



Adapter for WebSphere MQ Integrator Broker 사용자 안내서

버전 2.5.x



Adapter for WebSphere MQ Integrator Broker 사용자 안내서

버전 2.5.x

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 91 페이지의 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

이 문서 개정판은 새 개정판에서 별도로 명시하지 않는 한 IBM WebSphere Business Integration Adapter for WebSphere MQ Integrator, 버전 2.4.x 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견은 ibmkspoe@kr.ibm.com으로 전자 우편을 보내십시오. 사용자의 의견을 기다립니다.

IBM에 정보를 보내는 경우, IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

© Copyright International Business Machines Corporation 2000, 2003. All rights reserved.

목차

이 책의 정보	v
이 책의 사용자	v
이 책을 사용하기 전에	v
관련 서적	v
인쇄 규칙	vi
이 릴리스의 새로운 기능	vii
릴리스 2.5.x의 새로운 기능	vii
릴리스 2.4.x의 새로운 기능	vii
릴리스 2.3.x의 새로운 기능	vii
릴리스 2.2.x의 새로운 기능	viii
릴리스 2.1.x의 새로운 기능	viii
릴리스 1.5.x의 새로운 기능	viii
릴리스 1.4.x의 새로운 기능	ix
릴리스 1.3.x의 새로운 기능	ix
제 1 장 개요	1
Adapter for WebSphere MQ Integrator Broker	2
커넥터 구조	4
응용프로그램 커넥터 통신 메소드	5
이벤트 처리	7
보증된 이벤트 전달	9
Business Object 요청	9
Verb 처리	10
로케일 종속 데이터 처리	15
제 2 장 커넥터 설치 및 구성	17
설치 작업 개요	17
커넥터 및 관련 파일 설치	17
설치된 파일 구조	17
메시지 플로우 수정	19
커넥터 구성	21
다중 커넥터 인스턴스 작성	26
대기열 URI(Uniform Resource Identifiers)	27
Meta Object 속성 구성	29
시작 파일 구성	41
시작	42
제 3 장 Business Object 작성 또는 수정	45
커넥터 Business Object 구조	45
오류 처리	49
추적	50

제 4 장 문제점 해결	53
시작 시 문제점	53
이벤트 처리	53
부록 A. 커넥터의 표준 구성 등록 정보	55
신규 및 삭제된 등록 정보	55
표준 커넥터 등록 정보 구성	55
표준 등록 정보 요약	57
표준 구성 등록 정보	60
부록 B. Connector Configurator	73
Connector Configurator 개요	73
Connector Configurator 시작	74
System Manager에서 Configurator 실행	75
커넥터 특정 등록 정보 템플릿 작성	75
새 구성 파일 작성	78
기존 파일 사용	79
구성 파일 완료	80
구성 파일 등록 정보 설정	81
구성 파일 저장	88
구성 파일 변경	88
구성 완료	89
국제화된 환경에서 Connector Configurator 사용	89
주의사항	91
프로그래밍 인터페이스 정보	92
상표 및 서비스표	93

이 책의 정보

IBM^(R) WebSphere^(R) Business Integration Adapter 포트폴리오는 주요 e-business 기술, 엔터프라이즈 응용프로그램, 레거시 및 메인프레임 시스템의 통합 연결성을 제공합니다. 제품 세트에는 비즈니스 프로세스 통합용 구성요소를 사용자 정의, 작성 및 관리하는 도구와 템플릿이 포함됩니다.

이 책에서는 Adapter for WebSphere MQ Integrator Broker의 설치, 구성 및 Business Object 개발에 대해 설명합니다.

이 책의 사용자

이 책은 고객의 지역에서 WebSphere Business Integration 제품을 지원하고 관리하는 컨설턴트, 개발자 및 시스템 관리자를 위한 것입니다.

이 책을 사용하기 전에

이 책의 사용자는 WebSphere Business Integration System, Business Object와 협업 개발 및 WebSphere MQ Integrator(MQ Integrator Broker) 응용프로그램에 익숙해야 합니다.

관련 서적

이 제품과 함께 제공되는 문서 세트에는 모든 WebSphere Business Integration Adapter 설치에 공통되는 사양 및 구성요소에 대한 설명과 특정 구성요소에 관한 참조 자료가 수록되어 있습니다.

이 책에는 다른 두 권의 책인 *Windows용 시스템 설치 안내서* 또는 *UNIX용 시스템 설치 안내서* 및 *System Implementation Guide for WebSphere InterChange Server*에 대한 여러 참조 내용이 있습니다. 이 책을 인쇄하면 위의 두 책도 인쇄할 필요를 느낄 수 있습니다.

다음 사이트에서 관련 서적을 설치할 수 있습니다.

- 일반 어댑터 정보, WebSphere 메시지 브로커(WebSphere MQ Integrator, WebSphere MQ Integrator Broker, WebSphere Business Integration Message Broker)와 함께 어댑터를 사용하는 경우 및 WebSphere Application Server와 함께 어댑터를 사용하는 경우에는 다음 웹 사이트를 참조하십시오.

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

- InterChange Server와 함께 어댑터를 사용하는 경우에는 다음 웹 사이트를 참조하십시오.

<http://www.ibm.com/websphere/integration/wicsserver/infocenter>

<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>

- 메시지 브로커(WebSphere MQ Integrator Broker, WebSphere MQ Integrator 및 WebSphere Business Integration Message Broker)에 대한 자세한 정보는 다음 웹 사이트를 참조하십시오.

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>

- WebSphere Application Server에 대한 자세한 정보는 다음 웹 사이트를 참조하십시오.

<http://www.ibm.com/software/webservers/appserv/library.html>

이 사이트에서는 문서를 다운로드 및 설치하고 보는 데 필요한 간단한 지시사항을 제공합니다.

인쇄 규칙

이 책에서는 다음 규칙을 사용합니다.

courier 글꼴	명령어, 파일 이름, 사용자가 입력한 정보, 시스템이 화면에 인쇄한 정보와 같은 리터럴 값을 표시합니다.
굵은체	처음 나타나는 새 항목을 표시합니다.
가울임체	변수 이름이나 상호 참조를 표시합니다.
파란색 윤곽선	온라인 매뉴얼을 볼 때만 볼 수 있는 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 건너 뛰려면 윤곽선 내부를 누르십시오.
{ }	구문 행에서 중괄호 안에는 하나만 선택해야 하는 옵션 세트가 표시됩니다.
[]	구문 행에서 대괄호 안에는 선택적 매개변수가 표시됩니다.
...	구문 행에서 말줄임표(...)는 이전 매개변수의 반복을 표시합니다. 예를 들어, option[,...]은 쉼표로 구분되는 여러 옵션을 입력할 수 있음을 의미합니다.
< >	이름 지정 규칙에서 꺾쇠괄호는 <server_name><connector_name> tmp.log에서와 같이 이름의 개별 요소를 묶어 이들을 서로 구별합니다.
/, \	이 책에서 백슬래시(\)는 디렉토리 경로의 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시 대신 슬래시(/)를 사용합니다. 모든 제품 경로 이름은 제품이 사용자 시스템에 설치된 디렉토리과 관련됩니다.
%text% 및 \$text	퍼센트(%) 부호 내의 텍스트는 Windows text 시스템 변수나 사용자 변수의 값을 표시합니다. UNIX 환경에서 동등한 표시법은 \$text이며, text UNIX 환경 변수의 값을 표시합니다.
ProductDir	제품이 설치된 디렉토리를 표시합니다.

이 릴리스의 새로운 기능

릴리스 2.5.x의 새로운 기능

커넥터는 이제 다음 플랫폼에서 실행됩니다.

- Microsoft Windows 2000
- Solaris 7, 8 또는 AIX 5.1, 5.2 또는 HP UX 11.i

2.5.0 버전으로 시작할 때 Adapter for WebSphere MQ는 더 이상 Microsoft Windows NT에서 지원되지 않습니다.

이 안내서에서 어댑터 설치 정보가 이동되었습니다. 해당 정보의 새 위치는 2장을 참조하십시오.

릴리스 2.4.x의 새로운 기능

어댑터는 이제 통합 브로커로 WebSphere Application Server를 사용할 수 있습니다. 자세한 정보는 2 페이지의 『브로커 호환성』 항목을 참조하십시오.

커넥터는 이제 다음 플랫폼에서 실행됩니다.

- Microsoft Windows NT 4.0 서비스 팩 6A 또는 Windows 2000
- Solaris 7, 8 또는 AIX 5.1, 5.2 또는 HP UX 11.i

릴리스 2.3.x의 새로운 기능

2003년 3월에 갱신되었습니다. "CrossWorlds"라는 이름은 더 이상 전체 시스템을 설명하거나 구성요소 또는 도구 이름을 수정하는 데 사용되지 않습니다. 다른 경우는 이전과 거의 동일합니다. 예를 들어, "CrossWorlds System Manager"는 이제 "System Manager"이고, "CrossWorlds InterChange Server"는 "WebSphere InterChange Server"입니다.

이제 Data Handler와 입력 대기열을 연관시킬 수 있습니다. 자세한 정보는 30 페이지의 『InputQueues에 Data Handler 맵핑』을 참조하십시오.

이제 어댑터는 WebSphere MQ Event Broker, WebSphere MQ Integrator Broker 및 WebSphere InterChange Server 통합 브로커를 지원합니다.

보증된 이벤트 전달 기능이 강화되었습니다. 자세한 정보는 9 페이지의 『보증된 이벤트 전달』을 참조하십시오.

릴리스 2.2.x의 새로운 기능

진행 중 대기열이 더 이상 필요 없으며 사용 불가능할 수도 있습니다. 자세한 정보는 25 페이지의 『InProgressQueue』를 참조하십시오.

커넥터는 MQSeries 5.1, 5.2와 5.3을 통해 응용프로그램과의 상호 운영성을 지원합니다. 자세한 정보는 3 페이지의 『어댑터 종속성』을 참조하십시오.

커넥터가 이제 Business Object 처리를 위해 UseDefaults 등록 정보를 가집니다. 자세한 정보는 25 페이지의 『UseDefaults』를 참조하십시오.

커넥터는 이제 Data Handler가 명시적으로 하나의 Verb를 Business Object에 지정하지 않으면 기본 Verb를 적용할 수 있습니다. 자세한 정보는 23 페이지의 『DefaultVerb』를 참조하십시오.

ReplyToQueue는 이제 ReplyToQueue 커넥터 등록 정보가 아니라 동적 하위 Meta Object를 통해 지정됩니다. 자세한 정보는 37 페이지의 『JMS 머리글, MQ Integrator Broker 메시지 등록 정보 및 동적 하위 Meta Object 속성』을 참조하십시오.

어댑터는 메시지 선택자를 사용하여 주어진 요청에 대해 응답 메시지를 식별하는 방법을 식별하고 필터하고, 아니면 제어할 수 있습니다. 이 JMS 성능은 동기 요청 처리에만 적용됩니다. 자세한 정보는 10 페이지의 『동기 전달』을 참조하십시오.

릴리스 2.1.x의 새로운 기능

커넥터가 국제화되었습니다. 자세한 정보는 15 페이지의 『로케일 종속 데이터 처리』 및 55 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

이 안내서에서는 ICS에서 이 어댑터를 사용하는 방법에 대한 정보를 제공합니다.

주: 보증된 이벤트 전달 기능을 사용하려면 ICS 릴리스 4.1.1.2를 설치해야 합니다.

릴리스 1.5.x의 새로운 기능

IBM WebSphere Business Integration Adapter for MQ Integrator는 Connector for MQ Integrator를 포함합니다. 이 어댑터는 ICS(InterChange Server) 통합 브로커로 작동합니다. 통합 브로커는 이기종의 응용프로그램 세트 통합을 수행하는 응용프로그램으로서, 데이터 라우팅을 포함하는 서비스를 제공합니다. 어댑터는 다음을 포함합니다.

- MQ Integrator에 고유한 응용프로그램 구성요소
- 샘플 Business Object
- 다음으로 구성된 IBM WebSphere Adapter Framework
 - 커넥터 프레임워크

- 개발 도구(Business Object Designer 및 IBM CrossWorlds System Manager 포함)
- API(CDK 포함)

이 매뉴얼에서는 ICS에서 이 어댑터를 사용하는 방법에 대한 정보를 제공합니다.

중요: 커넥터가 국제화되지 않았으므로, ISO Latin-1 데이터만이 처리되도록 보증할 수 없는 경우에는 ICS 버전 4.1.1에 대해 실행하지 마십시오.

커넥터가 IBM CrossWorlds 4.1.x 시스템에서 사용 가능해졌습니다.

릴리스 1.4.x의 새로운 기능

이 책의 1.4.x 릴리스에서 결함을 수정하고 IBM CrossWorlds 하부 구조 릴리스 버전 4.0.0과의 호환성을 제공하기 위해 약간의 변경이 수행되었습니다.

릴리스 1.3.x의 새로운 기능

이 책의 1.3.x 릴리스에는 다음과 같이 새 기능과 제품 향상을 위한 정보가 있습니다.

- Create, Update 및 Delete 조작을 확인하기 위한 동기 요청 및 응답 처리를 위한 지원.
- Retrieve, Retrieve By Content 및 Exist 조작의 지원.
- 포함 오류는 물론 정상적으로 처리되고 미등록된 메시지를 포함하는 메시지의 전체 아카이브.
- 둘 이상의 Business Object로 동일한 메시지 형식을 지정하기 위한 확장 성능.
- 이제 커넥터는 완전한 URI 없이도 로컬 대기열을 식별할 수 있습니다. 이에 따라, "URI" 접미부는 더 이상 InputQueueURI, InProgressQueueURI, UnsubscribedURI 및 ErrorQueueURI와 같은 커넥터 등록 정보의 일부가 아닙니다.
- 커넥터 Meta Object의 추가 기본 변환 등록 정보. 이에 따라, 커넥터 등록 정보 DefaultOutputQueueURI가 제거되었습니다.

제 1 장 개요

- 4 페이지의 『커넥터 구조』
- 5 페이지의 『응용프로그램 커넥터 통신 메소드』
- 7 페이지의 『이벤트 처리』
- 9 페이지의 『보증된 이벤트 전달』
- 9 페이지의 『Business Object 요청』
- 10 페이지의 『Verb 처리』
- 15 페이지의 『로케일 종속 데이터 처리』

Connector for MQ Integrator Broker는 WebSphere Business Integration Adapter for WebSphere MQ Integrator Broker의 런타임 구성요소입니다. 커넥터는 WebSphere MQ Integration Broker가 WebSphere MQ 메시지의 형식으로 데이터를 전송 및 수신하는 응용프로그램으로 Business Object를 교환할 수 있도록 합니다.

커넥터는 IBM의 MQ Integrator Broker 버전 2.2 제품 및 IBM WebSphere Business Integration System Data Handler를 사용하여 메시지를 Business Object로 변환하고 Business Object를 메시지로 변환합니다. 따라서 커넥터는 WebSphere Business Integration System과 IBM WebSphere MQ Integrator System 간의 브릿지를 제공합니다. 이 장에서는 커넥터 구성요소 및 관련된 비즈니스 통합 시스템 구조에 대해 설명합니다.

커넥터는 응용프로그램 특정 구성요소 및 커넥터 프레임워크로 구성됩니다. 응용프로그램 특정 구성요소에는 특정 응용프로그램으로 조정된 코드가 있습니다. 코드가 모든 커넥터에 공통인 커넥터 프레임워크는 통합 브로커와 특정 응용프로그램 구성요소 간의 매개자로 작용합니다. 커넥터 프레임워크는 통합 브로커와 특정 응용프로그램 구성요소 간에 다음과 같은 서비스를 제공합니다.

- Business Object 수신 및 전송
- 시작 및 관리 메시지의 교환 관리

이 책에는 특정 응용프로그램 구성요소 및 커넥터 프레임워크에 대한 정보가 있습니다. 이 구성요소를 모두 커넥터로 참조합니다.

커넥터에 대한 통합 브로커의 관계에 대한 자세한 정보는 *IBM WebSphere InterChange Server System Administration Guide*를 참조하십시오.

주: 모든 WebSphere Business Integration Adapter는 통합 브로커로 작동합니다. Connector for WebSphere MQ Integrator Broker는 다음 항목으로 작동합니다.

- *IBM WebSphere InterChange Server*에 대한 기술적 소개에서 설명한 InterChange Server 통합 브로커
- *Implementing Adapters with WebSphere Application Server*에 설명되어 있는 WAS(WebSphere Application Server) 통합 브로커

주: WebSphere MQ Integrator Broker, 버전 2.2는 WebSphere MQ 대기열 간의 메시지를 변환하고 라우트하는 메시지 브로커 제품입니다. 이 기술은 응용프로그램이 잠재적인 원격 대기열로 메시지를 전달하고 메시지를 수신하여 비동기적으로 통신할 수 있게 합니다. MQ Integrator Broker에서의 주요 변경사항은 사용자 정의 논리에 근거하여 메시지를 형식화, 저장 및 라우트하는 기능을 추가하는 메시지 플로우의 추가입니다.

Adapter for WebSphere MQ Integrator Broker

어댑터를 설치, 구성 및 사용하기 전에 해당 환경 요구사항을 이해해야 합니다.

- 『브로커 호환성』
- 3 페이지의 『어댑터 플랫폼』
- 3 페이지의 『어댑터 종속성』
- 4 페이지의 『로케일 종속 데이터』

브로커 호환성

어댑터가 사용하는 어댑터 프레임워크는 어댑터가 통신하는 통합 브로커(또는 브로커)의 버전과 호환 가능해야 합니다. WebSphere MQ Integrator Broker의 어댑터 2.5 버전은 다음과 같은 어댑터 프레임워크 및 통합 브로커에서 지원됩니다.

- 어댑터 프레임워크: WebSphere Business Integration Adapter Framework, 버전:
 - 2.1
 - 2.2
 - 2.3.x
 - 2.4
- 통합 브로커:
 - WebSphere InterChange Server, 버전:
 - 4.11
 - 4.2
 - 4.2.1
 - 4.2.x
 - WebSphere Studio Application Developer Integration Edition, 버전 5.0.1이 있는 WebSphere Application Server Enterprise, 버전 5.0.2

예외사항은 릴리스 정보를 참조하십시오.

주: 통합 브로커 설치 지시사항 및 전제조건은 다음 문서를 참조하십시오.

WebSphere ICS(InterChange Server)는 *UNIX용 시스템 설치 안내서* 또는 *Windows용 시스템 설치 안내서*를 참조하십시오.

메시지 브로커(WebSphere MQ Integrator Broker, WebSphere MQ Integrator 및 WebSphere Business Integration Message Broker)는 *Implementing Adapters with WebSphere Message Brokers* 및 메시지 브로커의 설치 문서를 참조하십시오. 이 중 일부는 웹 사이트 <http://www.ibm.com/software/integration/mqfamily/library/manualsa/>에서 찾을 수 있습니다. WebSphere Application Server는 *Implementing Adapters with WebSphere Application Server* 및 다음 웹 사이트에 있는 문서를 참조하십시오. <http://www.ibm.com/software/webservers/appserv/library.html>

어댑터 플랫폼

어댑터는 다음 플랫폼에서 지원됩니다.

- Windows 2000
- AIX 5.1, 5.2
- Solaris 7, 8
- HP-UX 11i

어댑터 종속성

어댑터에는 다음 소프트웨어 전제조건 및 기타 종속성이 있습니다.

- 커넥터는 WebSphere MQ 또는 WebSphere MQ 5.1, 5.2를 통해 응용프로그램과의 상호 운영성을 지원합니다.¹ 따라서 이 소프트웨어 릴리스 중 하나를 설치해야 합니다.

주: 어댑터는 WebSphere MQ 5.3에서 SSL(Secure Socket Layer)을 지원하지 않습니다. 어댑터 프레임워크 통합 브로커 통신에 적절한 WebSphere MQ 소프트웨어 버전은 해당 플랫폼(Windows/UNIX)에 대한 설치 안내서를 참조하십시오.

- 또한 IBM WebSphere MQ Java 클라이언트 라이브러리가 있어야 합니다.

1. 환경이 문자 세트 변환을 위해 가져올 때 변환(convert-on-the-get) 방법론을 구현할 경우, IBM에서 최신 MA88(JMS 클래스)을 다운로드해야 합니다. 패치 레벨은 최소한 5.2.2(WebSphere MQ 버전 5.2의 경우) 이상이어야 합니다. 이렇게 하면 지원되지 않는 인코딩 오류를 막을 수 있습니다.

로케일 종속 데이터

커넥터가 2바이트 문자 세트를 지원하고 지정된 언어로 메시지 텍스트를 전달하도록 국제화되었습니다. 커넥터가 한 문자 코드를 사용하는 위치에서 다른 문자 세트를 사용하는 위치로 데이터를 전송할 때 커넥터는 문자 변환을 수행하여 데이터 의미를 유지합니다.

JVM(Java Virtual Machine) 내의 Java 런타임 환경은 Unicode 문자 코드 세트에 데이터를 표시합니다. 유니코드에는 알려진 문자 코드 세트(단일 바이트와 다중 바이트 모두)로 된 문자의 인코딩이 들어 있습니다. WebSphere Business Integration System의 대부분의 구성요소는 Java로 작성됩니다. 따라서 데이터가 대부분의 통합 구성요소 사이에서 전송될 때 문자 변환은 필요하지 않습니다.

해당 국가 또는 지역에서 해당 언어로 오류 및 정보 메시지를 로그하려면 사용자 환경에 대한 Locale 표준 구성 등록 정보를 구성하십시오. 구성 등록 정보에 대한 자세한 정보는 55 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

커넥터 구조

그림 1에서는 커넥터 구성요소를 보여주며 WebSphere Business Integration System 내에서의 MQSI 메시지 브로커에 대한 커넥터 구성요소의 관계를 표시합니다. 커넥터는 메타 데이터 구동입니다. 메시지 라우팅 및 형식 변환은 이벤트 폴링 기술로 시작됩니다. 커넥터는 엔터프라이즈 메시징 시스템에 액세스하기 위해 API인 JMS(Java™ Message Service)을 MQ에서 구현한 것을 사용합니다.

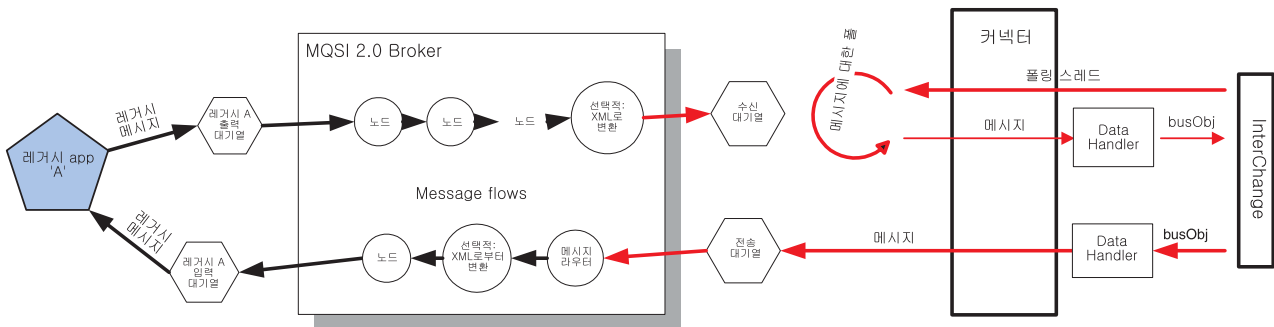


그림 1. 커넥터의 구조

커넥터는 데이터에 대한 변경사항이 발생할 때 WebSphere MQ 메시지를 발행하거나 수신하는 응용프로그램과 협업이 비동기적으로 Business Object를 교환하게 합니다.

커넥터는 대기열에서 WebSphere MQ 메시지를 검색하며, Data Handler를 호출하여 메시지를 해당 Business Object로 변환한 후 협업으로 전달합니다. 반대로, 커넥터는 협업에서 Business Object를 수신하고, 동일한 Data Handler를 사용하여 WebSphere MQ 메시지로 변환한 후, 메시지를 WebSphere MQ 대기열로 전달합니다.

메시지 처리 시 커넥터를 구성하여 Data Handler를 사용할 수 있습니다. 그러나 MQSI 메시지 브로커는 XML 형식으로 구문 분석 가능한 메시지를 변환할 수 있으므로, XML에서 모든 메시지를 전달할 수 있게 커넥터를 구성하는 것이 가장 바람직합니다. 이것은 처리를 위해 XML Data Handler를 구현하는 것을 의미합니다. 자세한 정보는 *Data Handler* 안내서를 참조하십시오.

메시지 처리에서 사용되는 Business Object 및 Verb의 유형은 WebSphere MQ 메시지 머리글에 포함된 FORMAT 필드에 근거합니다. 커넥터는 Meta Object 항목을 사용하여 Business Object 이름 및 Verb를 판별합니다. 사용자는 Business Object 이름 및 Verb를 저장하도록 Meta Object를 구성하여 WebSphere MQ 메시지 머리글 FORMAT 필드 텍스트와 연관시킵니다.

메시지가 입력 대기열에서 검색되면, 커넥터는 FORMAT 텍스트 필드와 연관된 Business Object 이름을 찾습니다. 그런 다음 Business Object 이름과 함께 메시지가 Data Handler로 전달됩니다. Business Object가 정상적으로 메시지 내용으로 채워지면, 커넥터는 Business Object가 등록되어 있는지 여부를 참조하도록 확인한 후 `getAppEvents()` 메소드를 사용하여 InterChange Server로 전달합니다.

응용프로그램 커넥터 통신 메소드

커넥터는 JMS(Java Message Service)의 IBM WebSphere MQ 구현을 이용합니다. JMS는 엔터프라이즈 메시징 시스템에 액세스하기 위한 개방 표준 API입니다. 비즈니스 응용프로그램이 비동기적으로 비즈니스 데이터 및 이벤트를 전송하고 수신하도록 디자인되었습니다.

메시지 요청

그림 2에서는 메시지 요청 통신에 대해 설명합니다. `doVerbFor()` 메소드가 협업에서 Business Object를 수신하면, 커넥터는 Business Object를 Data Handler로 전달합니다. Data Handler는 Business Object를 MQSI 적합 메시지로 변환하여 대기열로 발행합니다. 여기에서 JMS 레이어는 대기열 세션을 열고 메시지를 라우트하도록 적절한 호출을 작성합니다.

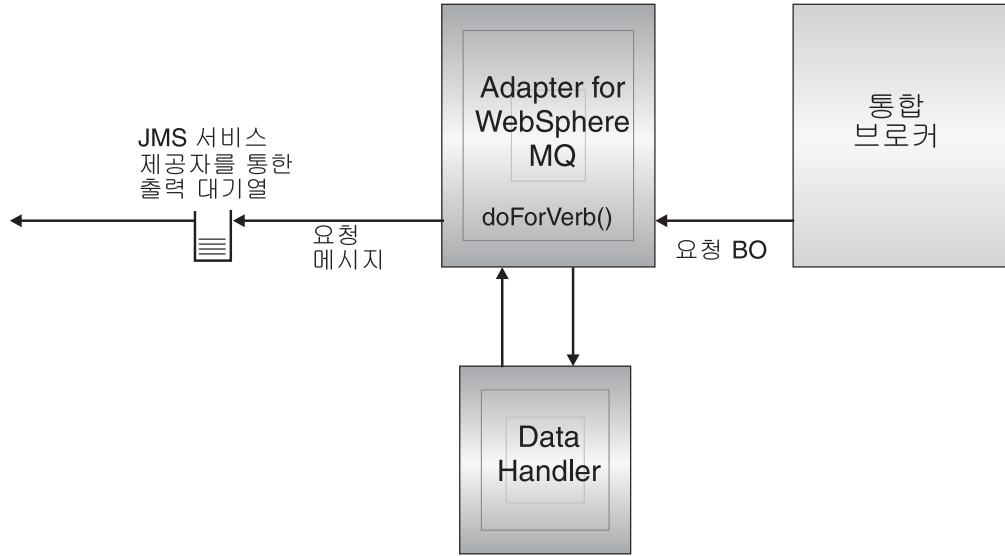


그림 2. 응용프로그램 커넥터 통신 메소드: 메시지 요청

메시지 리턴

그림 3에서는 메시지 리턴 방향을 보여줍니다. pollForEvents() 메소드는 입력 대기열에서 적용 가능한 다음 메시지를 검색합니다. 처리가 완료될 때까지 메시지가 남아 있는 진행 중인 대기열에 메시지를 넣습니다. 커넥터는 MQ Integrator Broker Meta Object를 사용하여, 커넥터는 메시지 유형이 지원되는지 여부를 우선 판별합니다. 그렇게 하면, 커넥터는 Data Handler로 메시지를 전달하며, Data Handler는 Business Object로 메시지를 변환합니다. 설정된 Verb는 메시지 유형에 설정된 변환 등록 정보를 반영합니다. 그런 다음 커넥터는 Business Object가 협업으로 등록되었는지 여부를 판별합니다. 그렇게 하면, getAppEvents() 메소드는 Business Object를 InterChange Server로 전달하고 메시지는 진행 중인 대기열에서 제거됩니다.

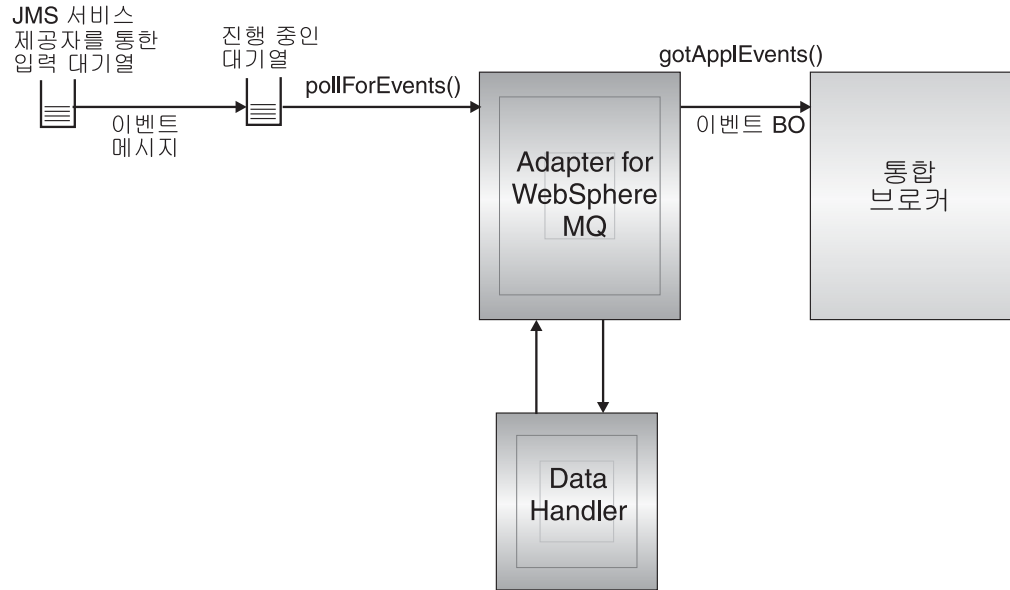


그림 3. 응용프로그램 커넥터 통신 메소드: 메시지 전달

이벤트 처리

이벤트 공고의 경우, 커넥터는 데이터베이스 트리거 이외의 응용프로그램이 대기열에 쓰는 이벤트를 발견합니다. 응용프로그램 또는 기타 MQ를 사용 가능한 소프트웨어가 WebSphere MQ 메시지를 생성하여 MQ 메시지 대기열에 이를 저장할 때 이벤트가 발생합니다.

검색

커넥터는 pollForEvents() 메소드를 사용하여 정기적으로 메시지에 대한 MQ 대기열을 폴링합니다. 커넥터가 메시지를 찾으면, MQ 대기열에서 메시지를 검색하고 조사하여 해당 형식을 판별합니다. 형식이 커넥터 Meta Object에 정의된 경우, 커넥터는 Data Handler를 사용하여 Verb가 있는 적절한 Business Object를 생성합니다. 이벤트 장애 시나리오에 대해서는 49 페이지의 『오류 처리』를 참조하십시오.

커넥터는 입력 대기열에 대한 트랜잭션 세션을 처음에 열어 메시지를 처리합니다. 이 트랜잭션 접근 방법은 커넥터가 정상적으로 Business Object를 제출했지만 대기열에서 트랜잭션을 확약하는 데 실패했으므로 Business Object가 협업으로 두 번 전달될 수 있는 기회를 허용합니다. 이러한 문제점을 피하기 위해, 커넥터는 모든 메시지를 진행 중인 대기열로 이동시킵니다. 여기에서 메시지는 처리가 완료될 때까지 보류됩니다. 커넥터가 처리 중 예기치 않게 종료한 경우, 메시지는 원래 입력 대기열로 복귀되는 대신 진행 중인 대기열에 남습니다.

주: JMS 서비스 제공자가 있는 트랜잭션 세션은 이벤트가 대기열에서 제거되기 전에 대기열에 있는 모든 요청 조치가 수행되고 예약되도록 요구합니다. 이에 따라, 커넥터가 대기열에서 메시지를 검색할 때, 다음 세 가지 사항이 발생할 때까지 검색을 예약하지 않습니다. 1) 메시지가 Business Object로 변환됩니다. 2) gotAppEvents() 메소드에 의해 Business Object가 InterChange Server로 전달됩니다. 3) 리턴 값이 수신됩니다.

복구

초기화 시, 커넥터는 커넥터 종료 때문에 완전히 처리되지 않은 메시지에 대한 진행 중인 대기열을 검사합니다. 커넥터 구성 등록 정보 InDoubtEvents는 시작 시 실패, 재처리, 무시 또는 오류 로그와 같은 메시지의 복구를 처리하는 네 가지 옵션 중 하나를 지정하게 합니다.

시작 시 실패

시작 시 실패 옵션 사용 시, 커넥터가 초기화 중에 진행 중인 대기열에서 메시지를 찾은 경우 오류를 로그하고 즉시 종료합니다. 메시지를 조사하고 이 메시지를 모두 삭제하거나 다른 대기열로 이동시키는 것 중 하나의 적절한 조치를 취하는 것은 사용자 또는 시스템 관리자의 책임입니다.

재처리

재처리 옵션 사용 시, 커넥터가 초기화 중에 진행 중인 대기열에서 메시지를 찾은 경우 차후 폴링 중 처음에 이들 메시지를 처리합니다. 진행 중인 대기열의 모든 메시지가 처리되면, 커넥터가 입력 대기열에서 메시지 처리를 시작합니다.

무시

무시 옵션 사용 시, 커넥터가 초기화 중에 진행 중인 대기열에서 메시지를 찾은 경우 커넥터는 메시지를 무시하지만 종료하지 않습니다.

오류 로그

오류 로그 옵션 사용 시, 커넥터가 초기화 중에 진행 중인 대기열에서 메시지를 찾은 경우 커넥터는 오류를 로그하지만 종료하지 않습니다.

아카이브

커넥터 등록 정보 ArchiveQueue가 지정되고 유효한 대기열을 식별하는 경우, 커넥터는 정상적으로 처리된 모든 메시지의 사본을 아카이브 대기열에 놓습니다. ArchiveQueue가 비정의된 경우, 메시지를 처리 후 버립니다. 등록되지 않은 아카이브 또는 오류가 있는 메시지에 대한 자세한 정보는 49 페이지의 『오류 처리』를 참조하십시오.

주: JMS 규칙에 따라 검색된 메시지는 다른 대기열로 즉시 발행될 수 없습니다. 메시지의 아카이브와 재전달을 사용 가능하게 하기 위해 커넥터는 먼저 원본의 본문 및 머리글(적용 가능하면)을 중복하는 두 번째 메시지를 생성합니다. JMS 서비스 제

공자와의 충돌을 피하기 위해, JMS 필수 필드만이 중복됩니다. 이에 따라, 형식 필드는 아카이브되거나 다시 전달된 메시지에 대해 복사된 유일한 추가 메시지 등록 정보입니다.

보증된 이벤트 전달

보증된 이벤트 전달 기능을 사용하면 커넥터 프레임워크에서 이벤트가 유실되지 않아 커넥터의 이벤트 저장소, JMS 이벤트 저장소 및 목적지의 JMS 대기열 간에 이벤트가 두 번 전송되지 않도록 할 수 있습니다. JMS를 사용하려면 커넥터의 `DeliveryTransport` 표준 등록 정보를 JMS로 구성해야 합니다. 따라서 구성된 커넥터는 JMS 전송을 사용하며, 커넥터와 통합 브로커 간의 모든 후속 통신은 이 전송을 통해 발생합니다. JMS 전송으로 메시지는 결국 목적지에 전달됩니다. 이 역할은, 트랜잭션 대기열 세션이 시작되면 예약이 발행될 때까지 메시지가 캐시화하고 장애가 발생하거나 롤백이 발행되면 메시지를 버리는 것입니다.

주: 보증된 이벤트 전달 기능을 사용하지 않으면 커넥터가 이벤트를 공개한 시간(커넥터가 `pollForEvents()` 메소드 내에서 `getApplEvent()` 메소드를 호출한 시간)과 커넥터가 이벤트 레코드를 삭제하여(또는 “이벤트 게시” 상태로 갱신하여) 이벤트 저장소를 갱신한 시간 사이에 작은 창에서 장애가 발생할 수 있습니다. 이 창에서 장애가 발생하면 이벤트는 전송되지만 이벤트 레코드는 이벤트 저장소에 “진행 중”인 상태로 남아 있습니다. 커넥터가 다시 시작될 때 커넥터는 이벤트 저장소에 남아 있는 이 이벤트 레코드를 발견하여 전송합니다. 따라서 이벤트는 두 번 전송됩니다.

JMS 이벤트 저장소의 사용 여부에 관계없이 JMS 사용 가능 커넥터에 보증된 이벤트 전달 기능을 구성할 수 있습니다. 보증된 이벤트 전달을 위한 커넥터를 구성하려면 *Connector Development Guide for Java*의 지시사항을 참조하십시오.

커넥터 프레임워크가 ICS 통합 브로커로 Business Object를 전달할 수 없는 경우, 오브젝트는 `FaultQueue(UnsubscribedQueue 및 ErrorQueue가 아닌)`에 놓이고 상태 표시기 및 문제점에 대한 설명을 기록합니다. `FaultQueue` 메시지는 MQRFH2 형식으로 작성됩니다.

Business Object 요청

InterChange Server가 Business Object를 `doVerbFor()` 메소드로 전송할 때 Business Object 요청이 처리됩니다. 커넥터는 구성된 Data Handler를 사용하여 Business Object를 WebSphere MQ 메시지로 변환하여 발행합니다. Data Handler의 요구사항을 제외하고 처리된 Business Object의 유형에 대한 요구사항은 없습니다.

Verb 처리

커넥터는 각 Business Object의 Verb에 근거하여 협업에 의해 전달된 Business Object를 처리합니다. 커넥터는 Business Object 핸들러 및 doForVerb() 메소드를 사용하여 커넥터가 지원하는 Business Object를 처리합니다. 커넥터는 다음과 같은 Business Object Verb를 지원합니다.

- Create
- Update
- Delete
- Retrieve
- Exists
- Retrieve by Content

주: Create, Update 및 Delete Verb가 있는 Business Object는 비동기적으로 또는 동기적으로 발행될 수 있습니다. 기본 모드는 비동기입니다. 커넥터는 Retrieve, Exists 또는 Retrieve By Content Verb를 사용하는 Business Object의 비동기 전달을 지원하지 않으므로, 이에 따라 Retrieve, Exists 또는 Retrieve By Content Verb의 경우 기본 모드는 동기입니다.

Create, Update 및 Delete

Create, Update 및 Delete Verb를 사용하는 Business Object 처리는 오브젝트가 비동기적 또는 동기적으로 발행되는지의 여부에 따라 다릅니다.

비동기 전달

이것은 Create, Update 및 Delete Verb가 있는 Business Object의 기본 전달 모드입니다. 메시지는 Data Handler를 사용하여 Business Object에서 작성된 후 출력 대기열로 기록됩니다. 메시지가 전달된 경우, 커넥터는 BON_SUCCESS를 리턴하며, 그렇지 않으면 BON_FAIL을 리턴합니다.

주: 커넥터에는 메시지를 수신했는지 여부나 조치를 취했는지 여부를 검증하는 방식이 없습니다.

동기 전달

replyToQueue가 커넥터 등록 정보에 정의되었으며 responseTimeout이 Business Object의 변환 등록 정보에 있는 경우, 커넥터는 동기 모드에서 요청을 발행합니다. 그런 다음 커넥터는 수신 응용프로그램에 의해 적절한 조치가 취해졌는지 검증하기 위해 응답을 기다립니다.

MQ Integrator Broker의 경우, 커넥터는 처음으로 표 1에 표시된 대로 머리글로 메시지를 발행합니다.

표 1. 요청 메시지 설명자 머리글(MQMD)

필드	설명	값
Format	형식 이름	변환 등록 정보에 정의되고 IBM 요구사항에 맞도록 8자로 잘린 출력 형식(예: MQSTR)
MsgType	메시지 유형	MQMT_DATAGRAM*
Report	요청된 보고서 메시지의 옵션	응답 메시지가 예상되면, 이 필드는 다음과 같이 채워집니다. MQRO_PAN*은 처리가 성공적인 경우 긍정적인 조치 보고서가 필요함을 표시하며, MQRO_NAN*은 처리에 실패한 경우 부정적인 조치 보고서가 필요함을 표시하며, MQRO_COPY_MSG_ID_TO_CORREL_ID*는 생성된 보고서의 상관 ID가 원래 발행된 요청의 메시지 ID와 동일해야 함을 표시합니다.
ReplyToQ	응답 대기열의 이름	응답 메시지가 예상되면 이 필드는 커넥터 등록 정보 ReplyToQueue의 값으로 채워집니다.
Persistence	메시지 지속성	MQPER_PERSISTENT*
Expiry	메시지 지속 시간	MQEI_UNLIMITED*

별표(*)는 IBM에서 정의한 상수를 나타냅니다.

표 1에 설명된 메시지 머리글 다음에는 메시지 본문이 나옵니다. 메시지 본문은 Data Handler를 사용하여 직렬화된 Business Object입니다.

보고서 필드는 긍정 조치 보고서와 부정 조치 보고서 모두가 수신 응용프로그램에서 예상됨을 표시하기 위해 설정됩니다. 메시지를 발행한 스레드는 수신 응용프로그램이 요청을 처리할 수 있었는지 여부를 표시하는 응답 메시지를 기다립니다.

응용프로그램이 커넥터로부터 동기 요청을 수신하면, Business Object를 처리하며 2, 3 및 4에 설명된 보고서 메시지를 발행합니다.

표 2. 응답 메시지 설명자 머리글(MQMD)

필드	설명	값
Format	형식 이름	변환 등록 정보에 정의된 대로 busObj의 입력 형식
MsgType	메시지 유형	MQMT_REPORT*

별표(*)는 IBM에서 정의한 상수를 나타냅니다.

표 3. 응답 메시지 채우기

Verb	피드백 필드	메시지 본문
create, update 또는 delete	SUCCESS	(선택적) 변경사항을 반영하는 직렬화된 Business Object
	VALCHANGE	
	VALDUPES	(선택적) 오류 메시지
	FAIL	

표 4. MQ Integrator Broker 피드백 코드 및 WebSphere Business Integration System 응답 값

MQ Integrator 피드백 코드	동등한 WebSphere Business Integration System 응답*
MQFB_PAN 또는 MQFB_APPL_FIRST	SUCCESS
MQFB_NAN 또는 MQFB_APPL_FIRST + 1	FAIL

표 4. MQ Integrator Broker 피드백 코드 및 WebSphere Business Integration System 응답 값 (계속)

MQ Integrator 피드백 코드	동등한 WebSphere Business Integration System 응답*
MQFB_APPL_FIRST + 2	VALCHANGE
MQFB_APPL_FIRST + 3	VALDUPES
MQFB_APPL_FIRST + 4	MULTIPLE_HITS
MQFB_APPL_FIRST + 5	FAIL_RETRIEVE_BY_CONTENT
MQFB_APPL_FIRST + 6	BO_DOES_NOT_EXIST
MQFB_APPL_FIRST + 7	UNABLE_TO_LOGIN(커넥터 에이전트가 즉시 종료됨)
MQFB_APPL_FIRST + 8	APP_RESPONSE_TIMEOUT

*자세한 정보는 *Connector Development Guide*를 참조하십시오.

Business Object를 처리할 수 있는 경우, 응용프로그램은 MQFB_PAN(또는 특정 WebSphere Business Integration System 값)으로 설정된 피드백 필드를 사용하여 보고서 메시지를 작성합니다. 선택적으로 응용프로그램은 변경사항이 있는 직렬화된 Business Object로 메시지 본문을 채웁니다. Business Object를 처리할 수 없는 경우, 응용프로그램은 MQFB_NAN(또는 특정 WebSphere Business Integration System 값)으로 설정된 피드백 필드를 사용하여 보고서 메시지를 작성한 후 선택적으로 오류 메시지를 메시지 본문에 포함시킵니다. 어느 경우에도, 응용프로그램은 메시지의 correlationID 필드를 커넥터 메시지의 messageID로 설정하며 replyTo 필드로 지정된 대기열로 발행합니다.

응답 메시지 검색에서 커넥터는 기본적으로 응답의 correlationID를 요청 메시지의 messageID에 대응시킵니다. 그런 다음 커넥터는 요청을 발행한 스레드에 알립니다. 응답의 피드백 필드에 따라, 커넥터는 메시지 본문에서 Business Object 또는 오류 메시지 중 하나를 예상합니다. Business Object가 예상되었지만 메시지 본문이 채워지지 않은 경우, 커넥터는 요청 조건의 InterChange Server에서 원래 발행된 동일한 Business Object를 단순히 리턴합니다. 오류 메시지가 예상되었지만 메시지 본문이 채워지지 않은 경우, 응답 코드와 함께 일반 오류 메시지가 InterChange Server로 리턴됩니다. 그러나 어댑터가 메시지 선택자를 사용하여 주어진 요청에 대해 응답 메시지를 식별하는 방법을 식별하고 필터하고, 아니면 제어할 수도 있습니다. 이 메시지 선택자 기능은 JMS 기능입니다. 이 기능은 동기 요청 처리에만 적용되고 아래에 설명되어 있습니다.

메시지 선택자를 사용하여 응답 메시지 필터: 커넥터가 동기 요청 처리에 대한 Business Object를 수신하자마자 Verb의 특정 응용프로그램 정보에 있는 response_selector 문자열의 존재를 확인합니다. response_selector가 정의되어 있지 않으면, 커넥터가 위에 설명된 대로 상관 ID를 사용하여 응답 메시지를 식별합니다.

response_selector가 정의되어 있으면, 커넥터가 다음 구문의 이름값 쌍을 예상합니다.


```
response_selector=JMSCorrelationID LIKE'selectorstring'
```

메시지 *selectorstring*이 응답을 고유하게 구별해야 하고 값은 아래에 있는 예와 같이 작은따옴표(')로 묶여 있습니다.

```
response_selector=JMSCorrelationID LIKE 'Oshkosh'
```

위에 있는 예에서 어댑터가 요청 메시지를 발행한 후에 "Oshkosh"와 같은 *correlationID*가 있는 응답 메시지에 대한 ReplyToQueue를 모니터링합니다. 어댑터는 먼저 이 메시지 선택자와 일치하는 메시지를 검색한 후, 그것을 응답으로서 디스패치합니다.

선택적으로 어댑터가 런타임 대체를 수행하며, 이는 각 요청에 대한 고유한 메시지 선택자를 생성하게 합니다. 메시지 선택자 대신 중괄호로 묶은 정수 양식으로 위치 표시기를 지정합니다(예: '{1}'). 그런 다음 콜론을 추가하고 대체에 사용할 속성 목록을 쉼표로 구분하여 붙입니다. 위치 표시기에 있는 정수가 대체에 사용할 속성에 대한 색인 역할을 합니다. 예를 들어, 다음 메시지 선택자는

```
response_selector=JMSCorrelationID LIKE '{1}': MyDynamicMO.CorrelationID
```

어댑터에게 {1}을 선택자 뒤에 오는 첫 번째 속성의 값으로 바꾸라고 알려줍니다(이 경우에는 MyDynamicMO라는 하위 오브젝트의 CorrelationId라는 속성). CorrelationID 속성에 123ABC라는 값이 있으면 어댑터가 다음 기준으로 작성된 메시지 선택자를 생성하고 사용하여

```
JMSCorrelation LIKE '123ABC'
```

응답 메시지를 식별합니다.

다음과 같이 여러 대체를 지정할 수도 있습니다.

```
response_selector=PrimaryId LIKE '{1}' AND AddressId LIKE '{2}' :  
PrimaryId, Address[4].AddressId
```

이 예에서는 어댑터가 {1}을 최상위 레벨 Business Object의 PrimaryId 속성 값으로 바꾸고 {2}를 하위 컨테이너 오브젝트 Address의 5번째 위치의 AddressId 값으로 바꿀 것입니다. 이러한 접근 방법으로 응답 메시지에 있는 Business Object 및 Meta Object의 아무 값이나 참조할 수 있습니다. Address[4].AddressId를 사용하여 검색을 좀더 자세히 수행하는 방법에 대한 자세한 정보는 JCDK API 매뉴얼(getAttribute 메소드)을 참조하십시오.

런타임 시 다음 중 어떤 것이든 발생하면 오류가 보고됩니다.

- '{}' 기호 사이에 정수가 아닌 값을 지정하는 경우
- 속성이 정의되어 있지 않은 색인을 지정하는 경우

- Business Object 또는 Meta Object에 지정된 속성이 존재하지 않는 경우
- 속성 경로의 구문이 잘못된 경우

예를 들어, 메시지 선택자에 리터럴 값인 '{' 또는 '}'를 포함한 경우, '{{' 또는 "{}"를 각각 사용할 수 있습니다. 속성값에 이들 문자를 놓을 수도 있습니다. 이 경우 첫 번째 "{"는 필요없습니다. 이탤 문자를 사용하는 다음 예를 고려하십시오.

```
response_selector=JMSCorrelation LIKE '{1}' and CompanyName='A{{P':MyDynamicMO.CorrelationID
```

커넥터가 다음과 같이 이 메시지 선택자를 분석할 것입니다.

```
JMSCorrelationID LIKE '123ABC' and CompanyName='A{P'
```

커넥터가 속성 값에서 특수 문자(예: '{', '}', ':', ';' 또는 ':')를 발견하면, 조회 문자열로 직접 삽입됩니다. 이렇게 하면 조회 문자열에 특정 응용프로그램 정보 분리문자 역할도 하는 특수 문자를 포함할 수 있습니다.

다음 예에서는 리터럴 문자열 대체가 속성 값에서 추출되는 방법을 보여줍니다.

```
response_selector=JMSCorrelation LIKE '{1}' and CompanyName='A{{P':MyDynamicMO.CorrelationID
```

MyDynamicMO.CorrelationID가 {A:B}C;D 값을 포함하면, 커넥터가 다음과 같이 메시지 선택자를 분석할 것입니다.

```
JMSCorrelationID LIKE '{A:B}C;D' and CompanyName='A{P'
```

응답 선택자 코드에 대한 자세한 정보는 JMS 1.0.1 스펙을 참조하십시오.

Retrieve, Exists 및 Retrieve By Content

Retrieve, Exists 및 Retrieve By Content Verb가 있는 Business Object는 동기 전달만을 지원합니다. 커넥터는 작성, 갱신 및 삭제에 대해 정의된 동기 전달에 대해서와 같이 이들 Verb로 Business Object를 처리합니다. 그러나 Retrieve, Exists 및 Retrieve By Content Verb를 사용할 때, responseTimeout 및 replyToQueue가 필요합니다. 또한 Retrieve By Content 및 Retrieve Verb의 경우, 트랜잭션을 완료하려면 메시지 본문이 직렬화된 Business Object로 채워져야 합니다.

표 5에서는 이 Verb에 대한 응답 메시지를 보여줍니다.

표 5. 응답 메시지 채우기

Verb	피드백 필드	메시지 본문
Retrieve 또는 RetrieveByContent	FAIL FAIL_RETRIEVE_BY_CONTENT	(선택적) 오류 메시지
	MULTIPLE_HITS SUCCESS	직렬화된 Business Object

표 5. 응답 메시지 채우기 (계속)

Verb	피드백 필드	메시지 본문
Exist	FAIL	(선택적) 오류 메시지
	SUCCESS	

로케일 종속 데이터 처리

커넥터가 2바이트 문자 세트를 지원하고 지정된 언어로 메시지 텍스트를 전달하도록 국제화되었습니다. 커넥터가 한 문자 코드를 사용하는 위치에서 다른 문자 세트를 사용하는 위치로 데이터를 전송할 때 커넥터는 문자 변환을 수행하여 데이터 의미를 유지합니다.

JVM(Java Virtual Machine) 내의 Java 런타임 환경은 Unicode 문자 코드 세트로 데이터를 표시합니다. Unicode는 가장 잘 알려진 문자 코드 세트(단일 바이트와 다중 바이트 모두)로 된 문자의 인코딩을 포함합니다. WebSphere Business Integration System의 대부분의 구성요소는 Java로 작성됩니다. 따라서 데이터가 대부분의 통합 구성요소 사이에서 전송될 때 문자 변환은 필요하지 않습니다.

해당 국가 또는 지역에서 해당 언어로 오류 및 정보 메시지를 로그하려면 사용자 환경에 대한 Locale 표준 구성 등록 정보를 구성하십시오. 구성 등록 정보에 대한 자세한 정보는 55 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

제 2 장 커넥터 설치 및 구성

- 『설치 작업 개요』
- 『커넥터 및 관련 파일 설치』
- 『설치된 파일 구조』
- 19 페이지의 『메시지 플로우 수정』
- 21 페이지의 『커넥터 구성』
- 27 페이지의 『대기열 URI(Uniform Resource Identifiers)』
- 29 페이지의 『Meta Object 속성 구성』
- 41 페이지의 『시작 파일 구성』
- 42 페이지의 『시작』

이 장에서는 커넥터를 설치하고 구성하는 방법과 커넥터와 작업하도록 메시지 플로우를 구성하는 방법에 대해 설명합니다.

설치 작업 개요

Connector for MQ Integrator Broker를 설치하려면 다음 작업을 수행해야 합니다.

- 통합 브로커 설치 WebSphere Business Integration System 설치와 통합 브로커 시작을 포함하는 이 작업은 사용자 브로커 및 운영 체제의 설치 문서에서 설명됩니다.
- 어댑터 및 관련 파일 설치 이 작업은 소프트웨어 패키지에서 어댑터의 파일을 사용자 시스템으로 설치하는 것을 포함합니다. 『커넥터 및 관련 파일 설치』를 참조하십시오.

커넥터 및 관련 파일 설치

WebSphere Business Integration Adapter 제품에 대한 자세한 정보는 다음 사이트의 WebSphere Business Integration Adapters Infocenter에서 *WebSphere Business Integration Adapters 설치 안내서*를 참조하십시오.

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

설치된 파일 구조

아래의 섹션에서는 설치 후 제품의 경로 및 파일명에 대해 설명합니다.

Windows 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

유틸리티는 *ProductDir*\connectors\WebSphereMQIntegratorBroker 디렉토리에 커넥터 에이전트를 설치하고, 시작 메뉴에 커넥터 에이전트의 바로 가기를 추가합니다.

표 6에서는 커넥터에서 사용하는 Windows 파일 구조에 대해 설명하며, 설치 프로그램을 통해 커넥터를 설치하도록 선택할 때 자동으로 설치된 파일을 보여줍니다.

표 6. 커넥터를 위해 설치된 Windows 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors\WebSphereMQIntegratorBroker\CWebSphereMQIntegratorBroker.jar	WebSphere MQ Integrator 커넥터에서만 사용되는 클래스가 들어 있습니다
connectors\WebSphereMQIntegratorBroker\start_WebSphereMQIntegratorBroker.bat	커넥터의 시작 스크립트(Windows 2000)
connectors\messages\WebSphereMQIntegratorBrokerConnector.txt	커넥터의 메시지 파일
repository\WebSphereMQIntegratorBroker\CN_WebSphereMQIntegratorBroker.txt	커넥터의 저장소 정의
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\LegacyItem.txt	샘플 메시지
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\Sample_Integrator_Broker_Workspace.xml	샘플 플로우를 포함하는 MQ Integrator Broker 작업공간
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\Sample_Event_Broker_Workspace.xml	샘플 플로우를 포함하는 MQ Event Broker 작업공간
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\WebSphereMQIntegratorBrokerConnector.cfg	샘플 커넥터 구성 파일
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\PortConnector.cfg	샘플 포트 커넥터 구성 파일
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\Sample_WMQIB_LegacyItem.xsd	샘플 XML 스키마
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\Sample_WMQIB_LegacyItem_XMLDoc.xsd	샘플 XML 스키마
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\Sample_WMQIB_MO_Config.xsd	샘플 Meta Object
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\Sample_WMQIB_MO_DataHandler.xsd	샘플 Data Handler Meta Object
connectors\WebSphereMQIntegratorBroker\samples\LegacyItem\Sample_WMQIB_MO_DataHandler_XMLConfig.xsd	샘플 XML Data Handler 구성 스키마 파일

주: 모든 제품 경로 이름은 제품이 사용자 시스템에 설치된 디렉토리와 관련됩니다.

UNIX 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

유틸리티는 *ProductDir*/connectors/MQSIV2 디렉토리에 커넥터 에이전트를 설치합니다.

표 7에서는 커넥터에서 사용하는 UNIX 파일 구조에 대해 설명하며, 설치 프로그램을 통해 커넥터를 설치하도록 선택할 때 자동으로 설치된 파일을 보여줍니다.

표 7. 커넥터에 설치된 UNIX 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors/WebSphereMQIntegratorBroker/CWebSphereMQIntegratorBroker.jar	WebSphere MQ Integrator 커넥터에서만 사용되는 클래스가 들어 있습니다

표 7. 커넥터에 설치된 UNIX 파일 구조 (계속)

<i>ProductDir</i> 의 서브디렉토리	설명
connectors/WebSphereMQIntegratorBroker/start_WebSphereMQIntegratorBroker.sh	커넥터의 시스템 시작 스크립트. 이 스크립트는 일반 커넥터 관리자 스크립트에서 호출됩니다. 시스템 관리자의 커넥터 구성 화면을 누르면, 설치 프로그램은 이 커넥터 관리자 스크립트의 사용자 정의된 램퍼를 작성합니다. 커넥터를 시작하고 중지하려면 이 사용자 정의된 램퍼를 사용하십시오.
connectors/messages/MQSIV2Connector.txt	커넥터의 메시지 파일
repository/MQSIV2/CN_MQSIV2.txt	커넥터의 저장소 정의
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/LegacyItem.txt	샘플 메시지
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/Sample_Integrator_Broker_Workspace.xml	샘플 플로우를 포함하는 MQ Integrator Broker 작업공간
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/Sample_Event_Broker_Workspace.xml	샘플 플로우를 포함하는 MQ Event Broker 작업공간
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/WebSphereMQIntegratorBrokerConnector.cfg	샘플 커넥터 구성 파일
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/PortConnector.cfg	샘플 포트 커넥터 구성 파일
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/Sample_WMQIB_LegacyItem.xsd	샘플 XML 스키마
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/Sample_WMQIB_LegacyItem_XMLDoc.xsd	샘플 XML 스키마
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/Sample_WMQIB_MO_Config.xsd	샘플 Meta Object
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/Sample_WMQIB_MO_DataHandler.xsd	샘플 Data Handler Meta Object
connectors/WebSphereMQIntegratorBroker/samples/LegacyItem/Sample_WMQIB_MO_DataHandler_XMLConfig.xsd	샘플 XML Data Handler 구성 스키마 파일

주: 모든 제품 경로 이름은 제품이 사용자 시스템에 설치된 디렉토리과 관련됩니다.

메시지 플로우 수정

각 수신 메시지가 Data Handler에 해당하는 형식으로 변환되도록 메시지 플로우를 수정해야 합니다. 이 변환은 메시지가 커넥터의 입력 대기열로 발행되기 전에 일어나야 합니다.

예를 들어, XML Data Handler를 구성한 경우, 메시지를 커넥터의 입력 대기열로 발행하고, 메시지를 XML로 변환하기 전에 메시지 플로우를 수정해야 합니다. 이 수정을 위해서는(XML Data Handler의 경우), 계산 노드를 메시지 플로우 끝에 추가하십시오. 계산 노드는 그림 4에 표시된 ESQ문을 포함해야 합니다.

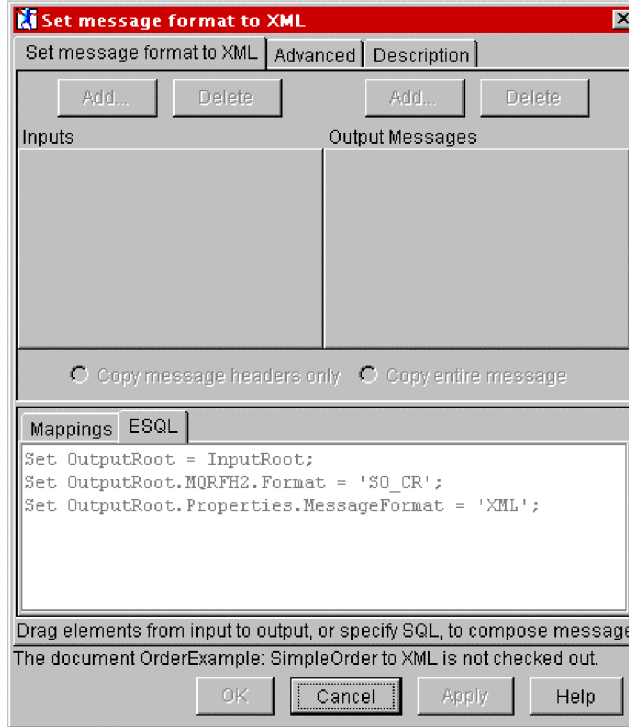


그림 4. 메시지 형식을 XML로 설정

그림 4에서는 커넥터가 이해할 수 있는 형식으로 수신 메시지를 변환하도록 구성된 계산 노드의 샘플 보기를 보여줍니다. 이 계산 노드가 사용 가능해지면, 원래 메시지를 나타내는 XML 문서가 커넥터 입력 대기열로 발행됩니다.

그림 4의 ESQL 텍스트 영역에 있는 필드는 다음과 같습니다.

```
Set OutputRoot = InputRoot;
```

이것은 출력용 메시지를 복사합니다.

```
Set OutputRoot.MQRFH2.Format = 'SO-CR';
```

이것은 커넥터가 이 형식을 점검하여 적절하게 메시지를 변환하도록 보장합니다.

```
SET OutputRoot.Properties.MessageFormat = 'XML';
```

이것은 전달 중 메시지가 XML로 변환되어야 함을 MQ Integrator Broker에 나타냅니다.

주: MQ Integrator Broker MRM(Message Repository Manager)에서 사용자 정의 형식을 정의한 경우, 메시지 형식을 단순히 XML로 설정하여 레거시 형식을 XML로 변환할 수 있습니다. 이 형식은 MQRFH2와는 다릅니다. OutputRoot.Properties.MessageFormat은 MRM과 관련되지만, OutputRoot.MQRFH2.Format은 메시지를 수신 중인 응용프로그램에 대한 메시지 형식을 지정하는 데 사용됩니다.

커넥터 구성

커넥터에는 두 유형의 구성 등록 정보(표준 구성 등록 정보 및 어댑터 특정 구성 등록 정보)가 있습니다. 어댑터를 실행하기 전에 이러한 등록 정보의 값을 설정해야 합니다.

Connector Configurator를 사용하여 커넥터 등록 정보를 구성할 수 있습니다.

- Connector Configurator 및 단계별 절차에 대해서는 73 페이지의 부록 B 『Connector Configurator』를 참조하십시오.
- 표준 커넥터 등록 정보 설명에 대해서는 『표준 커넥터 등록 정보』 및 55 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.
- 커넥터 특정 등록 정보 설명은 『커넥터 특정 등록 정보』를 참조하십시오.

커넥터는 시작할 때 구성 값을 얻습니다. 런타임 세션 중, 하나 이상의 커넥터 등록 정보의 값을 변경하려고 할 수 있습니다. 일부 커넥터 구성 등록 정보(예: AgentTraceLevel)에 대한 변경사항은 즉시 적용됩니다. 기타 커넥터 구성요소에 대한 변경사항을 적용하려면 변경 이후에 구성요소를 다시 시작하거나 시스템을 다시 시작해야 합니다. 등록 정보가 동적인지(즉시 적용) 또는 정적(커넥터 구성요소 다시 시작 또는 시스템 다시 시작 중 하나가 필요)인지 판별하려면 System Manager의 커넥터 등록 정보 창에서 갱신 메소드 열을 참조하십시오.

표준 커넥터 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이 등록 정보에 대한 문서는 55 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

주: Connector Configurator에서 구성 등록 정보를 설정할 때, 브로커 유형 등록 정보를 사용하여 브로커를 지정합니다. 설정되면, 브로커 관련 등록 정보는 Connector Configurator 창에 나타납니다.

커넥터 특정 등록 정보

커넥터 특정 구성 등록 정보는 런타임 시에 커넥터 에이전트에 필요한 정보를 제공합니다. 또한 커넥터 특정 등록 정보는 에이전트를 다시 코드화하여 재빌드할 필요없이 커넥터 에이전트 내의 논리 또는 정적 정보를 변경하는 방식을 제공합니다.

표 8에서는 커넥터에 대한 커넥터 특정 구성 등록 정보를 나열합니다. 등록 정보의 설명에 대해서는 다음 절을 참조하십시오.

표 8. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수
ApplicationPassword	로그인 암호		아니오
ApplicationUserName	로그인 사용자 ID		아니오
ArchiveQueue	정상적으로 처리된 메시지의 사본이 전송되는 대기열	MQWFCONN.ARCHIVE	아니오
CCSID	대기열 관리자 연결에 대한 문자-세트	null	예
Channel	MQ 서버 커넥터 채널		예
ConfigurationMetaObject	구성 Meta Object의 이름		예
DataHandlerClassName	Data Handler 클래스 이름	com.crossworlds.DataHandlers. text.xml	예
DataHandlerConfigMO	Data Handler Meta Object	MQ_DataHandler_Default	예
DataHandlerMimeType	MIME 유형의 파일	text/xml	아니오
DefaultVerb	커넥터에 의해 지원되는 모든 Verb	Create	
ErrorQueue	미처리 메시지의 대기열	MQWFCONN.ERROR	아니오
HostName	WebSphere MQ 서버		아니오
InDoubtEvents	FailOnStartup Reprocess IgnoreLogError	Reprocess	아니오
InputQueue	풀 대기열	MQWFCONN.IN	예
InProgressQueue	진행 중 이벤트 대기열	MQWFCONN.IN_PROGRESS	아니오
PollQuantity	InputQueue 등록 정보에 지정된 각 대기열에서 검색할 메시지 수	1	아니오
Port	WebSphere MQ 리스너용으로 설정된 포트		아니오
ReplyToQueue	커넥터가 요청을 발행할 때 응답 메시지가 전달되는 대기열	MQWFCONN.REPLYTO	아니오
UnsubscribedQueue	등록되지 않은 메시지가 전송되는 대기열	MQWFCONN.UNSUBSCRIBE	아니오
UseDefaults	true 또는 false	false	

ApplicationPassword

MQ Integrator Broker에 로그인하기 위해 UserID로 사용되는 암호입니다.

기본값 = 없음

ApplicationPassword가 공백으로 남아 있거나 제거된 경우, 커넥터는 MQ Integrator Broker에서 제공하는 기본 암호를 사용합니다.*

ApplicationUserName

MQ Integrator Broker에 로그인하기 위해 Password로 사용되는 사용자 ID입니다.

기본값 = 없음

ApplicationUserName이 공백으로 남아 있거나 제거된 경우, 커넥터는 MQ Integrator Broker에서 제공하는 기본 사용자 ID를 사용합니다.*

ArchiveQueue

정상적으로 처리된 메시지의 사본이 전송되는 대기열입니다.

기본값 = MQWFCONN.ARCHIVE

CCSID

대기열 관리자 연결에 대한 문자 세트입니다. 이 등록 정보의 값은 대기열 URI에 있는 CCSID 등록 정보 값과 일치해야 합니다. 27 페이지의 『대기열 URI(Uniform Resource Identifiers)』를 참조하십시오.

기본값 = null

Channel

커넥터가 MQ Integrator Broker와 통신하는 MQ 서버 커넥터 채널입니다.

기본값 = 없음

Channel이 공백으로 남아 있거나 제거된 경우, 커넥터는 MQ Integrator Broker에서 제공하는 기본 서버 채널을 사용합니다.*

ConfigurationMetaObject

커넥터의 구성 정보가 들어 있는 Meta Object의 이름입니다.

기본값 = 없음

DataHandlerClassName

메시지를 Business Object로 변환하거나 역으로 변환할 때 사용할 Data Handler 클래스입니다.

기본값 = com.crossworlds.DataHandlers.text.xml

DataHandlerConfigMO

구성 정보를 제공하기 위해 Data Handler로 전달되는 Meta Object입니다.

기본값 = MO_DataHandler_Default

DataHandlerMimeType

특정 MIME 유형에 근거한 Data Handler를 요청하게 합니다.

기본값 = text/xml

DefaultVerb

Verb가 풀링하는 동안 Data Handler에 의해 설정되지 않았으면, 수신 Business Object 내에서 설정할 Verb를 지정합니다.

기본값 = Create

ErrorQueue

처리할 수 없는 메시지가 전송되는 대기열입니다.

기본값 = queue://CrossWorlds.QueueManager/ERROR

HostName

MQ Integrator Broker를 호스트하는 서버의 이름입니다.

기본값 = 없음

InDoubtEvents

예기치 않은 커넥터 종료로 인해 완전히 처리되지 않은 진행 중 이벤트를 처리하는 방식을 지정합니다. 초기화 중에 진행 중인 대기열에서 이벤트를 찾은 경우 취할 수 있는 네 가지 조치 중 하나를 선택하십시오.

- FailOnStartup. 오류를 로그하고 즉시 종료합니다.
- Reprocess. 우선 나머지 이벤트를 처리한 후, 입력 대기열에서 메시지를 처리합니다.
- Ignore. 진행 중인 대기열에서 모든 메시지를 무시합니다.
- LogError. 오류를 로그하지만 종료하지 않습니다.

기본값 = Reprocess

InputQueue

새 메시지의 커넥터에 의해 폴링될 메시지 대기열. 커넥터는 여러 세미콜론으로 구분된 대기열 이름을 승인합니다. 예를 들어, 세 개의 대기열(MyQueueA, MyQueueB 및 MyQueueC)을 폴링하려면 커넥터 구성 등록 정보 *InputQueue*의 값이 MyQueueA;MyQueueB;MyQueueC와 같습니다.

커넥터는 라운드 로빈 방식으로 대기열을 폴링하고 각 대기열에서 메시지의 *pollQuantity* 수까지 검색합니다. 예를 들어, *pollQuantity*가 2와 같고, MyQueueA에 두 개의 메시지가 있으며, MyQueueB에는 한 개의 메시지가 있으며 MyQueueC에는 5개의 메시지가 있는 경우, 커넥터는 다음과 같은 방식으로 메시지를 검색합니다.

*pollQuantity*가 2이므로, 커넥터는 *pollForEvents*에 대한 호출마다 각 대기열에서 최대 두 개까지의 메시지를 검색합니다. 첫 번째 주기(2개 중 1)의 경우, 커넥터는 각 MyQueueA, MyQueueB 및 MyQueueC에서 첫 번째 메시지를 검색합니다. 그러면 폴링의 처음 폴링 라운드를 완료하며, *pollQuantity*가 1인 경우에는 커넥터가 중지합니다. *pollQuantity*가 2이므로, 커넥터는 폴링의 두 번째 라운드(2개 중 2)를 시작하며 MyQueueA 및 MyQueueC에서 각각 하나의 메시지를 검색합니다.(MyQueueB는 지금 비어 있으므로 건너뛸니다.) 모든 대기열 2x 각각을 폴링한 후에 메소드 *pollForEvents*에 대한 호출이 완료됩니다. 다음은 메시지 검색의 순서입니다.

1. MyQueueA에서 메시지 하나
2. MyQueueB에서 메시지 하나
3. MyQueueC에서 메시지 하나
4. MyQueueA에서 메시지 하나

5. MyQueueB는 지금 비어 있으므로 건너뛰

6. MyQueueC에서 메시지 하나

기본값 = `queue://CrossWorlds.QueueManager/IN`

InProgressQueue

처리 중 메시지가 보류된 메시지 대기열. 커넥터에 특정한 등록 정보에서 기본 InProgressQueue 이름을 제거하기 위해 System Manager를 사용하여 커넥터가 이 대기열 없이 작동하도록 구성할 수 있습니다. 이렇게 하면 이벤트가 보류 중일 때 커넥터가 종료되어 있으면 이벤트 전달이 손상될 수 있다는 경고가 시작할 때 표시됩니다.

기본값= `queue://CrossWorlds.QueueManaer/IN_PROGRESS`

PollQuantity

pollForEvents 스캔 중 InputQueue 등록 정보에 지정된 각 대기열에서 검색할 메시지 수입니다.

기본값 =1

Port

MQ Integrator Broker 리스너용으로 설정된 포트입니다.

기본값 = 없음

ReplyToQueue

커넥터가 요청을 발행할 때 응답 메시지가 전달되는 대기열입니다.

기본값 = `queue://CrossWorlds.QueueManager/REPLYTO`

UnsubscribedQueue

등록되지 않은 메시지가 전송되는 대기열입니다.

기본값 = `queue://CrossWorlds.QueueManager/UNSUBSCRIBED`

주: *잘못되었거나 알지 못할 수 있으므로 MQ Integrator Broker가 제공하는 값을 항상 확인하십시오. 그런 경우, 내재적으로 값을 지정하십시오.

UseDefaults

Create 조작에서 UseDefaults가 true로 설정되면, 커넥터가 isRequired Business Object 속성마다 유효한 값 또는 기본값이 제공되었는지를 확인합니다. 값이 제공되면, Create 조작이 성공합니다. 매개변수가 false로 설정되어 있으면, 커넥터가 유효한 값만 확인하고 그것이 제공되지 않으면 Create 조작이 실패합니다. 기본값은 false입니다.

다중 커넥터 인스턴스 작성

복수의 커넥터 인스턴스를 작성하는 것은 여러 가지 면에서 사용자 정의 커넥터를 작성하는 방법과 동일합니다. 다음 단계를 수행하여 복수의 커넥터 인스턴스를 작성하여 실행하도록 시스템을 설정할 수 있습니다. 복수의 커넥터 인스턴스를 작성하려면 다음을 수행해야 합니다.

- 커넥터 인스턴스의 새 디렉토리를 작성하십시오.
- 필수 Business Object 정의가 있는지 확인하십시오.
- 새 커넥터 정의 파일을 작성하십시오.
- 새 시작 스크립트를 작성하십시오.

새 디렉토리 작성

각 커넥터 인스턴스의 커넥터 디렉토리를 작성해야 합니다. 다음과 같이 이 커넥터 디렉토리의 이름을 지정해야 합니다.

```
ProductDir\connectors\connectorInstance
```

여기서 connectorInstance는 고유하게 커넥터 인스턴스를 식별합니다.

커넥터에 커넥터 특정 Meta Object가 있는 경우, 커넥터 인스턴스의 Meta Object를 작성해야 합니다. Meta Object를 파일로 저장할 경우, 이 디렉토리를 작성한 후 파일을 다음 디렉토리에 저장하십시오.

```
ProductDir\repository\connectorInstance
```

Business Object 정의 작성

각 커넥터 인스턴스에 대한 Business Object 정의가 아직 프로젝트 내에 존재하지 않으면 이를 정의해야 합니다.

1. 초기 커넥터와 연관된 Business Object 정의를 수정해야 할 경우, 해당 파일을 복사한 다음 Business Object Designer를 사용하여 가져오십시오. 초기 커넥터에 대한 모든 파일을 복사할 수 있습니다. 파일을 변경해야 할 경우 이름만 바꾸십시오.
2. 초기 커넥터용 파일은 다음 디렉토리에 있어야 합니다.

```
ProductDir\repository\initialConnectorInstance
```

작성한 모든 추가 파일은 ProductDir\repository의 적절한 connectorInstance 서브디렉토리에 있어야 합니다.

커넥터 정의 작성

Connector Configurator에서 커넥터 인스턴스의 구성 파일(커넥터 파일)을 작성합니다. 이를 수행하려면 다음 단계를 따르십시오.

1. 초기 커넥터의 구성 파일(커넥터 정의)을 복사한 후 이름을 바꾸십시오.

2. 커넥터 인스턴스에 지원되는 Business Object(및 연관된 Meta Object)가 올바르게 나열되는지 확인하십시오.
3. 커넥터 등록 정보를 적절하게 사용자 정의하십시오.

시작 스크립트 작성

시작 스크립트를 작성하려면 다음을 수행하십시오.

1. 초기 커넥터의 시작 스크립트를 복사한 다음 시작 스크립트 이름에 커넥터 디렉토리의 이름이 포함되도록 이름을 지정하십시오.
dirname
2. 시작 스크립트를 26 페이지의 『새 디렉토리 작성』에서 작성한 커넥터 디렉토리에 배치하십시오.
3. 시작 스크립트 단축 아이콘을 작성하십시오(Windows 전용).
4. 초기 커넥터의 단축 아이콘 텍스트를 복사한 후 새 커넥터 인스턴스의 이름과 일치하도록 초기 커넥터의 이름을 변경하십시오(명령행에서).

이제 통합 서버에서 두 커넥터 인스턴스를 동시에 실행할 수 있습니다.

사용자 정의 사용자 정의 작성에 대한 자세한 정보는 *Connector Development Guide for C++* 또는 *Connector Development for Java*를 참조하십시오.

대기열 URI(Uniform Resource Identifiers)

대기열의 URI는 순서 queue://로 시작하며 다음 항목이 뒤따릅니다.

- 대기열이 상주하는 대기열 관리자의 이름
- 또다른 /
- 대기열의 이름
- 선택적으로, 나머지 대기열 등록 정보를 설정하기 위한 이름-값 쌍의 목록.

예를 들어, 다음 URI는 대기열 관리자 crossworlds.queue.manager에서 IN 대기열에 연결하며 모든 메시지를 우선순위가 5인 WebSphere MQ 메시지로 전송합니다.

```
queue://crossworlds.queue.manager/IN?targetClient=1&persistence=5
```

표 9에서는 대기열 URI의 등록 정보 이름을 보여줍니다.

표 9. MQ Integrator Broker 대기열 URI 등록 정보

등록 정보 이름	설명	값
expiry	메시지 지속 시간(ms)	0 = 무제한 양수 = 제한시간(ms)

표 9. MQ Integrator Broker 대기열 URI 등록 정보 (계속)

등록 정보 이름	설명	값
priority	메시지의 우선순위	0-9, 여기서 1은 최고의 우선순위입니다. -1의 값은 등록 정보가 대기열의 구성으로 판별되어야 함을 의미합니다. -2 값은 커넥터가 자체 기본값을 사용할 수 있음을 지정합니다.
persistence	메시지가 디스크로 '고정'되어야 하는지 여부	1 = 비지속적 2 = 지속적
CCSID	목적지의 문자 세트	-1의 값은 등록 정보가 대기열의 구성으로 판별되어야 함을 의미합니다. -2 값은 커넥터가 자체 기본값을 사용할 수 있음을 지정합니다. 정수 - 기본 MQ Integrator Broker 문서에 나열된 유효한 값. 이 값은 CCSID 캐릭터 특정 구성 등록 정보의 값과 일치해야 합니다. 23 페이지의 『CCSID』를 참조하십시오.
targetClient	수신 응용프로그램이 JMS를 준수하는지 여부	0 = JMS(MQRFH2 머리글)
encoding	숫자 필드 표시 방법	1 = MQ(MQMD 머리글만) 기본 MQ Integrator Broker 문서에 설명된 정수 값

주: 커넥터에는 문자 세트(CCSID)의 제어나 MQMessages의 데이터 인코딩 속성이 없습니다. 데이터가 메시지 버퍼에서 검색되거나 메시지 버퍼로 전달될 때 데이터 변환이 적용되므로, 커넥터는 JMS의 IBM WebSphere MQ 구현에 따라 데이터를 변환합니다(IBM MQSeries Java 클라이언트 라이브러리 문서 참조). 이에 따라, 이 변환은 옵션 MQGMO_CONVERT를 사용하여 기본 MQSeries 또는 WebSphere MQ, API에 의해 수행되는 변환과 양방향으로 동등한 변환이어야 합니다. 커넥터는 변환 프로세스에서 차이점 또는 실패를 제어하지 않습니다. 커넥터는 추가 수정 없이 WebSphere MQ에서 지원하는 인코딩 또는 임의 CCSID의 메시지 데이터를 검색할 수 있습니다. 특정 CCSID 또는 인코딩의 메시지를 전달하려면 출력 대기열이 완전한 URI여야 하며 CCSID 및 encoding의 값을 지정해야 합니다. 커넥터는 MQMessage 전달을 위해 데이터를 인코딩할 때 (JMS API를 통해) 정보를 사용하는 WebSphere MQ에 이 정보를 전달합니다. 종종 IBM의 웹 사이트에서 IBM MQSeries Java 클라이언트 라이브러리의 최신 버전을 다운로드하여 CCSID 및 인코딩에 대한 지원의 부족을 해결할 수 있습니다. CCSID 및 인코딩에 특정한 문제점이 지속되는 경우, WebSphere Business Integration System 기술 지원에 문의하여 커넥터를 실행하기 위한 대체 JVM(Java Virtual Machine)을 사용하는 가능성을 논의하십시오.

Meta Object 속성 구성

Connector for MQ Integrator Broker는 두 가지 종류의 Meta Object를 인식하고 읽을 수 있습니다.

- 정적 커넥터 Meta Object
- 동적 하위 Meta Object

동적 하위 Meta Object의 속성 값은 정적 Meta Object의 속성 값을 대체 및 중복합니다.

정적 Meta Object

MQ Integrator Broker 구성 Meta Object는 여러 Business Object에 정의된 변환 등록 정보의 목록으로 구성됩니다. Business Object의 변환 등록 정보를 정의하려면 우선 문자열 속성을 작성하고 구문 busObj_verb를 사용하여 이름 지정하십시오. 예를 들어, Create Verb로 고객 오브젝트의 변환 등록 정보를 정의하려면 Customer_Create라는 속성을 작성하십시오. 속성의 응용프로그램 특정 정보에서 실제 변환 등록 정보를 지정합니다.

주: 정적 Meta Object가 지정되지 않은 경우, 커넥터는 폴링 중 주어진 메시지 형식을 특정 Business Object 유형으로 매핑할 수 없습니다. 이때, 커넥터는 Business Object를 지정하지 않고 메시지 텍스트를 구성된 Data Handler로 전달합니다. Data Handler가 텍스트에만 근거하여 Business Object를 작성할 수 없는 경우, 커넥터는 이 메시지 형식이 인식되지 않음을 나타내는 오류를 보고합니다.

표 10에서는 Meta Object 등록 정보에 대해 설명합니다.

표 10. MQ Integrator Broker Meta Object 등록 정보

등록 정보 이름	설명
InputFormat	InputFormat은 주어진 Business Object와 연관시킬 메시지 형식입니다. 검색된 메시지가 이러한 형식인 경우, 가능하면 주어진 Business Object로 변환됩니다. Business Object에 대해 이 형식이 지정되지 않은 경우, 커넥터는 주어진 Business Object에 대한 등록 전달을 처리하지 않습니다. 기본 변환 등록 정보를 사용할 때 이 등록 정보는 설정하지 마십시오. 이 등록 정보의 값은 수신되는 메시지를 Business Object와 대조하는 데 사용됩니다.
InputQueue	커넥터가 새 메시지를 발견하기 위해 폴링하는 입력 대기열. 주: 커넥터 특정 등록 정보의 InputQueue 등록 정보는 어댑터가 폴링하는 대기열을 정의합니다. 이것은 어댑터가 폴링할 대기열을 판별하는 데 사용하는 유일한 등록 정보입니다. 정적 MO에서 InputQueue 등록 정보 및 InputFormat 등록 정보는 특정 Business Object에 정해진 메시지를 매핑하는 어댑터의 기준 역할을 할 수 있습니다. InputQueues에 Data Handler 매핑에 대한 정보는 30 페이지의 『InputQueues에 Data Handler 매핑』 항목을 참조하십시오.
OutputFormat	OutputFormat은 주어진 Business Object에서 작성된 메시지에 설정됩니다. OutputFormat이 지정되지 않은 경우, 입력 형식이 사용됩니다. (사용 가능한 경우)
OutputQueue	OutputQueue는 주어진 Business Object에서 파생된 메시지가 전달된 출력 대기열입니다.
ResponseTimeout	응답을 기다릴 때 시간 종료되기 전에 기다리는 시간(ms)을 표시합니다. 이것이 정의되어 있지 않거나 0보다 작은 값을 갖는 경우, 커넥터는 응답을 기다리지 않고 즉시 SUCCESS를 리턴합니다.

표 10. MQ Integrator Broker Meta Object 등록 정보 (계속)

등록 정보 이름	설명
TimeoutFatal	이 등록 정보가 정의되고 True의 값을 가진 경우, ResponseTimeout에 의해 지정된 시간 내에 응답이 수신되지 않으면 커넥터는 APP_RESPONSE_TIMEOUT을 리턴합니다. 응답 메시지를 기다리는 기타 모든 스레드는 APP_RESPONSE_TIMEOUT을 즉시 InterChange Server로 리턴합니다. 이로써 InterChange Server가 커넥터를 종료합니다.

그리고 Default로 이름 지정된 예약 등록 정보는 Meta Object에서 정의될 수 있습니다. 이 등록 정보가 있으면, 해당 응용프로그램 특정 정보는 모든 Business Object 등록 정보에 대한 기본값을 지정합니다.

표 11. Customer_Create의 MQ Integrator 정적 Meta Object 구조

등록 정보 이름	응용프로그램 특정 텍스트
Default	OutputFormat=CUST_OUT; OutputQueue=QueueA; ResponseTimeout=10000; TimeoutFatal=False

응용프로그램 특정 정보

응용프로그램 특정 정보는 세미콜론으로 구분되는 이름-값 쌍 형식으로 구성됩니다. 예를 들면, 다음과 같습니다.

```
InputFormat=CUST_IN;OutputFormat=CUST_OUT
```

InputQueues에 Data Handler 맵핑

정적 Meta Object의 응용프로그램 특정 정보에 있는 nputQueue 등록 정보를 사용하여 Data Handler와 입력 대기열을 연관시킬 수 있습니다. 이 기능은 형식 및 변환 요구사항이 다른 여러 거래 상대와 관계될 때 유용합니다. 이를 위해서는 다음을 수행해야 합니다.

1. 커넥터 특정 등록 정보(24 페이지의 『InputQueue』 참조)를 사용하여 하나 이상의 입력 대기열을 구성하십시오.
2. 각 입력 대기열의 경우, 응용프로그램 특정 정보의 대기열 관리자, 입력 대기열 이름, Data Handler 클래스 이름 및 MIME 유형을 지정하십시오.

예를 들어, 정적 Meta Object의 다음 속성은 Data Handler와 CompReceipts 입력 대기열을 연관시킵니다.

```
[Attribute]
Name = Cust_Create
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = InputQueue=//queue.manager/CompReceipts;DataHandlerClassName=com.crossworlds.DataHandlers.MQ.disposition_notification;DataHandlerMimeType=
```

```
message/  
disposition_notification  
IsRequiredServerBound = false  
[End]
```

입력 형식 과부하

메시지 검색 시, 커넥터는 보통 입력 형식을 하나의 특정 Business Object 및 Verb 조합과 일치시킵니다. 그런 다음 커넥터는 Business Object 이름 및 메시지의 내용을 Data Handler로 전달합니다. Data Handler가 메시지 내용이 사용자가 예상하는 Business Object에 해당하는지 검증합니다.

그러나 동일한 입력 형식이 둘 이상의 Business Object에 정의된 경우, 데이터를 Data Handler로 전달하기 전에 데이터가 어떤 Business Object를 표시하는지 커넥터가 판별할 수 없습니다. 이러한 경우, 커넥터는 메시지 내용만을 Data Handler로 전달한 후 생성된 Business Object에 근거하여 변환 등록 정보를 찾습니다. 이에 따라, Data Handler는 메시지 내용에만 근거하여 Business Object를 판별해야 합니다.

생성된 Business Object의 Verb가 설정되지 않은 경우, 커넥터는 임의의 Verb로 이 Business Object에 정의된 변환 등록 정보를 검색합니다. 변환 등록 정보 세트가 하나만 있는 경우, 커넥터는 지정된 Verb를 지정합니다. 더 많은 등록 정보가 있는 경우, Verb를 구별할 수 없으므로 커넥터는 메시지 지정에 실패합니다.

샘플 Meta Object

아래 Meta Object는 Create, Update, Delete 및 Retrieve Verb를 사용하여 고객 Business Object를 변환하도록 커넥터를 구성합니다. 속성 Default가 Meta Object에 정의되어 있음에 유의하십시오. 이에 따라, 커넥터는 이 속성의 변환 등록 정보를 사용합니다(OutputQueue=CustomerQueue1;ResponseTimeout=5000;TimeoutFatal=true). 그러므로 속성에 의해 다르게 지정되지 않는 한, 커넥터는 모든 Business Object를 CustomerQueue1 대기열로 발행한 후 응답 메시지를 기다립니다. 응답이 5000 초 이내에 도착하지 않으면, 커넥터가 즉시 종료됩니다.

Create Verb를 사용한 고객 오브젝트: 속성 Customer_Create는 형식 NEW의 임의의 메시지가 Create Verb를 사용하여 고객 Business Object로 변환되어야 함을 커넥터에 나타냅니다. 출력 형식은 정의되어 있지 않으므로, 커넥터는 입력에 정의된 형식을 사용하여 이 오브젝트 Verb 조합을 나타내는 메시지를 전송합니다(이 경우 NEW).

Update Verb 및 Delete를 사용한 고객 오브젝트: 입력 형식 MODIFY는 Update Verb를 사용하는 Business Object 고객과 Delete Verb를 사용하는 Business Object 고객으로 모두 정의되어 과부하되어 있습니다. 이러한 형식의 검색 메시지를 정상적으로 처리하려면 식별할 Data Handler의 메시지 내용에 Business Object 이름 및 Verb가 포함되어야 합니다(『입력 형식 과부하』 참조). 요청 처리 조작의 경우, 출력 형식이 정의되어 있지 않으므로 커넥터는 입력 형식 MODIFY를 사용하여 어느 한쪽 Verb에 대한 메시지를 전송합니다.

Retrieve Verb를 사용한 고객 오브젝트: 속성 Customer_Retrieve는 Retrieve Verb를 사용한 고객 유형의 Business Object가 형식 Retrieve가 있는 메시지로 전송되도록 함을 지정합니다. 제한시간 전에 커넥터가 10000초 동안 기다리도록 기본 응답 시간이 대체되었음에 유의하십시오(응답이 수신되지 않는 경우에도 종료합니다).

```
[ReposCopy]
Version = 3.1.0
Repositories = 1cHyILNuPTc=
[End]
[BusinessObjectDefinition]
Name = Sample_MO
Version = 1.0.0

[Attribute]
Name = Default
Type = String
Cardinality = 1
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = false
AppSpecificInfo = OutputQueue=CustomerQueue1;ResponseTimeout=5000;TimeoutFatal=true
IsRequiredServerBound = false
[End]
[Attribute]
Name = Customer_Create
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = InputFormat=NEW
IsRequiredServerBound = false
[End]
[Attribute]
Name = Customer_Update
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = InputFormat=MODIFY
IsRequiredServerBound = false
[End]
[Attribute]
Name = Customer_Delete
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = InputFormat=MODIFY
IsRequiredServerBound = false
[End]
[Attribute]
Name = Customer_Retrieve
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
```

```

IsRequired = false
AppSpecificInfo = OutputFormat=RETRIEVE;ResponseTimeout=10000
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

```

동적 하위 Meta Object

정적 Meta Object를 통해 필요한 메타 데이터를 지정하기 어렵거나 가능하지 않은 경우, 커넥터는 각 Business Object 인스턴스에 대해 런타임 시 지정된 메타 데이터를 선택적으로 허용할 수 있습니다.

커넥터는 커넥터에 전달된 최상위 레벨 Business Object에 대한 하위로 추가된 동적 Meta Object에서 변환 등록 정보를 인식하고 읽습니다. 동적 하위 Meta Object의 속성 값은 커넥터를 구성하기 위해 사용되는 정적 Meta Object를 통해 지정할 수 있는 변환 등록 정보를 중복합니다.

동적 하위 Meta Object 등록 정보는 정적 Meta Object에 있는 등록 정보를 대체하므로, 동적 하위 Meta Object를 지정한 경우, 정적 Meta Object를 지정하는 커넥터 등록 정보를 포함할 필요가 없습니다. 이에 따라, 정적 Meta Object와 무관하게 동적 하위 Meta Object를 사용할 수 있으며 역으로도 가능합니다.

주: 커넥터는 동기 이벤트 전달 중에 협업 이름을 제공하기 위해 동적 하위 Meta Object의 사용을 지원하지 않습니다.

표 11 및 12에서는 Business Object Customer_Create에 대해 샘플 정적 및 동적 하위 Meta Object를 각각 보여줍니다. 응용프로그램 특정 정보는 세미콜론으로 구분된 이름-값 쌍으로 구성됩니다.

표 12. Customer_Create의 MQ Integrator 동적 Meta Object 구조

등록 정보 이름	값
OutputFormat	CUST_OUT
OutputQueue	QueueA
ResponseTimeout	10000
TimeoutFatal	False

커넥터는 cw_mo_conn 태그가 하위 Meta Object를 지정하는지 여부를 판별하기 위해 수신된 최상위 레벨 Business Object의 응용프로그램 특정 정보를 검사합니다. 그런 경우, 동적 하위 Meta Object 값은 정적 Meta Object에 지정된 값을 대체합니다.

폴링 시 동적 하위 Meta Object 채우기

폴링 중 검색된 메시지에 대한 자세한 정보를 협업에 제공하기 위해, 작성된 Business Object에 대해 이미 정의된 경우 커넥터는 동적 Meta Object의 특정 속성을 채웁니다.

표 13에서는 폴링을 위해 동적 하위 Meta Object를 구성할 수 있는 방식을 보여줍니다.

표 13. 폴링의 MQ Integrator 동적 Meta Object 구조

등록 정보 이름	샘플 값
InputFormat	CUST_IN
InputQueue	MYInputQueue
OutputFormat	CxIgnore
OutputQueue	CxIgnore
ResponseTimeout	CxIgnore
TimeoutFatal	CxIgnore

표 13에 표시된 대로, 추가 속성 Input_Format 및 InputQueue를 동적 하위 Meta Object에 정의할 수 있습니다. Input_Format은 수신된 메시지 형식으로 채워지지만, InputQueue 속성에는 주어진 메시지가 검색된 대기열의 이름이 있습니다. 이 등록 정보가 하위 Meta Object에 정의되지 않은 경우, 등록 정보는 채워지지 않습니다.

시나리오 예는 다음과 같습니다.

- 커넥터가 MyInputQueue 대기열에서 CUST_IN 형식으로 메시지를 검색합니다.
- 커넥터는 이 메시지를 고객 Business Object로 변환하고 Meta Object가 정의되었는지 여부를 판별하기 위해 응용프로그램 특정 텍스트를 검사합니다.
- Meta Object가 정의된 경우, 커넥터는 이 Meta Object의 인스턴스를 작성하며 InputQueue 및 InputFormat 속성을 이에 따라 채운 후, 협업을 사용 가능하게 하기 위해 Business Object를 공개합니다.

샘플 동적 하위 Meta Object

```
[BusinessObjectDefinition]
Name = MO_Sample_Config
Version = 1.0.0
```

```

[Attribute]
Name = OutputFormat
Type = String
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = false
DefaultValue = CUST
IsRequiredServerBound = false
[End]
[Attribute]
Name = OutputQueue
Type = String
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
DefaultValue = OUT
IsRequiredServerBound = false
[End]
[Attribute]
Name = ResponseTimeout
Type = String
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
DefaultValue = -1
IsRequiredServerBound = false
[End]
[Attribute]
Name = TimeoutFatal
Type = String
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
DefaultValue = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = InputFormat
Type = String
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = InputQueue
Type = String
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false

```

```

[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]
[BusinessObjectDefinition]
Name = Customer
Version = 1.0.0
AppSpecificInfo = cw_mo_conn=MyConfig

[Attribute]
Name = FirstName
Type = String
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = LastName
Type = String
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Telephone
Type = String
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false

```



```

[End]
[Attribute]
Name = MyConfig
Type = MO_Sample_Config
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

```

JMS 머리글, MQ Integrator Broker 메시지 등록 정보 및 동적 하위 Meta Object 속성

메시지 전송에 대한 자세한 정보와 더 많은 제어를 얻기 위해 동적 Meta Object에 속성을 추가할 수 있습니다. 이러한 속성을 추가하면 JMS 등록 정보를 수정하여 요청별로 ReplyToQueue를 제어하고(어댑터 등록 정보에 지정된 기본 ReplyToQueue를 사용하지 않고), 메시지 CorrelationID를 다시 대상으로 할 수 있습니다. 이 절에서는 이 속성과 이들이 어떻게 동기 모드와 비동기 모드 둘다에서 이벤트 공고 및 요청 처리에 영향을 미치는지에 대해 설명합니다.

JMS 및 MQ Integrator Broker 머리글 등록 정보를 반영하는 다음 속성들은 동적 Meta Object에서 인식됩니다.

표 14. 동적 Meta Object 머리글 속성

머리글 속성 이름	모드	해당 JMS 머리글
CorrelationID	읽기/쓰기	JMSCorrelationID
ReplyToQueue	읽기/쓰기	JMSReplyTo
DeliveryMode	읽기/쓰기	JMSDeliveryMode
Priority	읽기/쓰기	JMSPriority
Destination	읽기	JMSDestination
Expiration	읽기	JMSExpiration
MessageID	읽기	JMSMessageID
Redelivered	읽기	JMSRedelivered
TimeStamp	읽기	JMSTimeStamp
Type	읽기	JMSType
UserID	읽기	JMSXUserID
AppID	읽기	JMSXAppID
DeliveryCount	읽기	JMSXDeliveryCount
GroupID	읽기	JMSXGroupID
GroupSeq	읽기	JMSXGroupSeq
JMSProperties	읽기/쓰기	

읽기 전용 속성은 이벤트 공고 중에 메시지에서 읽어들이고 동적 Meta Object로 쓰이게 됩니다. 이 등록 정보는 요청을 처리하는 동안 응답 메시지가 발행될 때 동적 Meta Object를 채우기도 합니다. 읽기/쓰기 속성은 요청을 처리하는 동안 작성된 메시지 머리글에 설정되어 있습니다. 이벤트 공고 동안에 읽기/쓰기 속성은 메시지 머리글에서 읽어들이며 동적 Meta Object를 채웁니다.

이 속성의 해석 및 사용은 아래에 있는 절에 설명되어 있습니다.

주: 위에 있는 속성 중 어떠한 것도 필수가 아닙니다. 비즈니스 프로세스와 관련된 동적 Meta Object에 아무 속성이나 추가할 수 있습니다.

동기 이벤트 공고: 동기 이벤트 처리에서는 어댑터가 이벤트를 게시하고 응용프로그램으로 응답 메시지를 다시 전송하기 전에 통합 브로커의 응답을 기다립니다. 비즈니스 데이터에 대한 모든 변경사항이 리턴되는 응답 메시지에 반영됩니다. 어댑터가 이벤트를 게시하기 전에 비동기 이벤트 공고에 설명된 대로 동적 Meta Object를 채웁니다. 동적 Meta Object에 설정된 값은 아래에 설명된 대로 응답 발행 머리글에 반영됩니다.(동적 Meta Object에 있는 다른 모든 읽기 전용 머리글 속성은 무시됩니다.):

- **CorrelationID** 동적 Meta Object가 CorrelationID 속성을 포함하면, 그것을 기원 응용프로그램이 예상하는 값으로 설정해야 합니다. 응용프로그램이 CorrelationID를 사용하여 커넥터에서 리턴되는 메시지와 원래의 요청을 일치시킵니다.

CorrelationID의 예상하지 않거나 유효하지 않은 값은 문제점을 일으킵니다. 이 속성을 사용하기 전에 응용프로그램이 요청 메시지와 응답 메시지를 상관시키는 것을 처리하는 방법을 판별하는 것이 도움이 됩니다. 동기 요청에는 CorrelationID를 채우기 위한 네 가지 옵션이 있습니다.

1. 값을 변경하지 않고 그대로 둡니다. 응답 메시지의 CorrelationID는 요청 메시지의 CorrelationID와 동일합니다. 이것은 WebSphere MQ 옵션 MQRO_PASS_CORREL_ID와 동일합니다.
2. 값을 CxIgnore로 변경합니다. 커넥터는 기본적으로 요청의 메시지 ID를 응답의 CorrelationID로 복사합니다. 이것은 WebSphere MQ 옵션 MQRO_COPY_MSG_ID_TO_CORREL_ID와 동일합니다.
3. 값을 CxBlank로 변경합니다. 커넥터가 응답 메시지에서 CorrelationID를 설정하지 않습니다.
4. 값을 사용자 정의 값으로 변경합니다. 이렇게 하려면 응답을 처리하는 응용프로그램이 사용자 정의 값을 인식해야 합니다.

Meta Object에서 CorrelationID 속성을 정의하지 않으면, 커넥터가 자동으로 CorrelationID를 처리합니다.

- **ReplyToQueue** ReplyToQueue 속성에 다른 대기열을 지정하여 동적 Meta Object를 갱신하면, 커넥터가 지정하는 대기열로 응답 메시지를 전송합니다. 이것은 권장하지 않습니다. 요청 메시지에 특정 응답 대기열을 설정하는 응용프로그램은 해당 대기열에서 응답을 기다리고 있는 것으로 가정되기 때문에 커넥터가 다른 대기열로 응답 메시지를 전송하게 하면 통신에 방해가 될 수 있습니다.
- **JMS** 등록 정보 갱신된 Business Object가 커넥터로 리턴될 때 동적 Meta Object에 있는 JMS 등록 정보에 설정된 값이 응답 메시지에 설정됩니다.

비동기 요청 처리: 커넥터는 동적 Meta Object가 있으면 요청 메시지를 발행하기 전에 이를 사용하여 요청 메시지를 채웁니다. 커넥터가 요청 메시지를 전송하기 전에 다음 단계를 수행합니다.

1. CorrelationID 속성이 동적 Meta Object에 있으면, 커넥터가 아웃바운드 요청 메시지의 CorrelationID를 이 값으로 설정합니다.
2. ReplyToQueue 속성이 동적 Meta Object에 지정되어 있으면, 커넥터가 요청 메시지를 통해 이 대기열을 전달하고 이 대기열에서 응답을 기다립니다. 이렇게 하면 커넥터 구성 등록 정보에 지정된 ReplyToQueue 값을 대체할 수 있습니다. 음수의 ResponseTimeout(커넥터가 응답을 기다리지 않아야 한다는 것을 의미)을 추가로 지정하면, 커넥터가 실제로 응답을 기다리지 않아도 ReplyToQueue가 응답 메시지에 설정됩니다.

3. 속성 `DeliveryMode`를 2로 설정하면, 메시지가 계속 전송됩니다. 속성 `DeliveryMode`를 1로 설정하면, 메시지가 계속 전송되지 않습니다. 다른 값을 사용하면 커넥터가 실패합니다. MO에서 `DeliveryMode`를 지정하지 않은 경우, JMS 제공자는 지속성을 설정합니다.
4. 속성 `Priority`를 지정한 경우 커넥터는 전송하는 요청의 값을 설정합니다. `Priority` 속성의 가능한 값은 0 - 9입니다. 다른 값을 지정하려면 커넥터가 종료됩니다.
5. 동적 `Meta Object`에 `JMSProperties`가 지정되어 있으면, 하위 동적 `Meta Object`에 지정된 해당 JMS 등록 정보가 커넥터에서 전송하는 아웃바운드 메시지에 설정됩니다.

주: 동적 `Meta Object`에 있는 머리글 속성이 정의되어 있지 않거나 `CxIgnore`를 지정하면, 커넥터가 해당 기본 설정값을 따릅니다.

동기 요청 처리: 커넥터는 동적 `Meta Object`가 있으면 요청 메시지를 발행하기 전에 이를 사용하여 요청 메시지를 채웁니다. 동적 `Meta Object`에 머리글 속성이 있으면, 커넥터가 그것을 응답 메시지에 있는 새 값으로 채웁니다. 커넥터가 응답 메시지를 수신한 후에 다음 단계를 수행합니다.(`Meta Object`를 전송과 관련된 데이터로 채우는 것뿐만 아니라)

1. `CorrelationID` 속성이 동적 `Meta Object`에 있으면, 어댑터가 이 속성을 응답 메시지에 지정된 `JMSCorrelationID`로 갱신합니다.
2. `ReplyToQueue` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 응답 메시지에 지정된 `JMSReplyTo`의 이름으로 갱신합니다.
3. `DeliveryMode` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 메시지의 `JMSDeliveryMode` 머리글 필드 값으로 갱신합니다.
4. `Priority` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 메시지의 `JMSPriority` 머리글 필드 값으로 갱신합니다.
5. `Destination` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 응답 메시지에 지정된 `JMSDestination`의 이름으로 갱신합니다.
6. `Expiration` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 메시지의 `JMSExpiration` 머리글 필드 값으로 갱신합니다.
7. `MessageID` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 메시지의 `JMSMessageID` 머리글 필드 값으로 갱신합니다.
8. `Redelivered` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 메시지의 `JMSRedelivered` 머리글 필드 값으로 갱신합니다.
9. `TimeStamp` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 메시지의 `JMSTimeStamp` 머리글 필드 값으로 갱신합니다.
10. `Type` 속성이 동적 `Meta Object`에 정의되어 있으면, 어댑터가 이 속성을 메시지의 `JMSType` 머리글 필드 값으로 갱신합니다.

11. UserID 속성이 동적 Meta Object에 정의되어 있으면, 어댑터가 이 속성을 메시지의 JMSXUserID 머리글 필드 값으로 갱신합니다.
12. AppID 속성이 동적 Meta Object에 정의되어 있으면, 어댑터가 이 속성을 메시지의 JMSXAppID 머리글 필드 값으로 갱신합니다.
13. DeliveryCount 속성이 동적 Meta Object에 정의되어 있으면, 어댑터가 이 속성을 메시지의 JMSXDeliveryCount 머리글 필드 값으로 갱신합니다.
14. GroupID 속성이 동적 Meta Object에 정의되어 있으면, 어댑터가 이 속성을 메시지의 JMSXGroupID 머리글 필드 값으로 갱신합니다.
15. GroupSeq 속성이 동적 Meta Object에 정의되어 있으면, 어댑터가 이 속성을 메시지의 JMSXGroupSeq 머리글 필드 값으로 갱신합니다.
16. JMSProperties 속성이 동적 Meta Object에 정의되어 있으면, 어댑터가 하위 오브젝트에 정의된 어떠한 등록 정보든지 응답 메시지에 있는 값으로 갱신합니다. 하위 오브젝트에 정의된 등록 정보가 메시지에 존재하지 않으면, 값이 CxBlank로 설정되어 있습니다.

주: 동적 Meta Object를 사용하여 요청 메시지에 설정된 CorrelationID를 변경하는 것이 어댑터가 응답 메시지를 식별하는 방법에 영향을 미치지 않습니다. (어댑터는 기본적으로 모든 응답 메시지의 CorrelationID가 어댑터에 의해 전송된 요청의 메시지 ID와 같다고 예상합니다.)

오류 처리: JMS 등록 정보를 메시지에서 읽어들이고 메시지에 쓰일 수 없으면, 커넥터는 오류를 로그하고 요청 또는 이벤트가 실패합니다. 사용자가 지정한 ReplyToQueue가 존재하지 않거나 액세스되지 않으면, 커넥터가 오류를 로그하고 요청에 실패합니다. CorrelationID가 유효하지 않거나 설정될 수 없으면, 커넥터가 오류를 로그하고 요청에 실패합니다. 모든 경우에 로그된 메시지는 커넥터 메시지 파일에서 옵니다.

시작 파일 구성

Connector for WebSphere MQ Integrator를 시작하기 전에 시작 파일을 구성해야 합니다.

Windows

Windows 플랫폼에 맞게 커넥터 구성을 완료하려면

start_WebSphereMQIntegratorBroker.bat 파일을 수정해야 합니다.

1. start_WebSphereMQIntegratorBroker.bat 파일을 여십시오.
2. “Set the directory containing your WebSphereMQ Java client libraries”로 시작하는 섹션으로 이동한 후, WebSphere MQ Java 클라이언트 라이브러리의 위치를 지정하십시오.

UNIX

UNIX 플랫폼에 맞게 커넥터 구성을 완료하려면

`start_WebSphereMQIntegratorBroker.sh` 파일을 수정해야 합니다.

1. `start_start_WebSphereMQIntegratorBroker.sh` 파일을 여십시오.
2. “Set the directory containing your WebSphere MQ Java client libraries”로 시작하는 섹션으로 이동한 후, WebSphere MQ Java 클라이언트 라이브러리의 위치를 지정하십시오.

시작

커넥터 시작

커넥터는 해당 커넥터 시작 스크립트를 사용하여 명시적으로 시작해야 합니다. 시작 스크립트는 커넥터의 런타임 디렉토리에 있어야 합니다.

`ProductDir\connectors\connName`

여기서 `connName`은 커넥터를 식별합니다. 시작 스크립트의 이름은 표 15에 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.

표 15. 커넥터의 시작 스크립트

운영 체제	시작 스크립트
UNIX 기반 시스템	<code>connector_manager_connName</code>
Windows	<code>start_connName.bat</code>

다음 방법 중 하나를 사용하여 커넥터 시작 스크립트를 호출할 수 있습니다.

- Windows 시스템의 경우, 시작 메뉴에서 다음을 수행하십시오.

프로그램 > **IBM WebSphere Business Integration Adapters** > 어댑터 > 커넥터를 선택하십시오. 기본적으로 프로그램 이름은 “IBM WebSphere Business Integration Adapters”입니다. 그러나 이 이름을 사용자 정의할 수 있습니다. 대안으로, 커넥터에 대한 데스크탑 단축 아이콘을 작성할 수 있습니다.

- 명령행에서 커넥터를 시작할 경우에는 다음을 수행하십시오.

- Windows 시스템의 경우:

```
start_connName connName brokerName [-configFile ]
```

- UNIX 기반 시스템의 경우:

```
connector_manager_connName -start
```

여기서 `connName`은 커넥터의 이름이며 `brokerName`은 다음과 같이 통합 브로커를 식별합니다.

- WebSphere InterChange Server의 경우 `brokerName`에 ICS 인스턴스의 이름을 지정하십시오.

- WebSphere 메시지 브로커(WebSphere MQ Integrator, WebSphere MQ Integrator Broker 또는 WebSphere Business Integration Message Broker) 또는 WebSphere Application Server의 경우, *brokerName*에 브로커를 식별하는 문자열을 지정하십시오.

주: Windows 시스템의 WebSphere 메시지 브로커 또는 WebSphere Application Server에서는 -c 옵션 다음에 커넥터 구성 파일의 이름이 포함되어야 합니다. ICS의 경우 -c는 선택적입니다.

- System Manager를 시작할 때 실행되는 Adaptor Monitor(WebSphere Business Integration Adapters 제품 전용)에서 다음을 수행할 수 있습니다.
이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.
- System Monitor(WebSphere InterChange Server 제품 전용)에서 다음을 수행할 수 있습니다.
이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.
- Windows 시스템에서 Windows 서비스로 시작하도록 커넥터를 구성할 수 있습니다.
이 경우, Windows 시스템이 시동하거나(자동 서비스인 경우) Windows 서비스 창을 통해 서비스를 시작할 때(수동 서비스인 경우) 커넥터가 시작됩니다.

명령행 시작 옵션을 포함하여 커넥터를 시작하는 방법에 대한 자세한 정보는 다음 문서 중 하나를 참조하십시오.

- WebSphere InterChange Server에 대한 자세한 정보는 *System Administration Guide*를 참조하십시오.
- WebSphere Message Broker에 대한 자세한 정보는 *Implementing Adapters with WebSphere Message Brokers*를 참조하십시오.
- WebSphere Application Server에 대한 자세한 정보는 *Implementing Adapters with WebSphere Application Server*를 참조하십시오.

커넥터 중지

커넥터를 중지하는 방법은 다음과 같이 커넥터를 시작한 방법에 따라 다릅니다.

- 해당 커넥터 시작 스크립트를 사용하여 명령행에서 커넥터를 시작한 경우에는 다음을 수행하십시오.
 - Windows 시스템의 경우, 시작 스크립트를 호출하면 커넥터에 대한 별도의 “콘솔” 창이 작성됩니다. 이 창에서 “Q”를 입력한 다음 Enter를 눌러 커넥터를 중지하십시오.
 - UNIX 기반 시스템에서, 커넥터는 백그라운드에서 실행되므로 별도의 창이 없습니다. 대신 커넥터를 중지하려면 다음 명령을 실행하십시오.


```
connector_manager_connName -stop
```

여기서 *connName*은 커넥터의 이름입니다.

- System Manager를 시작할 때 실행되는 Adapt Monitor(WebSphere Business Integration Adapters 제품 전용)에서 다음을 수행할 수 있습니다.
이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.
- System Monitor(WebSphere InterChange Server 제품 전용)에서 다음을 수행할 수 있습니다.
이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.
- Windows 시스템에서 Windows 서비스로 시작하도록 커넥터를 구성할 수 있습니다.
이 경우, Windows 시스템이 종료되면 커넥터가 중지됩니다.

제 3 장 Business Object 작성 또는 수정

- 『커넥터 Business Object 구조』
- 49 페이지의 『오류 처리』
- 50 페이지의 『추적』

커넥터는 샘플 Business Object만을 제공합니다. 시스템 통합자, 컨설턴트 또는 고객은 Business Object를 빌드해야 합니다.

커넥터는 메타 데이터 구동 커넥터입니다. Business Object에서 메타 데이터는 Business Object 정의에 저장되고 커넥터가 응용프로그램과 상호작용하는 데 도움이 되는 응용 프로그램에 대한 데이터입니다. 메타 데이터 구동 커넥터는 커넥터에서 하드 코드된 명령어보다는 Business Object 정의로 인코드된 메타 데이터를 근거로 하여 지원하는 각 Business Object를 처리합니다.

Business Object 메타 데이터에는 Business Object의 구조, 해당 속성 등록 정보의 설정 및 응용프로그램 특정 정보의 내용이 포함됩니다. 커넥터는 메타 데이터 구동이므로, 커넥터 코드를 수정할 필요 없이 새로운 Business Object 또는 수정된 Business Object를 처리할 수 있습니다. 사실상, 커넥터는 MQ Integrator Broker 응용프로그램에 대한 수정에 의해 영향을 받지 않아야 합니다. 그러나 커넥터의 구성된 Data Handler는 Business Object의 구조, 오브젝트 카디널리티, 응용프로그램 특정 정보의 형식 및 Business Object의 데이터베이스 표시에 대한 가정을 작성합니다. 따라서 MQ Integrator Broker용 Business Object를 작성하거나 수정할 때, 사용자 수정이 커넥터가 따르도록 디자인된 규칙을 준수해야 하고, 그렇지 않으면 커넥터가 새로운 Business Object 또는 수정된 Business Object를 제대로 처리할 수 없습니다.

이 장에서는 커넥터가 Business Object를 처리하는 방식에 대해 설명하며 커넥터가 작성한 가정에 대해 설명합니다. 새 Business Object를 구현하는 지침으로 이 정보를 사용할 수 있습니다.

커넥터 Business Object 구조

커넥터 설치 이후에, MQ Integrator Broker용 Business Object를 작성해야 합니다. 구성된 Data Handler에 의해 부과된 요구사항 이외에 Business Object의 구조에 대한 요구사항은 없습니다. 커넥터가 처리하는 Business Object는 InterChange Server에서 허용하는 모든 이름을 가질 수 있습니다. 이름 지정 규칙에 대한 자세한 정보는 *Naming IBM WebSphere InterChange Server Components*를 참조하십시오.

커넥터는 대기열에서 메시지를 검색하여 메시지 내용으로 Business Object(Meta Object에서 정의됨)를 채우려고 시도합니다. 엄밀히 말해서, 커넥터는 Business Object 구조

를 제어하지도 Business Object 구조에 영향을 미치지도 않습니다. 이러한 사항이 커넥터의 Data Handler 요구사항과 마찬가지로 Meta Object 정의의 기능입니다. 실제로, Business Object 레벨 응용프로그램 정보는 없습니다. Business Object를 검색하고 전달할 때 커넥터의 기본 역할은 오류에 대한 메시지 대 Business Object(그리고 반대의 경우도 포함) 프로세스를 모니터링하는 것입니다.

샘플 Business Object 등록 정보

이 절에서는 이름-값 Data Handler를 사용한 커넥터의 샘플 Business Object 등록 정보에 대해 설명합니다.

```
[ReposCopy]
Version = 3.0.0
[End]
[BusinessObjectDefinition]
Name = Example_SimpleOrder
Version = 1.0.0
AppSpecificInfo = m_SimpleOrder

[Attribute]
Name = ProductCode
Type = String
Cardinality = 1
MaxLength = 3
IsKey = true
IsForeignKey = false
IsRequired = true
AppSpecificInfo = e_ProductCode;type=pcdata;
DefaultValue = W02
IsRequiredServerBound = false
[End]

[Attribute]
Name = ProductDescription
Type = String
Cardinality = 1
MaxLength = 20
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = e_ProductDescription;type=pcdata;
DefaultValue = Deluxe Widget
IsRequiredServerBound = false
[End]
[Attribute]
Name = ProductQuantity
Type = Integer
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = true
AppSpecificInfo = e_ProductQuantity;type=pcdata;
DefaultValue = 1
IsRequiredServerBound = false
[End]

[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
```

```

IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

[BusinessObjectDefinition]
Name = Example_SimpleOrder_MRM
Version = 1.0.0
AppSpecificInfo = MRM

[Attribute]
Name = xmlns
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = xmlns;type=attribute
DefaultValue = www.mrmnames.net/DIN94VC078001
IsRequiredServerBound = false
[End]
[Attribute]
Name = SimpleOrder
Type = Example_SimpleOrder
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = false
AppSpecificInfo = m_SimpleOrder
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create

```

```

[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

[BusinessObjectDefinition]
Name = Example_SimpleOrder_XMLDoc
Version = 1.0.0

[Attribute]
Name = XMLDeclaration
Type = String
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
AppSpecificInfo = type=pi;
DefaultValue = xml version="1.0"
IsRequiredServerBound = false
[End]

[Attribute]
Name = Doctype
Type = String
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
AppSpecificInfo = type=doctype;
DefaultValue = DOCTYPE MRM PUBLIC "www.mrmnames.net/DIN94VC078001" "DIN94VC078001"
IsRequiredServerBound = false
[End]

[Attribute]
Name = MRM_Wrapper
Type = Example_SimpleOrder_MRM
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = true
AppSpecificInfo = MRM
IsRequiredServerBound = false
[End]

[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

```

```

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

```

오류 처리

커넥터에서 생성한 모든 오류 메시지는 MQSIV2Connector.txt라는 메시지 파일에 저장됩니다. (파일의 이름은 LogFileName 표준 커넥터 구성 등록 정보로 판별됩니다.) 각 오류의 오류 번호 다음에는 오류 메시지가 있습니다.

```

Message number
Message text

```

커넥터는 다음 절에서 설명한 대로 특정 오류를 처리합니다.

응용프로그램 제한시간

다음 경우에 오류 메시지 ABON_APPRESPONSETIMEOUT이 리턴됩니다.

- 커넥터가 메시지 검색 중 JMS 서비스 제공자에 대한 연결을 설정할 수 없는 경우
- 커넥터가 정상적으로 Business Object를 메시지로 변환하지만 연결이 끊어져서 전송 대기열로 전달할 수 없는 경우
- 커넥터가 메시지를 발행하지만 변환 등록 정보 TimeoutFatal이 True인 Business Object에 대한 응답을 대기 중 시간이 제한된 경우
- 커넥터가 APP_RESPONSE_TIMEOUT 또는 UNABLE_TO_LOGIN과 같은 리턴 코드로 응답 메시지를 수신한 경우

등록되지 않은 Business Object

다음의 경우에 커넥터는 UnsubscribedQueue 등록 정보에서 지정한 대기열로 메시지를 전달합니다.

- 커넥터에서 등록되지 않은 Business Object와 연관된 메시지를 검색하거나, NO_SUBSCRIPTION_FOUND 코드가 getAppEvent() 메소드에 의해 리턴되었을 때.
- 커넥터는 메시지를 검색하지만 FORMAT 필드의 텍스트를 Business Object 이름과 연관시킬 수 없을 경우

주: UnsubscribedQueue가 정의되지 않은 경우, 등록되지 않은 메시지는 버립니다.

활성화하지 않은 커넥터

gotAppEvent() 메소드가 CONNECTOR_NOT_ACTIVE 코드를 리턴할 때, pollForEvents() 메소드는 APP_RESPONSE_TIMEOUT 코드를 리턴하며 이벤트는 InProgress 대기열에 남아 있게 됩니다.

Data handler 변환

Data Handler가 메시지를 Business Object로 변환하는 데 실패하거나 Business Object에 특정한 처리 오류가 발생한 경우(JMS 제공자와 반대로), 메시지는 ErrorQueue에서 지정한 대기열로 전달됩니다. ErrorQueue가 정의되지 않은 경우, 오류 때문에 처리될 수 없는 메시지는 버립니다.

Data Handler가 Business Object를 메시지로 변환하는 데 실패한 경우, BON_FAIL이 리턴됩니다.

추적

추적은 커넥터 작동을 엄밀하게 따르도록 설정할 수 있는 선택적 디버깅 기능입니다. 기본적으로 추적 메시지는 STDOUT에 쓰여집니다. 추적 메시지 구성에 대한 자세한 정보는 제2장의 커넥터 구성 등록 정보를 참조하십시오. 추적을 사용 가능하게 하고 설정하는 방법을 포함하여 추적에 대한 자세한 정보는 *Connector Development Guide for Java*를 참조하십시오.

다음에 오는 내용은 커넥터 추적 메시지에 대한 권장 내용입니다.

- 레벨 0 이 레벨은 커넥터 버전을 식별하는 추적 메시지로 사용됩니다.
- 레벨 1 폴링 스레드가 입력 대기열에서 새 메시지를 발견할 때마다 기록 또는 처리되는 각 Business Object에 대한 주요 정보를 제공하는 추적 메시지로 이 레벨을 사용하십시오.
- 레벨 2 gotAppEvent() 또는 executeCollaboration() 중 하나에서 Business Object가 InterChange Server로 게시될 때마다 로그하는 추적 메시지로 이 레벨을 사용하십시오.
- 레벨 3 데이터 형식(예: XML) 대 Business Object 및 Business Object 대 데이터 형식 변환에 대한 정보를 제공하거나 출력 대기열로의 메시지 전달에 대한 정보를 제공하는 추적 메시지로 이 레벨을 사용하십시오.
- 레벨 4 커넥터가 함수를 입력하거나 종료할 때 식별하는 추적 메시지로 이 레벨을 사용하십시오.
- 레벨 5 커넥터 초기화를 표시하거나, 응용프로그램에서 실행되는 명령문을 표

시하거나, 메시지를 대기열에서 제거하거나 대기열로 넣거나 또는 Business Object 덤프를 기록할 때마다 표시하는 추적 메시징으로 이 레벨을 사용하십시오.

제 4 장 문제점 해결

이 장에서는 커넥터 시작 또는 실행 시 발생할 수 있는 문제점에 대해 설명합니다.

시작 시 문제점

문제점

커넥터가 초기화 중 갑자기 종료되고 다음 메시지가 보고되었습니다.
Exception in thread "main" java.lang.
NoClassDefFoundError: javax/jms/JMSEException...

커넥터가 초기화 중 갑자기 종료되고 다음 메시지가 보고되었습니다.
Exception in thread "main" java.lang.
NoClassDefFoundError: com/ibm/mq/jms/
MQConnectionFactory...

커넥터가 초기화 중 갑자기 종료되고 다음 메시지가 보고되었습니다.
Exception in thread "main" java.lang.
NoClassDefFoundError: javax/naming/Referenceable...

커넥터가 초기화 중 갑자기 종료되고 다음 예외가 보고되었습니다.
java.lang.UnsatisfiedLinkError: no mqjbd01 in shared library
path

커넥터에서 MQJMS2005: failed to create MQQueueManager for ':'
메시지를 보고합니다.

잠재적 솔루션 / 설명

커넥터가 IBM MQSeries Java 클라이언트 라이브러리에서
jms.jar 파일을 찾을 수 없습니다. start_connector.bat
파일의 MQSERIES_JAVA_LIB 변수가 IBM MQSeries Java
클라이언트 라이브러리 폴더를 가리키는지 확인하십시오.
커넥터가 IBM MQSeries Java 클라이언트 라이브러리에서
com.ibm.mqjms.jar 파일을 찾을 수 없습니다.
start_connector.bat 파일의 MQSERIES_JAVA_LIB 변수
가 IBM MQSeries Java 클라이언트 라이브러리 폴더를 가
리키는지 확인하십시오.

커넥터가 IBM MQSeries Java 클라이언트 라이브러리에서
jndi.jar 파일을 찾을 수 없습니다. start_connector.bat
파일의 MQSERIES_JAVA_LIB 변수가 IBM MQSeries Java
클라이언트 라이브러리 폴더를 가리키는지 확인하십시오.
커넥터가 필수 런타임 라이브러리(mqjbd01.dll [Windows]
또는 libmqjbd01.so [UNIX])를 IBM MQSeries Java 클
라이언트 라이브러리에서 찾을 수 없습니다. 사용자 경로가
IBM MQSeries Java 클라이언트 라이브러리 폴더를 포함하
는지 확인하십시오.

명시적으로 HostName, Channel 및 Port 등록 정보에 대한
값을 설정합니다.

이벤트 처리

문제점

커넥터는 MQRFH2 머리글로 모든 메시지를 전달합
니다.

커넥터 Meta Object에 정의된 형식에 관계없이 전달
시 커넥터는 모든 메시지 형식을 8자로 자릅니다.

잠재적 솔루션 / 설명

MQMD WebSphere MQ 머리글로만 메시지를 전달하려면 ?targetClient=1
을 출력 대기열 URI의 이름에 추가하십시오. 예를 들어, 메시지를 대기열
queue://my.queue.manager/OUT으로 출력하려면 URI를 queue://my.queue.
manager/OUT?targetClient=1로 변경하십시오. 자세한 정보는 17 페이지의 제
2 장 『커넥터 설치 및 구성』을 참조하십시오.

이는 커넥터가 아닌 WebSphere MQ MQMD 메시지 머리글의 한계입니다.

문제점

pollForEvents 중에, 다음 JMS 예외가 보고된 후에 커넥터가 종료됩니다.

MQJMS1000: Failed to create JMS message

pollForEvents 중에, 다음 JMS 예외가 보고된 후에 커넥터가 종료됩니다.

MQJMS1052: Unrecognised [sic] JMS Message class

잠재적 솔루션 / 설명

이 오류는 커넥터 자체가 아닌 MQSeries Java API와 관련되어 나타납니다. WebSphere MQ Integrator Broker 제품 자체와 동일한 시스템에서 커넥터를 실행할 때 종종 발생합니다. 문제점을 해결하려면, 다음을 수행해야 합니다.

1. %MQSIV2_INSTALL_FOLDER%\Dependencies\JRE_122_4.zip 파일을 %CROSSWORLDS%\connectors\MQSIV2\Dependencies\jre_122_Re14. 폴더에 압축을 해제하십시오.
2. %CROSSWORLDS%\connectors\MQSIV2\start_MQSIV2.bat를 열고 다음 두 행의 주석을 해제하십시오.
oset PATH=%CONNDIR%\Dependencies\jre_122_Re14\bin...
oset JAVA=%CONNDIR%\Dependencies\jre_122_Re14\lib\rt.jar
3. 커넥터를 다시 시작하십시오.

이 오류는 JMS 준수 응용프로그램에서 기원하는 메시지에 대해 WebSphere MQ Integrator Broker가 작성한 수정 결과일 수 있습니다. 이를 정정하려면, 다음과 같은 SQL 문을 WebSphere MQ Integrator Broker 계산 노드에 추가하여 문제가 되는 메시지에서 나머지 JMS 정보를 제거하십시오.

```
SET OutputRoot.MQRFH2.jms = null;
```

부록 A. 커넥터의 표준 구성 등록 정보

이 부록에서는 WebSphere Business Integration Adapter의 커넥터 구성요소에 대한 표준 구성 등록 정보에 대해합니다. 다음 통합 브로커에서 실행되는 커넥터에 대한 정보를 다룹니다.

- WebSphere InterChange Server(ICS)
- WebSphere MQ Integrator, WebSphere MQ Integrator Broker 및 WebSphere Business Integration Message Broker를 통털어 WMQI(WebSphere Message Brokers)라고 합니다.
- WebSphere Application Server(WAS)

모든 커넥터가 이들 표준 등록 정보를 전부 사용하는 것은 아닙니다. Connector Configurator에서 통합 브로커를 선택할 때, 이 브로커에서 실행되는 어댑터에 구성해야 하는 표준 등록 정보의 목록이 표시됩니다.

커넥터 특정 등록 정보에 대한 정보는 관련 어댑터 사용자 안내서를 참조하십시오.

주: 이 책에서 백슬래시(\)는 디렉토리 경로의 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체하고 각 운영 체제의 규칙을 따르십시오.

신규 및 삭제된 등록 정보

다음 표준 등록 정보가 이 릴리스에 추가되었습니다.

새 등록 정보

- XMLNameSpaceFormat

삭제된 등록 정보

- RestartCount
- RHF2MessageDomain

표준 커넥터 등록 정보 구성

어댑터 커넥터에는 두 가지 유형의 구성 등록 정보가 있습니다.

- 표준 구성 등록 정보
- 커넥터 특정 구성 등록 정보

이 절에서는 표준 구성 등록 정보에 대해 설명합니다. 커넥터 특정 구성 등록 정보에 대한 정보는 해당 어댑터 사용자 안내서를 참조하십시오.

Connector Configurator 사용

System Manager에서 액세스하는 Connector Configurator에서 커넥터 등록 정보를 구성합니다. Connector Configurator 사용에 대한 자세한 정보는 Connector Configurator 부록을 참조하십시오.

주: Connector Configurator와 System Manager는 Windows 시스템에서만 실행됩니다. UNIX 시스템에서 커넥터를 실행 중인 경우, Windows 시스템에 이러한 도구가 설치되어 있어야 합니다. UNIX에서 실행하는 커넥터에 커넥터 등록 정보를 설정하려면, Windows 시스템에서 System Manager를 시작하고, UNIX 통합 브로커에 연결한 후 커넥터용 Connector Configurator를 가져와야 합니다.

등록 정보 값 설정 및 갱신

등록 정보 필드의 기본 길이는 255자입니다.

커넥터는 다음 순서를 사용하여 등록 정보의 값을 판별합니다(가장 높은 숫자가 다른 값을 대체함).

1. 기본값
2. 저장소(WebSphere InterChange Server가 통합 브로커일 경우에만)
3. 로컬 구성 파일
4. 명령행

커넥터는 시작할 때 구성값을 확보합니다. 런타임 세션 중에 하나 이상의 커넥터 등록 정보 값을 변경하면, 등록 정보의 갱신 메소드가 변경사항의 적용 방법을 판별합니다. 표준 커넥터 등록 정보에 대한 갱신 메소드에는 네 가지가 있습니다.

- 동적
System Manager에 저장된 후 변경사항이 즉시 적용됩니다. 커넥터가 WebSphere Message Broker와 같은 독립형 모드에서 작동 중이면(System Manager와 무관하게), 구성 파일을 통해서만 등록 정보를 변경할 수 있습니다. 이 경우에는 동적 갱신이 가능하지 않습니다.
- 구성요소 다시 시작
커넥터가 중지된 다음 System Manager에서 다시 시작된 후에만 변경사항이 적용됩니다. 응용프로그램 특정 구성요소나 통합 브로커를 중지하고 다시 시작할 필요는 없습니다.
- 서버 다시 시작
응용프로그램 특정 구성요소와 통합 브로커를 중지하고 다시 시작한 후에만 변경사항이 적용됩니다.
- 에이전트 다시 시작(ICS에만 해당)
응용프로그램 특정 구성요소를 중지하고 다시 시작한 후에만 변경사항이 적용됩니다.

특정 등록 정보의 갱신 방법을 결정하려면, Connector Configurator 창의 갱신 메소드 열이나 아래 등록 정보 요약 테이블의 갱신 메소드 열을 참조하십시오.

표준 등록 정보 요약

표 16은 표준 커넥터 구성 등록 정보에 대한 빠른 참조를 제공합니다. 모든 커넥터가 다음 등록 정보를 모두 사용하는 것은 아니며 표준 등록 정보 종속성은 RepositoryDirectory를 기본으로 하기 때문에 등록 정보 설정은 통합 브로커에 따라 달라질 수 있습니다.

커넥터를 실행하기 전에 이러한 등록 정보 중 일부의 값을 설정해야 합니다. 각 등록 정보에 대한 설명을 보려면 다음 절을 참조하십시오.

표 16. 표준 구성 등록 정보 요약

등록 정보 이름	가능한 값	기본값	갱신 메소드	참고
AdminInQueue	올바른 JMS 대기열 이름	CONNECTORNAME /ADMININQUEUE	구성요소 다시 시작	전달 전송이 JMS임
AdminOutQueue	올바른 JMS 대기열 이름	CONNECTORNAME/ADMINOUTQUEUE	구성요소 다시 시작	전달 전송이 JMS임
AgentConnections	1 - 4	1	구성요소 다시 시작	전달 전송이 MQ 또는 IDL인 경우: 저장소 디렉토리가 <REMOTE>임
AgentTraceLevel	0 - 5	0	동적	
ApplicationName	응용프로그램 이름	커넥터 응용프로그램 이름에 대해 지정된 값	구성요소 다시 시작	
BrokerType	ICS, WMQI, WAS			
CharacterEncoding	ascii7, ascii8, SJIS, Cp949, GBK, Big5, Cp297, Cp273, Cp280, Cp284, Cp037, Cp437 주: 이것은 지원되는 값의 서브 세트입니다.	ascii7	구성요소 다시 시작	
ConcurrentEventTriggeredFlows	1 - 32,767	1	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>임
ContainerManagedEvents	값이 없음 또는 JMS	값이 없음	구성요소 다시 시작	전달 전송이 JMS임
ControllerStoreAndForwardMode	true 또는 false	True	동적	저장소 디렉토리가 <REMOTE>임
ControllerTraceLevel	0 - 5	0	동적	저장소 디렉토리가 <REMOTE>임
DeliveryQueue		CONNECTORNAME/DELIVERYQUEUE	구성요소 다시 시작	JMS 전송만
DeliveryTransport	MQ, IDL 또는 JMS	JMS	구성요소 다시 시작	저장소 디렉토리가 로컬이면 값은 JMS 뿐임

표 16. 표준 구성 등록 정보 요약 (계속)

등록 정보 이름	가능한 값	기본값	갱신 메소드	참고
DuplicateEventElimination	True 또는 False	False	구성요소 다시 시작	JMS 전송만: Container Managed Events가 <NONE> 이어야 함
FaultQueue		CONNECTORNAME/FAULTQUEUE	구성요소 다시 시작	JMS 전송만
jms.FactoryClassName	CxCommon.Messaging.jms .IBMMQSeriesFactory 또는 CxCommon.Messaging .jms.SonicMQFactory 또는 Java 클래스 이름	CxCommon.Messaging.jms. IBMMQSeriesFactory	구성요소 다시 시작	JMS 전송만
jms.MessageBrokerName	actoryClassName이 IBM일 경우, crossworlds.queue. manager를 사용하십시오. FactoryClassName 이 Sonic이면 localhost:2506을 사용하십 시오.	crossworlds.queue.manager	구성요소 다시 시작	JMS 전송만
jms.NumConcurrentRequests	양의 정수	10	구성요소 다시 시작	JMS 전송만
jms.Password	올바른 암호		구성요소 다시 시작	JMS 전송만
jms.UserName	올바른 이름		구성요소 다시 시작	JMS 전송만
JvmMaxHeapSize	힙 크기(MB)	128M	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>임
JvmMaxNativeStackSize	스택 크기(KB)	128K	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>임
JvmMinHeapSize	힙 크기(MB)	1M	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>임
ListenerConcurrency	1 - 100	1	구성요소 다시 시작	전달 전송이 MQ여야 함
Locale	en_US, ja_JP, ko_KR, zh_CN, zh_TW, fr_FR, de_DE, it_IT, es_ES, pt_BR 주: 이것은 지원되는 로케일의 서브세트입니다.	en_US	구성요소 다시 시작	
LogAtInterchangeEnd	True 또는 False	False	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>여야 함
MaxEventCapacity	1 - 2147483647	2147483647	동적	저장소 디렉토리가 <REMOTE>여야 함
MessageFileName	경로 또는 파일 이름	InterchangeSystem.txt	구성요소 다시 시작	
MonitorQueue	올바른 대기열 이름	CONNECTORNAME/MONITORQUEUE	구성요소 다시 시작	JMS 전송만: DuplicateEvent Elimination이 True 이어야 함

표 16. 표준 구성 등록 정보 요약 (계속)

등록 정보 이름	가능한 값	기본값	갱신 메소드	참고
OADAutoRestartAgent	True 또는 False	False	동적	저장소 디렉토리가 <REMOTE>여야 함
OADMaxNumRetry	양수	1000	동적	저장소 디렉토리가 <REMOTE>여야 함
OADRetryTimeInterval	분 단위의 양수 값	10	동적	저장소 디렉토리가 <REMOTE>여야 함
PollEndTime	HH:MM	HH:MM	구성요소 다시 시작	
PollFrequency	밀리초 단위의 양수 no(폴링을 사용 불가능하게 할 경우), key(문자 p가 카백터 명령 창에 입력될 때 폴하기 위해)	10000	동적	
PollQuantity	1 - 500	1	에이전트 다시 시작	JMS 전송만: 컨테이너이고 관리 이벤트가 지정되는 경우에만
PollStartTime	HH:MM(HH는 0 - 23, MM은 0 - 59)	HH:MM	구성요소 다시 시작	
RepositoryDirectory	메타 데이터 저장소의 위치		에이전트 다시 시작	ICS의 경우: <REMOTE> WebSphere MQ 메 시지 브로커 및 WAS의 경우: C:\crossworlds\ repository로 설정됨
RequestQueue	올바른 JMS 대기열 이름	CONNECTORNAME/REQUESTQUEUE	구성요소 다시 시작	전달 전송이 JMS임
ResponseQueue	올바른 JMS 대기열 이름	CONNECTORNAME/RESPONSEQUEUE	구성요소 다시 시작	전달 전송이 JMS: 저장소 디렉토리가 <REMOTE>인 경우 에만 필수
RestartRetryCount	0 - 99	3	동적	
RestartRetryInterval	분별 가능한 값(분): 1 - 2147483547	1	동적	
SourceQueue	올바른 WebSphere MQ 이름	CONNECTORNAME/SOURCEQUEUE	에이전트 다시 시작	전달 전송이 JMS 및 컨테이너이고 관 리 이벤트가 지정되 는 경우에만
SynchronousRequestQueue		CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	구성요소 다시 시작	전달 전송이 JMS임
SynchronousRequestTimeout	0 - 임의의 수(밀리초)	0	구성요소 다시 시작	전달 전송이 JMS임

표 16. 표준 구성 등록 정보 요약 (계속)

등록 정보 이름	가능한 값	기본값	갱신 메소드	참고
SynchronousResponseQueue		CONNECTORNAME / SYNCHRONOUSRESPONSEQUEUE	구성요소 다시 시작	전달 전송이 JMS임
WireFormat	CwXML, CwBO	CwXML	에이전트 다시 시작	저장소 디렉토리가 <REMOTE>가 아 닐 경우에는 CwXML: 저장소 디렉토리가 <REMOTE>인 경우 에는 CwBO
WsifSynchronousRequest 제한시간	0 - 임의의 숫자 (밀리초)	0	구성요소 다시 시작	WAS만 해당됨
XMLNameSpaceFormat	short, long	short	에이전트 다시 시작	WebSphere MQ 메 시지 브로커 및 WAS만 해당됨

표준 구성 등록 정보

이 절에서는 표준 커넥터 구성 등록 정보를 나열하고 각각을 정의합니다.

AdminInQueue

관리 메시지를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMININQUEUE입니다.

AdminOutQueue

관리 메시지를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMINOUTQUEUE입니다.

AgentConnections

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

AgentConnections 등록 정보는 orb.init[]에 의해 열려 있는 ORB 연결 수를 제어합니다.

기본적으로, 이 등록 정보의 값은 1로 설정되어 있으며 이 기본값을 변경할 필요가 없습니다.

AgentTraceLevel

응용프로그램 특정 구성요소의 추적 메시지 레벨입니다. 기본값은 0입니다. 구성요소는 설정된 추적 레벨이나 그 이하의 레벨에서 적용 가능한 모든 추적 메시지를 전달합니다.

ApplicationName

커넥터의 응용프로그램을 고유하게 식별하는 이름입니다. System Manager는 이 이름을 사용하여 WebSphere Business Integration System 환경을 모니터링합니다. 이 등록 정보는 사용자가 커넥터를 실행하기 전에 값을 가지고 있어야 합니다.

BrokerType

사용 중인 통합 브로커 유형을 식별합니다. 옵션에는 ICS, WebSphere Message Broker(WMQI, WMQIB 또는 WBIMB) 또는 WAS가 있습니다.

CharacterEncoding

문자(예: 영문자, 숫자 표시 또는 구두점 표시)에서 숫자값으로 맵핑하는 데 사용되는 문자 코드 세트를 지정합니다.

주: Java 기반 커넥터에서는 이 등록 정보를 사용하지 않습니다. C++ 커넥터는 현재 이 등록 정보에 값 `ascii7`을 사용합니다.

기본적으로 지원되는 문자 인코딩의 서브세트만 드롭 목록에 표시됩니다. 기타 지원되는 값을 드롭 목록에 추가하려면, 제품 디렉토리에 있는 `\Data\Std\stdConnProps.xml` 파일을 직접 수정해야 합니다. 자세한 정보는 Connector Configurator의 부록을 참조하십시오.

ConcurrentEventTriggeredFlows

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

이벤트 전달을 위해 커넥터에서 동시에 처리할 수 있는 Business Object 수를 판별합니다. 이 속성값을 동시에 맵핑하여 전달할 Business Object 수로 설정하십시오. 예를 들어, 5개의 Business Object가 동시에 처리되도록 하려면 이 등록 정보의 값을 5로 설정하십시오. 기본값은 1입니다.

이 등록 정보를 1보다 큰 값으로 설정하면, 소스 응용프로그램의 커넥터가 동시에 여러 이벤트 Business Object를 맵핑하고 이들을 여러 협업 인스턴스에 동시에 전달할 수 있습니다. 특히 Business Object가 복잡한 맵을 사용하는 경우, 통합 브로커로 Business Object를 전달하는 속도가 빨라집니다. 협업에 대한 Business Object 도착률이 증가하면 시스템에서 전체 성능이 향상될 수 있습니다.

전체 플로우에 대한 동시 처리를 구현하려면(소스 응용프로그램에서 목적지 응용프로그램으로) 다음을 수행해야 합니다.

- 동시 이벤트 최대 수 등록 정보를 다중 스레드를 사용할 만큼 충분히 설정하여, 다중 스레드를 사용하도록 협업을 구성하십시오.
- 목적지 응용프로그램의 응용프로그램 특정 구성요소가 요청을 동시에 처리할 수 있는지 확인하십시오. 즉, 이 구성요소가 다중 스레드 방식이거나 Connector Agent

Parallelism을 사용할 수 있고 다중 프로세스에 맞게 구성되어 있어야 합니다.
Parallel Process Degree 구성 등록 정보를 1보다 큰 값으로 설정하십시오.

ConcurrentEventTriggeredFlows 등록 정보는 단일 스레드이고 순차적으로 수행되는 커넥터 폴링에 영향을 주지 않습니다.

ContainerManagedEvents

이 등록 정보는 JMS 이벤트 저장소가 있는 JMS 사용 커넥터에서 보증된 이벤트 전달을 제공하게 하며, 여기에서 이벤트가 소스 대기열에서 제거되고 단일 JMS 트랜잭션으로 목적지 대기열에 위치합니다.

기본값은 값 없음입니다.

ContainerManagedEvents가 JMS로 설정되면, 보증된 이벤트 전달이 가능하도록 다음 등록 정보를 구성해야 합니다.

- PollQuantity = 1 ~ 500
- SourceQueue = CONNECTORNAME/SOURCEQUEUE

또한 MimeType, DHClass 및 DataHandlerConfigMOName(선택적) 등록 정보로 데이터 핸들러를 구성해야 합니다. 이러한 값을 설정하려면 Connector Configurator의 **Data Handler** 탭을 사용하십시오. Data Handler 탭 아래의 값의 필드는 ContainerManagedEvents를 JMS로 설정한 경우에만 표시됩니다.

주: ContainerManagedEvents가 JMS로 설정되면, 커넥터는 pollForEvents() 메소드를 호출하지 않으므로 해당 메소드의 기능을 사용할 수 없습니다.

이 등록 정보는 DeliveryTransport 등록 정보가 JMS 값으로 설정되어 있는 경우에만 나타납니다.

ControllerStoreAndForwardMode

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

목적지 응용프로그램 특정 구성요소가 사용 불가능함을 발견한 후 커넥터 제어기의 작동을 설정합니다.

이 등록 정보가 true로 설정되고 이벤트가 ICS에 도달할 때 목적지 응용프로그램 특정 구성요소가 사용 불가능한 경우, 커넥터 제어기는 응용프로그램 특정 구성요소에 대한 요청을 차단합니다. 응용프로그램 특정 구성요소가 작동하게 되면, 제어기는 요청을 전달합니다.

그러나 커넥터 제어기가 서비스 호출 요청을 전달한 후 목적지 응용프로그램의 응용프로그램 특정 구성요소가 사용 불가능하게 되면, 커넥터 제어기가 요청에 실패합니다.

이 등록 정보가 false로 설정된 경우, 커넥터 제어기는 목적지 응용프로그램 특정 구성요소가 사용 불가능함을 발견하는 즉시 모든 서비스 호출 요청에 실패하기 시작합니다.

기본값은 true입니다.

ControllerTraceLevel

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

커넥터 제어기의 추적 메시지 레벨입니다. 기본값은 0입니다.

DeliveryQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

Business Object를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/DELIVERYQUEUE입니다.

DeliveryTransport

이벤트 전달에 대한 전송 메커니즘을 지정합니다. 가능한 값은 WebSphere MQ의 경우 MQ, CORBA IIOP의 경우 IDL 또는 Java Messaging Service의 경우 JMS입니다.

- 브로커 유형이 ICS인 경우, DeliveryTransport 등록 정보의 값은 MQ, IDL 또는 JMS이며, 기본값은 IDL입니다.
- RepositoryDirectory가 로컬 디렉토리이면 JMS만이 값이 될 수 있습니다.

DeliveryTransport 등록 정보에 구성된 값이 MQ 또는 IDL일 경우, 커넥터는 CORBA IIOP를 통해 서비스 호출 요청과 관리 메시지를 전송합니다.

WebSphere MQ 및 IDL

한 제품만을 사용해야 하는 경우가 아니면, 이벤트 전달 전송에 IDL 대신 WebSphere MQ를 사용하십시오. WebSphere MQ는 IDL을 통해 다음과 같은 장점을 제공합니다.

- 비동기 통신:
WebSphere MQ는 서버를 사용할 수 없는 경우에도 응용프로그램 특정 구성요소가 이벤트를 폴링하여 지속적으로 저장할 수 있게 합니다.
- 서버측 성능:
WebSphere MQ는 서버측에서 더 빠른 성능을 제공합니다. 최적화된 모드에서, WebSphere MQ는 실제 이벤트가 WebSphere MQ 대기열에 남아 있는 반면 저장소 데이터베이스에는 이벤트에 대한 포인터만 저장합니다. 이로 인해 잠재적으로 큰 이벤트를 저장소 데이터베이스에 기록하지 않아도 됩니다.
- 에이전트측 성능:
WebSphere MQ는 응용프로그램 특정 구성요소측에서 더 빠른 성능을 제공합니다.

커넥터의 폴링 스레드가 WebSphere MQ를 사용하여 이벤트를 선택하고, 이를 커넥터의 대기열에 넣은 후 다음 이벤트를 선택합니다. 커넥터의 폴링 스레드가 이벤트를 선택하고 네트워크에서 서버 프로세스로 이동하며 지속적으로 이벤트를 저장소 데이터베이스에 저장한 후 다음 이벤트를 선택해야 하는 IDL보다 더 빠릅니다.

JMS

JMS(Java Messaging Service)를 사용하여 커넥터와 클라이언트 커넥터 프레임워크 간의 통신을 사용 가능하게 합니다.

전달 전송으로 JMS를 선택하면 추가 JMS 등록 정보인 `jms.MessageBrokerName`, `jms.FactoryClassName`, `jms.Password` 및 `jms.UserName`이 Connector Configurator에 표시됩니다. 이들 등록 정보 중 처음 두 가지는 이 전송에 필요합니다.

중요: 다음 환경에서 JMS 전송 메커니즘을 커넥터에 사용하는 경우에는 메모리 제한이 있을 수 있습니다.

- AIX 5.0
- WebSphere MQ 5.3.0.1
- ICS가 통합 브로커일 때

이 환경에서는 WebSphere MQ 클라이언트 내의 메모리 사용으로 인해 커넥터 제어기(서버측) 및 커넥터(클라이언트측) 모두를 시작하는 것이 어려울 수 있습니다. 설치 시 768M 미만의 프로세스 힙 크기를 사용하는 경우, IBM은 다음의 설정을 권장합니다.

- CWSHaredEnv.sh 스크립트의 LDR_CNTRL 환경 변수.

이 스크립트는 제품 디렉토리 아래의 `\bin` 디렉토리에 상주합니다. 텍스트 편집기를 사용하여 다음 행을 CWSHaredEnv.sh 스크립트의 첫 번째 행으로 추가하십시오.

```
export LDR_CNTRL=MAXDATA=0x30000000
```

이 행은 힙 메모리 사용을 최대 768MB(3세그먼트 * 256MB)로 제한합니다. 프로세스 메모리가 이 한계 이상으로 증가하면 페이지 스와핑이 발생할 수 있습니다. 이는 시스템의 성능을 저하시킬 수 있습니다.

- IPCCBaseAddress 등록 정보를 값 11 또는 12로 설정. 이 등록 정보에 대한 자세한 정보는 *UNIX용 시스템 설치 안내서*를 참조하십시오.

DuplicateEventElimination

이 등록 정보를 true로 설정할 때, JMS 사용 커넥터는 중복된 이벤트가 전달 대기열로 전달되지 않게 보장할 수 있습니다. 이 기능을 사용하려면 응용프로그램 특정 코드에서 Business Object의 **ObjectEventId** 속성으로 설정된 커넥터에 고유한 이벤트 ID가 있어야 합니다. 이는 커넥터 개발 중에 수행됩니다.

또한 이 등록 정보는 false로 설정될 수 있습니다.

주: DuplicateEventElimination이 true로 설정될 때, 보증된 이벤트 전달이 사용 가능하도록 MonitorQueue 등록 정보를 구성해야 합니다.

FaultQueue

메시지를 처리하는 동안 커넥터에서 오류가 발생한 경우, 커넥터는 상태 표시기 및 문제점 설명과 함께 이 등록 정보에 지정한 대기열로 메시지를 이동시킵니다.

기본값은 CONNECTORNAME/FAULTQUEUE입니다.

JvmMaxHeapSize

에이전트의 최대 힙 크기(MB)입니다. 이 등록 정보는 RepositoryDirectory 값이 <REMOTE>인 경우에만 적용 가능합니다.

기본값은 128m입니다.

JvmMaxNativeStackSize

에이전트의 최대 기본 스택 크기(KB)입니다. 이 등록 정보는 RepositoryDirectory 값이 <REMOTE>인 경우에만 적용 가능합니다.

기본값은 128k입니다.

JvmMinHeapSize

에이전트의 최소 힙 크기(MB)입니다. 이 등록 정보는 RepositoryDirectory 값이 <REMOTE>인 경우에만 적용 가능합니다.

기본값은 1m입니다.

jms.FactoryClassName

JMS 제공자가 인스턴스로 생성할 클래스 이름을 지정합니다. 전달 전송 메커니즘(DeliveryTransport)으로 JMS를 선택할 때 반드시 이 커넥터 등록 정보를 설정해야 합니다.

기본값은 CxCommon.Messaging.jms.IBMMQSeriesFactory입니다.

jms.MessageBrokerName

JMS 제공자가 사용할 브로커 이름을 지정합니다. 전달 전송 메커니즘(DeliveryTransport)으로 JMS를 선택할 때 반드시 이 커넥터 등록 정보를 설정해야 합니다.

기본값은 crossworlds.queue.manager입니다.

jms.NumConcurrentRequests

동시에 커넥터에 전송할 수 있는 최대 동시 서비스 호출 요청 수를 지정합니다. 최대값에 일단 도달하면, 새 서비스 호출이 차단되고 계속 진행하기 전에 다른 요청이 완료될 때까지 대기합니다.

기본값은 10입니다.

jms.Password

JMS 제공자의 암호를 지정합니다. 이 등록 정보에 대한 값은 선택적입니다.

기본값은 없습니다.

jms.UserName

JMS 제공자의 사용자 이름을 지정합니다. 이 등록 정보에 대한 값은 선택적입니다.

기본값은 없습니다.

ListenerConcurrency

이 등록 정보는 ICS가 통합 브로커일 때 MQ Listener의 멀티스레딩을 지원합니다. 이는 데이터베이스에 대한 복수 이벤트의 일괄처리 쓰기를 가능하게 하여 시스템 성능을 향상시킵니다. 기본값은 1입니다.

이 등록 정보는 MQ 전송을 사용하는 커넥터에만 적용됩니다. DeliveryTransport 등록 정보를 MQ로 설정해야 합니다.

Locale

언어 코드, 국가 또는 지역, 그리고 선택적으로 연관된 문자 코드 세트를 지정합니다. 이 등록 정보의 값은 데이터의 배열 및 정렬 순서, 날짜 및 시간 형식, 화폐 스펙에서 사용되는 기호와 같은 문화적 규약을 판별합니다.

로케일 이름의 형식은 다음과 같습니다.

ll_TT.codeset

여기서

ll

2문자 언어 코드(보통 소문자)

TT

2문자 국가 또는 지역 코드(보통 대문자)

codeset

연관된 문자 코드 세트의 이름입니다. 이름의 이 부분은 대체로 선택적입니다.

기본적으로, 지원되는 로케일의 서브세트만 드롭 목록에 표시됩니다. 기타 지원되는 값을 드롭 목록에 추가하려면, 제품 디렉토리에 있는 \Data\Std\stdConnProps.xml 파일을 직접 수정해야 합니다. 자세한 정보는 Connector Configurator의 부록을 참조하십시오.

기본값은 en_US입니다. 커넥터가 국제화되지 않은 경우, 이 등록 정보의 유일한 올바른 값은 en_US입니다. 특정 커넥터가 국제화되었는지 여부를 판별하려면 다음 웹사이트의 커넥터 버전 목록을 참조하십시오.

<http://www.ibm.com/software/websphere/wbiadapters/infocenter> 또는
<http://www.ibm.com/websphere/integration/wicsserver/infocenter>

LogAtInterchangeEnd

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

통합 브로커의 로그 목적지에 오류를 기록할지 여부를 지정합니다. 브로커의 로그 목적지에 기록하면, 전자 우편 공고가 커져서 오류 또는 심각한 오류가 발생할 때 InterchangeSystem.cfg 파일에 지정된 MESSAGE_RECIPIENT에 대한 전자 우편 공고를 생성합니다.

예를 들어, 커넥터에서 응용프로그램에 대한 연결이 끊어질 때 LogAtInterChangeEnd가 true로 설정되면, 전자 우편 메시지가 지정된 메시지 수신자에게 전송됩니다. 기본값은 false입니다.

MaxEventCapacity

제어기 버퍼의 최대 이벤트 수입입니다. 이 등록 정보는 플로우 제어에서 사용되며 RepositoryDirectory 등록 정보의 값이 <REMOTE>인 경우에만 적용 가능합니다.

값은 1 - 2147483647의 양수일 수 있습니다. 기본값은 2147483647입니다.

MessageFileName

커넥터 메시지 파일의 이름입니다. 메시지 파일의 표준 위치는 \connectors\messages입니다. 메시지 파일이 표준 위치에 없으면 절대 경로에 메시지 파일 이름을 지정하십시오.

커넥터 메시지 파일이 존재하지 않을 경우, 커넥터는 InterchangeSystem.txt를 메시지 파일로 사용합니다. 이 파일은 제품 디렉토리에서 발견됩니다.

주: 특정 커넥터에 자체 메시지 파일이 있는지 판별하려면 개별 어댑터 사용자 안내서를 참조하십시오.

MonitorQueue

중복 이벤트를 모니터하기 위해 커넥터에서 사용하는 논리 대기열입니다. 이 등록 정보는 DeliveryTransport 등록 정보 값이 JMS이고 DuplicateEventElimination이 TRUE로 설정되는 경우에만 사용됩니다.

기본값은 CONNECTORNAME/MONITORQUEUE입니다.

OADAutoRestartAgent

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

커넥터가 자동 및 원격 다시 시작 기능을 사용할지 여부를 지정합니다. 이 기능은 비정상 종료 이후에 커넥터를 다시 시작하거나 System Monitor로부터 원격 커넥터를 시작할 때 MQ 트리거 OAD(Object Activation Daemon)를 사용합니다.

자동 및 원격 다시 시작 기능을 사용 가능하게 하려면 이 등록 정보를 true로 설정해야 합니다. MQ 트리거 OAD 기능을 구성하는 방법에 대해서는 *Windows용 설치 안내서* 또는 *UNIX용 설치 안내서*를 참조하십시오.

기본값은 false입니다.

OADMaxNumRetry

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

비정상 종료 후 MQ 트리거 OAD가 자동으로 커넥터를 다시 시작하려고 시도하는 최대 횟수를 지정합니다. 이 등록 정보를 적용하려면 OADAutoRestartAgent 등록 정보를 true로 설정해야 합니다.

기본값은 1000입니다.

OADRetryTimeInterval

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

MQ 트리거 OAD에 대한 재시도 시간 간격에 시간(분 단위)을 지정합니다. 이 재시도 시간 간격 이내에 커넥터 에이전트가 다시 시작되지 않으면 커넥터 제어기는 OAD에게 커넥터 에이전트를 다시 시작하도록 요청합니다. OAD는 OADMaxNumRetry 등록 정보에 지정된 횟수만큼 이 재시도 프로세스를 반복합니다. 이 등록 정보를 적용하려면 OADAutoRestartAgent 등록 정보를 true로 설정해야 합니다.

기본값은 10입니다.

PollEndTime

이벤트 대기열 폴링을 중지할 시간입니다. 형식은 HH:MM이며, 여기서 HH는 0 - 23의 시간, MM은 0 - 59의 분을 나타냅니다.

이 등록 정보에 올바른 값을 제공해야 합니다. 기본값은 HH:MM이지만, 변경해야 합니다.

PollFrequency

각 폴링 조치 사이의 시간값입니다. PollFrequency를 다음 값 중 하나로 설정하십시오.

- 폴링 조치 사이의 밀리초 수.
- 단어 key - 사용자가 커넥터의 명령 프롬프트 창에 영문자 p를 입력할 경우에만 커넥터가 폴링하도록 합니다. 단어는 소문자로 입력하십시오.
- 단어 no - 커넥터가 폴링하지 않게 합니다. 단어는 소문자로 입력하십시오.

기본값은 10000입니다.

중요: 일부 커넥터는 이 등록 정보 사용에 제한을 가지고 있습니다. 특정 커넥터가 이 등록 정보를 사용하는지 여부를 판별하려면 해당 어댑터 안내서의 설치 및 구성 장을 참조하십시오.

PollQuantity

커넥터가 폴링해야 하는 응용프로그램으로부터의 항목 수를 지정합니다. 어댑터에 폴링 수를 설정하기 위한 커넥터 특정 등록 정보가 있으면, 커넥터 특정 등록 정보에 설정된 값이 표준 등록 정보 값을 대체합니다.

PollStartTime

이벤트 대기열 폴링을 시작할 시간입니다. 형식은 HH:MM이며, 여기서 HH는 0 - 23의 시간, MM은 0 - 59의 분을 나타냅니다.

이 등록 정보에 올바른 값을 제공해야 합니다. 기본값은 HH:MM이지만, 변경해야 합니다.

RequestQueue

Business Object를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTOR/REQUESTQUEUE입니다.

RepositoryDirectory

Business Object 정의의 메타 데이터를 저장하는 XML 스키마 문서를 커넥터가 읽는 저장소의 위치입니다.

통합 브로커가 ICS이면, 커넥터가 InterChange Server 저장소에서 정보를 얻으므로 이 값을 <REMOTE>로 설정해야 합니다.

통합 브로커가 WebSphere Message Broker 또는 WAS이면 이 값을 <로컬 디렉토리>로 설정해야 합니다.

ResponseQueue

DeliveryTransport가 JMS인 경우에만 적용 가능하며 RepositoryDirectory가 <REMOTE>인 경우에만 필수입니다.

커넥터 프레임워크에서 통합 브로커로 응답 메시지를 전달하는 JMS 응답 대기열을 지정합니다. 통합 브로커가 ICS이면 서버는 요청을 전송한 후 JMS 응답 대기열에서 응답을 기다립니다.

RestartRetryCount

커넥터가 커넥터 자체를 다시 시작하려고 하는 횟수를 지정합니다. 병렬 커넥터에 사용한 경우, 마스터 커넥터 응용프로그램 특정 구성요소가 슬레이브 커넥터 응용프로그램 특정 구성요소를 다시 시작하려고 시도하는 횟수를 지정합니다.

기본값은 3입니다.

RestartRetryInterval

커넥터가 커넥터 자체를 다시 시작하려고 시도하는 간격을 분 수로 지정합니다. 병렬 커넥터에 사용한 경우, 마스터 커넥터 응용프로그램 특정 구성요소가 슬레이브 커넥터 응용프로그램 특정 구성요소를 다시 시작하려고 시도하는 간격을 지정합니다. 가능한 값은 1 - 2147483647입니다.

기본값은 1입니다.

SourceQueue

DeliveryTransport가 JMS이고 ContainerManagedEvents가 지정된 경우에만 적용 가능합니다.

JMS 이벤트 저장소를 사용하는 JMS 사용 커넥터에 대해 보증된 이벤트 전달을 지원하는 커넥터 프레임워크의 JMS 소스 대기열을 지정합니다. 자세한 정보는 62 페이지의 『ContainerManagedEvents』를 참조하십시오.

기본값은 CONNECTOR/SOURCEQUEUE입니다.

SynchronousRequestQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

커넥터 프레임워크에서 브로커로의 동기 응답을 요구하는 요청 메시지를 전달합니다. 이 대기열은 커넥터가 동기 실행을 사용할 경우에만 필요합니다. 동기 실행을 사용할 때, 커넥터 프레임워크는 메시지를 SynchronousRequestQueue로 전송하고

SynchronousResponseQueue에서 브로커의 응답을 기다립니다. 커넥터에 전송되는 응답 메시지는 원래 메시지 ID와 일치하는 Correlation ID를 유지합니다.

기본값은 CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE입니다.

SynchronousResponseQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

브로커에서 커넥터 프레임워크에 이르는 동기 요청에 대한 응답으로 전송되는 응답 메시지를 전달합니다. 이 대기열은 커넥터가 동기 실행을 사용할 경우에만 필요합니다.

기본값은 CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE입니다.

SynchronousRequestTimeout

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

커넥터가 동기 요청에 대한 응답을 기다리는 시간(분)을 지정합니다. 지정된 시간 내에 응답이 수신되지 않을 경우, 커넥터는 원래 동기 요청 메시지를 오류 메시지와 함께 결합 대기열로 이동합니다.

기본값은 0입니다.

WireFormat

전송 시 메시지 형식입니다.

- RepositoryDirectory가 로컬 디렉토리이면 설정은 CwXML입니다.
- RepositoryDirectory의 값이 <REMOTE>이면 설정은 CwBO입니다.

WsifSynchronousRequest 제한시간

WAS 통합 브로커에만 해당됩니다.

커넥터가 동기 요청에 대한 응답을 기다리는 시간(분)을 지정합니다. 지정된 시간 내에 응답이 수신되지 않을 경우, 커넥터는 원래 동기 요청 메시지를 오류 메시지와 함께 결합 대기열로 이동합니다.

기본값은 0입니다.

XMLNamespaceFormat

WebSphere Message Broker 및 WAS 통합 브로커에만 해당됩니다.

사용자가 XML 형식의 Business Object 정의에 짧고 긴 이름 공간을 지정할 수 있도록 하는 강력한 등록 정보입니다.

기본값은 short입니다.

부록 B. Connector Configurator

이 부록에서는 Connector Configurator를 사용하여 어댑터의 구성 등록 정보 값을 설정하는 방법에 대해 설명합니다.

Connector Configurator를 사용하여 다음을 수행할 수 있습니다.

- 커넥터 구성을 위한 커넥터 특정 등록 정보 템플릿 작성
- 구성 파일 작성
- 구성 파일에 등록 정보 설정

주:

이 책에서 백슬래시(\)는 디렉토리 경로의 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체하고 각 운영 체제의 규칙을 따르십시오.

이 부록에서 다루는 주제는 다음과 같습니다.

- 73 페이지의 『Connector Configurator 개요』
- 74 페이지의 『Connector Configurator 시작』
- 75 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 78 페이지의 『새 구성 파일 작성』
- 81 페이지의 『구성 파일 등록 정보 설정』
- 89 페이지의 『국제화된 환경에서 Connector Configurator 사용』

Connector Configurator 개요

Connector Configurator를 사용하면 다음 통합 브로커에서 사용하는 어댑터의 커넥터 구성요소를 구성할 수 있습니다.

- WebSphere InterChange Server(ICS)
- WebSphere MQ Integrator, WebSphere MQ Integrator Broker 및 WebSphere Business Integration Message Broker를 통털어 WMQI(WebSphere Message Brokers)라고 합니다.
- WebSphere Application Server(WAS)

Connector Configurator를 사용하여 다음을 수행할 수 있습니다.

- 커넥터 구성을 위한 커넥터 특정 등록 정보 템플릿 작성
- 커넥터 구성 파일 작성: 설치한 각 커넥터당 하나의 구성 파일을 작성해야 합니다.
- 구성 파일에 등록 정보 설정
커넥터 템플릿의 등록 정보에 설정된 기본값을 수정해야 할 경우가 있습니다. 또한

필요한 경우, 메시징, 로깅 및 추적, Data Handler 매개변수 지정과 더불어 지원되는 Business Object 정의와 ICS와 함께 협업에 사용하기 위한 맵을 지정해야 합니다.

Connector Configurator를 실행하는 모드와, 사용하는 구성 파일 유형은 실행 중인 통합 브로커에 따라 다를 수 있습니다. 예를 들어, WMQ가 브로커인 경우, System Manager 내에서가 아니라 직접 Connector Configurator를 실행합니다(75 페이지의 『독립형 모드로 Configurator 실행』 참조).

커넥터 구성 등록 정보에는 표준 구성 등록 정보(모든 커넥터가 가지고 있는 등록 정보) 및 커넥터 특정 등록 정보(특정 응용프로그램이나 기술에 따라 커넥터에 필요한 등록 정보)가 둘 다 포함됩니다.

표준 등록 정보는 모든 커넥터에서 사용되기 때문에, 이러한 등록 정보를 처음부터 정의할 필요는 없습니다. 파일을 작성하는 즉시 Connector Configurator는 이들을 구성 파일로 통합합니다. 그러나 Connector Configurator에서 각 표준 등록 정보 값을 설정해야 합니다.

모든 브로커와 모든 구성에서 표준 등록 정보의 범위가 동일하지 않을 수도 있습니다. 일부 등록 정보는 다른 등록 정보에 특정 값이 제공되는 경우에만 사용 가능합니다. Connector Configurator의 표준 등록 정보 창에는 특정 구성에 사용 가능한 등록 정보가 표시됩니다.

그러나 커넥터 특정 등록 정보의 경우 먼저 등록 정보를 정의한 다음 해당 값을 설정해야 합니다. 특정 어댑터의 커넥터 특정 등록 정보 템플릿을 작성하여 이를 수행합니다. 시스템에 템플릿이 이미 설정되어 있으면 이 템플릿을 사용하면 됩니다. 그렇지 않은 경우, 76 페이지의 『새 템플릿 작성』의 단계에 따라 새 템플릿을 설정하십시오.

주: Connector Configurator는 Windows 환경에서만 실행합니다. UNIX 환경에서 커넥터를 실행하고 있는 경우, Windows에서 Connector Configurator를 사용하여 구성 파일을 수정한 다음 파일을 UNIX 환경으로 복사하십시오.

Connector Configurator 시작

다음 두 가지 모드 중 하나로 Connector Configurator를 시작하고 실행할 수 있습니다.

- 독립형 모드로 독립적으로 시작 및 실행
- System Manager에서 시작 및 실행

독립형 모드로 Configurator 실행

Connector Configurator를 독립적으로 실행하고 브로커와 무관하게 커넥터 구성 파일에 대해 작업할 수 있습니다.

이를 수행하려면 다음 단계를 따르십시오.

- 시작 > 프로그램에서 **IBM WebSphere InterChange Server > IBM WebSphere Business Integration Toolset > 개발 > Connector Configurator**를 누르십시오.
- 파일 > 새로 작성 > 구성 파일을 선택하십시오.
- 시스템 연결 통합 브로커 옆에 있는 풀다운 메뉴를 누르면 브로커에 따라 ICS, WebSphere Message Brokers 또는 WAS를 선택할 수 있습니다.

Connector Configurator를 별도로 실행하도록 선택하여 파일을 생성한 다음 System Manager에 연결하여 파일을 System Manager 프로젝트에 저장할 수 있습니다(80 페이지의 『구성 파일 완료』 참조).

System Manager에서 Configurator 실행

System Manager에서 Connector Configurator를 실행할 수 있습니다.

Connector Configurator를 실행하려면 다음을 수행하십시오.

1. System Manager를 여십시오.
2. System Manager 창에서 통합 구성요소 라이브러리 아이콘을 펼치고 커넥터를 강조 표시하십시오.
3. System Manager 메뉴 표시줄에서 도구 > **Connector Configurator**를 누르십시오. Connector Configurator 창이 열리고 새 커넥터 대화 상자가 표시됩니다.
4. 시스템 연결 통합 브로커 옆에 있는 풀다운 메뉴를 누르면 브로커에 따라 ICS, WebSphere Message Brokers 또는 WAS를 선택할 수 있습니다.

기존 구성 파일을 편집하려면 다음을 수행하십시오.

1. System Manager 창에서 커넥터 폴더에 나열된 구성 파일 중 하나를 선택하고 이를 마우스 오른쪽 단추로 누르십시오. Connector Configurator가 열리고 맨 위에 통합 브로커 유형 및 파일 이름과 함께 구성 파일이 표시됩니다.
2. 표준 등록 정보 탭을 눌러 이 구성 파일에 포함되어 있는 등록 정보를 보십시오.

커넥터 특정 등록 정보 템플릿 작성

커넥터에 맞는 구성 파일을 작성하려면, 시스템 제공 표준 등록 정보 외에 커넥터 특정 등록 정보 템플릿이 필요합니다.

커넥터의 커넥터 특정 등록 정보에 맞는 새 템플릿을 작성하거나, 기존 파일을 템플릿으로 사용할 수 있습니다.

- 새 템플리트를 작성하려면 76 페이지의 『새 템플리트 작성』을 참조하십시오.
- 기존 파일을 사용하려면 기존 템플리트를 수정하여 새 이름으로 저장하기만 하면 됩니다.

새 템플리트 작성

이 절에서는 템플리트에서 등록 정보를 작성하고, 이러한 등록 정보의 일반 특성 및 값을 정의하며, 등록 정보 사이에 종속성을 지정하는 방법에 대해 설명합니다. 그런 다음 템플리트를 저장하여 새 커넥터 구성 파일을 작성하기 위한 기본으로 사용합니다.

템플리트를 작성하려면 다음을 수행하십시오.

1. 파일 > 새로 작성 > 커넥터 특정 등록 정보 템플리트를 누르십시오.
2. 다음 필드가 있는 커넥터 특정 등록 정보 템플리트 대화 상자가 나타납니다.
 - 템플리트 및 이름
커넥터를 식별하는 고유한 이름을 입력하거나, 이 템플리트를 사용할 커넥터 유형을 입력하십시오. 템플리트에서 새 구성 파일을 작성하기 위해 대화 상자를 열 때 이 이름이 다시 표시됩니다.
 - 이전 템플리트 및 수정할 기존 템플리트 선택
현재 사용 가능한 모든 템플리트 이름이 템플리트 이름 화면에 표시됩니다.
 - 템플리트의 커넥터 특정 등록 정보 정의를 보려면 템플리트 이름 화면에서 해당 템플리트의 이름을 선택하십시오. 해당 템플리트에 포함된 등록 정보 정의 목록이 템플리트 미리보기 화면에 나타납니다. 등록 정보 정의가 커넥터에 필요한 것과 유사한 기존 템플리트를 사용자 템플리트를 위한 출발점으로 사용할 수 있습니다.
3. 템플리트 이름 화면에서 템플리트를 선택하고, 이름 찾기 필드에 해당 템플리트 이름을 입력하고(또는 템플리트 이름에서 선택사항을 강조표시하고), 다음을 누르십시오.

커넥터에서 사용하는 커넥터 특정 등록 정보를 표시하는 템플리트가 보이지 않으면 템플리트를 작성해야 합니다.

일반 특성 지정

다음을 누르면 템플리트를 선택할 수 있는 등록 정보 - 커넥터 특정 등록 정보 템플리트 대화 상자가 나타납니다. 대화 상자에는 정의된 등록 정보의 일반 특성과 값 제한사항에 관한 탭이 있습니다. 일반 화면에는 다음 필드가 있습니다.

- 일반
 - 등록 정보 유형
 - 갱신 메소드
 - 설명

- 플래그
표준 플래그
- 사용자 정의 플래그
플래그

등록 정보의 일반 특성을 선택한 후, **값** 탭을 누르십시오.

값 지정

값 탭을 사용하여 등록 정보에 대한 최대 길이, 최대 복수 값, 기본값 또는 값 범위를 설정할 수 있습니다. 편집 가능한 값도 허용됩니다. 이를 수행하려면 다음 단계를 따르십시오.

1. 값 탭을 누르십시오. 일반 표시 패널이 값 표시 패널로 바뀝니다.
2. 등록 정보 편집 화면에서 등록 정보의 이름을 선택하십시오.
3. 최대 길이 및 최대 복수 값에 대한 필드에서 필요한 변경을 수행하십시오. 다음 단계에 설명된 대로 등록 정보의 등록 정보 값 대화 상자를 열 때까지는 변경사항이 승인되지 않습니다.
4. 값 테이블의 왼쪽 맨 위 구석에 있는 상자를 마우스 오른쪽 단추로 누른 다음 추가를 누르십시오. 등록 정보 값 대화 상자가 나타납니다. 등록 정보 유형에 따라, 대화 상자에서 값을 입력하거나 값과 범위를 모두 입력할 수 있습니다. 적절한 값이나 범위를 입력하고 확인을 누르십시오.
5. 값 패널이 최대 길이 및 최대 복수 값에서 수행한 모든 변경사항을 표시하기 위해 새로 고쳐집니다. 세 개의 열이 있는 다음 테이블이 나타납니다.

값 열은 사용자가 등록 정보 값 대화 상자에 입력한 값과, 사용자가 작성한 모든 이전 값을 표시합니다.

기본값 열을 사용하면 임의의 값을 기본값으로 지정할 수 있습니다.

값 범위는 등록 정보 값 대화 상자에서 입력한 범위를 표시합니다.

값을 작성한 후 값이 격자에 나타나면, 테이블 화면에서 이를 편집할 수 있습니다. 테이블의 기존 값을 변경하려면 행 번호를 눌러 전체 행을 선택하십시오. 그런 다음 값 필드를 마우스 오른쪽 단추로 누르고 **값** 편집을 누르십시오.

종속성 설정

일반 및 값 탭에서 변경을 수행한 후, 다음을 누르십시오. 종속성 - 키벡터 특정 등록 정보 템플릿 대화 상자가 나타납니다.

종속 등록 정보는 다른 등록 정보의 값이 특정 조건에 부합하는 경우에만 템플릿에 포함되어 구성 파일에 사용되는 등록 정보입니다. 예를 들어, PollQuantity는 JMS가 전송 메커니즘이고 DuplicateEventElimination이 True로 설정되는 경우에만 템플릿에 나타납니다.

등록 정보가 종속되도록 지정하고 종속 조건을 설정하려면, 다음을 수행하십시오.

1. 사용 가능한 등록 정보 화면에서 종속될 등록 정보를 선택하십시오.
2. 등록 정보 선택 필드에서 드롭 다운 메뉴를 사용하여 조건 값을 보유할 등록 정보를 선택하십시오.
3. 조건 연산자 필드에서 다음 중 하나를 선택하십시오.
 - == (같음)
 - != (같지 않음)
 - > (초과)
 - < (미만)
 - >= (이상)
 - <= (이하)
4. 조건 값 필드에 종속 등록 정보를 템플릿에 포함하기 위해 필요한 값을 입력하십시오.
5. 사용 가능한 등록 정보 화면에서 종속 등록 정보가 강조표시된 상태에서 화살표를 눌러서 종속 등록 정보 화면으로 이동하십시오.
6. 완료 버튼을 누르십시오. Connector Configurator가 Connector Configurator를 설치한 \bin 디렉토리의 \data\app에 XML 문서로 입력한 정보를 저장합니다.

새 구성 파일 작성

새 구성 파일을 작성하는 첫 단계는 통합 브로커를 선택하는 것입니다. 선택하는 브로커에 따라 구성 파일에 표시될 등록 정보가 달라집니다.

브로커를 선택하려면 다음을 수행하십시오.

- Connector Configurator 홈 메뉴에서 파일 > 새로 작성 > 커넥터 구성을 누르십시오. 새 커넥터 대화 상자가 나타납니다.
- 통합 브로커 필드에서 ICS, WebSphere Message Brokers 또는 WAS 연결을 선택하십시오.
- 이 장의 뒷부분에서 설명된 대로 새 커넥터 창에서 나머지 필드를 완료하십시오.

다음 작업도 수행할 수 있습니다.

- System Manager 창에서 커넥터 폴더를 마우스 오른쪽 단추로 누르고 새 커넥터 작성을 선택하십시오. Connector Configurator가 열리고 새 커넥터 대화 상자가 표시됩니다.

커넥터 특정 템플릿에서 구성 파일 작성

커넥터 특정 템플릿을 작성한 후, 이 템플릿을 사용하여 구성 파일을 작성할 수 있습니다.

1. 파일 > 새로 작성 > 커넥터 구성을 누르십시오.

2. 다음 필드가 있는 새 커넥터 대화 상자가 나타납니다.

- 이름

커넥터의 이름을 입력하십시오. 이름은 대소문자를 구분합니다. 입력하는 이름은 고유해야 하며 시스템에 설치된 커넥터의 파일 이름과 일치해야 합니다.

중요: Connector Configurator는 사용자가 입력한 이름의 철자를 확인하지 않습니다. 사용자가 이름이 정확한지 확인해야 합니다.

- 시스템 연결

ICS, WebSphere Message Brokers 또는 WAS를 누르십시오.

- 커넥터 특정 등록 정보 템플릿

커넥터에 맞게 설계된 템플릿의 이름을 입력하십시오. 사용 가능한 템플릿이 템플릿 이름 화면에 표시됩니다. 템플릿 이름 화면에서 이름을 선택할 경우 등록 정보 템플릿 미리보기 화면이 해당 템플릿에 정의된 커넥터 특정 등록 정보를 표시합니다.

사용하려는 템플릿을 선택하고 확인을 누르십시오.

3. 구성 중인 커넥터의 구성 화면이 표시됩니다. 제목 표시줄이 통합 브로커와 커넥터 이름을 표시합니다. 모든 필드값을 입력하여 지금 정의를 완료하거나, 파일을 저장했다가 나중에 필드를 완성할 수 있습니다.

4. 파일을 저장하려면 파일 > 저장 > 파일에 또는 파일 > 저장 > 프로젝트에를 누르십시오. 프로젝트를 저장하려면 System Manager가 실행 중이어야 합니다.

파일로 저장하는 경우 파일 커넥터 저장 대화 상자가 나타납니다. 파일 유형으로 *.cfg를 선택하고 파일 이름 필드에서 이름의 철자 및 대소문자가 올바른지 확인한 후, 파일을 저장할 디렉토리로 이동하고 저장을 누르십시오. Connector Configurator의 메시지 패널에 있는 상태 화면은 구성 파일이 성공적으로 작성되었음을 표시합니다.

중요: 여기에서 설정하는 디렉토리 경로 및 이름은 커넥터에 대해 시작 파일에 제공한 커넥터 구성 파일 경로 및 이름과 일치해야 합니다.

5. 커넥터 정의를 완료하려면, 이 장의 뒷부분에 설명된 대로 Connector Configurator 창의 각 탭에 있는 필드에 값을 입력하십시오.

기존 파일 사용

다음 중 하나 이상의 형식으로 기존 파일을 사용할 수 있습니다.

- 커넥터 정의 파일.

이 파일은 특정 커넥터의 등록 정보와 적용 가능한 기본값을 나열한 텍스트 파일입니다. 일부 커넥터는 전달 패키지의 \repository 디렉토리에 이러한 파일을 포함합니다. (이 파일의 확장자는 보통 .txt입니다. 예를 들어, XML 커넥터의 경우에는 CN_XML.txt입니다.)

- ICS 저장소 파일.
커넥터의 이전 ICS 구현에서 사용된 정의는 해당 커넥터의 구성에 사용된 저장소 파일에서 사용할 수도 있습니다. 이러한 파일의 확장자는 일반적으로 `.in` 또는 `.out`입니다.
- 커넥터의 이전 구성 파일.
이러한 파일의 확장자는 일반적으로 `*.cfg`입니다.

이 파일 소스는 커넥터의 커넥터 특정 등록 정보의 전부 또는 대부분을 포함할 수 있지만, 커넥터 구성 파일은 이 장의 뒷부분에 설명된 대로 파일을 열고 등록 정보를 설정할 때까지 완료되지 않습니다.

기존 파일을 사용하여 커넥터를 구성하려면, Connector Configurator에서 파일을 열어 구성을 변경한 다음 파일을 다시 저장해야 합니다.

디렉토리에서 `*.txt`, `*.cfg` 또는 `*.in` 파일을 열려면 다음 단계를 따르십시오.

1. Connector Configurator에서 **파일 > 열기 > 파일에서**를 누르십시오.
2. **파일 커넥터 열기 대화 상자**에서 다음 파일 유형 중 하나를 선택해서 사용 가능한 파일을 보십시오.
 - 구성(`*.cfg`)
 - ICS 저장소(`*.in`, `*.out`)
저장소 파일이 ICS 환경에서 커넥터를 구성하는 데 사용된 경우 이 옵션을 선택하십시오. 저장소 파일은 여러 개의 커넥터 정의를 포함할 수 있으며, 모든 커넥터 정의는 사용자가 파일을 열 때 표시됩니다.
 - 모든 파일(`*.*`)
`*.txt` 파일이 커넥터용 어댑터 패키지로 전달되거나 확장자가 다른 정의 파일을 사용할 수 있는 경우 이 옵션을 선택하십시오.
3. 디렉토리 화면에서 적당한 커넥터 정의 파일로 이동하고 이를 선택한 후 열기를 누르십시오.

System Manager 프로젝트에서 커넥터 구성을 열려면 다음 단계를 따르십시오.

1. System Manager를 시작하십시오. System Manager가 시작된 경우에만 구성을 System Manager에서 열거나 System Manager에 저장할 수 있습니다.
2. Connector Configurator를 시작하십시오.
3. **파일 > 열기 > 프로젝트에서**를 누르십시오.

구성 파일 완료

프로젝트에서 구성 파일이나 커넥터를 열면, Connector Configurator 창에 현재 속성 및 값이 있는 구성 화면이 나타납니다.

구성 화면의 제목에는 파일에 지정된 커넥터 이름과 통합 브로커가 표시됩니다. 브로커가 올바른지 확인하십시오. 그렇지 않을 경우, 커넥터를 구성하기 전에 브로커 값을 변경하십시오. 이를 수행하려면 다음 단계를 따르십시오.

1. 표준 등록 정보 탭에서 브로커 유형 등록 정보에 대한 값 필드를 선택하십시오. 드롭 다운 메뉴에서 ICS, WMQI 또는 WAS를 선택하십시오.
2. 표준 등록 정보 탭에 선택한 브로커와 연관된 등록 정보가 표시됩니다. 지금 파일을 저장하거나 84 페이지의 『지원되는 Business Object 정의 지정』에 설명된 대로 나머지 구성 필드를 완성할 수 있습니다.
3. 구성을 완료했을 때, 파일 > 저장 > 프로젝트에 또는 파일 > 저장 > 파일에를 누르십시오.

파일에 저장할 경우, 확장자로 *.cfg를 선택하고 파일의 올바른 위치를 선택한 후 저장을 누르십시오.

여러 커넥터 구성이 열려 있으면 파일에 모두 저장을 눌러서 모든 구성을 파일에 저장하거나, 프로젝트에 모두 저장을 눌러서 모든 커넥터 구성을 System Manager 프로젝트에 저장하십시오.

Connector Configurator는 파일을 저장하기 전에 모든 필수 표준 등록 정보의 값이 설정되었는지 확인합니다. 필수 표준 등록 정보 값이 누락된 경우, Connector Configurator는 유효성 검증 실패를 알리는 메시지를 표시합니다. 구성 파일을 저장하려면 해당 등록 정보에 값을 제공해야 합니다.

구성 파일 등록 정보 설정

새 커넥터 구성 파일을 작성하여 이름을 지정하거나 기존 커넥터 구성 파일을 열 때, Connector Configurator는 필수 구성값의 카테고리에 대한 탭이 있는 구성 화면을 표시합니다.

Connector Configurator는 다음 범주에서 모든 브로커에서 실행 중인 커넥터에 대한 등록 정보의 값이 필요합니다.

- 표준 등록 정보
- 커넥터 특정 등록 정보
- 지원되는 Business Object
- 추적/로그 파일 값
- Data Handler(이벤트 전달이 보장되는 JMS 메시지를 사용하는 커넥터의 경우 적용 가능)

주: 데이터를 Business Object로 변환하는 Data Handler의 구성에서 JMS 메시지를 사용하는 커넥터의 경우, 추가 카테고리가 표시될 수 있습니다.

ICS에서 실행되는 커넥터의 경우, 다음 등록 정보의 값도 필요합니다.

- 연관된 맵
- 자원
- 메시징(적용 가능할 경우)

중요: Connector Configurator는 영어 또는 영어 이외의 문자 세트의 등록 정보 값을 승인합니다. 그러나 표준 및 커넥터 특정 등록 정보의 이름과 지원되는 Business Object의 이름은 영어 문자 세트만을 사용해야 합니다.

표준 등록 정보는 다음과 같이 커넥터 특정 등록 정보와 다릅니다.

- 커넥터의 표준 등록 정보는 커넥터의 응용프로그램 특정 구성요소와 해당 브로커 구성 요소에서 모두 공유됩니다. 모든 커넥터에는 동일한 표준 등록 정보 세트가 있습니다. 이러한 등록 정보는 각 어댑터 안내서의 부록 A에 설명되어 있습니다. 일부는 변경할 수 있지만 모든 값을 변경할 수는 없습니다.
- 응용프로그램 특정 등록 정보는 커넥터의 응용프로그램 특정 구성요소, 즉 응용프로그램과 직접 상호작용하는 구성요소에만 적용됩니다. 각 커넥터에는 커넥터의 응용프로그램에 고유한 응용프로그램 특정 등록 정보가 있습니다. 이러한 등록 정보 중 일부는 기본값을 제공하며, 일부는 기본값을 제공하지 않습니다. 일부 기본값은 수정할 수 있습니다. 각 어댑터 안내서의 설치 및 구성 장에서는 응용프로그램 특정 등록 정보와 권장값에 대해 설명합니다.

표준 등록 정보 및 커넥터 특정 등록 정보에 대한 값은 구성 가능한 필드를 표시하기 위해 색상을 사용하여 코딩됩니다.

- 회색 배경의 필드는 표준 등록 정보를 나타냅니다. 값을 변경할 수 있으나 이름을 변경하거나 등록 정보를 제거할 수는 없습니다.
- 흰색 배경의 필드는 응용프로그램 특정 등록 정보를 나타냅니다. 이들 등록 정보는 응용프로그램 또는 커넥터의 특정 필요에 따라 다릅니다. 값을 변경하고 이 등록 정보를 삭제할 수 있습니다.
- 값 필드는 구성 가능합니다.
- 갱신 메소드 필드는 정보용이므로 구성할 수 없습니다. 이 필드는 값이 변경된 등록 정보를 활성화하기 위해 필요한 조치를 지정합니다.

표준 커넥터 등록 정보 설정

표준 등록 정보의 값을 변경하려면 다음을 수행하십시오.

1. 값을 설정하려는 필드를 누르십시오.
2. 값을 입력하거나, 값이 나타나는 경우 드롭 다운 메뉴에서 선택하십시오.
3. 표준 등록 정보의 모든 값을 입력한 후, 다음 중 하나를 수행할 수 있습니다.
 - 변경사항을 취소하고 원래 값을 보존한 상태에서 Connector Configurator를 종료하려면, 파일 > 종료를 누르고(또는 창을 닫고) 변경사항을 저장할 것인지 묻는 메시지가 표시될 때 아니오를 누르십시오.

- Connector Configurator에서 다른 카테고리의 값을 입력하려면 카테고리 탭을 선택하십시오. 표준 등록 정보(또는 다른 카테고리)에 입력하는 값은 사용자가 다음 카테고리로 이동할 때 보존됩니다. 사용자가 창을 닫을 때 모든 카테고리에 입력한 값을 전체적으로 저장할 것인지 버릴 것인지 묻는 메시지가 표시됩니다.
- 변경된 값을 저장하려면, 파일 > 종료를 누르고(또는 창을 닫고) 변경사항을 저장할지 묻는 메시지가 표시되면 예를 누르십시오. 또는 파일 메뉴나 도구 모음에서 저장 > 파일에를 누르십시오.

응용프로그램 특정 구성 등록 정보 설정

응용프로그램 특정 구성 등록 정보의 경우 등록 정보 이름을 추가하거나 변경하고, 값을 구성하고, 등록 정보를 삭제하고, 등록 정보를 암호화할 수 있습니다. 기본 등록 정보 길이는 255자입니다.

1. 격자의 왼쪽 맨 윗부분을 마우스 오른쪽 단추로 누르십시오. 팝업 메뉴 표시줄이 표시됩니다. 등록 정보를 추가하려면 추가를 누르십시오. 하위 등록 정보를 추가하려면 상위 행 번호를 마우스 오른쪽 단추로 누른 다음 하위 추가를 누르십시오.
2. 등록 정보 또는 하위 등록 정보의 값을 입력하십시오.
3. 등록 정보를 암호화하려면 암호화 상자를 선택하십시오.
4. 82 페이지의 『표준 커넥터 등록 정보 설정』에 설명한 대로 변경사항을 저장하거나 취소하십시오.

각 등록 정보에 대해 표시되는 갱신 메소드는 변경된 값을 활성화하기 위해 구성요소 또는 에이전트를 다시 시작해야 하는지 여부를 표시합니다.

중요: 사전 설정된 응용프로그램 특정 커넥터 등록 정보 이름을 변경하면 커넥터가 실패할 수 있습니다. 커넥터를 응용프로그램에 연결하거나 제대로 실행하려면 특정 등록 정보 이름이 필요할 수 있습니다.

커넥터 등록 정보 암호화

응용프로그램 특정 등록 정보는 등록 정보 편집 창에서 암호화 선택란을 선택하여 암호화할 수 있습니다. 값의 암호를 해독하려면 암호화 선택란을 선택 해제하고 검증 대화 상자에 올바른 값을 입력한 다음 확인을 누르십시오. 입력된 값이 맞는 경우, 값은 암호 해독되고 표시됩니다.

각 커넥터에 대한 어댑터 안내서에는 각 등록 정보 및 해당 기본값의 목록과 설명이 들어 있습니다.

등록 정보의 값이 여러 개일 경우, 등록 정보의 첫 번째 값에 대한 암호화 선택란이 나타납니다. 암호화를 선택하면 모든 등록 정보 값이 암호화됩니다. 등록 정보의 여러 값에 대한 암호를 해독하려면, 등록 정보의 첫 번째 값에 대한 암호화 선택란을 선택 해제하고 검증 대화 상자에 새 값을 입력하십시오. 입력값이 일치하면, 모든 복수 값이 암호 해독됩니다.

갱신 메소드

56 페이지의 『등록 정보 값 설정 및 갱신』의 부록 커넥터에 대한 표준 구성 등록 정보에 있는 갱신 메소드에 대한 설명을 참조하십시오.

지원되는 Business Object 정의 지정

Connector Configurator의 지원되는 **Business Object** 탭을 사용하여 커넥터가 사용할 Business Object를 지정하십시오. 일반 Business Object와 응용프로그램 특정 Business Object를 모두 지정하고, Business Object 사이의 맵에 대한 연관을 지정해야 합니다.

주: 일부 커넥터에서는 해당 응용프로그램으로 이벤트 공고 또는 추가 구성(Meta Object 사용)을 수행하기 위해 특정 Business Object가 지원되도록 지정해야 합니다. 자세한 정보는 *Connector Development Guide for C++* 또는 *Connector Development Guide for Java*를 참조하십시오.

ICS가 브로커인 경우

Business Object 정의가 커넥터에서 지원되도록 지정하거나 기존 Business Object 정의의 지원 설정을 변경하려면, 지원되는 **Business Objects** 탭을 누르고 다음 필드를 사용하십시오.

Business Object 이름: System Manager가 실행 중인 커넥터에서 Business Object 정의가 지원되도록 지정하려면 다음을 수행하십시오.

1. **Business Object** 이름 목록의 빈 필드를 누르십시오. System Manager 프로젝트에 존재하는 모든 Business Object 정의를 보여주는 드롭 다운 목록이 표시됩니다.
2. Business Object를 눌러 추가하십시오.
3. Business Object에 대한 에이전트 지원(아래에 설명됨)을 설정하십시오.
4. Connector Configurator 창의 파일 메뉴에서 프로젝트에 저장을 누르십시오. 추가된 Business Object 정의에 지정된 지원을 포함하여 변경된 커넥터 정의가 System Manager의 프로젝트에 저장됩니다.

지원되는 목록에서 Business Object를 삭제하려면 다음을 수행하십시오.

1. Business Object 필드를 선택하려면 Business Object의 왼쪽에 있는 번호를 누르십시오.
2. Connector Configurator 창의 편집 메뉴에서 행 삭제를 누르십시오. 목록 화면에서 Business Object는 제거됩니다.
3. 파일 메뉴에서 프로젝트에 저장을 누르십시오.

지원되는 목록에서 Business Object를 삭제하면 커넥터 정의가 변경되고 삭제된 Business Object가 이 커넥터의 이 구현에서는 사용할 수 없게 됩니다. 커넥터 코드에 영향을 주거나 System Manager에서 Business Object 정의 자체를 제거하지는 않습니다.

에이전트 지원: Business Object가 에이전트 지원을 갖는 경우, 시스템은 커넥터 에이전트를 통해 응용프로그램으로 데이터를 전달하기 위해 해당 Business Object를 사용하려 시도합니다.

일반적으로 커넥터의 응용프로그램 특정 Business Object는 해당 커넥터의 에이전트에서 지원되지만 일반 Business Object는 지원되지 않습니다.

커넥터 에이전트가 Business Object를 지원한다는 것을 표시하려면 에이전트 지원 상자를 선택하십시오. Connector Configurator 창은 에이전트 지원 선택에 대해 유효성을 검증하지 않습니다.

최대 트랜잭션 레벨: 커넥터의 최대 트랜잭션 레벨은 커넥터가 지원하는 최상위 트랜잭션 레벨입니다.

대부분의 커넥터의 경우 최상의 노력만이 가능한 선택사항입니다.

트랜잭션 레벨 변경 사항을 적용하려면 서버를 다시 시작해야 합니다.

WebSphere Message Broker가 브로커인 경우

독립형 모드(System Manager에 연결되지 않음)로 작업 중인 경우 Business Object 이름을 수동으로 입력해야 합니다.

System Manager가 실행 중이면, 지원되는 **Business Objects** 탭의 **Business Object** 이름 열에서 빈 상자를 선택할 수 있습니다. 커넥터가 속하는 통합 구성요소 라이브러리 프로젝트에서 사용 가능한 Business Object 목록이 있는 콤보 상자가 나타납니다. 목록에서 원하는 Business Object를 선택하십시오.

WebSphere Business Integration Message Broker 5.0의 경우에는 메시지 세트 ID가 선택적 필드이므로 ID를 제공할 경우 고유하지 않아도 됩니다. 그러나 WebSphere MQ Integrator 및 Integrator Broker 2.1의 경우에는 고유한 ID를 제공해야 합니다.

WAS가 브로커인 경우

WebSphere Application Server를 브로커 유형으로 선택할 경우, Connector Configurator는 메시지 세트 ID가 필요하지 않습니다. 지원되는 **Business Object** 탭이 지원되는 Business Object에 대해서만 **Business Object** 이름 열을 표시합니다.

독립형 모드(System Manager에 연결되지 않은 채)로 작업 중인 경우 Business Object 이름을 수동으로 입력해야 합니다.

System Manager가 실행 중이면, 지원되는 Business Object 탭의 Business Object 이름 열에서 빈 상자를 선택할 수 있습니다. 커넥터가 속하는 통합 구성요소 라이브러리 프로젝트에서 사용 가능한 Business Object의 목록이 포함된 콤보 상자가 나타납니다. 이 목록에서 원하는 Business Object를 선택하십시오.

연관된 맵(ICCS에만 해당)

각 커넥터는 현재 WebSphere InterChange Server에서 활성 상태인 Business Object 정의 및 그와 연관된 맵의 목록을 지원합니다. 이 목록은 연관된 맵 탭을 선택할 때 나타납니다.

Business Object 목록에는 에이전트가 지원하는 응용프로그램 특정 Business Object와 제어기가 등록 협업으로 전송하는 해당 일반 오브젝트가 있습니다. 맵의 연관은 응용프로그램 특정 Business Object를 일반 Business Object로 변환하거나 일반 Business Object를 응용프로그램 특정 Business Object로 변환하는 데 사용할 맵을 결정합니다.

특정 소스 및 목적지 Business Object에 고유하게 정의된 맵을 사용 중인 경우, 화면을 열면 맵은 이미 해당 Business Object와 연관되어 있으므로 변경할 필요가(또는 변경할 수) 없습니다.

지원되는 Business Object에서 둘 이상의 맵이 사용 가능한 경우, 사용해야 하는 맵에 Business Object를 명시적으로 바인드해야 합니다.

연관된 맵 탭은 다음 필드를 표시합니다.

- **Business Object** 이름

지원되는 **Business Object** 탭에서 지정한 대로, 이 커넥터가 지원하는 Business Object입니다. 지원되는 Business Object 탭 아래에서 추가 Business Object를 지정하면, 해당 Business Object는 Connector Configurator 창의 파일 메뉴에서 프로젝트에 저장을 선택하여 변경사항을 저장한 후 이 목록에 반영됩니다.

- 연관된 맵

커넥터가 지원하는 Business Object에서 사용하기 위해 시스템에 설치된 모든 맵이 화면에 표시됩니다. 각 맵에 대한 소스 Business Object가 **Business Object** 이름 화면에서 맵 이름의 왼쪽에 표시됩니다.

- 명시

어떤 경우에는 연관된 맵을 명시적으로 바인드해야 할 수도 있습니다.

명시적 바인딩은 지원되는 특정 Business Object에 둘 이상의 맵이 존재하는 경우에만 필요합니다. ICCS는 시동할 때 자동으로 각 커넥터에 대해 각 지원되는 Business Object에 맵을 바인드하려 시도합니다. 여러 개의 맵이 입력으로 같은 Business Object를 사용할 경우, 서버는 다른 맵의 상위 세트인 하나의 맵을 찾아서 바인드하려고 합니다.

다른 맵의 상위 세트인 맵이 없을 경우, 서버는 Business Object를 단일 맵에 바인드할 수 없으므로 사용자가 명시적으로 바인딩을 설정해야 합니다.

맵을 명시적으로 바인드하려면 다음을 수행하십시오.

1. 명시 열에서, 바인드하려는 맵에 대한 선택란에 체크 표시를 하십시오.
2. Business Object와 연관시키려는 맵을 선택하십시오.

3. Connector Configurator 창의 파일 메뉴에서 프로젝트에 저장을 누르십시오.
4. 프로젝트를 ICS에 전개하십시오.
5. 변경사항이 적용되도록 서버를 다시 시동하십시오.

자원(ICS)

자원 탭에서는 커넥터 에이전트가 커넥터 에이전트 병렬화를 사용하여 동시에 다중 프로세스를 처리할 수 있는지 여부와 그 범위를 판별하는 값을 설정할 수 있습니다.

모든 커넥터가 이 기능을 지원하지는 않습니다. 대개 다중 프로세스보다 다중 스레드를 사용하는 것이 더 효율적이므로 다중 스레드되도록 Java로 설계된 커넥터 에이전트를 실행 중인 경우 이 기능을 사용하지 않는 것이 좋습니다.

메시징(ICS)

메시징 등록 정보는 MQ를 DeliveryTransport 표준 등록 정보의 값으로 설정하고 ICS를 브로커 유형으로 설정한 경우에만 사용 가능합니다. 이 등록 정보는 커넥터가 대기열을 사용하는 방법에 영향을 미칩니다.

추적/로그 파일 값 설정

커넥터 구성 파일이나 커넥터 정의 파일을 열 때 Connector Configurator는 해당 파일의 로깅 및 추적 값을 기본값으로 사용합니다. Connector Configurator에서 값을 변경할 수 있습니다.

로깅 및 추적 값을 변경하려면 다음을 수행하십시오.

1. 추적/로그 파일 탭을 누르십시오.
2. 로깅 또는 추적에 대해 다음 중 하나 또는 모두에 메시지를 기록하도록 선택할 수 있습니다.

- 콘솔에(STDOUT):

로깅 또는 추적 메시지를 STDOUT 화면에 기록합니다.

주: Windows 플랫폼에서 실행 중인 커넥터의 경우 추적/로그 파일 탭에서 STDOUT 옵션만을 사용할 수 있습니다.

- 파일에:

로깅 또는 추적 메시지를 사용자가 지정하는 파일에 기록합니다. 파일을 지정하려면, 디렉토리 단추(말줄임표)를 누르고 원하는 위치로 이동한 후 파일 이름을 제공하고 저장을 누르십시오. 로깅 또는 추적 메시지는 사용자가 지정하는 위치 및 파일에 기록됩니다.

주: 로깅 및 추적 파일은 단순한 텍스트 파일입니다. 파일 이름을 설정할 때 원하는 파일 확장자를 사용할 수 있습니다. 그러나 추적 파일의 경우, 사용자

시스템에 상주할 수 있는 다른 파일과의 혼동을 피하기 위해 .trc보다는 .trace 확장자를 사용하는 것이 좋습니다. 파일 로깅의 경우에는 .log 및 .txt가 일반적인 파일 확장자입니다.

Data Handler

Data Handler 섹션은 DeliveryTransport의 JMS 값과 ContainerManagedEvents의 JMS 값을 지정한 경우에만 구성에 사용할 수 있습니다. 모든 어댑터가 Data Handler를 사용하지는 않습니다.

이들 등록 정보에 사용할 값에 대해서는 부록 A, 표준 등록 정보의 ContainerManagedEvents 설명을 참조하십시오. 자세한 정보는 *Connector Development Guide for C++* 또는 *Connector Development Guide for Java*를 참조하십시오.

구성 파일 저장

커넥터 구성이 완료되면 커넥터 구성 파일을 저장하십시오. Connector Configurator는 이 파일을 구성 중에 선택한 브로커 모드로 저장합니다. Connector Configurator의 제목 표시줄에는 현재 사용하고 있는 브로커 모드(ICS, WMQI 또는 WAS)가 항상 표시됩니다.

파일은 XML 문서로 저장됩니다. XML 문서를 다음과 같은 세 가지 방식으로 저장할 수 있습니다.

- System Manager에서, 통합 구성요소 라이브러리에 *.con 확장자를 갖는 파일로 저장.
- 지정하는 디렉토리에 저장.
- 독립형 모드에서, 디렉토리 폴더에 *.cfg 확장자를 갖는 파일로 저장.

System Manager의 프로젝트 사용 및 전개에 대한 자세한 정보는 다음 구현 안내서를 참조하십시오.

- ICS: *Implementation Guide for WebSphere InterChange Server*
- WebSphere Message Brokers의 경우: *Implementing Adapters with WebSphere Message Brokers*
- WAS: *Implementing Adapters with WebSphere Application Server*

구성 파일 변경

기존 구성 파일의 통합 브로커 설정을 변경할 수 있습니다. 그러면 파일을 템플릿로 사용하여 다른 브로커와 함께 사용할 수 있는 새 구성 파일을 작성할 수 있습니다.

주: 통합 브로커를 전환하는 경우에는 브로커 모드 등록 정보 외에 다른 구성 등록 정보를 변경할 필요가 있습니다.

기존 구성 파일 내에서 브로커 선택을 변경하려면 다음을 수행하십시오(선택적).

- Connector Configurator에서 기존 구성 파일을 여십시오.
- 표준 등록 정보 탭을 선택하십시오.
- 표준 등록 정보 탭의 브로커 유형 필드에서 브로커에 적합한 값을 선택하십시오.
현재 값을 변경할 때, 등록 정보 화면의 사용 가능한 탭 및 필드 선택사항은 즉시 변경되어 사용자가 선택한 새 브로커와 관련된 필드와 탭만을 표시합니다.

구성 완료

커넥터에 대한 구성 파일을 작성하고 수정한 후 커넥터가 시작할 때 해당 구성 파일을 찾을 수 있는지 확인하십시오.

이렇게 하려면 커넥터에 사용되는 시작 파일을 열고 커넥터 구성 파일에 사용되는 위치 및 파일 이름이 사용자가 파일에 부여한 이름 및 사용자가 파일을 저장한 디렉토리 또는 경로와 정확하게 일치하는지 확인하십시오.

국제화된 환경에서 Connector Configurator 사용

Connector Configurator는 국제화되었으며 구성 파일과 통합 브로커 사이의 문자 변환을 처리할 수 있습니다. Connector Configurator는 기본 인코딩을 사용합니다. 구성 파일에 기록할 때 UTF-8 인코딩을 사용합니다.

Connector Configurator는 다음 위치에서 영어 이외의 문자를 지원합니다.

- 모든 값 필드
- 로그 파일 및 추적 파일 경로(추적/로그 파일 탭에 지정됨)

CharacterEncoding 및 Locale 표준 구성 등록 정보에 대한 드롭 목록은 지원되는 값의 서브세트만을 표시합니다. 드롭 목록에 다른 값을 추가하려면 제품 디렉토리에 있는 \Data\Std\stdConnProps.xml 파일을 수동으로 수정해야 합니다.

예를 들어, 로케일 en_GB를 Locale 등록 정보에 대한 값 목록에 추가하려면, stdConnProps.xml 파일을 열고 아래의 행을 굵은체 유형으로 추가하십시오,

```
<Property name="Locale" isRequired="true" updateMethod="component restart">
  <ValidType>String</ValidType>
  <ValidValues>
    <Value>ja_JP</Value>
    <Value>ko_KR</Value>
    <Value>zh_CN</Value>
    <Value>zh_TW</Value>
    <Value>fr_FR</Value>
    <Value>de_DE</Value>
    <Value>it_IT</Value>
    <Value>es_ES</Value>
    <Value>pt_BR</Value>
    <Value>en_US</Value>
    <Value>en_GB</Value>
```

```
        <DefaultValue>en_US</DefaultValue>
    </ValidValues>
</Property>
```

주의사항

IBM은 다른 국가에서는 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 언급하는 것이 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 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.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 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 통지없이 언제든지 개선 및(또는) 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 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.

(1) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및
(2) 교환된 정보의 상호 이용을 목적으로 정보를 원하는 프로그램 라이선스 사용자는
다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들어, 사용료 지불 등)에 따라 사용할 수 있습니다.

이 정보에 기술된 라이선스가 있는 프로그램 및 이 프로그램에 대해 사용 가능한 모든 라이선스가 있는 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 레벨 상태의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한, 일부 성능은 추정치일 수도 있으므로 실제 결과는 다를 수 있습니다. 이 문서의 사용자는 해당 데이터를 사용자의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 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 위해 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

IBM의 향후 방향 또는 의도에 관한 모든 언급은 별도의 통지없이 변경될 수 있습니다.

프로그래밍 인터페이스 정보

프로그래밍 인터페이스 정보(제공될 경우)는 이 프로그램을 사용하여 응용프로그램 소프트웨어를 작성하는 것을 돕기 위한 것입니다.

범용 프로그래밍 인터페이스를 사용하면 이 프로그램 도구 서비스를 확보하는 응용프로그램 소프트웨어를 작성할 수 있습니다.

그러나 이 정보는 또한 진단, 수정 및 성능 조정에 대한 정보를 포함할 수 있습니다. 진단, 수정 및 성능 조정에 대한 정보는 사용자의 응용프로그램 소프트웨어를 디버그하는 데 도움을 주기 위해 제공됩니다.

경고: 진단, 수정 및 성능 조정에 대한 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용해서는 안됩니다.

상표 및 서비스표

다음 용어는 미국 또는 기타 국가에서 사용되는 IBM Corporation의 상표 또는 등록 상표입니다.

IBM
IBM 로고
AIX
CrossWorlds
DB2
DB2 Universal Database
Domino
Lotus
Lotus Notes
MQIntegrator
MQSeries
Tivoli
WebSphere

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

MMX, Pentium 및 ProShare는 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.

Java 및 모든 Java 기반의 상표는 Sun Microsystems, Inc.의 상표 또는 등록상표입니다.

기타 회사, 제품 또는 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.



WebSphere Business Integration Adapter Framework V2.4.0

IBM