX는 호출 전후에 호출됩니다.

- 비동기 메시지 이용을 처리하는 동안 CSQCAPX는 세 가지 상황에서 호출됩니다.
 - IBM MQ 루틴 CSQAVICD를 호출하기 전, 콜백 루틴에 메시지를 전달하기 전에 메시지에 데이터 변환이 필요한 경우입니다.
 - 콜백 루틴에 대한 호출 전후에.

이러한 호출의 경우 CSQCAPX에 전달된 매개변수 목록의 MQXP_PCOPYPARM 필드가 참조하는 IBM MQ MQI 매개변수 목록에는 콜백 루틴(MQMD, MQGMO, 버퍼 및 컨텍스트)으로 전달되는 동일한 데이터가 포함됩니다.

메시지 특성 및 메시지 핸들(MQCRTMH, MQDLTMH, MQSETMP, MQINQMP, MQDLTMP, MQMHBUF 및 MQBUFMH)에 대한 함수 호출에 대해 CSQCAPX가 호출되지 않습니다.

TCB 전환의 과도한 사용을 줄이려면 엑시트 프로그램이 THREADSAFE로 정의되어 있는지 확인해야 합니다.

엑시트 프로그램은 호출이 실행되기 전에 한 번 호출되고 호출이 실행된 후에 한 번 호출됩니다. 엑시트 호출의 "이전" 유형에서 엑시트 프로그램은 MQI 호출의 매개변수를 수정하거나 호출을 완전히 비활성화하거나 호출을 처리할 수 있습니다. 호출이 처리되면 호출이 완료한 후 엑시트가 다시 호출됩니다.

엑시트 프로그램은 순환적이지 않습니다. 엑시트 안에 작성된 MQI 호출은 엑시트 프로그램을 두 번 호출하지 않습니다.

CICS-MQ 교차 엑시트 프로그램과의 통신

호출된 후 CICS-MQ 교차 엑시트 프로그램은 DFHEICAP라는 필드가 가리키는 CICS COMMAREA(COMMunication AREA)의 매개변수 목록을 전달합니다.

CICS 실행 인터페이스 블록 필드 EIBCALEN은 이 영역의 길이를 표시합니다. 이 COMMAREA(COMMunication AREA)의 구조는 다음과 같이 교차 엑시트 프로그램에서 정의됩니다.

* 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)
*
```

필드 *MQXP_PXPB* 는 엑시트 매개변수 블록, *MQXP*를 가리킵니다.

필드 *MQXP_PCOPYPARM* 는 호출 매개변수 주소의 배열입니다. 예를 들어, 애플리케이션에서 매개변수 P1, P2 또는 P3을 사용하여 MQI 호출을 실행하면, COMMAREA(COMMunication AREA)에는 다음이 포함됩니다.

```
PXPB, PP1, PP2, PP3
```

여기서 *P* 는 포인터(주소)를 나타내며 *XPB*는 엑시트 매개변수 블록입니다.

사용자 고유의 CICS-MQ API 교차 엑시트 프로그램 작성

사용자 고유의 프로그램을 위한 프레임워크로서 CICS와 함께 제공되는 샘플 API 교차 엑시트 프로그램을 사용할 수 있습니다. 엑시트 프로그램을 스레드 안전으로 작성하고 엑시트 프로그램을 스레드 안전으로 선언해야 합니다.

이 태스크 정보

성능상의 이유로 프로그램을 어셈블러 언어로 작성하십시오. CICS가 지원하는 다른 언어로 작성하는 경우, 사용자 고유의 데이터 정의 파일을 제공해야 합니다. 프로그램을 AMODE(31) 및 RMODE(ANY)로 링크 편집하십시오. 교차 엑시트 프로그램으로 전달되는 MQXP 매개변수 목록의 레이아웃은 프로그램에서 정의됩니다.

프로시저

- 엑시트는 CICS-MQ 코드의 확장으로 작성됩니다. 엑시트가 MQI를 사용하는 z/OS용 IBM MQ 프로그램 또는 트랜잭션을 중단시키지 않는지 확인하십시오.
CSQ, DFH 또는 CK의 접두부로 표시됩니다.
- 프로그램은 CICS 태스크 관련 사용자 엑시트 프로그램이 사용할 수 있는 모든 API(예: IMS 및 Db2®와 CICS)를 사용할 수 있습니다.
엑시트 프로그램 및 엑시트 프로그램이 호출하는 프로그램에서 스레드 안전 CICS 명령만 사용해야 합니다.
- 프로그램은 MQCONN, MQCONNX 및 MQDISC를 제외한 모든 MQI 호출을 사용할 수 있습니다.
그러나 엑시트 프로그램의 MQI 호출은 엑시트 프로그램을 두 번 호출하지 않습니다.
- MQI 호출 후에 호출되면 프로그램은 호출에 의해 설정된 완료 및 이유 코드를 검사하고 수정할 수 있습니다.
- 애플리케이션이 발행한 MQI 호출 이름을 찾으려면 MQXP 구조의 **ExitCommand** 필드를 검사하십시오. 호출에서 매개변수의 수를 찾으려면 **ExitParmCount** 필드를 조사하십시오.
16바이트 **ExitUserArea** 필드를 사용하여 애플리케이션이 확보한 동적 저장영역의 주소를 저장할 수 있습니다. 이 필드는 엑시트 사용에서 유지되며 CICS 태스크와 동일한 수명을 갖습니다.
- 프로그램은 **ExitResponse** 필드에서 MQXCC_SUPPRESS_FUNCTION 또는 MQXCC_SKIP_FUNCTION을 리턴하여 MQI 호출의 실행을 비활성화할 수 있습니다.
호출이 실행되고 호출이 완료된 후 엑시트 프로그램이 다시 호출되려면 엑시트 프로그램이 MQXCC_OK를 리턴해야 합니다.
- 프로그램은 **EXEC CICS SYNCPOINT** 또는 **EXEC CICS SYNCPOINT ROLLBACK** 명령을 발행할 수 있습니다. 이러한 명령은 엑시트가 사용된 시점까지 태스크가 수행한 모든 업데이트를 커밋하거나 롤백합니다.

- 매개변수 MQXP_EXITCALLPROG는 commarea를 통해 모듈로 전달되는 parmlist에 포함되며 모듈을 호출한 프로그램의 이름을 포함합니다. CSQCAPX는 URM(User Replaceable Module)으로 호출되므로 EXEC CICS ASSIGN INVOKINGPROG 명령은 공백을 리턴합니다. EXEC CICS ASSIGN INVOKINGPROG 명령에서 수신된 출력 형식과 동일한 형식으로 호출 프로그램의 이름을 포함하므로 매개변수 MQXP_EXITCALLPROG를 대신 사용할 수 있습니다. 이 매개변수는 MXQP_VERISON 매개변수의 값이 2인 경우 사용할 수 있습니다.
- 프로그램은 **EXEC CICS RETURN** 명령을 실행하여 종료해야 합니다. XCTL 명령으로 제어를 전송하지 않아야 합니다.
- 엑시트 프로그램이 호출하는 엑시트 프로그램 및 프로그램에 대한 PROGRAM 자원 정의를 작성하십시오. 엑시트 프로그램 CSQCAPX를 호출해야 합니다.

PROGRAM 자원 정의에서 다음 설정을 지정하십시오.

- a) 엑시트 프로그램 및 엑시트 프로그램이 호출하는 모든 프로그램에 대해 CONCURRENCY(THREADSAFE) 또는 CONCURRENCY(REQUIRED)를 지정하십시오.
엑시트 프로그램 및 엑시트 프로그램이 호출하는 프로그램에서 스레드 안전 CICS 명령만 사용해야 합니다.
- b) CICS Transaction Server 저장영역 보호 기능을 사용 중인 경우, 엑시트 프로그램 및 제어를 통과하는 모든 프로그램에 대해 EXECKEY(CICS)를 지정하여 CICS 실행 키에서 실행되도록 하십시오.
추가 정보는 [엑시트 프로그램 및 CICS 스토리지 보호 기능의 내용](#)을 참조하십시오.
- c) 통신 영역의 매개변수는 주소이므로 엑시트 프로그램을 원격 프로그램이 아닌 CICS 시스템의 로컬로 정의해야 합니다.

CICS는 그룹 DFHMQ에서 CSQCAPX에 대한 샘플 프로그램 정의를 제공합니다. 이 프로그램 정의는 CONCURRENCY(THREADSAFE) 및 EXECKEY(CICS)를 지정합니다. 엑시트 프로그램에 대한 프로그램 정의를 위한 기초로서 이 샘플을 사용할 수 있습니다.

결과

CICS 시스템은 CICS가 z/OS용 IBM MQ에 연결될 때 엑시트 프로그램을 로드하려고 합니다. 이 시도가 성공하면 DFHMQ0301I 메시지가 CKQC 패널 또는 시스템 콘솔로 전송됩니다. 예를 들면 로드 모듈이 DFHRPL 연결의 라이브러리에 없는 경우 로드가 성공하지 못하면, 메시지 DFHMQ0315E는 CKQC 패널 또는 시스템 콘솔로 전송됩니다.

샘플 API 교차 엑시트 프로그램, CSQCAPX

샘플 엑시트 프로그램은 어셈블러 언어 프로그램으로서 제공됩니다. 소스 파일(CSQCAPX)은 *thlqual.SDFHSAMP*로 제공되며, 여기서 *thlqual*은 설치에 사용되는 상위 레벨 규정자입니다. 이 소스 파일에는 프로그램 로직을 설명하는 의사 코드가 포함됩니다.

샘플 프로그램에는 사용자 고유의 엑시트 프로그램을 작성할 때 사용할 수 있는 초기화 코드 및 레이아웃이 포함되어 있습니다.

샘플은 이러한 태스크를 수행하는 방법을 보여줍니다.

- 엑시트 매개변수 블록 설정
- 호출 및 엑시트 매개변수 블록 주소 지정
- 엑시트가 호출되는 MQI 호출 판별
- MQI 호출 처리 후 엑시트가 호출되는지 판별
- CICS 임시 저장영역 큐에 메시지를 배치하십시오.
- 재진입을 유지하도록 동적 저장영역 획득에 매크로 DFHEIENT 사용
- CICS exec 제어 블록에 DFHEIBLK 사용
- 오류 조건 트랩
- 호출자로 제어 리턴

샘플 엑시트 프로그램은 CICS 임시 저장영역 큐(CSQ1EXIT)에 메시지를 기록하여 엑시트 오퍼레이션을 표시합니다. 메시지는 MQI 호출 후 엑시트가 호출되는지 여부를 표시합니다. MQI 호출 후 엑시트가 호출되면 메시

지에 호출이 리턴한 완료 코드 및 이유 코드가 포함됩니다. 샘플은 CMQXA 매크로의 이름 지정된 상수를 사용하여 입력 유형을 확인합니다(즉, 해당 호출 전후에).

샘플은 모니터링을 수행하지 않지만, 시간소인이 있는 메시지를 처리 중인 호출 유형을 표시하는 CICS 큐에 배치합니다. 메시지는 MQI의 성능과 엑시트 프로그램의 올바른 작동을 표시합니다.

샘플 엑시트 프로그램은 프로그램이 실행되는 동안 작성된 각 MQI 호출에 대해 6개의 EXEC CICS 호출을 실행합니다. 이 엑시트 프로그램을 사용하면 CICS-MQ 애플리케이션의 성능이 저하됩니다.

CICS-MQ API 교차 엑시트 사용

CSQCAPX에 대해 제공된 프로그램 정의는 STATUS(DISABLED)를 지정합니다. 엑시트 프로그램을 사용하면 tly reduce IBM MQ 성능이 크게 저하될 수 있으므로 이 프로그램은 사용 안함 상태로 설치됩니다. API 교차 엑시트를 사용하려면 먼저 사용 가능하게 설정해야 합니다.

이 태스크 정보

CICS-MQ 어댑터 제어판, 명령행 또는 CICS 애플리케이션 프로그램을 사용하여 임시로 API 교차 엑시트를 활성화할 수 있습니다. CICS 자원 정의를 수정하여 API 교차 엑시트를 영구적으로 사용 설정할 수 있습니다.

프로시저

- CICS-MQ 어댑터 제어판을 사용하여 API 교차 엑시트를 임시로 활성화하려면 다음 단계를 수행하십시오.
 - CKQC를 입력하고 Enter를 눌러 CICS-MQ 어댑터 제어판에 액세스하십시오.
 - 메뉴 표시줄에서 **Connection**을 선택하십시오.
 - 메뉴에서 **Modify** 조치를 선택하십시오.
 - Modification Options** 보조 매개변수 창에서 **Enable API Exit**를 선택하고 Enter를 눌러 API 교차 엑시트의 상태를 사용으로 변경하십시오.
 화면에 이 프로세스가 다음과 같이 표시됩니다.

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		
		Modification Options 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

- CICS 명령행에서 API 교차 엑시트를 임시로 활성화하려면 이러한 방법 중 하나를 선택하십시오.
 - CICS 마스터 터미널에서 명령 CEMT S PROGRAM(CSQCAPX) ENABLED를 발행하십시오.
 - 명령행에서 명령 CKQC MODIFY N E를 발행하십시오. 옵션 E는 연결을 사용으로 설정합니다. 옵션 N은 연결 통계와 관련이 있고 필수지만 효과는 없습니다.
 - CICS 애플리케이션 프로그램을 사용하여 API 교차 엑시트를 임시로 활성화하려면 EXEC CICS LINK 명령을 발행하여 어댑터 재설정 프로그램, DFHMQRS(또는 호환성을 위해 유지되는 CSQCRST)에 링크하고 옵션이 E인 CKQC MODIFY 명령을 발행하십시오.
- 다음 예는 이를 수행하는 방법을 보여줍니다.

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

MODIFY 명령은 4개의 후행 공백과 구분자로 다른 공백을 채워야 합니다(CICS(r) 애플리케이션 프로그램의 CKQC 명령 참조). 옵션 N은 연결 통계와 관련이 있고 필수지만 효과는 없습니다.

- API 교차 엑시트가 영구적으로 사용 설정된 상태로 실행하려면 다음 단계를 수행하십시오.
 - a) CICS-제공 그룹 DFHMQ에서 CSQCAPX의 정의를 사용자 고유의 그룹으로 복사하십시오.
 - b) 상태를 DISABLED에서 ENABLED로 변경하여 CSQCAPX 정의를 변경하고 새 그룹을 설치하십시오.
 - c) 그룹이 grouplist의 DFHMQ 뒤에 설치되었는지 확인하십시오.

CICS-MQ API 교차 엑스트 사용 안함

IBM MQ 성능을 향상시키기 위해 더 이상 필요하지 않을 때 CICS-MQ API 교차 엑스트를 사용 안함으로 설정할 수 있습니다.

이 태스크 정보

CICS-MQ 어댑터 제어판, 명령행 또는 CICS 애플리케이션 프로그램을 사용하여 API 교차 엑시트를 사용 안함으로 설정할 수 있습니다.

프로시저

1. CICS-MQ 어댑터 제어판을 사용하여 API 교차 엑시트를 사용 안함으로 설정하려면 다음 단계를 수행하십시오.
 - a) CKQC를 입력하고 Enter를 눌러 CICS-MQ 어댑터 제어판에 액세스하십시오.
 - b) 메뉴 표시줄에서 **Connection**을 선택하십시오.
 - c) 메뉴에서 **Modify** 조치를 선택하십시오.
 - d) **Modification Options** 보조 매개변수 창에서 **Disable API Exit**를 선택하고 Enter를 눌러 API 교차 엑시트의 상태를 사용 안함으로 변경하십시오.
 화면에 이 프로세스가 다음과 같이 표시됩니다.

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 style="border: 1px dashed black; padding: 5px;"> <p style="text-align: center;">Modification Options</p> <p>Select modify option. Then press Enter.</p> <p>4 1. Reset statistics</p> <p>2. Enable API Exit</p> <p>3. Disable API Exit</p> <p>F1=Help F12=Cancel</p> </div>

F1=Help F3=Exit

2. CICS 명령행에서 API 교차 엑시트를 사용 안함으로 설정하려면 이러한 방법 중 하나를 선택하십시오.
 - CICS 마스터 터미널에서 명령 CEMT S PROGRAM(CSQCAPX) DISABLED를 발행하십시오.
 - 명령행에서 명령 CKQC MODIFY N D를 발행하십시오. 옵션 D는 연결을 사용 안함으로 설정합니다. 옵션 N은 연결 통계와 관련이 있고 필수지만 효과는 없습니다.
3. CICS 애플리케이션 프로그램을 사용하여 API 교차 엑시트를 사용 안함으로 설정하려면 EXEC CICS LINK 명령을 발행하여 어댑터 재설정 프로그램, DFHMQRS(또는 호환성을 위해 유지되는 CSQCRST)에 링크하고 옵션이 D인 CKQC MODIFY 명령을 발행하십시오.
다음 예는 이를 수행하는 방법을 보여줍니다.

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

MODIFY 명령은 4개의 후행 공백과 구분자로 다른 공백을 채워야 합니다([CICS\(r\) 애플리케이션 프로그램의 CKQC 명령 참조](#)). 옵션 N은 연결 통계와 관련이 있고 필수지만 효과는 없습니다.

제 9 장 CICS-MQ 어댑터 문제점 해결

CICS-MQ 어댑터에 문제점이 있는 경우 CICS는 원인을 진단하는 데 도움이 되는 다양한 정보를 제공합니다.

필요한 정보 및 현재 작업 중인 환경에 따라 CEMT 또는 **EXEC CICS INQUIRE** 명령, CICS-MQ 어댑터 제어판, CKQC DISPLAY 명령 또는 CICSplex SM을 사용하여 CICS-MQ 연결에 대한 정보를 표시할 수 있습니다. 이러한 여러 가지 방법에 대한 정보는 [CICS-MQ 연결에 대한 정보 표시](#)의 내용을 참조하십시오.

CICS-MQ 어댑터 제어판에서 CKQC 트랜잭션은 IBM MQ에 대한 연결을 사용하는 개별 태스크의 세부사항과 이러한 태스크의 상태(예: GET WAIT)를 표시할 수 있습니다. [CICS-MQ 연결을 사용하는 태스크 표시](#)에 이 정보를 표시하는 방법이 설명되어 있습니다.

CICS 실행 진단 기능(CEDF)은 모든 CICS API 서비스에 대한 호출을 트래핑할 뿐 아니라 각 MQI 호출에 대한 CICS-MQ 어댑터의 시작 및 종료를 트래핑합니다.

CICS-MQ 어댑터는 A000 - A1FF 범위의 AP 도메인 추적점을 사용하며, 추적 항목이 표준 CICS 추적 대상에 기록됩니다. CICS-MQ 어댑터에서 발행되는 추적점의 콘텐츠는 [CICS IBM MQ 추적점에 문서화](#)되어 있습니다. 오류 조건에 대한 예외 추적 항목은 무조건 생성됩니다. 비예외 추적 항목은 자원 관리자 인터페이스(RI) 및 자원 관리자 어댑터(RA) 레벨 1 및 레벨 2 추적으로 제어됩니다. CICS 추적에 대한 자세한 정보는 [CETR - 추적 제어](#)의 내용을 참조하십시오.

IBM MQ 대기

태스크가 자원 유형 MQseries, WMQ_INIT 또는 WMQCDISC에서 대기 중인 경우 CICS-MQ 어댑터가 태스크를 일시중단한 것입니다.

자원 유형 MQseries

WAIT 옵션이 MQGET 호출과 함께 사용되었고 사용 가능한 메시지가 없어서 CICS-IBM MQ MQ 어댑터 (DFHMQTRU 모듈)가 태스크를 CICS 대기로 전환했습니다. 대기에 사용된 자원 이름은 GETWAIT입니다. 디스패처의 WAIT_MVS 함수가 이 대기에 사용되며 워크로드 관리를 위한 대기 유형은 OTHER_PRODUCT입니다. 태스크를 영구 제거할 수 있습니다.

자원 유형 WMQ_INIT

CICS-IBM MQ 초기화 프로그램인 DFHMQIN1은 DFHMQIN2가 완료될 때까지 이 대기를 발행합니다. 디스패처의 WAIT_OLDC 함수가 이 대기에 사용되며 워크로드 관리를 위한 대기 유형은 MISC입니다. 태스크를 영구 제거할 수 있습니다.

자원 유형 WMQCDISC

SET MQCONN NOTCONNECTED 명령이 WAIT 또는 FORCE 옵션과 함께 실행되었으며 DFHMQTM 모듈은 IBM MQ를 사용하는 사용자 태스크 수가 0에 도달할 때까지 대기합니다. 자원 이름은 CICS 시스템의 설치된 MQCONN 자원 정의의 이름으로 지정됩니다. 디스패처의 WAIT_OLDC 함수가 이 대기에 사용되며 워크로드 관리를 위한 대기 유형은 MISC입니다. 태스크를 영구 제거할 수 있습니다.

CICS-MQ 연결 종료 시 발생하는 상황

CICS 및 IBM MQ 간의 연결에는 정상(또는 순서에 따른) 종료 및 강제 종료 등 두 가지 유형의 종료가 있습니다. 연결 종료는 운영자 조치, CICS 종료 또는 IBM MQ 큐 관리자에서 발생할 수 있습니다.

[112 페이지](#)의 표 12에는 연결이 활성 상태일 때 어댑터가 다양한 양식의 연결 종료를 처리하는 방법이 요약되어 있습니다. 연결이 활성 상태가 아닐 때(예: 연결이 정지된 후) CICS 또는 IBM MQ가 종료되면 조치가 수행되지 않고 메시지가 발행되지 않습니다.

표 12. CICS 어댑터 연결 종료	
종료 방법	어댑터에서 처리하는 방법
연결의 정지된 종료(EXEC CICS SET MQCONN NOTCONNECTED BUSY(WAIT NOWAIT), CEMT SET MQCONN NOTCONNECTED 또는 CKQC STOP)	어댑터의 상태를 <i>Quiescing</i> 으로 표시합니다. 활성 태스크 및 대기 중인 태스크를 모두 완료할 수 있습니다. 동기점을 허용합니다. 새 태스크에서의 호출을 허용하지 않습니다. 마지막 태스크가 IBM MQ에서 연결 끊기를 시작합니다.
연결 강제 종료(EXEC CICS SET MQCONN NOTCONNECTED BUSY(FORCE), CEMT SET MQCONN FORCENOTCON 또는 CKQC STOP FORCE)	어댑터의 상태를 <i>StoppingForce</i> 로 표시합니다. IBM MQ에서 연결을 끊습니다. 먼저 CKTI의 인스턴스를 포함하여 IBM MQ에서 대기 중인 태스크를 재개한 후 WebSphere MQ에 액세스한 모든 인플라이트 태스크를 강제 제거합니다.
CICS 웹 종료	DFHMQ0411I 메시지를 발행합니다. 연결의 정상 종료(quiesced shutdown)를 시작합니다.
CICS 즉시 종료	DFHMQ0410I 메시지를 발행합니다. IBM MQ를 사용하는 모든 인플라이트 태스크가 백아웃됩니다.
CICS 이상 종료	DFHMQ0412I 메시지를 발행합니다.
IBM MQ 정상 종료(quiesced shutdown)	연결의 정상 종료(quiesced shutdown)를 시작합니다.
IBM MQ 이상 종료 또는 강제 종료	연결의 강제 종료를 시작합니다.

정상(또는 순서에 따른) 종료

연결을 정상 종료(quiesced shutdown)하면 인터페이스가 닫히기 전에 각 CICS 트랜잭션이 종료될 수 있습니다. 이 방법을 사용하는 경우 CICS에 다시 연결할 때 인다우트 작업 단위가 없다고 예상할 수 있습니다.

다음 각 상황에서 정상 종료(quiesced shutdown)가 발생합니다.

- CICS 터미널 조작자가 **EXEC CICS** 또는 **CEMT SET MQCONN NOTCONNECTED** 명령을 실행하거나 **CKQC STOP** 명령을 실행합니다. CICS 및 큐 관리자는 활성 상태로 남아 있습니다.
- CICS 터미널 조작자가 **CEMT PERFORM SHUTDOWN** 명령을 실행합니다.
- 큐 관리자가 다음 명령을 통해 정지됩니다.

```
+CSQ1 STOP QMGR MODE(QUIESCE)
```

이 명령은 큐 관리자를 중지하고 현재 식별된 태스크가 정상 처리를 계속할 수 있게 하며 새 태스크가 큐 관리자에 대해 자신을 식별하도록 허용하지 않습니다. CICS는 활성 상태로 남아 있습니다.

강제 종료

연결을 강제 종료하면 큐 관리자에 연결된 CICS 트랜잭션이 비정상적으로 종료될 수 있습니다. 따라서 시스템이 다시 연결될 때 인다우트 작업 단위가 있을 수 있습니다.

다음 각 상황에서 강제 종료가 발생합니다.

- CICS 터미널 조작자가 **EXEC CICS SET MQCONN NOTCONNECTED** 명령을 FORCE 옵션과 함께 실행하거나 **CEMT SET MQCONN FORCENOTCON** 명령 또는 **CKQC STOP FORCE** 명령을 실행합니다.
- CICS 터미널 조작자가 CICS 즉시 종료 명령을 실행합니다.

```
CEMT PERFORM SHUTDOWN IMMEDIATE
```

큐 관리자는 활성 상태로 남아 있습니다. 이 명령에 대한 정보는 **CEMT PERFORM SHUTDOWN**의 내용을 참조하십시오.

- IBM MQ 강제 종료 명령이 실행됩니다.

```
+CSQ1 STOP QMGR MODE(FORCE) or +CSQ1 STOP QMGR MODE(RESTART)
```

CICS는 활성 상태로 남아 있습니다.

- IBM MQ 이상 종료가 발생합니다. CICS는 활성 상태로 남아 있습니다.
- CICS 이상 종료가 발생합니다. 큐 관리자는 활성 상태로 남아 있습니다.

큐 관리자를 중지할 때 발생하는 상황

큐 관리자가 정상적으로 중지되면 IBM MQ가 모든 활동을 순서대로 중지합니다. 정상(Quiesce), 강제 또는 다시 시작 모드를 사용하여 IBM MQ를 중지할 수 있습니다.

영향이 [113 페이지의 표 13](#)이 나와 있습니다.

표 13. QUIESCE, FORCE 및 RESTART 모드로 큐 관리자 중지			
스레드 유형	QUIESCE	FORCE	RESTART
활성 스레드	완료될 때까지 실행	백아웃	백아웃
새 스레드	시작할 수 있음	허용되지 않음	허용되지 않음
새 연결	허용되지 않음	허용되지 않음	허용되지 않음

CICS에서는 현재 스레드가 작업 단위 끝까지만 실행됩니다. CICS를 사용하는 경우 정상(quiesce) 모드에서 큐 관리자를 중지하면 CICS-MQ 어댑터가 중지되므로 활성 태스크에 둘 이상의 작업 단위가 포함되어 있으면 태스크가 완료될 때까지 실행되지 않아도 됩니다.

어떤 모드에서든 큐 관리자를 중지할 때 단계는 다음과 같습니다.

1. 연결이 종료됩니다.
2. IBM MQ가 명령 승인을 중단합니다.
3. IBM MQ가 페이지 세트에 대한 미해결 업데이트가 완료되었는지 확인합니다.
4. 다시 시작 상태 바이트 주소(RBA)가 z/OS 콘솔 로그에 기록될 수 있도록 IBM MQ에서 내부적으로 DISPLAY USAGE 명령이 실행됩니다.
5. 종료 체크포인트가 작성되고 IBM MQ 부트스트랩 데이터 세트(BSDS)가 업데이트됩니다.

강제 또는 다시 시작 모드에서 큐 관리자를 중지하면 새 스레드가 할당되지 않으며 연결된 스레드에 대한 작업이 롤백됩니다. 이 모드를 사용하면 커밋 처리 단계 사이에 있는 스레드에 대한 인다우트 작업 단위가 작성될 수 있습니다. 이러한 작업 단위는 IBM MQ가 제어 CICS, IMS 또는 RRS 서브시스템과 다시 연결될 때 해결됩니다.

정상(Quiesce) 모드는 인다우트 작업 단위에 영향을 미치지 않습니다. 모든 인다우트 작업 단위가 인다우트 상태로 남아 있습니다.

자동 재연결 및 재동기화

CICS가 IBM MQ에 연결되고 큐 관리자가 중지되면 중지가 발견된 후 CICS-MQ 어댑터가 다시 연결하려고 시도합니다. 이 연결 요청에서는 이전 연결 요청에서 사용된 것과 동일한 연결 매개변수를 사용합니다.

단일 큐 관리자를 연결 대상으로 지정한 경우 중지가 발견된 후 CICS가 10초 동안 대기한 다음 다시 연결하려고 시도합니다. 큐 관리자가 10초 이내에 다시 시작되지 않으면 CICS가 자동으로 다시 연결되어 큐 관리자가 다시 시작될 때까지 연결 요청이 지연됩니다.

큐 공유 그룹을 연결 대상으로 지정한 경우 중지가 발견된 즉시 CICS가 다시 연결하려고 시도합니다. CICS가 연결을 복원하기 위해 수행하는 조치는 마지막 큐 관리자에 대한 미해결 작업 단위가 있는지 여부와 재동기화에 대해 지정한 설정에 따라 다릅니다.

큐 공유 그룹을 연결 대상으로 지정한 경우 MQCONN 자원 정의의 RESYNCMEMBER 속성을 사용하여 CICS에 대한 적절한 재동기화 조치를 선택할 수 있습니다. 재동기화는 IBM MQ에 대한 연결이 끊어지고 CICS가 마지막 큐 관리자에 대해 미해결 작업 단위를 보유하고 있을 때 발생합니다. CICS가 동일한 큐 관리자에 다시 연결하기 위해 대기하는지 여부 또는 CICS가 동일한 큐 관리자에 다시 연결하려고 한 번 시도하지만 이 시도에 실패하는

경우 그룹의 다른 적격 큐 관리자에 연결하는지 여부를 선택할 수 있습니다. 큐 관리자가 현재 CICS 리전과 동일한 LPAR에서 활성화된 경우 CICS 리전에 연결할 수 있습니다.

또는 IBM MQ가 CICS에 대한 그룹 복구 단위를 지원하는 경우 **RESYNCMEMBER(GROUPRESYNC)** 옵션을 사용할 수 있습니다. GROUPRESYNC 옵션을 사용하면 CICS는 큐 공유 그룹의 로컬 IBM MQ 큐 관리자에 연결하고 이전에 연결된 큐 관리자에 관계없이 모든 미해결 인다우트 작업 단위를 해결합니다. 미해결 작업 단위를 해결하려면 CICS가 이전 연결에 사용된 동일한 MQCONN 자원 정의를 사용하여 다시 연결해야 합니다. 자세한 정보는 [CICS에서 인다우트 작업 단위를 해결하는 방법을 참조하십시오](#).

CICS-MQ 어댑터가 다시 시작될 때 발생하는 상황

연결이 끊어질 때마다 어댑터가 다시 연결 프로세스 중에 다시 시작 단계를 거쳐야 합니다. 다시 시작 단계에서는 자원을 재동기화합니다. CICS와 IBM MQ의 재동기화를 통해 인다우트 작업 단위를 식별하고 해결할 수 있습니다.

재동기화는 다음 요청으로 인해 발생할 수 있습니다.

- 분산 큐잉 컴포넌트로부터의 명시적 요청
- IBM MQ에 연결할 때 암시적 요청

IBM MQ에 연결하여 재동기화가 발생하는 경우 이벤트 시퀀스는 다음과 같습니다.

1. 연결 프로세스가 IBM MQ에서 인다우트라고 간주하는 작업 단위(UOW) 목록을 얻습니다.
2. UOW ID가 콘솔의 DFHM0313I 메시지에 표시됩니다.
3. UOW ID가 CICS에 전달됩니다.
4. CICS가 각 인다우트 UOW ID에 대한 재동기화 태스크(CRSY)를 시작합니다.
5. 각 인다우트 UOW에 대한 태스크의 결과가 콘솔에 표시됩니다.

연결 프로세스 중에 표시되는 다음 메시지를 확인해야 합니다.

DFHM0313I

UOW가 인다우트 상태임을 표시합니다.

DFHM0400I

UOW를 식별하며 다음 메시지 중 하나가 다음에 옵니다.

- DFHM0402I 또는 DFHM0403I는 UOW가 성공적으로 해결되었음을 표시합니다(커밋되거나 백아웃 됨).
- DFHM0404E, DFHM0405E, DFHM0406E 또는 DFHM0407E는 UOW가 해결되지 않았음을 표시합니다.

DFHM0409I

모든 UOW가 성공적으로 해결되었음을 표시합니다.

DFHM0408I

일부 UOW가 성공적으로 해결되지 않았음을 표시합니다.

DFHM0314I

*로 강조표시된 UOW ID는 자동으로 해결되지 않음을 경고합니다. 이러한 UOW는 다시 시작될 때 분산 큐잉 컴포넌트에서 명시적으로 해결되어야 합니다.

DFHM0313I 메시지 총계는 DFHM0402I + DFHM0403I 메시지 총계와 같아야 합니다. 총계가 같지 않으면 연결 프로세스가 일부 UOW를 해결할 수 없습니다. 해결할 수 없는 UOW는 CICS의 문제점(예: 콜드 스타트) 또는 IBM MQ의 문제점이나 분산 큐잉으로 인해 발생합니다. 이러한 문제점이 해결되면 연결을 끊었다가 다시 연결하여 다른 재동기화를 시작할 수 있습니다.

또는 RESOLVE INDOUBT 명령과 DFHM0400I 메시지에 표시되는 UOW ID를 사용하여 자체적으로 각 미해결 UOW를 해결할 수 있습니다. 그런 다음 연결 끊기 및 연결을 시작하여 CICS에서 작업 단위 디스크립터를 정리해야 합니다. 수동으로 UOW를 해결하려면 UOW의 올바른 결과를 알아야 합니다.

IBM MQ에서 미해결 UOW와 연관된 모든 메시지를 잠그기 때문에 일괄처리, TSO 또는 CICS 태스크가 이러한 메시지에 액세스할 수 없습니다.

CICS가 실패하고 긴급 다시 시작이 필요한 경우 CICS 시스템의 GENERIC APPLID를 변경하지 마십시오. 변경한 후 IBM MQ에 다시 연결하면 IBM MQ가 새 CICS 인스턴스를 다른 CICS로 처리하므로(APPLID가 다르게 때문) IBM MQ의 데이터 무결성을 보장할 수 없습니다. 그러면 인다우트 해결이 잘못된 CICS 로그를 기반으로 합니다.

작업 단위가 IBM MQ에서 처리되지 않은 경우 이는 작업 단위를 해결할 수 없다는 것을 의미하므로 RESYNCMEMBER 설정을 변경하지 마십시오. CICS에서 보류된 작업 단위는 자원 관리자 규정자로 식별됩니다. RESYNCMEMBER(GROUPRESYNC)가 사용될 때 규정자는 큐 공유 그룹의 이름이고 그렇지 않은 경우 사용되는 규정자는 개별 큐 관리자의 이름입니다.

CICS에서 인다우트 작업 단위를 해결하는 방법

CICS-MQ 어댑터의 기능 중 하나는 CICS와 IBM MQ 간의 데이터를 동기화된 상태로 유지하는 것입니다. CICS에 연결되어 있는 동안 큐 관리자가 비정상적으로 종료되면 IBM MQ에서 인식하지 못하는 상태로 CICS가 작업을 커밋하거나 백아웃할 수 있습니다. 큐 관리자가 다시 시작되면 해당 작업이 인다우트 상태가 됩니다.

CICS에 대한 연결이 다시 시작되거나 다시 연결될 때까지 IBM MQ가 이러한 작업 단위를 해결할 수 없습니다 (즉, IBM MQ 자원에 대한 변경사항을 커밋하거나 백아웃할 수 없음).

CICS가 동일한 IBM MQ 큐 관리자에 다시 연결하면 CICS-MQ 어댑터 시작 중에 인다우트 작업 단위를 해결하는 프로세스가 시작됩니다. 이 프로세스의 단계는 다음과 같습니다.

- CICS-MQ 어댑터가 IBM MQ로부터 이 연결 ID에 대한 인다우트 작업 단위의 목록을 요청합니다.
- 어댑터가 인다우트 작업 단위 목록을 수신하여 해결을 위해 CICS에 전달합니다.
- CICS가 이 목록의 항목을 자체 로그의 항목과 비교합니다. CICS가 자체 목록에서 각 인다우트 작업 단위에 대해 수행한 조치를 판별합니다.

IBM MQ는 해결된 모든 단위에 대해 필요에 따라 큐를 업데이트하고 해당 잠금을 해제합니다. 작업 단위가 인다우트로 처리 지연된 경우, 즉 CICS 자체에서 해결할 수 없는 경우 작업 단위가 개별적으로 해결됩니다. 처리 지연된 작업 단위에 대한 자세한 정보는 [처리 지연된 작업 단위의 내용](#)을 참조하십시오.

인다우트 작업 단위의 해결은 CICS 자원에 영향을 미치지 않습니다. IBM MQ는 CICS를 복구 조정자로 간주하고 WebSphere MQ가 다시 시작될 때 로그 레코드가 커밋의 시작을 표시했는지 여부에 따라 각 단위를 자동으로 커밋하거나 백아웃합니다. WebSphere MQ가 다시 연결되는 동안 인다우트 오브젝트가 존재하면 CICS 자원이 잠기지 않습니다.

해결되지 않은 작업 단위

연결에 IBM MQ 큐 공유 그룹을 사용 중이고 연결에 대한 MQCONN 정의에 RESYNCMEMBER(NO)를 지정한 경우 CICS가 한 번만 동일한 큐 관리자에 다시 연결하려고 시도합니다. 해당 시도가 실패하면 CICS가 큐 공유 그룹의 적합한 멤버에 연결합니다. 큐 관리자가 현재 CICS 리전과 동일한 LPAR에서 활성화된 경우 CICS 리전에 연결할 수 있습니다. CICS가 다른 큐 관리자에 연결되면 인다우트 작업 단위를 해결할 수 없으며 해결되지 않은 작업 단위가 남아 있는 경우 CICS에서 경고 메시지 DFHMQ2064를 발행합니다.

일부 상황에서는 CICS가 동일한 큐 관리자에 다시 연결하지만 인다우트 작업 단위를 해결하기 위해 IBM MQ 프로세스를 실행할 수 없습니다. 이러한 상황에서는 CICS가 다음 오류 메시지 중 하나를 발행합니다.

- DFHMQ0404E
- DFHMQ0405E
- DFHMQ0406E
- DFHMQ0407E

다음에 DFHMQ0408I 메시지가 옵니다.

다시 시작한 후에도 해결되지 않은 단위가 남아 있으면 [116 페이지의 『CICS 작업 단위를 수동으로 해결하는 방법』](#)에서 설명된 방법으로 해결하십시오.

IBM MQ는 CICS에 대한 그룹 복구 단위를 지원하며 WebSphere MQ 7.1 이상을 사용하면 새 옵션 RESYNCMEMBER(GROUPRESYNC)를 사용할 수 있습니다. GROUPRESYNC 옵션을 사용하면 CICS가 큐 공유 그룹의 로컬 IBM MQ 큐 관리자에 연결합니다. 큐 관리자는 IBM MQ에서 선택되며 인다우트 작업 단위가 있는 큐 관리자에 관계없이 모든 인다우트 작업 단위를 해결하도록 CICS에 요청합니다. 작업 단위가 IBM MQ에서 처리되지 않은 경우 이는 작업 단위를 해결할 수 없다는 것을 의미하므로 RESYNCMEMBER 설정을 변경하지 마십시오.

시오. CICS에서 보류된 작업 단위는 자원 관리자 규정자로 식별됩니다. RESYNCMEMBER(GROUPRESYNC)가 사용될 때 규정자는 큐 공유 그룹의 이름이고 그렇지 않은 경우 사용되는 규정자는 개별 큐 관리자의 이름입니다.

- RESYNCMEMBER(GROUPRESYNC)를 지정하고 이전 연결이 RESYNCMEMBER(YES) 또는 RESYNCMEMBER(NO)가 포함된 MQCONN 정의를 사용했으며 IBM MQ에 미해결 인다우트 작업 단위가 있으면 RESYNCMEMBER의 이전 설정으로 되돌리지 않고 해당 인다우트 작업 단위를 해결할 수 없습니다. CICS가 경고 메시지 DFHMQ2065를 발행합니다.
- RESYNCMEMBER(YES) 또는 RESYNCMEMBER(NO)를 지정하고 이전 연결이 RESYNCMEMBER(GROUPRESYNC)가 포함된 MQCONN 정의를 사용했으며 IBM MQ에 미해결 인다우트 작업 단위가 있으면 RESYNCMEMBER의 이전 설정으로 되돌리지 않고 해당 인다우트 작업 단위를 해결할 수 없습니다. CICS가 경고 메시지 DFHMQ2066을 발행합니다.

CICS 작업 단위를 수동으로 해결하는 방법

CICS-MQ 어댑터가 비정상적으로 종료되면 CICS 및 IBM MQ가 이상 종료의 원인이 된 서브시스템에 따라 동적으로 또는 다시 시작 중에 인다우트 목록을 빌드합니다. CICS가 IBM MQ에 연결할 때 하나 이상의 작업 단위가 해결되지 않았을 수 있습니다. CICS에서 해결할 수 없는 작업 단위는 IBM MQ 명령을 사용하여 수동으로 해결해야 합니다.

이 태스크 정보

일부 작업 단위가 해결되지 않은 경우 다음 메시지 중 하나가 콘솔로 전송됩니다.

- DFHMQ0404E
- DFHMQ0405E
- DFHMQ0406E
- DFHMQ0407E
- DFHMQ0408I
- DFHMQ2064
- DFHMQ2065
- DFHMQ2066

CICS는 연결 시작 중에 해결되지 않은 작업 단위의 세부사항을 보존합니다. IBM MQ에서 제공하는 목록에 더 이상 나타나지 않을 때 항목이 영구 제거됩니다.

CICS에서 해결할 수 없는 작업 단위는 IBM MQ 명령을 사용하여 수동으로 해결해야 합니다. 이 수동 프로시저는 작동 오류나 소프트웨어 문제점으로 인해 자동 해결이 불가능한 경우에만 필요하므로 거의 사용되지 않습니다. 인다우트 해결 중에 발견된 불일치를 조사해야 합니다.

작업 단위를 해결하려면 다음을 수행하십시오.

프로시저

1. 다음 명령을 사용하여 IBM MQ에서 작업 단위의 목록을 얻으십시오.

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

다음 메시지가 수신됩니다.

```
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 연결의 경우 NID(기점 ID)는 CICS applid와 동기점 로그 항목이 기록될 때 CICS에서 제공된 고유 번호로 구성됩니다. 이 고유 번호는 동기점 처리 시 CICS 시스템 로그와 IBM MQ 로그 모두에 기록된 레코드에 저장됩니다. CICS에서는 이 값을 작업 토큰이라고 합니다.

2. CICS 로그에서 특정 작업 단위와 관련된 항목을 스캔하십시오.

- 작업 토큰 필드(JCSRMTKN)가 네트워크 ID에서 얻은 값과 동일한 태스크 관련 설치에 대한 PREPARE 레코드를 찾으십시오.
네트워크 ID는 IBM MQ가 DISPLAY CONN 명령 출력에서 제공됩니다.
- 작업 단위에 대한 CICS 로그의 PREPARE 레코드는 CICS 태스크 번호를 제공합니다. 이 CICS 태스크의 로그에 있는 기타 모든 항목은 이 번호를 사용하여 찾을 수 있습니다.
- 로그를 스캔할 때 CICS 저널 인쇄 유틸리티 DFHJUP를 사용할 수 있습니다.
이 프로그램 사용에 대한 세부사항은 [배치 작업\(DFHJUP\)을 사용하여 로그 스트림 읽기의 내용을 참조하십시오](#).

3. IBM MQ 로그에서 NID가 특정 작업 단위와 관련된 레코드를 스캔하십시오. 그런 다음 이 레코드의 URID를 사용하여 이 작업 단위에 대한 나머지 로그 레코드를 확보하십시오.

IBM MQ 로그 레코드 인쇄 프로그램(CSQ1LOGP)을 사용하여 로그를 스캔할 수 있습니다.

IBM MQ 로그를 스캔할 때 IBM MQ 시작 메시지 CSQJ001I에서 이 세션의 시작 RBA를 제공합니다.

4. 필요한 경우 IBM MQ에서 인다우트 해결을 수행하십시오. IBM MQ RESOLVE INDOUBT 명령을 사용하여 작업 단위에 대한 작업 조치를 수행하도록 IBM MQ에 지시할 수 있습니다. 특정 connection-name과 연관된 모든 스레드를 복구하려면 NID(*) 옵션을 사용하십시오.

이 명령은 스레드가 커밋되었는지 아니면 백아웃되었는지를 표시하는 다음 메시지 중 하나를 생성합니다.

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED
```

결과

인다우트 해결을 수행할 때 IBM MQ 자원만 영향을 받기 때문에 CICS 및 어댑터는 작업 단위를 커밋하거나 백아웃하기 위해 IBM MQ에 대해 수행하는 명령을 인식하지 못합니다. 그러나 CICS는 IBM MQ에서 해결되지 않은 인다우트 스레드에 대한 세부사항을 보존합니다. 제공된 목록이 비어 있거나 세부사항이 있는 CICS의 작업 단위가 목록에 포함되지 않은 경우 이 정보가 영구 제거됩니다.

트리거가 작동하지 않는 경우

임시 MQ 오류(예: 큐 관리자가 복구할 수 있는 결합 기능 장애)로 인해 트리거 모니터가 큐 관리자로부터 메시지를 가져올 수 없는 경우 트리거 모니터가 최대 1시간 동안 1분 간격으로 요청을 가져오려고 재시도합니다. 장애가 1시간 넘게 지속되면 트리거 모니터가 종료됩니다.

트리거 모니터가 프로그램을 시작할 수 없거나 큐 관리자가 트리거 메시지를 전달할 수 없는 경우 프로그램이 트리거되지 않습니다. 예를 들어, 프로세스 오브젝트의 applid에 프로그램이 백그라운드에서 시작되도록 지정되어 야 합니다. 그렇지 않으면, 트리거 모니터가 프로그램을 시작할 수 없습니다.

트리거 메시지를 작성했지만 이니시에이션 큐에 넣을 수 없는 경우(예를 들어, 큐가 가득 찼거나 트리거 메시지 길이가 이니시에이션 큐에 대해 지정된 최대 메시지 길이보다 긴 경우) 트리거 메시지를 대신 데드 레터(미배달 메시지) 큐에 넣습니다.

데드-레터 큐에 넣기 조작을 성공적으로 완료할 수 없는 경우 트리거 메시지가 삭제되고 경고 메시지가 z/OS 콘솔이나 시스템 운영자에게 전송되거나 오류 로그에 추가됩니다.

트리거 메시지를 데드-레터 큐에 넣으면 해당 큐에 대한 트리거 메시지가 생성될 수 있습니다. 데드-레터 큐에 메시지를 추가하면 이 두 번째 트리거 메시지가 삭제됩니다.

프로그램이 성공적으로 트리거되었지만 큐에서 메시지를 수신하기 전에 이상 종료된 경우 추적 유틸리티(예: 프로그램이 CICS에서 실행되는 경우 CICS AUXTRACE)를 사용하여 실패 원인을 찾으십시오.

제 10 장 CICS-MQ 브릿지 문제점 해결

오류 발생 시 CICS-MQ 브릿지가 수행하는 특정 조치 및 브릿지에 대한 문제점을 식별하고 해결하기 위해 수행할 수 있는 조치가 있습니다.

CICS-MQ 브릿지가 오류를 어떻게 처리합니까?

CICS-MQ 브릿지를 사용하면 IBM MQ 애플리케이션이 MQ 브릿지 큐에 MQ 요청 메시지를 넣어서(MQPUT), CICS DPL 프로그램 또는 CICS 3270 트랜잭션을 실행할 수 있습니다. CICS-MQ 브릿지는 MQGET 요청을 발행하여 큐에서 메시지를 검색한 후 MQ 메시지 헤더가 없는 메시지인 메시지 콘텐츠를 대상 프로그램 또는 트랜잭션에 전달합니다.

임시 MQ 오류(예: 큐 관리자가 복구할 수 있는 결함 기능 장애)로 인해 CICS-MQ 브릿지가 큐 관리자에서 메시지를 가져올 수 없는 경우 CICS-MQ 브릿지가 최대 1시간 동안 1분 간격으로 요청을 가져오려고 재시도합니다. 장애가 1시간 넘게 지속되면 CICS-MQ 브릿지가 종료됩니다.

일반적으로 애플리케이션 또는 트랜잭션이 메시지를 처리하고 응답을 CICS-MQ 브릿지에 리턴합니다. 요청 메시지에 회신 큐가 지정된 경우 CICS-MQ 브릿지가 MQ 응답 메시지 헤더를 이 응답에 추가하고 MQPUT 요청을 통해 이 MQ 응답 메시지를 회신 큐에 넣습니다.

애플리케이션 또는 트랜잭션이 오류(예: 요청의 데이터가 누락되었거나 올바르지 않음)를 발견하면 CICS-MQ 브릿지에 응답 메시지를 리턴할 수 있습니다. CICS-MQ 브릿지가 정상 응답을 처리하는 것과 동일한 방식으로 이 오류 응답을 처리합니다.

CICS-MQ 브릿지는 IBM MQ 애플리케이션이 처리할 수 없거나 처리하지 않는 일부 오류를 처리할 수 없습니다. 예를 들어, 다음과 같습니다.

- 메시지 헤더가 올바르지 않은 문자 코딩을 지정합니다.
- 메시지에 MQ 또는 MQ 브릿지 요구사항을 준수하지 않는 메시지 헤더가 포함되어 있습니다.
- 트랜잭션 또는 해당 프로그램이 설치되지 않았습니.
- 메시지의 보안 신임 정보가 트랜잭션 진행을 허용하지 않습니다.
- 예외 조건으로 인해 애플리케이션 또는 브릿지 자체의 비정상 종료(ABEND)가 발생합니다.

일반적으로 이러한 오류가 발생하는 경우 CICS-MQ 브릿지가 다음 단계를 수행합니다.

1. DFHMQ07nn 메시지를 CSMT 트랜잭션 데이터 큐, CICS 작업 로그 또는 둘 모두에 씁니다.
2. 트랜잭션 덤프를 작성합니다.
3. **SYNCPPOINT ROLLBACK**을 발행하십시오.
4. 브릿지가 재시도하지 않는 요청 메시지를 처리합니다.

세부사항은 120 페이지의 『오류가 발생한 경우 CICS-MQ 브릿지가 수행하는 조치』의 내용을 참조하십시오.

MQ 브릿지가 작업 단위 롤백을 어떻게 수행합니까?

표준 동기점 MQGET은 되돌릴 수 있습니다. 작업 단위가 SYNCPPOINT를 발행하면 MQ가 큐에서 메시지를 제거하는 작업을 완료합니다. 작업 단위가 **SYNCPPOINT ROLLBACK**을 발행하면 MQ는 큐에 메시지를 리턴합니다.

일시적인 조건의 경우 이 메커니즘을 통해 CICS-MQ가 자동으로 복구할 수 있습니다. MQ가 요청 메시지를 브릿지 큐에 리턴하고 CICS-MQ 브릿지가 동일한 메시지를 다시 가져와서 처리합니다.

그러나 일시적이지 않은 오류의 경우 이 메커니즘은 MQ 만기 처리가 메시지를 삭제할 때까지 동일한 요청 메시지를 무의미하게 반복적으로 다시 처리하여 시스템 자원을 사용합니다.

이 문제점을 해결하기 위해 CICS-MQ 브릿지가 재시도 횟수를 제한할 수 있습니다. 이는 CICS-MQ 브릿지가 일시적인 조건에서 자동으로 복구될 수 있는 기회를 제공하며 일시적이지 않은 조건의 경우 무한 재시도를 방지합니다.

브릿지 큐의 백아웃 임계값(BOTHRESH) 속성은 최대 재시도 수, 즉 **SYNCPOINT ROLLBACK**이 동일한 메시지를 브릿지 큐에 리턴할 수 있는 최대 횟수를 지정합니다. 기본값은 0이며, 이는 **SYNCPOINT ROLLBACK**이 브릿지 큐에 메시지를 리턴하지 않음을 의미합니다.

CICS-MQ 브릿지가 재시도하지 않는 메시지를 어떻게 처리합니까?

CICS-MQ 브릿지가 단순히 메시지를 브릿지 큐에 다시 넣을 수 없는 경우(예를 들어, 백아웃 수가 임계값보다 큰 메시지) 브릿지가 다음과 같이 메시지를 처리합니다.

1. 백아웃 큐가 큐 관리자에 정의된 경우 브릿지가 먼저 메시지를 백아웃 큐에 넣으려고 시도합니다.
2. 1단계가 실패하거나 불가능하면 데드 레터 큐가 큐 관리자에 정의된 경우 브릿지가 대신 데드 레터 큐를 사용하려고 시도합니다.
3. 2단계가 실패하거나 불가능한 경우 브릿지가 메시지의 지속성 속성을 확인합니다.
 - 메시지가 지속적이지 않으면 브릿지가 메시지를 버립니다.
 - 메시지가 지속적이면 브릿지가 비정상적으로 종료됩니다.

CICS-MQ 브릿지를 어떻게 디버깅할 수 있습니까?

CICS-MQ 브릿지가 예상대로 메시지를 처리하지 않거나, 브릿지 태스크가 이상 종료되거나 예기치 않게 종료된 경우 [122 페이지의 『CICS-MQ 브릿지 디버깅』](#)의 문제점 해결 가이드라인에 따라 브릿지를 디버깅할 수 있습니다.

오류가 발생한 경우 CICS-MQ 브릿지가 수행하는 조치

CICS-MQ 브릿지가 오류를 발견하면 일련의 조치를 수행합니다.

[121 페이지의 그림 12](#)에서는 CICS-MQ가 오류를 처리하기 위해 수행하는 일련의 조치를 설명합니다.

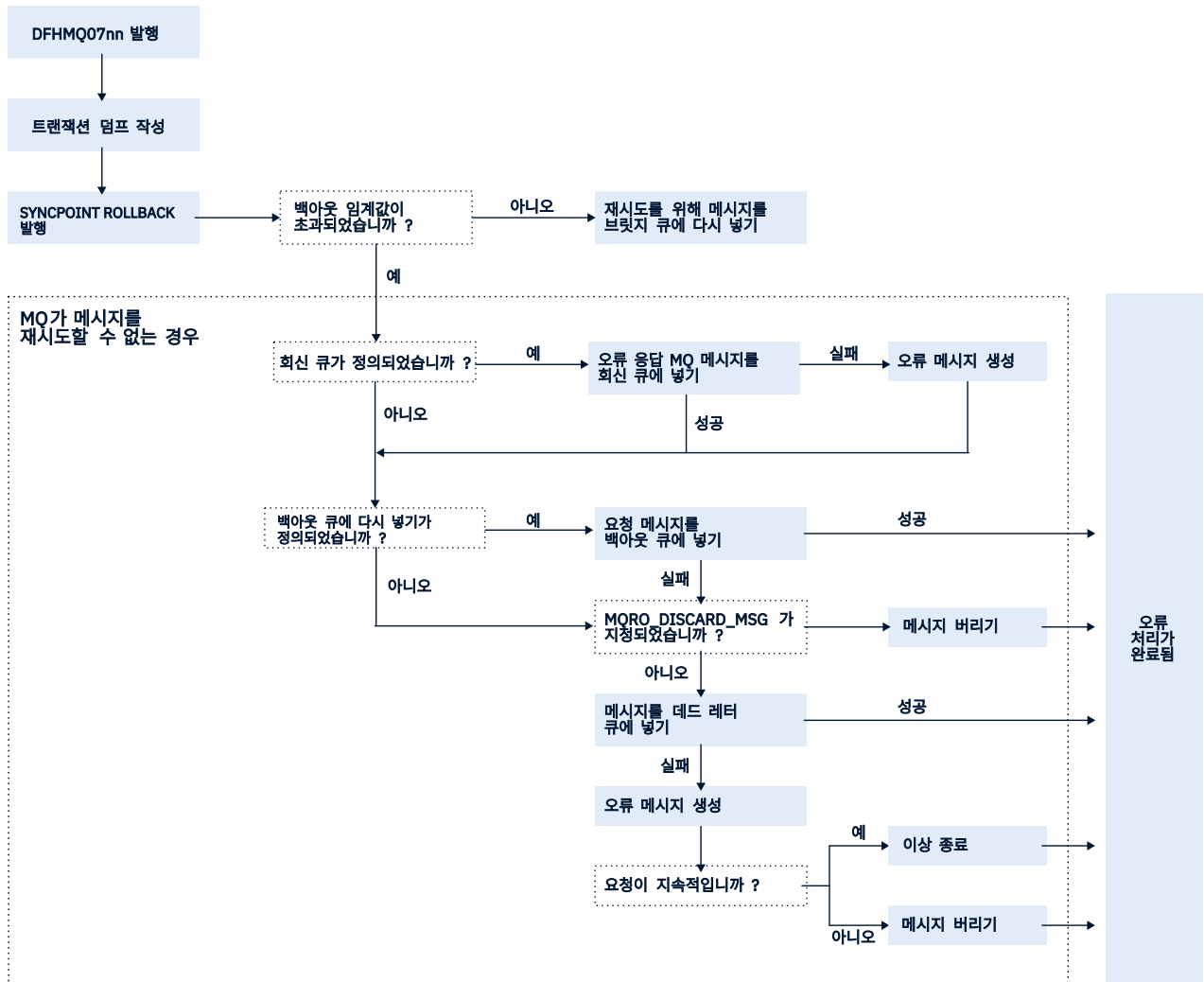


그림 12. CICS-MQ 오류 처리 워크플로우

오류가 발생하면 CICS-MQ 브릿지가 다음 조치를 수행합니다.

1. DFHMQ07nn 메시지를 CSMT 트랜지언트 데이터 큐, CICS 작업 로그 또는 둘 모두에 씁니다.
2. 트랜잭션 덤프를 작성합니다.
3. SYNCPOINT ROLLBACK을 실행합니다.
4. 메시지의 백아웃 수가 아직 백아웃 임계값에 도달하지 않은 경우 CICS-MQ 브릿지가 요청 메시지를 다시 가져와서 처리를 재시도할 수 있도록 SYNCPOINT ROLLBACK 처리 시 요청 메시지를 브릿지 큐에 다시 넣습니다. 이 오류에 대한 처리가 완료됩니다.

메시지의 백아웃 수가 백아웃 임계값을 초과하는 경우 [122 페이지의 『5』 단계](#) 및 [122 페이지의 『6』 단계](#)를 계속합니다.

5. 요청 메시지가 회신 큐를 지정하는 경우 오류 응답 MQ 메시지를 빌드하여 회신 큐에 넣습니다. 이 오류 응답 메시지에 대한 넣기 조작이 실패하면 오류 메시지를 생성합니다. 이 오류에 대한 처리가 완료됩니다.

참고:

- 요청 메시지에 포함되지 않은 경우에도 이 오류 응답 MQ 메시지에 MQCIH가 포함됩니다.
 - CICS-MQ 브릿지는 요청 메시지에서 MQCIH_PASS_EXPIRATION 및 MQRO_PASS_DISCARD_AND_EXPIRY 옵션을 확인하고 처리합니다.
6. 브릿지 큐에 대해 정의된 백아웃 큐가 있으면 요청 메시지를 백아웃 큐에 넣습니다. 넣기 조작이 성공하면 이 오류에 대한 처리가 완료됩니다.

넣기 조작이 실패하거나 브릿지 큐에 대해 정의된 백아웃 큐가 없는 경우 [122 페이지의 『7』 단계](#)를 계속합니다.
 7. 메시지를 데드 레터 큐에 넣지 않도록 브릿지에 지시하는 MQRO_DISCARD_MSG가 요청 메시지에 지정되어 있지 않으면 메시지를 데드 레터 큐에 넣습니다.

넣기 조작이 실패하면 브릿지가 오류 메시지를 생성하고 요청이 지속되는 경우 비정상적으로 종료하거나 (ABEND) 메시지가 지속되지 않는 경우 메시지를 버립니다.

이 오류에 대한 처리가 완료됩니다.

참고: 데드 레터 큐에 넣을 때 요청 메시지가 MQDLH와 함께 추가됩니다. 결과 메시지가 데드 레터 큐에 비해 너무 길면 결과 메시지가 잘릴 수 있습니다.

CICS-MQ 브릿지 디버깅

CICS-MQ 브릿지가 예상대로 메시지를 처리하지 않거나, 브릿지 태스크가 이상 종료되거나 예기치 않게 종료된 경우 다음과 같은 문제점 해결 가이드라인이 브릿지를 디버깅하는 데 도움이 됩니다.

발생한 오류가 무엇입니까?

- [122 페이지의 『메시지를 브릿지 요청 큐에 넣었지만\(PUT\) 브릿지 모니터에서 처리하지 않습니다.』](#)
- [123 페이지의 『브릿지 모니터가 요청 큐에서 인바운드 메시지를 가져왔지만 CICS DPL 프로그램 또는 CICS 트랜잭션을 실행하는 데 실패했습니다.』](#)
- [123 페이지의 『CICS-MQ 브릿지 태스크 이상 종료』](#)
- [124 페이지의 『CICS-MQ 브릿지 모니터가 예기치 않게 종료됨』](#)

메시지를 브릿지 요청 큐에 넣었지만(PUT) 브릿지 모니터에서 처리하지 않습니다.

1. 브릿지 모니터가 실행 중인지 확인하십시오. **CEMT I TASK**를 실행하고 CKBR 또는 브릿지 모니터에 사용 중인 다른 트랜잭션 ID를 찾으십시오.

브릿지 모니터가 실행 중이지 않고 트리거될 것으로 예상되는 경우 브릿지 요청 큐의 트리거 옵션이 올바른지 확인하십시오. 큐 속성 TRIGTYPE(FIRST)를 사용하십시오.

브릿지 모니터가 실행 중이었지만 더 이상 실행되지 않는 경우 브릿지 모니터가 실행될 것으로 예상하는 모든 CICS 리전에서 CICS 리전 CSMT 및 작업 로그의 출력을 확인하여 오류로 인해 브릿지 모니터가 종료되었는지 확인하십시오.

2. 브릿지 요청 큐가 QSGDISP(SHARED)로 정의된 경우 INDXTYPE(CORRELID)도 지정했는지 확인하십시오.
3. 처리되지 않은 인바운드 메시지를 찾아보고 MQMD.MsgId 및 MQMD.CorrelId의 값이 올바른지 확인하십시오. 이 메시지가 작업 단위 또는 의사 대화의 첫 번째 메시지인 경우 MQMD.CorrelId가 MQCI_NEW_SESSION 값으로 설정되고 MQMD.MsgId가 MQMI_NONE(2진 0)으로 설정되어야 합니다.
4. 이 메시지가 작업 단위 또는 의사 대화의 첫 번째 메시지가 아닌 경우 애플리케이션이 이전 응답 메시지에서 가능한 오류를 충분히 검사했는지 확인하십시오. 최소한 MQCIH에서 다음 필드를 검사해야 합니다.
 - MQCIH.ReturnCode
 - MQCIH.CompCode
 - MQCIH.TaskEndStatus
 - MQCIH.AbendCode
 - MQCIH.ErrorOffset

[맨 위로 이동...](#)

브릿지 모니터가 요청 큐에서 인바운드 메시지를 가져왔지만 CICS DPL 프로그램 또는 CICS 트랜잭션을 실행하는데 실패했습니다.

1. CICS MSGUSR 로그의 출력을 확인하십시오. 이 출력은 거의 언제나 DPL 프로그램 또는 트랜잭션을 실행하는데 실패한 이유를 보고합니다. 일반적인 이유는 다음과 같습니다.
 - 프로그램 또는 트랜잭션이 CICS에 정의되지 않았습니다. CEDA를 사용하여 프로그램 또는 트랜잭션을 정의하고 브릿지 애플리케이션을 다시 실행하십시오.
 - 프로그램 또는 트랜잭션을 실행할 권한이 충분하지 않습니다. CICS-MQ 브릿지에서 사용되는 인증 레벨을 제어하는 방법에 대한 세부사항은 [CICS-MQ 브릿지 설정](#)에서 제공됩니다.
2. 브릿지 모니터에서 응답 큐로 전송된 메시지를 확인하십시오. 오류가 발생한 경우 MQCIH.Format 필드가 MQFMT_STRING으로 설정되고 오류 메시지가 벡터 대신 MQCIH에 추가되었을 수 있습니다.
3. 데드 레터 큐를 확인하여 브릿지 모니터가 응답 메시지를 이 큐로 전송했는지 확인하십시오. 그러한 경우 MQMD.MsgId 및 MQMD.CorrelId의 값이 올바르면 MQDLH.Reason의 값을 확인하십시오. 이 값은 일반적으로 실패 이유를 표시하는 피드백 코드로 설정됩니다.

CICS-MQ 브릿지에 특정한 코드를 포함한 피드백 코드에 대한 정보는 [CICS 메시지](#)의 내용을 참조하십시오.

[맨 위로 이동...](#)

CICS-MQ 브릿지 태스크 이상 종료

이상 종료 코드는 MQCIH.AbendCode 필드의 아웃바운드 메시지에 설정됩니다. 또한 CICS MSGUSR 로그의 출력은 실패한 브릿지 태스크에 대한 이상 종료 코드를 보고합니다.

이상 종료는 [CICS 메시지](#)에 문서화되어 있습니다.

일부 공통 이상 종료 코드를 다음과 같이 처리할 수 있습니다.

ABRG

인바운드 메시지에 올바르지 않은 브릿지 기능 토큰이 지정되었습니다. 첫 번째 인바운드 메시지는 항상 MQCIH.Facility 필드에 MQCFAC_NONE 값을 지정하고 MQCIH.FacilityKeepTime에 0이 아닌 값을 지정해야 합니다. CICS가 MQCIH.Facility 필드에 기능 토큰을 리턴하며 의사 대화의 모든 후속 인바운드 메시지에서 이 값을 사용할 수 있습니다.

ABXH

버퍼 옵션이 있는 수신이 지정된 경우 brmq_re_buffer_indicator가 N으로 설정되었거나 brmq_re_buffer_indicator가 Y로 설정되고 버퍼 옵션이 없는 수신이 지정되었습니다.

MBRJ

MQCIH에 올바르지 않은 데이터가 있습니다. MQCIH 필드의 값을 확인하여 범위를 벗어난 값을 찾으십시오. 길이 불일치로 인해 MBRJ가 발생할 수도 있습니다. 예를 들어, brmq_vector_length의 값과 데이터 벡터의 길이가 일치하지 않거나 CICS 헤더 및 벡터에 충분한 데이터가 없는 경우입니다.

MBRN

메시지가 예상보다 더 짧습니다. 모든 벡터 구조에는 하나 또는 두 개의 데이터 길이 필드가 있습니다. 첫 번째는 모든 벡터에 대한 표준 헤더의 첫 번째 전자 필드이며 가변 길이 데이터를 포함하여 벡터의 전체 길이와 같아야 합니다. 일부 벡터에는 가변 길이 데이터의 길이만 제공하는 다른 전자 길이 필드도 포함되어 있습니다. 이러한 값이 존재하는 것보다 더 많은 데이터를 표시하는 경우 브릿지 태스크가 MBRN과 함께 이상 종료됩니다.

MBRO 및 MBRP

가변 길이 데이터가 아닌 벡터 구조에 오류가 있습니다. MQCIH 필드 ERROROFFSET는 오류가 발생한 필드의 오프셋을 제공합니다. [Link3270 메시지 형식](#)에 설명된 대로 허용되는 값과 비교하여 벡터의 필드 값을 확인하십시오.

[맨 위로 이동...](#)

CICS-MQ 브릿지 모니터가 예기치 않게 종료됨

일부 오류로 인해 브릿지 모니터 트랜잭션 CKBR이 예기치 않게 종료될 수 있습니다. 트리거된 큐를 사용하여 모니터를 시작하고 메시지가 브릿지 요청 큐에 남아 있으면 CKTI 트랜잭션이 CKBR을 다시 시작하려고 시도할 수 있습니다. 원래 오류가 지속되면 CKBR 실패가 반복될 수 있습니다. 반복을 정지하려면 기본 문제점을 진단하고 수정하는 동안 요청 큐의 TriggerControl 속성을 설정 해제하십시오.

큐 또는 CICS 트랜잭션에 액세스할 수 있는 충분한 권한이 없는 경우 또는 데드 레터 큐에 쓸 수 없거나 CICS 또는 IBM MQ 서비스를 실행할 때 문제점이 발생하는 경우 브릿지 모니터가 실패할 수 있습니다.

[맨 위로 이동...](#)

제 11 장 MQCIH – CICS-MQ 브릿지 헤더

MQCIH 구조는 z/OS용 IBM MQ를 통해 CICS-MQ 브릿지에 전송된 메시지의 시작 부분에 표시될 수 있는 정보를 설명합니다.

가용성

이 시스템에 연결된 AIX®, HP-UX, z/OS, Solaris, Linux®, Windows, IBM MQ 클라이언트.

C++ 애플리케이션의 경우 ImqCICSBridgeHeader 클래스가 MQCIH 데이터 구조의 특정 기능을 캡슐화합니다. 136 페이지의 『CICS-MQ 브릿지의 IBM MQ C++ 메시지 헤더』의 내용을 참조하십시오.

형식 이름

MQFMT_CICS.

버전

MQCIH의 현재 버전은 MQCIH_VERSION_2입니다. 지원되는 프로그래밍 언어에 대해 제공되는 INCLUDE 파일 및 헤더, COPY는 MQCIH_VERSION_2로 설정된 버전 필드의 초기값과 함께 MQCIH의 최신 버전을 포함합니다.

MQCIH_VERSION_2 미만 버전의 경우에는 CursorPosition, ErrorOffset, InputItem, Reserved4 필드가 존재하지 않습니다.

문자 세트 및 인코딩

MQCIH 구조 및 애플리케이션 메시지 데이터에 사용되는 문자 세트와 인코딩에는 다음 특수 조건이 적용됩니다.

- CICS-MQ 브릿지 큐를 소유한 큐 관리자에 연결하는 애플리케이션은 큐 관리자의 문자 세트와 인코딩으로 MQCIH 구조를 제공해야 합니다. 이 경우에는 MQCIH 구조의 데이터 변환이 수행되지 않기 때문입니다.
- 기타 큐 관리자에 연결하는 애플리케이션은 지원되는 문자 세트와 인코딩으로 MQCIH 구조를 제공할 수 있습니다. CICS-MQ 브릿지 큐를 소유한 큐 관리자에 연결된 수신 메시지 채널 에이전트가 MQCIH 구조를 변환합니다.
- MQCIH 구조 다음의 애플리케이션 메시지 데이터는 MQCIH 구조와 동일한 문자 세트 및 인코딩을 사용해야 합니다. MQCIH 구조의 CodedCharSetId 및 Encoding 필드를 사용하여 애플리케이션 메시지 데이터의 문자 세트와 인코딩을 지정할 수는 없습니다. 데이터가 큐 관리자에서 지원되는 기본 제공 형식 중 하나가 아닌 경우 애플리케이션 메시지 데이터를 변환하려면 데이터 변환 종료를 제공해야 합니다.

용도

애플리케이션에 초기값과 동일한 값이 필요하며 브릿지가 AUTH=LOCAL 또는 AUTH=IDENTIFY로 실행 중이면 메시지에서 MQCIH 구조를 생략할 수 있습니다. 다른 모든 경우에는 구조가 존재해야 합니다.

브릿지는 버전 1 또는 버전 2 MQCIH 구조 중 하나를 채택하지만 3270 트랜잭션의 경우에는 버전 2 구조를 사용해야 합니다.

애플리케이션은 요청 필드로 문서화된 필드의, 브릿지로 전송되는 메시지에 적합한 값이 있는지 확인해야 합니다. 이 필드는 브릿지의 입력으로 사용됩니다.

응답 필드로 문서화된 필드는 CICS-MQ 브릿지를 통해 브릿지가 애플리케이션에 전송하는 응답 메시지에 설정됩니다. 오류 정보는 ReturnCode, Function, CompCode, Reason, AbendCode 필드에 리턴되지만 모든 경우에 모두가 설정되는 것은 아닙니다. 다음 표는 ReturnCode의 여러 다른 값에 대해 어느 필드가 설정되는지 보여줍니다.

표 14. MQCIH 구조의 오류 정보 필드 콘텐츠				
ReturnCode	Function	CompCode	Reason	AbendCode
MQCRC_OK	–	–	–	–

표 14. MQCIH 구조의 오류 정보 필드 콘텐츠 (계속)				
ReturnCode	Function	CompCode	Reason	AbendCode
MQCRC_BRIDGE_ERROR	–	–	MQFB_CICS_*	–
MQCRC_MQ_API_ERROR MQCRC_BRIDGE_TIMEOUT	MQ 호출 이름	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

초기값 및 언어 선언

이 표에는 CICS-MQ 브릿지 헤더(MQCIH)의 필드 초기값 및 MQCIH 구조에서의 값 순서가 표시됩니다.

표 1. MQCIH의 필드 초기값		
필드 이름	상수 이름	상수 값
StrucId	MQCIH_STRUC_ID	'CIHb'
Version	MQCIH_VERSION_2	2
StrucLength	MQCIH_LENGTH_2	180
Encoding	없음	0
CodedCharSetId	없음	0
Format	MQFMT_NONE	공백
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	없음	0
ADSDescriptor	MQCADSD_NONE	0
ConversationalTask	MQCCT_NO	0
TaskEndStatus	MQCTES_NOSYNC	0
Facility	MQCFAC_NONE	널
Function	MQCFUNC_NONE	공백
AbendCode	없음	공백
Authenticator	없음	공백

표 1. MQCIH의 필드 초기값
(계속)

필드 이름	상수 이름	상수 값
Reserved1	없음	공백
ReplyToFormat	MQFMT_NONE	공백
RemoteSysId	없음	공백
RemoteTransId	없음	공백
TransactionId	없음	공백
FacilityLike	없음	공백
AttentionId	없음	공백
StartCode	MQCSC_NONE	공백
CancelCode	없음	공백
NextTransactionId	없음	공백
Reserved2	없음	공백
Reserved3	없음	공백
CursorPosition	없음	0
ErrorOffset	없음	0
InputItem	없음	0
Reserved4	없음	0

참고:

1. b 기호는 단일 공백 문자를 나타냅니다.
2. C 프로그래밍 언어에서, 매크로 변수 MQCIH_DEFAULT는 표에 나열된 값을 포함합니다. 다음과 같이 이 값을 사용하여 구조에서 필드의 초기값을 제공하십시오.

```
MQCIH MyCIH = {MQCIH_DEFAULT};
```

AbendCode (MQCHAR4)

이 필드에 리턴된 값은 ReturnCode 필드의 값이 MQCRC_APPLICATION_ABEND 또는 MQCRC_BRIDGE_ABEND인 경우에만 의미가 있습니다. 해당하는 경우 AbendCode에 CICS ABCODE 값이 포함됩니다.

이 필드는 응답 필드입니다. 이 필드의 길이는 MQ_ABEND_CODE_LENGTH에 제공됩니다. 이 필드의 초기값은 4개의 공백 문자입니다.

ADSDescriptor (MQLONG)

이 지표는 SEND 및 RECEIVE BMS 요청에 ADS 디스크립터를 보낼지 여부를 지정합니다.

다음 값이 정의됩니다.

MQCADSD_NONE

ADS 디스크립터를 보내거나 받지 않습니다.

MQCADSD_SEND

ADS 디스크립터를 보냅니다.

MQCADSD_RECV

ADS 디스크립터를 받습니다.

MQCADSD_MSGFORMAT

ADS 디스크립터에 메시지 형식을 사용합니다. 이 옵션은 ADS 디스크립터의 긴 양식을 사용하여 ADS 디스크립터를 보내거나 받습니다. 긴 양식에는 4바이트 경계에 맞추어진 필드가 있습니다.

다음과 같이 *ADSDescriptor* 필드를 설정하십시오.

- ADS 디스크립터를 사용 중이 아니면 필드를 MQCADSD_NONE으로 설정하십시오.
- 각 환경에 동일한 CCSID로 ADS 디스크립터를 사용 중인 경우 필드를 MQCADSD_SEND 및 MQCADSD_RECV의 합으로 설정하십시오.
- 각 환경에 상이한 CCSID의 ADS 디스크립터를 사용 중이면 필드를 MQCADSD_SEND, MQCADSD_RECV, MQCADSD_MSGFORMAT의 합으로 설정하십시오.

이 필드는 3270 트랜잭션에만 사용되는 요청 필드입니다. 이 필드의 초기값은 MQCADSD_NONE입니다.

AttentionId (MQCHAR4)

이 필드는 트랜잭션이 시작될 때 AID 키의 초기값입니다.

값은 1바이트 값이며, 왼쪽으로 정렬됩니다.

이 요청 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 길이는 MQ_ATTENTION_ID_LENGTH에 제공됩니다. 이 필드의 초기값은 4개의 공백입니다.

Authenticator (MQCHAR8)

이 필드는 비밀번호 또는 패스토크를 보유합니다. 사용자-ID 인증이 CICS-MQ 브릿지에 대해 활성화된 경우 Authenticator는 사용자 ID와 함께 MQMD ID 컨텍스트에 사용되어 메시지 송신자를 인증합니다.

이 필드는 요청 필드입니다. 이 필드의 길이는 MQ_AUTHENTICATOR_LENGTH에 지정됩니다. 이 필드의 초기값은 8개의 공백입니다.

CancelCode (MQCHAR4)

이 필드는 트랜잭션(일반적으로 더 많은 데이터를 요청하는 대화 트랜잭션)을 중지하는 데 사용되는 이상 종료 코드를 보유합니다. 그 외의 경우 이 필드는 공백으로 설정됩니다.

이 필드는 3270 트랜잭션에만 사용되는 요청 필드입니다. 이 필드의 길이는 MQ_CANCEL_CODE_LENGTH에 제공됩니다. 이 필드의 초기값은 4개의 공백입니다.

CodedCharSetId (MQLONG)

이 필드는 예약되어 있으며 그 값은 유의하지 않습니다.

이 필드의 초기값은 0(영)입니다.

CompCode (MQLONG)

이 필드의 리턴되는 값은 ReturnCode에 따라 다릅니다.

[125 페이지의 표 14](#)의 내용을 참조하십시오.

이 필드는 응답 필드입니다. 이 필드의 초기값은 MQCC_OK입니다.

ConversationalTask (MQLONG)

이 지표는 태스크가 자세한 정보에 대한 요청을 발행하도록 허용할지 또는 태스크를 이상 종료할지 여부를 지정합니다.

값은 다음 중 하나여야 합니다.

MQCCT_YES

태스크가 대화 가능합니다. 다수의 메시지를 사용하여 변환에 대한 벡터를 제공할 수 있는 경우 이 값을 지정하십시오.

MQCCT_NO

태스크가 대화 가능하지 않습니다. 이 CICS 트랜잭션에 필요한 모든 입력 벡터가 입력 메시지에 제공되면 이 값을 지정하십시오.

이 요청 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 초기값은 MQCCT_NO입니다.

CursorPosition (MQLONG)

이 필드는 트랜잭션이 시작될 때 초기 커서 위치를 보유합니다. 그 후에는 대화 트랜잭션의 경우 RECEIVE 벡터에 커서 위치가 있습니다.

이 요청 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 초기값은 0입니다. MQCIH_VERSION_2 미만 버전의 경우에는 이 필드가 존재하지 않습니다.

Encoding (MQLONG)

이 필드는 예약되어 있으며 그 값은 유의하지 않습니다.

이 필드의 초기값은 0(영)입니다.

ErrorOffset (MQLONG)

이 필드는 브릿지 종료로 인해 발견된 올바르지 않은 데이터의 위치를 보유합니다. 이 필드는 메시지의 시작부터 올바르지 않은 데이터 위치까지의 오프셋을 제공합니다.

이 응답 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 초기값은 0입니다. MQCIH_VERSION_2 미만 버전의 경우에는 이 필드가 존재하지 않습니다.

Facility (MQBYTE8)

이 필드는 8바이트 브릿지 기능 토큰을 보유합니다. 브릿지 기능 토큰은 다수의 트랜잭션이 유사 대화에서 동일한 브릿지 기능(가상 3270 터미널)을 사용할 수 있게 합니다.

유사 대화의 첫 번째 메시지에(또는 메시지에만) MQCFAC_NONE 값을 설정하십시오. 이 값은 CICS가 이 메시지에 대해 새 브릿지 기능을 할당하도록 지시합니다. 0이 아닌 FacilityKeepTime 값이 입력 메시지에 지정되면 브릿지 기능 토큰이 응답 메시지에서 리턴됩니다. 그러면 유사 대화의 후속 입력 메시지는 동일한 브릿지 기능 토큰을 사용해야 합니다.

다음 특수 값이 정의됩니다.

MQCFAC_NONE

기능 토큰이 지정되지 않습니다.

C 프로그래밍 언어의 경우 MQCFAC_NONE_ARRAY 상수도 정의됩니다. 이 상수는 MQCFAC_NONE과 값이 동일하지만 문자열이 아닌 문자의 배열입니다.

이 필드는 3270 트랜잭션에만 사용되는 요청 및 응답 필드입니다. 이 필드의 길이는 MQ_FACILITY_LENGTH에 제공됩니다. 이 필드의 초기값은 MQCFAC_NONE입니다.

FacilityKeepTime (MQLONG)

이 필드는 브릿지 기능이 사용자 트랜잭션 종료 후에 유지되는 시간(초)을 지정합니다.

유사 대화의 예상 지속 시간에 해당되는 값을 지정하십시오. 유사 비대화식 트랜잭션의 경우에는 값이 0이어야 합니다. 비대화식 트랜잭션의 경우 값이 0이어야 합니다.

이 요청 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 초기값은 0(영)입니다.

FacilityLike (MQCHAR4)

이 필드는 브릿지 기능의 모델로 사용할 설치된 터미널의 이름을 지정합니다.

공백 값은 FacilityLike를 브릿지 트랜잭션 프로파일 정의에서 가져오거나 기본값이 사용됨을 의미합니다.

이 요청 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 길이는 MQ_FACILITY_LIKE_LENGTH에 제공됩니다. 이 필드의 초기값은 4개의 공백입니다.

Flags (MQLONG)

이 필드는 메시지의 플래그를 보유합니다.

이 필드의 값은 다음 중 하나여야 합니다.

MQCIH_NONE

플래그가 없습니다.

MQCIH_PASS_EXPIRATION

응답 메시지에 다음이 포함됩니다.

- 요청 메시지와 동일한 만기 보고서 옵션
 - 브릿지의 처리 시간에 대한 조정 없이, 요청 메시지의 나머지 만기 시간
- 이 값을 생략하면 만기 시간이 무제한으로 설정됩니다.

MQCIH_REPLY_WITHOUT_NULLS

DPL 프로그램이 리턴한 COMMAREA의 끝에 있는 후미 널('X'00')을 제외시키기 위해 CICS DPL 프로그램 요청의 응답 메시지 길이가 조정됩니다. 이 값이 설정되지 않으면 널이 유의할 수 있으며 전체 COMMAREA가 리턴됩니다.

MQCIH_SYNC_ON_RETURN

DPL 요청에 대한 CICS 링크에 SYNCONRETURN 옵션을 사용합니다. 이 옵션을 사용하면 프로그램이 다른 CICS 리전에 제공되는 경우 완료될 때 CICS가 동기점을 잡습니다. 브릿지는 요청을 제공할 CICS 리전을 지정하지 않으며, 이는 CICS 프로그램 정의 또는 워크로드 밸런싱 기능으로 제어됩니다.

이 필드는 요청 필드입니다. 이 필드의 초기값은 MQCIH_NONE입니다.

Format (MQCHAR8)

이 필드는 MQCIH 구조 다음의 데이터의 IBM MQ 형식 이름을 보유합니다.

MQPUT 또는 MQPUT1 호출에서 애플리케이션은 이 필드를 데이터에 적합한 값으로 설정해야 합니다. 이 필드를 코딩하는 규칙은 MQMD에서 Format 필드를 코딩하는 규칙과 동일합니다.

ReplyToFormat 필드의 값이 MQFMT_NONE인 경우 이 형식 이름은 응답 메시지에 사용도 사용됩니다.

- DPL 요청의 경우 Format은 COMMAREA의 형식 이름이어야 합니다.
- 3270 요청의 경우 Format은 DFHMQDCI 또는 CSCQBDCI여야 하며, 브릿지가 응답 메시지에 대한 DFHMQDCO 또는 CSCQBDCO에 형식을 설정합니다.

이 형식의 데이터 변환 종료는 종료를 실행할 큐 관리자에 설치해야 합니다.

요청 메시지가 오류 응답 메시지를 생성하는 경우 오류 응답 메시지의 형식 이름은 MQFMT_STRING입니다.

이 필드는 요청 필드입니다. 이 필드의 길이는 MQ_FORMAT_LENGTH에 지정됩니다. 이 필드의 초기값은 MQFMT_NONE입니다.

Function (MQCHAR4)

이 필드의 리턴되는 값은 ReturnCode에 따라 다릅니다.

125 페이지의 표 14의 내용을 참조하십시오. Function에 IBM MQ 호출 이름이 포함된 경우 다음 값이 가능합니다.

MQCFUNC_MQCONN

MQCONN 호출

MQCFUNC_MQGET

MQGET 호출

MQCFUNC_MQINQ

MQINQ 호출

MQCFUNC_MQOPEN

MQOPEN 호출

MQCFUNC_MQPUT

MQPUT 호출

MQCFUNC_MQPUT1

MQPUT1 호출

MQCFUNC_NONE

호출 없음

모든 경우에 C 프로그래밍 언어의 경우 상수 MQCFUNC_*_ARRAY도 정의됩니다. 이 상수는 해당 MQCFUNC_* 상수와 값이 동일하지만 문자열이 아닌 문자의 배열입니다.

이 필드는 응답 필드입니다. 이 필드의 길이는 MQ_FUNCTION_LENGTH에 제공됩니다. 이 필드의 초기값은 MQCFUNC_NONE입니다.

GetWaitInterval (MQLONG)

이 필드는 UOWControl 값이 MQCUOWC_FIRST일 때에 적용됩니다. 브릿지가 발행한 MQGET 호출이 메시지로 시작된 작업 단위의 두 번째 및 후속 요청 메시지를 대기하는 대략의 시간(밀리초)을 송신 애플리케이션이 지정할 수 있습니다. 이 값은 브릿지에 사용된 기본 대기 간격을 대체합니다.

다음 특수 값을 사용할 수 있습니다.

MQCGWI_DEFAULT

기본 대기 간격. 이 값을 사용하면 CICS-MQ 브릿지가 시작할 때 지정된 시간 동안 브릿지가 대기합니다.

MQWI_UNLIMITED

무제한 대기 간격.

이 필드는 요청 필드입니다. 이 필드의 초기값은 MQCGWI_DEFAULT입니다.

InputItem (MQLONG)

이 필드는 예약되어 있습니다. 값은 0이어야 합니다.

MQCIH_VERSION_2 미만 버전의 경우 이 필드가 존재하지 않습니다.

LinkType (MQLONG)

이 필드는 브릿지가 링크하려 시도하는 오브젝트의 유형을 표시합니다.

값은 다음 중 하나여야 합니다.

MQCLT_PROGRAM

DPL 프로그램

MQCLT_TRANSACTION

3270 트랜잭션

이 필드는 요청 필드입니다. 이 필드의 초기값은 MQCLT_PROGRAM입니다.

NextTransactionId (MQCHAR4)

이 필드는 사용자 트랜잭션을 통해 리턴되는(일반적으로 **EXEC CICS RETURN TRANSID**를 통해) 다음 트랜잭션의 이름을 보유합니다. 다음 트랜잭션이 존재하지 않는 경우 이 필드는 공백으로 설정됩니다.

이 응답 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 길이는 MQ_TRANSACTION_ID_LENGTH에 제공됩니다. 이 필드의 초기값은 4개의 공백입니다.

OutputDataLength (MQLONG)

이 필드는 응답 메시지의 클라이언트에 리턴할 사용자 데이터의 길이를 보유합니다. 이 길이는 8바이트 프로그램 이름을 포함합니다.

요청이 COMMAREA의 사용자 프로그램에 데이터를 전달 중이면 사용자 프로그램은 COMMAREA에 응답 데이터를 리턴할 수 있습니다. 응답 데이터는 요청 데이터보다 더 길 수 있습니다. 그러한 경우에는 OutputDataLength에 응답 데이터의 길이를 지정해야 합니다. 이 길이는 8바이트 프로그램 이름을 포함합니다.

요청이 채널/컨테이너의 사용자 프로그램에 데이터를 전달 중이면 사용자 프로그램은 컨테이너에 응답 데이터를 리턴할 수 있습니다. 컨테이너 자체가 보유한 데이터의 길이를 표시하므로 OutputDataLength가 사용되지 않습니다.

참고: 메시지의 사용자 데이터 길이는 MQCIH 구조를 제외한 메시지의 길이입니다.

요청 메시지의 사용자 데이터 길이가 OutputDataLength보다 작으면 LINK 명령의 DATALENGTH 옵션이 사용 됩니다. 이 옵션은 LINK를 다른 CICS 리전에 효율적으로 Function-Shipping할 수 있게 합니다.

다음 특수 값을 사용할 수 있습니다.

MQCODL_AS_INPUT

출력 길이가 입력 길이와 동일합니다.

응답이 요청되지 않은 경우에도 링크된 프로그램에 전달된 COMMAREA가 충분한 크기가 되도록 하기 위해 이 값이 필요할 수 있습니다.

이 요청 필드는 DPL 프로그램에만 사용됩니다. 이 필드의 초기값은 MQCODL_AS_INPUT입니다.

Reason (MQLONG)

이 필드의 리턴되는 값은 ReturnCode에 따라 다릅니다.

[125 페이지의 표 14](#)의 내용을 참조하십시오.

이 필드는 응답 필드입니다. 이 필드의 초기값은 MQRC_NONE입니다.

RemoteSysId (MQCHAR4)

이 필드는 요청을 처리 중인 CICS 시스템의 CICS 시스템 ID를 보유합니다. 이 필드가 공백이면 CICS 시스템 요청은 브릿지 모니터와 동일한 CICS 시스템에서 처리됩니다. 사용된 SYSID는 응답 메시지에 리턴됩니다.

3270 유사 대화의 경우 대화의 모든 후속 메시지가 초기 응답에 리턴된 원격 SYSID를 지정해야 합니다. 지정된 경우 SYSID의 특성은 다음과 같아야 합니다.

- 활성 상태여야 함
- IBM MQ 요청 큐에 대한 액세스 권한이 있어야 함
- 브릿지 모니터가 실행 중인 CICS 시스템에서 CICS ISC 링크가 액세스 가능해야 함

RemoteTransId (MQCHAR4)

이 필드는 선택적 요청 필드입니다. 지정된 경우 이 필드는 CICS START의 RTRANSID 값으로 사용됩니다.

이 필드의 길이는 MQ_TRANSACTION_ID_LENGTH에 제공됩니다.

ReplyToFormat (MQCHAR8)

이 필드는 현재 메시지에 대한 응답으로 전송되는 응답 메시지의 IBM MQ 형식 이름을 보유합니다.

요청이 COMMAREA의 사용자 프로그램에 데이터를 전달 중이면 사용자 프로그램은 COMMAREA에 응답 데이터를 리턴할 수 있습니다. 응답 데이터는 2진 또는 문자 형식일 수 있습니다. ReplyToFormat은 표시할 WebSphere® MQ 형식 이름을 포함해야 합니다.

요청이 채널 및 컨테이너의 사용자 프로그램에 데이터를 전달 중이면 사용자 프로그램은 컨테이너에 응답 데이터를 리턴할 수 있습니다. 컨테이너 자체가 보유한 데이터의 형식을 표시하므로 ReplyToFormat이 사용되지 않습니다.

이 필드를 코딩하는 규칙은 MQMD에서 Format 필드를 코딩하는 규칙과 동일합니다.

이 요청 필드는 DPL 프로그램에만 사용됩니다. 이 필드의 길이는 MQ_FORMAT_LENGTH에 지정됩니다. 이 필드의 초기값은 MQFMT_NONE입니다.

Reserved1 (MQCHAR8)

이 필드는 예약되어 있습니다. 값은 8개의 공백이어야 합니다.

Reserved2 (MQCHAR8)

이 필드는 예약되어 있습니다. 값은 8개의 공백이어야 합니다.

Reserved3 (MQCHAR8)

이 필드는 예약되어 있습니다. 값은 8개의 공백이어야 합니다.

Reserved4 (MQLONG)

이 필드는 예약되어 있습니다. 값은 0이어야 합니다.

MQCIH_VERSION_2 미만 버전의 경우 이 필드가 존재하지 않습니다.

ReturnCode (MQLONG)

이 필드는 브릿지가 수행한 처리의 결과를 설명하는 CICS 브릿지 모니터의 리턴 코드를 보유합니다.

Function, CompCode, Reason, AbendCode 필드에 추가 정보가 있을 수 있습니다. [125 페이지의 표 14](#)의 내용을 참조하십시오.

값은 다음 중 하나여야 합니다.

MQCRC_APPLICATION_ABEND

(5, X'005') 애플리케이션이 비정상적으로 종료되었습니다.

MQCRC_BRIDGE_ABEND

(4, X'004') CICS 브릿지가 비정상적으로 종료되었습니다.

MQCRC_BRIDGE_ERROR

(3, X'003') CICS 브릿지에서 오류를 발견했습니다.

MQCRC_BRIDGE_TIMEOUT

(8, X'008') 현재 작업 단위의 두 번째 또는 그 이후의 메시지가 지정된 시간 내에 수신되지 않았습니다.

MQCRC_CICS_EXEC_ERROR

(1, X'001') EXEC CICS 명령문에서 오류를 발견했습니다.

MQCRC_MQ_API_ERROR

(2, X'002') MQ 호출에서 오류를 발견했습니다.

MQCRC_OK

(0, X'000') 오류가 없습니다.

MQCRC_PROGRAM_NOT_AVAILABLE

(7, X'007') 프로그램을 사용할 수 없습니다.

MQCRC_SECURITY_ERROR

(6, X'006') 보안 오류가 발생했습니다.

MQCRC_TRANSID_NOT_AVAILABLE

(9, X'009') 트랜잭션을 사용할 수 없습니다.

이 필드는 응답 필드입니다. 이 필드의 초기값은 MQCRC_OK입니다.

StartCode (MQCHAR4)

이 지표는 브릿지가 START로 시작된 트랜잭션 또는 터미널 트랜잭션을 에뮬레이트하는지 여부를 지정합니다.

값은 다음 중 하나여야 합니다.

MQCSC_START
시작

MQCSC_STARTDATA
데이터 시작

MQCSC_TERMINPUT
터미널 입력

MQCSC_NONE
없음

모든 경우에 C 프로그래밍 언어의 경우 상수 MQCSC_*_ARRAY도 정의됩니다. 이 상수는 해당 MQCSC_* 상수와 값이 동일하지만 문자열이 아닌 문자의 배열입니다.

브릿지의 응답에서, 이 필드는 NextTransactionId 필드에 포함된 다음 트랜잭션 ID에 적합한 시작 코드로 설정됩니다. 다음 시작 코드가 응답에 사용 가능합니다.

- MQCSC_START
- MQCSC_STARTDATA
- MQCSC_TERMINPUT

이 필드는 요청 및 응답 필드입니다.

이 필드는 3270 트랜잭션에만 사용됩니다. 이 필드의 길이는 MQ_START_CODE_LENGTH에 제공됩니다. 이 필드의 초기값은 MQCSC_NONE입니다.

StrucId (MQCHAR4)

이 필드는 CICS 정보 헤더 구조의 ID를 보유합니다.

값은 다음과 같아야 합니다.

MQCIH_STRUC_ID
CICS 정보 헤더 구조의 ID입니다.

C 프로그래밍 언어의 경우 MQCIH_STRUC_ID_ARRAY 상수도 정의됩니다. 이 상수는 MQCIH_STRUC_ID와 값이 동일하지만 문자열이 아닌 문자의 배열입니다.

이 필드는 요청 필드입니다. 이 필드의 초기값은 MQCIH_STRUC_ID입니다.

StrucLength (MQLONG)

이 필드는 CICS 정보 헤더 구조의 길이를 지정합니다.

값은 다음 중 하나여야 합니다.

MQCIH_LENGTH_1
버전 1 CICS 정보 헤더 구조의 길이입니다.

MQCIH_LENGTH_2
버전 2 CICS 정보 헤더 구조의 길이입니다.

다음 상수는 현재 버전의 길이를 지정합니다.

MQCIH_CURRENT_LENGTH
현재 버전의 CICS 정보 헤더 구조의 길이입니다.

이 필드는 요청 필드입니다. 이 필드의 초기값은 MQCIH_LENGTH_2입니다.

TaskEndStatus (MQLONG)

이 필드는 태스크 종료 시 사용자 트랜잭션의 상태를 표시합니다.

다음 값 중 하나가 리턴됩니다.

MQCTES_NOSYNC

동기화되지 않습니다.

사용자 트랜잭션이 아직 완료되지 않았고 동기점을 잡지 못했습니다. 이 경우 MQMD의 MsgType 필드는 MQMT_REQUEST입니다.

MQCTES_COMMIT

작업 단위를 커밋합니다.

사용자 트랜잭션이 아직 완료되지 않았지만 첫 번째 작업 단위에 대한 동기점을 잡았습니다. 이 경우 MQMD의 MsgType 필드는 MQMT_DATAGRAM입니다.

MQCTES_BACKOUT

작업 단위를 백아웃합니다.

사용자 트랜잭션이 아직 완료되지 않았습니다. 현재 작업 단위가 백아웃됩니다. 이 경우 MQMD의 MsgType 필드는 MQMT_DATAGRAM입니다.

MQCTES_ENDTASK

태스크를 종료합니다.

사용자 트랜잭션이 종료(또는 이상 종료)되었습니다. 이 경우 MQMD의 MsgType 필드는 MQMT_REPLY입니다.

이 필드는 3270 트랜잭션에만 사용되는 응답 필드입니다. 이 필드의 초기값은 MQCTES_NOSYNC입니다.

TransactionId (MQCHAR4)

이 필드는 CICS에서 실행할 사용자 트랜잭션의 트랜잭션 ID 또는 CICS 프로그램을 실행할 트랜잭션 코드를 제공합니다.

LinkType 필드의 값이 MQCLT_TRANSACTION인 경우 TransactionId는 실행할 사용자 트랜잭션의 트랜잭션 ID이며, 이 경우 공백이 없는 값을 지정하십시오.

LinkType 필드의 값이 MQCLT_PROGRAM이면 TransactionId는 작업 단위의 모든 프로그램을 실행할 트랜잭션 코드입니다. 공백 값을 지정하는 경우에는 CICS DPL 브릿지 기본 트랜잭션 코드(CKBP)가 사용됩니다. 공백이 아닌 값이면 DFHMQBP0 또는 DFHMQBP3을 통해 초기 프로그램으로 CICS에 정의한 로컬 트랜잭션 ID, CKBP 또는 CKBC를 지정해야 합니다. TransactionId 필드는 UOWControl 필드의 값이 MQCUOWC_FIRST 또는 MQCUOWC_ONLY일 때에만 적용됩니다.

TransactionId 필드는 요청 필드입니다. 이 필드의 길이는 MQ_TRANSACTION_ID_LENGTH에 제공됩니다. 이 필드의 초기값은 4개의 공백입니다.

UOWControl (MQLONG)

이 필드는 CICS 브릿지가 수행한 작업 단위 처리를 제어합니다.

브릿지가 작업 단위에서 하나 이상의 프로그램 또는 단일 트랜잭션을 실행하도록 요청할 수 있습니다. 이 필드에는 CICS 브릿지가 작업 단위를 시작하거나, 현재 작업 단위에서 요청된 기능을 수행하거나, 커밋 또는 백아웃을 통해 작업 단위를 커밋하는지 여부가 표시됩니다. 데이터 전송 플로우를 최적화하기 위해 다양한 조합이 지원됩니다.

값은 다음 중 하나여야 합니다.

MQCUOWC_ONLY

작업 단위를 시작하고 기능을 수행한 후 작업 단위를 커밋합니다.

MQCUOWC_CONTINUE

현재 작업 단위의 추가 데이터입니다(3270만 해당).

MQCUOWC_FIRST

작업 단위를 시작하고 기능을 수행합니다.

MQCUOWC_MIDDLE

현재 작업 단위에서 기능을 수행합니다.

MQCUOWC_LAST

기능을 수행한 후 작업 단위를 커밋합니다.

MQCUOWC_COMMIT

작업 단위를 커밋합니다(DPL만 해당).

MQCUOWC_BACKOUT

작업 단위를 백아웃합니다(DPL만 해당).

이 필드는 요청 필드입니다. 이 필드의 초기값은 MQCUOWC_ONLY입니다.

Version (MQLONG)

이 필드는 CICS 정보 헤더 구조의 버전을 식별합니다.

값은 다음 중 하나여야 합니다.

MQCIH_VERSION_1

버전 1 CICS 정보 헤더 구조입니다.

MQCIH_VERSION_2

버전 2 CICS 정보 헤더 구조입니다.

최신 버전의 구조에만 존재하는 필드는 필드의 설명에서처럼 식별됩니다. 다음 상수는 현재 버전의 버전 번호를 지정합니다.

MQCIH_CURRENT_VERSION

CICS 정보 헤더 구조의 현재 버전입니다.

이 필드는 요청 필드입니다. 이 필드의 초기값은 MQCIH_VERSION_2입니다.

CICS-MQ 브릿지의 IBM MQ C++ 메시지 헤더

z/OS용 IBM MQ를 통해 CICS-MQ 브릿지로 메시지를 보내는 C++ 애플리케이션은 ImqCICSBridgeHeader 클래스를 사용하여 MQCIH 데이터 구조와 상호작용합니다.

다음 예는 CICS-MQ 브릿지 헤더 정보를 메시지에 추가하는 방법을 보여줍니다.

```
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 );

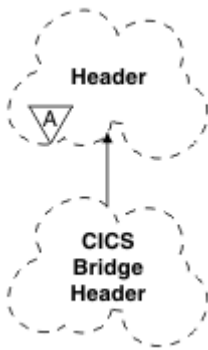
// Set up the CICS-MQ bridge header information.
// The reply-to format is often specified.
// Other attributes can be specified, but all have default values.
header.setReplyToFormat( /* ? */ );

// 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 클래스

ImqCICSBridgeHeader 클래스는 MQCIH 데이터 구조의 특정 기능을 캡슐화합니다. 이 클래스의 오브젝트는 z/OS용 IBM MQ를 통해 CICS-MQ 브릿지에 메시지를 보내는 애플리케이션에 사용됩니다.



이 클래스의 오브젝트 속성은 다음과 같습니다.

그림 13. *ImqCICSBridgeHeader* 클래스

ADS 디스크립터

송/수신 ADS 디스크립터. MQCADSD_NONE으로 디스크립터를 설정하십시오. 초기값은 MQCADSD_NONE입니다. 다음 추가 값이 가능합니다.

- MQCADSD_NONE
- MQCADSD_SEND
- MQCADSD_RECV
- MQCADSD_MSGFORMAT

어텐션 ID

AID 키입니다. 필드의 길이는 MQ_ATTENTION_ID_LENGTH여야 합니다.

인증자

RACF 비밀번호 또는 패스티켓입니다. 초기값은 MQ_AUTHENTICATOR_LENGTH 길이의 공백을 포함합니다.

브릿지 이상 종료 코드

MQ_ABEND_CODE_LENGTH 길이의 브릿지 이상 종료 코드입니다. 초기값은 4개의 공백 문자입니다. 이 필드의 리턴되는 값은 리턴 코드에 따라 다릅니다.

브릿지 취소 코드

브릿지 이상 종료 트랜잭션 코드입니다. 필드는 예약되어 있으며, 공백을 포함하고 MQ_CANCEL_CODE_LENGTH 길이어야 합니다.

브릿지 완료 코드

IBM MQ 완료 코드 또는 CICS EIBRESP 값을 포함할 수 있는 완료 코드입니다. 이 필드의 초기값은 MQCC_OK입니다. 이 필드의 리턴되는 값은 리턴 코드에 따라 다릅니다.

브릿지 오류 오프셋

브릿지 오류 오프셋입니다. 초기값은 0입니다. 이 속성은 읽기 전용입니다.

브릿지 이유 코드

이유 코드입니다. 이 필드는 IBM MQ 이유 또는 CICS EIBRESP2 값을 포함할 수 있습니다. 이 필드의 초기값은 MQRC_NONE입니다. 이 필드의 리턴되는 값은 리턴 코드에 따라 다릅니다.

브릿지 리턴 코드

브릿지의 리턴 코드입니다. 초기값은 MQCRC_OK입니다.

대화형 태스크

태스크가 대화형인지 여부를 표시합니다. 초기값은 MQCCT_NO입니다. 다음 추가 값이 가능합니다.

- MQCCT_YES
- MQCCT_NO

커서 위치

커서 위치입니다. 초기값은 0입니다.

기능 유지 시간

CICS 브릿지 기능 해제 시간입니다.

기능 선택

터미널 에뮬레이트 속성입니다. 이 필드의 길이는 MQ_FACILITY_LIKE_LENGTH여야 합니다.

기능 토큰

BVT 토큰 값입니다. 이 필드의 길이는 MQ_FACILITY_LENGTH여야 합니다. 초기값은 MQCFAC_NONE입니다.

함수

IBM MQ 호출 이름 또는 CICS EIBFN 함수를 포함할 수 있는 함수입니다. 이 필드의 초기값은 MQ_FUNCTION_LENGTH 길이의 MQCFUNC_NONE입니다. 이 필드의 리턴되는 값은 리턴 코드에 따라 다릅니다.

함수에 IBM MQ 호출 이름이 포함된 경우 다음 추가 값이 가능합니다.

- MQCFUNC_MQCONN
- MQCFUNC_MQGET
- MQCFUNC_MQINQ
- MQCFUNC_NONE
- MQCFUNC_MQOPEN
- MQCFUNC_PUT
- MQCFUNC_MQPUT1

Get 대기 간격

브릿지 태스크를 통해 발행된 MQGET 호출의 대기 간격입니다. 초기값은 MQCGWI_DEFAULT입니다. UOW 제어의 값이 MQCUOWC_FIRST일 때에만 필드가 적용됩니다. 다음 추가 값이 가능합니다.

- MQCGWI_DEFAULT
- MQWI_UNLIMITED

링크 유형

링크 유형입니다. 초기값은 MQCLT_PROGRAM입니다. 다음 추가 값이 가능합니다.

- MQCLT_PROGRAM
- MQCLT_TRANSACTION

다음 트랜잭션 ID

첨부할 다음 트랜잭션의 ID입니다. 이 필드는 MQ_TRANSACTION_ID_LENGTH 길이어야 합니다.

출력 데이터 길이

COMMAREA 데이터 길이입니다. 초기값은 MQCODL_AS_INPUT입니다.

회신 형식

응답 메시지의 형식 이름입니다. 초기값은 MQ_FORMAT_LENGTH 길이의 MQFMT_NONE입니다.

시작 코드

트랜잭션 시작 코드입니다. 이 필드는 MQ_START_CODE_LENGTH 길이어야 합니다. 초기값은 MQCSC_NONE입니다. 다음 추가 값이 가능합니다.

- MQCSC_START
- MQCSC_STARTDATA
- MQCSC_TERMINPUT
- MQCSC_NONE

태스크 종료 상태

태스크 종료 상태입니다. 초기값은 MQCTES_NOSYNC입니다. 다음 추가 값이 가능합니다.

- MQCTES_COMMIT
- MQCTES_BACKOUT
- MQCTES_ENDTASK
- MQCTES_NOSYNC

트랜잭션 ID

첨부할 트랜잭션의 ID입니다. 초기값은 공백을 포함하고 MQ_TRANSACTION_ID_LENGTH 길이어야 합니다. UOW 제어의 값이 MQCUOWC_FIRST 또는 MQCUOWC_ONLY일 때에만 필드가 적용됩니다.

UOW 제어

UOW 제어입니다. 초기값은 MQCUOWC_ONLY입니다. 다음 추가 값이 가능합니다.

- MQCUOWC_FIRST
- MQCUOWC_MIDDLE
- MQCUOWC_LAST
- MQCUOWC_ONLY
- MQCUOWC_COMMIT
- MQCUOWC_BACKOUT
- MQCUOWC_CONTINUE

버전

MQCIH 버전 번호입니다. 초기값은 MQCIH_VERSION_2입니다. 지원되는 다른 유일한 값은 MQCIH_VERSION_1입니다.

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 본 자료는 다른 언어로도 제공될 수 있습니다. 그러나 자료에 접근하기 위해서는 해당 언어로 된 제품 또는 제품 버전의 사본이 필요할 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 31FC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 31FC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 인물 또는 기업의 이름과 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 애플리케이션이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 애플리케이션 프로그래밍 인터페이스(API)에 부합하는 애플리케이션을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 본 샘플 프로그램은 일체의 보증 없이 "현상태대로" 제공됩니다. IBM은 귀하의 샘플 프로그램 사용과 관련되는 손해에 대해 책임을 지지 않습니다.

프로그래밍 인터페이스 정보

CICS에서는 프로그래밍 인터페이스로 간주될 수 있는 몇 가지 문서와 프로그래밍 인터페이스로 간주될 수 없는 몇 가지 문서를 제공합니다.

고객이 CICS Transaction Server for z/OS, 버전 5 릴리스 6 의 서비스를 얻는 프로그램을 작성하는 데 사용할 수 있는 프로그래밍 인터페이스는 온라인 제품 문서의 다음 절에 포함되어 있습니다.

- [애플리케이션 개발](#)
- [시스템 프로그램 개발](#)
- [CICS TS 보안](#)
- [외부 인터페이스 개발](#)
- [애플리케이션 개발 참조서](#)
- [참조: 시스템 프로그래밍](#)
- [참조: 연결성](#)

CICS Transaction Server for z/OS, 버전 5 릴리스 6 의 프로그래밍 인터페이스로 사용하지 않아야 하지만 프로그래밍 인터페이스로 오해할 수 있는 정보는 온라인 제품 문서의 다음 절에 포함되어 있습니다.

- [문제점 해결 및 지원](#)
- [CICS TS 진단 참조](#)

PDF 형식 매뉴얼의 CICS 문서에 액세스하려는 경우 고객이 CICS Transaction Server for z/OS, 버전 5 릴리스 6 의 서비스를 얻는 프로그램을 작성하는 데 사용할 수 있는 프로그래밍 인터페이스는 다음 매뉴얼에 포함되어 있습니다.

- 애플리케이션 프로그래밍 안내서 및 애플리케이션 프로그래밍 참조서
- Business Transaction Services
- 사용자 정의 안내서
- C++ OO 클래스 라이브러리
- 디버깅 도구 인터페이스 참조
- 분산 트랜잭션 프로그래밍 안내서
- 외부 인터페이스 안내서
- Front End Programming Interface 안내서
- IMS 데이터베이스 제어 안내서
- 설치 안내서
- 보안 안내서
- 제공 트랜잭션
- CICSplex SM 워크로드 관리
- CICSplex SM 관리 자원 사용법
- CICSplex SM 애플리케이션 프로그래밍 안내서 및 애플리케이션 프로그래밍 참조서

- CICS의 Java 애플리케이션

PDF 형식 매뉴얼의 CICS 문서에 액세스하는 경우 CICS Transaction Server for z/OS, 버전 5 릴리스 6 의 프로 그래밍 인터페이스로 사용하지 않아야 하지만 프로그래밍 인터페이스로 오해할 수 있는 정보는 다음 매뉴얼에 포함되어 있습니다.

- 데이터 영역
- 진단 참조
- 문제점 판별 안내서
- CICSplex SM 문제점 판별 안내서

상표

IBM, IBM 로고 및 ibm.com®은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또 는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹 [저작권 및 상표 정보](http://www.ibm.com/legal/copytrade.shtml)(www.ibm.com/legal/copytrade.shtml)에 있습니다.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표 또는 상표입니다.

Apache, Apache Axis2, Apache Maven, Apache Ivy, Apache Software Foundation(ASF) 로고 및 ASF 기능 로고는 Apache Software Foundation의 상표입니다.

Gradle 및 Gradlephant 로고는 미국 또는 기타 국가에서 사용되는 Gradle, Inc.의 등록상표입니다.

Intel, Intel 로고, Intel Inside Inside, Intel Inside 로고, Intel Centrino, Intel Centrino 로고, Celeron, Intel Xeon, Intel SpeedStep SpeedStep, Itanium 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 그 계열사의 상표 또는 등록상표입니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Red Hat® 및 Hibernate®는 미국 또는 기타 국가에서 사용되는 Red Hat, Inc.의 상표 또는 등록상표입니다.

Spring Boot는 미국 또는 기타 국가에서 사용되는Pivotal Software, Inc.의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Zowe™, Zowe 로고 및 Open Mainframe Project™는 The Linux Foundation의 상표입니다.

제품 문서의 이용 약관

다음 이용 약관에 따라 이 책을 사용할 수 있습니다.

적용성

본 이용 약관은 IBM 웹 사이트의 모든 이용 약관에 추가됩니다.

개인적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 개인적, 비상업적 용도로 복제할 수 있습니다. 귀 하는 IBM의 명시적 동의 없이 본 발행물 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습 니다.

상업적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 귀하 기업집단 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 귀하의 기업집단 외에서는 IBM의 명시적 동의 없이 이 책의 2차적 저작물을 만들거나 이 책 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

권한

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 본 발행물의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 이 책의 내용과 관련하여 아무런 보장을 하지 않습니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

IBM 온라인 개인정보처리방침

서비스 솔루션 소프트웨어를 비롯한 IBM 소프트웨어 제품(소프트웨어 오퍼링)은 제품 사용 정보 수집, 일반 사용자 편의성 향상, 일반 사용자와의 상호작용 조정 및 기타 목적을 위해 쿠키 또는 기타 기술을 사용할 수 있습니다. 많은 경우에 있어서, 소프트웨어 오퍼링은 개인 식별 정보를 수집하지 않습니다. IBM의 일부 소프트웨어 오퍼링은 귀하가 개인 식별 정보를 수집하도록 도울 수 있습니다. 본 소프트웨어 오퍼링이 쿠키를 사용하여 개인 식별 정보를 수집할 경우, 본 오퍼링의 쿠키 사용에 대한 특정 정보가 다음에 규정되어 있습니다:

CICSplex SM 웹 사용자 인터페이스(기본 인터페이스)의 경우:

이 소프트웨어 오퍼링은 배치된 구성에 따라 세션 관리, 인증, 사용자 편리성 개선, 기타 사용량의 추적이나 기능적인 용도로 각 사용자의 이름과 개인 정보를 수집하는 세션 및 지속적 쿠키를 사용할 수 있습니다. 이러한 쿠키를 사용하지 못하도록 할 수는 없습니다.

CICSplex SM 웹 사용자 인터페이스(데이터 인터페이스)의 경우:

본 소프트웨어 오퍼링은 배치된 구성에 따라 세션 관리, 인증 또는 기타 사용량 추적이나 기능의 용도로 각 사용자의 사용자 이름 및 개인 식별 정보를 수집하는 세션 쿠키를 사용할 수 있습니다. 이러한 쿠키를 사용하지 못하도록 할 수는 없습니다.

CICSplex SM 웹 사용자 인터페이스("hello world" 페이지)의 경우:

배치된 구성에 따라 이 소프트웨어 오퍼링은 개인 식별 정보를 수집하지 않는 세션 쿠키를 사용할 수 있습니다. 이러한 쿠키를 사용하지 못하도록 할 수는 없습니다.

CICS Explorer의 경우:

배치된 구성에 따라 이 소프트웨어 오퍼링은 세션 관리, 인증 및 싱글 사인온 구성을 위해 사용자의 사용자 이름 및 비밀번호를 수집하는 세션 및 지속적 환경 설정을 사용할 수 있습니다. 사인온 중에 선택란을 선택하여 사용자 비밀번호를 암호화된 양식으로 저장하면 사용자의 명시적인 조치에 의해서만 사용으로 설정할 수 있지만 이러한 환경 설정은 사용 안함으로 설정할 수 없습니다.

이 소프트웨어 오퍼링에 대해 배치된 구성이 고객님의 귀하에게 쿠키 및 기타 기술을 통해 일반 사용자로부터 개인적으로 식별 가능한 정보를 수집하는 기능을 제공하는 경우에는 공지사항 및 동의에 대한 요구사항을 포함하여 해당 데이터 콜렉션에 적용할 수 있는 법률에 대한 자체 법률 자문을 구해야 합니다.

해당 용도로 쿠키를 비롯한 다양한 기술을 사용하는 데 관한 자세한 정보는 [IBM 개인정보처리방침](#) 및 [IBM 온라인 개인정보처리방침](#)의 쿠키, 웹 비콘 및 기타 기술 절과 [IBM 소프트웨어 제품 및 SaaS\(Software-as-a-Service\) 개인정보 보호정책](#)을 참조하십시오.

색인

A

AbendCode 필드 [127](#)

ADS

CICS-MQ 브릿지 벡터에서 [92](#)

CICS-MQ 브릿지 벡터에서 사용됨 [88](#)

ADSDescriptor 필드 [127](#)

AttentionId 필드 [128](#)

Authenticator 필드 [128](#)

B

BMS

CICS-MQ 브릿지 사용 [87](#)

BRMQ

CICS-MQ 브릿지를 사용한 아웃바운드 구조 [77](#)

CICS-MQ 브릿지를 사용한 인바운드 구조 [77](#)

C

C++ [136](#)

CancelCode 필드 [128](#)

CEDF [111](#)

CEMT I TASK

CICS-MQ 브릿지를 사용한 예 [78](#)

CICS

시스템 정의(CSD) 데이터 세트 [15](#)

인다우트 작업 단위 해결 [115](#)

자동 설치 [6](#)

작업 단위 [116](#)

저장영역 보호 기능 [105](#)

CICS 3270 브릿지

아웃바운드 메시지 구조 [77](#)

애플리케이션 프로그래밍 [76](#)

인바운드 메시지 구조 [77](#)

MQCIH 필드 설정 [81](#)

CICS 실행 진단 기능 [111](#)

CICS 어댑터

정상 종료(quiesced shutdown) [111](#)

CICS 프로그램의 실행 키 [105](#)

CICS DPL 브릿지

메시지 구조 [73](#)

분산 환경의 트랜잭션 [86](#)

애플리케이션 프로그래밍 [75](#)

작업 단위(UOW) 관리 [83](#)

MsgId 및 CorrelId 관리 [83](#)

CICS-MQ 브릿지

3270 레거시 애플리케이션 [94](#)

동기점이 있는 트랜잭션 [77](#)

레거시 애플리케이션 [73](#)

벡터 사용 [76](#)

분산형 프로그래밍 [86](#)

시작 데이터가 있는 트랜잭션 [76](#)

아웃바운드 BRMQ 구조 [77](#)

애플리케이션 데이터 구조(ADS) [73](#), [92](#)

인바운드 BRMQ 구조 [77](#)

작업 단위(UOW) 관리 [84](#)

CICS-MQ 브릿지 (계속)

최적화되지 않는 예물레이션 예 [95](#)

최적화된 예물레이션 예 [96](#)

트랜잭션 [75](#)

BMS(Basic Mapping Support) [87](#)

CEMT I TASK 예 [78](#)

COMMAREA 데이터 [73](#)

DPL 프로그램 사용 [73](#)

MQCIH 필드 설정 [80](#)

MQMD 필드 설정 [79](#)

MsgId 및 CorrelId 관리 [84](#)

RECEIVE MAP 벡터 해석 [92](#)

SEND MAP 벡터 해석 [88](#)

z/OS의 애플리케이션 [73](#)

CICS-MQ 어댑터

CICS-MQ API 교차 엑시트 [103](#)

CICS-MQ API 교차 엑시트

샘플 [106](#)

호출 [104](#)

CSQCAPX [106](#)

CICS-WebSphere MQ 브릿지

튜닝 고려사항 [24](#)

CICS-WebSphere MQ 브릿지 튜닝 [24](#)

CICS-WebSphere MQ 어댑터

그림 [3](#)

자동 다시 연결 [113](#)

재동기화 [113](#)

컴포넌트 [3](#)

CICSplex SM 보기 [30](#)

CKAM 트랜잭션 [4](#)

CKBM 보안 [65](#)

CKCN 보안 [65](#)

CKDL 보안 [65](#)

CKDP 보안 [65](#)

CKQC

DISPLAY 명령 [40](#)

START 명령 [32](#)

STARTCKTI 명령 [54](#)

STOP 명령 [37](#)

STOPCKTI 명령 [56](#)

CKQC 트랜잭션

보안 [65](#)

CKQQ, 트랜지언트 데이터 큐 [29](#)

CKRS 보안 [65](#)

CKRT 보안 [65](#)

CKSD 보안 [65](#)

CKSG MCA 트랜잭션 [51](#)

CKSQ 보안 [65](#)

CKTI [53](#)

CKTI 트랜잭션

보안 [65](#)

시작 [51](#)

시작 자동화 [55](#)

중지 [55-57](#)

표시 [57](#)

CodedCharSetId 필드 [128](#)

COMMAREA 데이터

COMMAREA 데이터 (계속)
CICS-MQ 브릿지 사용 [73](#)
CompCode 필드 [128](#)
ConversationalTask 필드 [128](#)
CorrelId
CICS-MQ 브릿지로 관리 [83](#), [84](#)
CREATE MQCONN [29](#)
CREATE MQMONITOR [29](#)
CSD(CICS System Definition) 데이터 세트 [15](#)
CSQ1LOGP(로그 인쇄 유틸리티)
시작 RBA 찾기 [116](#)
CSQ4CVCN [100](#)
CSQ4CVCT [100](#)
CSQ4CVEV [100](#)
CSQ4CVPT
구문 [102](#)
CSQ4CVRG [100](#)
CSQ4SAMP [5](#), [15](#)
CSQCAPX 샘플 API 교차 엑시트 프로그램 [103](#)
CSQCAPX API 교차 엑시트 샘플 [106](#)
CSQCAT1 [5](#), [15](#)
CSQCKB [5](#), [15](#)
CSQCQCON [34](#)
CSQCSTUB [97](#)
CSQINP2
갱신 [15](#)
CursorPosition 필드 [129](#)

D

DFHMQ [5](#), [15](#)
DFHMQCOD 샘플 PLTPI 프로그램 [15](#), [20](#)
DISCARD MQCONN [29](#)
DPL 프로그램 [73](#)

E

Encoding 필드 [129](#)
ErrorOffset 필드 [129](#)
EXEC CICS LINK
COMMAREA 옵션 [19](#)
INPUTMSG 옵션 [29](#)

F

Facility 필드 [129](#)
FacilityKeepTime 필드 [129](#)
FacilityLike 필드 [129](#)
Flags 필드 [130](#)
Format 필드 [130](#)
Function 필드 [130](#)

G

GetWaitInterval 필드 [131](#)
GRPLIST 시스템 초기화 매개변수 [15](#)

I

ImqCICSBridgeHeader 클래스 [136](#)
InputItem 필드 [131](#)
INQUIRE MQCONN [29](#)
INQUIRE MQMONITOR [29](#)

IRC 및 CICS 어댑터 [15](#)

L

LinkType 필드 [131](#)

M

MQ 명령
DISPLAY CONN [116](#)
RESOLVE INDOUBT [116](#)
MQCB
샘플
개요 [99](#)
디자인 [100](#)
설정 [101](#)
MQCFUNC_* 값 [130](#)
MQCGWI_* 값 [131](#)
MQCIH
CICS 3270 브릿지로 필드 설정 [81](#)
CICS-MQ 브릿지로 필드 설정 [80](#)
MQCIH 구조 [125](#)
MQCIH_* 값 [134](#)
MQCIH_DEFAULT [126](#)
MQCLT_* 값 [131](#)
MQCODL_* 값 [132](#)
MQCONN
보안 [66](#)
MQCONN 명령 [29](#)
MQCRC_* 값 [133](#)
MQCTL
샘플
개요 [99](#)
디자인 [100](#)
설정 [101](#)
MQCUOWC_* 값 [135](#)
MQINI [16](#)
MQMD
CICS-MQ 브릿지로 필드 설정 [79](#)
MQMONITOR
보안 [66](#), [67](#)
MQMONITOR 명령 [29](#)
MsgId
CICS-MQ 브릿지로 관리 [83](#), [84](#)

N

NextTransactionId 필드 [131](#)
NID(네트워크 ID) [116](#)

O

OutputDataLength 필드 [132](#)

P

PLTPI(Program List Table Post Initialization)
연결 시작 [15](#), [20](#)

R

RDO(Resource Definition Online) [15](#)

Reason 필드 [132](#)
RECEIVE MAP 벡터
 CICS-MQ 브릿지로 해석 [92](#)
RemoteSysId 필드 [132](#)
RemoteTransId 필드 [132](#)
ReplyToFormat 필드 [132](#)
Reserved1 필드 [133](#)
Reserved2 필드 [133](#)
Reserved3 필드 [133](#)
Reserved4 필드 [133](#)
RESOLVE INDOUBT 명령, 잠긴 자원 잠금 해제 [116](#)
RESYNCMEMBER [113](#), [115](#)
ReturnCode 필드 [133](#)

S

SEND MAP 벡터
 CICS-MQ 브릿지로 해석 [88](#)
SET MQCONN [29](#)
SET MQMONITOR [29](#)
SIT(System Initialization Table)
 GRPLIST 매개변수 [15](#)
 PLTPI 매개변수 [15](#)
StartCode 필드 [133](#)
StrucId 필드 [134](#)
StrucLength 필드 [134](#)

T

TaskEndStatus 필드 [134](#)
TransactionId 필드 [135](#)
TYPETERM 정의, UCTRAN [32](#)

U

UCTRAN, TYPETERM 정의 [32](#)
UOWControl 필드 [135](#)

V

Version 필드 [136](#)

가

갱신
 CSQINP2 [15](#)
기본 맵핑 지원
 CICS-MQ 브릿지 사용 [87](#)

다

데이터 유형, 자세한 설명
 구조
 MQCIH [125](#)
동기점이 있는 트랜잭션
 CICS-MQ 브릿지 [77](#)

라

레거시 애플리케이션
 CICS-MQ 브릿지 사용 [73](#), [94](#)
로그 인쇄 유틸리티(CSQ1LOGP)

로그 인쇄 유틸리티(CSQ1LOGP) (계속)
 시작 RBA 찾기 [116](#)

마

메시지 구조
 CICS 3270 브릿지를 사용한 아웃바운드 [77](#)
 CICS 3270 브릿지를 사용한 인바운드 [77](#)
 CICS DPL 브릿지용 [73](#)
모니터 [10](#)
모니터링
 CICS 연결 활동 [42](#)

바

벡터
 CICS-MQ 브릿지로 사용 [76](#)
보안
 터미널 사용자 ID [51](#)
 CKSG 사용자 ID [51](#)
분산 환경의 트랜잭션
 CICS DPL 브릿지 [86](#)
브릿지 [10](#)
비동기 메시지 이용
 샘플
 개요 [99](#)
 디자인 [100](#)
 설정 [101](#)

사

사용자 엑시트 [103](#)

아

애플리케이션
 CICS-MQ 브릿지 사용 [73](#)
애플리케이션 데이터 구조
 CICS-MQ 브릿지 벡터에서 [92](#)
애플리케이션 프로그래밍
 CICS 3270 브릿지용 [76](#)
 CICS DPL 브릿지용 [75](#)
애플리케이션 프로그램
 CKQC DISPLAY [40](#)
에뮬레이션
 CICS-MQ 브릿지를 사용한 예 [95](#), [96](#)
엑시트 프로그램
 CICS-MQ 어댑터 [103](#)
연결
 시작
 CICS 애플리케이션 프로그램 [33](#), [34](#)
 EXEC CICS SET MQCONN [33](#)
 PLTPI 프로그램 [15](#), [20](#)
 중지
 CICS 명령행 [37](#)
 CICS 애플리케이션 프로그램 [38](#)
 EXEC CICS SET MQCONN [38](#)
연결 보안 [67](#)
온라인 자원 정의(RDO) [15](#)
이상 종료
 CICS 트랜잭션 연결을 끊는 중 [112](#)
인다우트 작업 단위
 CICS [114](#)

인다우트 작업 단위 (계속)
CICS에서 해결 [115](#)

자

자동 설치, CICS [6](#)
자원 유형
MQseries [111](#)
WMQ_INIT [111](#)
WMQCDISC [111](#)
작업
CICS, 수동으로 작업 단위 복구 [116](#)
작업 단위
인다우트 표시 [116](#)
CICS-MQ 브릿지 사용 [84](#)
CICS, 수동으로 복구 [116](#)
CICS에서 해결 [115](#)
작업 단위(UOW)
CICS DPL 브릿지로 관리 [83](#)
작업 토큰 [116](#)
작업, 단위 [116](#)
재동기화 [113](#), [115](#)

카

콜백 루틴
샘플
개요 [99](#)
디자인 [100](#)
설정 [101](#)
큐 관리자 중지 [113](#)

타

트랜잭션
CICS-MQ 브릿지 사용 [75](#)
트랜잭션 추적 [10](#)
트랜지언트 데이터 큐(TDQ), CKQQ [29](#)
트리거
작동하지 않는 경우 [117](#)

파

표시
CICS의 작업 단위 [116](#)
프로그램 목록 테이블(PLT) [15](#), [20](#)
프로그램 자동 설치, CICS [6](#)

하

해결
작업 단위 [116](#)

