

WebSphere Business Integration Adapters



Adapter for JMS 사용자 안내서

버전 2.5.x

WebSphere Business Integration Adapters



Adapter for JMS 사용자 안내서

버전 2.5.x

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 99 페이지의 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

이 문서의 이 개정판은 새 개정판에서 별도로 명시하지 않는 한 IBM WebSphere Business Integration Adapter for JMS, 버전 2.5.x 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견은 ibmkspe@kr.ibm.com으로 전자 우편을 보내주십시오. 고객의 의견을 기대합니다.

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.

이 책의 정보

IBM^(R) WebSphere^(R) Business Integration Adapter 포트폴리오는 주요 e-business 기술과 엔터프라이즈 응용프로그램의 통합 연결성을 제공합니다.

이 책에서는 IBM WebSphere Business Integration Adapter for JMS의 설치, 구성, Business Object 개발 및 문제점 해결에 대해 설명합니다.

이 책의 사용자

이 책은 고객의 지역에서 WebSphere Business Integration System을 지원하고 관리하는 컨설턴트, 개발자 및 System Manager를 위한 것입니다.

이 책을 읽기 전에

이 책의 사용자는 WebSphere Business Integration System, Business Object 및 협업 개발, JMS 응용프로그램에 친숙해야 합니다.

관련 서적

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

다음 사이트에서 관련 문서를 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 경우는 다음 웹 사이트를 참조하십시오.
 - <http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- InterChange Server와 함께 어댑터를 사용하는 경우는 다음 웹 사이트를 참조하십시오.
 - <http://www.ibm.com/websphere/integration/wicserver/infocenter>
 - <http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>

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

인쇄 규칙

이 책에서는 다음 규칙을 사용합니다.

<code>courier</code> 글꼴	명령어, 파일 이름, 입력하는 정보 또는 시스템이 화면에 인쇄하는 정보와 같은 리터럴 값을 표시합니다.
굵은체	처음 나타나는 새 용어를 표시합니다.
기울임체	변수 이름이나 상호 참조를 표시합니다.
파란색 윤곽선	온라인으로 매뉴얼을 보는 경우에만 볼 수 있는 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 바로 가려면 윤곽선 내부를 누르십시오.
{ }	구문 행에서, 중괄호는 그 중에서 단 하나만 선택해야 하는 옵션 세트를 묶습니다.
[]	구문 행에서, 대괄호는 선택적 매개변수를 묶습니다.
...	구문 행에서, 말줄임표는 이전 매개변수의 반복을 표시합니다. 예를 들어, <code>option[,...]</code> 은 쉼표로 구분되는 여러 옵션을 입력할 수 있습니다.
< >	이름 지정 규칙에서 꺾쇠 괄호는 <code><server_name><connector_name></code> <code>tmp.log</code> 에서처럼 이름의 각 요소를 서로 구별하기 위해 요소를 각각 묶습니다.
<i>ProductDir</i>	제품이 설치된 디렉토리를 표시합니다.
/, \	이 책에서 백슬래시(\)는 디렉토리 경로의 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체하십시오. 모든 WebSphere Business Integration System 경로 이름은 WebSphere Business Integration System이 시스템에 설치된 디렉토리에 상대적입니다.
UNIX:/Windows:	다음 중 하나로 시작하는 단락은 운영 체제 차이점을 나열하는 참고 사항을 표시합니다.
u	이 기호는 UNIX/Windows 단락의 끝을 표시합니다. 또한 여러 단락으로 된 참고사항 끝을 표시할 수도 있습니다.
<code>%text%</code> 및 <code>\$text</code>	퍼센트(%) 내의 텍스트는 Windows <code>text</code> 시스템 변수나 사용자 변수의 값을 표시합니다. UNIX 환경에서 이에 해당하는 표기는 <code>\$text</code> 로, 이는 <code>text</code> UNIX 환경 변수의 값을 표시합니다.

목차

이 책의 정보	iii
이 책의 사용자	iii
이 책을 읽기 전에	iii
관련 서적	iii
인쇄 규칙	iv
이 릴리스의 새로운 기능	vii
릴리스 2.5의 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 1.3.x의 새로운 기능	viii
릴리스 1.2.x의 새로운 기능	ix
릴리스 1.1.x의 새로운 기능	ix
제 1 장 Adapter for JMS 개요	1
Adapter for JMS 환경	2
Adapter for JMS 용어	3
Connector for JMS 구조	4
메시지 처리	4
제 2 장 어댑터 설치 및 구성	19
설치 작업	19
어댑터 및 관련 파일 설치	19
설치된 파일 구조	19
커넥터 구성	21
커넥터 등록 정보 구성	21
Meta Object 구성	29
시작 스크립트 구성	40
다중 커넥터 인스턴스 작성	40
커넥터 시작	42
커넥터 중지	43
제 3 장 Business Object 작성 또는 수정	45
커넥터 Business Object 구조	45
제 4 장 문제점 해결	47
오류 처리	47
추적	48
시작 문제점 수정	49
부록 A. 커넥터의 표준 구성 등록 정보	51
신규 및 삭제된 등록 정보	51

표준 커넥터 등록 정보 구성	51
표준 등록 정보 요약	53
표준 구성 등록 정보	56
부록 B. Connector Configurator	69
Connector Configurator 개요	69
Connector Configurator 시작	70
System Manager에서 Configurator 실행	71
커넥터 특정 등록 정보 템플릿 작성	71
새 구성 파일 작성	74
기존 파일 사용	75
구성 파일 완료	76
구성 파일 등록 정보 설정	77
구성 파일 저장	84
구성 파일 변경	84
구성 완료	85
국제화된 환경에서 Connector Configurator 사용	85
부록 C. 지습서	87
지습서 개요	87
환경 설정	88
시나리오 실행	90
부록 D. 항목 기반 및 대기열 기반 메시징 구성	95
대기열 기반 메시징 구성	95
항목 기반 메시징 구성	96
주의사항	99
프로그래밍 인터페이스 정보	100
상표 및 서비스표	101

이 릴리스의 새로운 기능

릴리스 2.5의 새로운 기능

이제 어댑터가 통합 브로커로 WebSphere Integration Message Broker를 사용할 수 있습니다. 자세한 정보는 2 페이지의 『브로커 호환성』을 참조하십시오.

2.5.x 버전으로 시작할 때 Adapter for JMS는 Microsoft Windows NT에 지원되지 않습니다.

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

이제 어댑터는 PTP(대기열 기반) 메시징 인터페이스뿐만 아니라 JMS 표준에 의해 정의된 Pub/Sub(항목 기반) 메시징 양식을 지원합니다. 어댑터의 단일 인스턴스는 하나의 메시징 양식만 지원합니다. 따라서 항목 및 대기열은 동일한 구성에 함께 지정될 수 없습니다. 그러나 하나 이상의 항목 기반 메시징 구현 인스턴스와 하나 이상의 대기열 기반 양식 인스턴스를 가진 어댑터의 다중 인스턴스를 실행하면 두 메시징 양식 모두가 지원될 수 있습니다.

릴리스 2.4.x의 새로운 기능

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

커넥터는 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 Handler를 입력 대기열과 연관시킬 수 있습니다. 자세한 정보는 34 페이지의 『Data Handler를 입력 목적지로 맵핑』을 참조하십시오.

보증된 이벤트 전달 기능이 향상되었습니다. 자세한 정보는 *Connector Development Guide for Java*를 참조하십시오.

릴리스 2.2.x의 새로운 기능

진행 중인 대기열이 더 이상 필요 없고 사용 불가능할 수도 있습니다. 자세한 정보는 26 페이지의 『InProgressDestination』을 참조하십시오.

ReplyToQueue가 이제 ReplyToQueue 커넥터 등록 정보가 아니라 동적 하위 Meta Object를 통해 지정됩니다. 자세한 정보는 38 페이지의 『JMS 머리글 및 동적 하위 Meta Object 속성』을 참조하십시오.

어댑터가 주어진 요청에 대해 응답 메시지를 식별하는 방법을 식별 및 필터하고, 아니면 제어하는 데 메시지 선택자를 사용할 수 있습니다. 이 JMS 성능은 동기 요청 처리에만 적용됩니다. 자세한 정보는 14 페이지의 『동기 처리』를 참조하십시오.

릴리스 2.1.x의 새로운 기능

커넥터가 국제화되었습니다. 자세한 정보는 3 페이지의 『로케일 종속 데이터』 및 51 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』 항목을 참조하십시오.

이 안내서에서는 ICS와 함께 이 어댑터를 사용하는 방법에 대한 정보를 제공합니다.

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

릴리스 1.3.x의 새로운 기능

IBM WebSphere Business Integration Adapter for JMS에는 Connector for JMS가 포함되어 있습니다. 이 어댑터는 ICS(InterChange Server) 통합 브로커와 함께 작동합니다. 이기종의 응용프로그램 세트 통합을 수행하는 응용프로그램인 통합 브로커는 데이터 라우팅을 포함하는 서비스를 제공합니다. 어댑터에는 다음 사항이 포함되어 있습니다.

- JMS에 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 매뉴얼은 ICS에서 이 어댑터를 사용하는 방법에 대한 정보를 제공합니다.

중요: 커넥터는 국제화되지 않았으므로 ISO Latin-1 데이터만 처리한다는 것이 확실하지 않은 경우에는 ICS 버전 4.1.1에 대해 커넥터를 실행하지 마십시오.

커넥터는 이제 AIX 4.3.3 패치 레벨 7에 대해 사용 가능합니다.

릴리스 1.2.x의 새로운 기능

이전 버전의 Connector for JMS에서는, JMS 메시지와 IBM CrossWorlds Business Object 사이에 데이터를 변환하기 위해 사용한 Data Handler가 DataHandlerConfigMO 및 DataHandlerMimeType 커넥터 등록 정보에 의해 판별됩니다. 여기에는 상이한 데이터 형식을 처리하기 위해 여러 가지의 커넥터 인스턴스가 필요하다는 제한이 있습니다. 릴리스 1.2.x에서, 커넥터는 이제 커넥터의 정적 Meta Object나 요청 Business Object의 동적 하위 Meta Object에서 이러한 등록 정보를 선택적으로 지정할 수 있도록 허용합니다. 세부사항은 29 페이지의 『Meta Object 구성』을 참조하십시오.

릴리스 1.1.x의 새로운 기능

이 책 릴리스에는 다음과 같은 새로운 기능 및 제품 개선사항에 대한 정보가 수록되어 있습니다.

- 커넥터는 커넥터에 전달되는 Business Object에 하위 요소로 추가된 동적 Meta Object에서 변환 등록 정보를 인식하여 읽습니다. 이 동적 하위 Meta Object의 속성 값은 커넥터를 구성하기 위해 사용되는 정적 Meta Object를 통해 이전에 지정 가능한 변환 등록 정보를 복제합니다. 정적 Meta Object를 지정하는 커넥터 등록 정보는 더 이상 필요하지 않지만 계속 사용할 수는 있습니다. 정적 Meta Object의 영향을 받지 않는 동적 하위 Meta Object, 또는 동적 하위 Meta Object의 영향을 받지 않는 정적 Meta Object를 사용할 수 있습니다.
- 커넥터는 커넥터 등록 정보인 *InputQueue*에 대해 여러 대기열 이름을 승인합니다. 커넥터는 라운드 로빙 방식으로 대기열을 폴링하고 각 대기열에서 *pollQuantity* 개수까지의 메시지를 검색합니다. 대기열 이름이 여러 개일 경우 세미콜론으로 구분합니다.

제 1 장 Adapter for JMS 개요

- 2 페이지의 『Adapter for JMS 환경』
- 3 페이지의 『Adapter for JMS 용어』
- 4 페이지의 『Connector for JMS 구조』
- 4 페이지의 『메시지 처리』

Connector for JMS는 IBM WebSphere Business Integration Adapter for JMS의 런타임 구성요소입니다. 커넥터를 통해 IBM WebSphere 통합 브로커는 Business Object를 JMS 메시지 양식으로 데이터를 전송하거나 수신하는 응용프로그램과 교환할 수 있습니다.

JMS는 엔터프라이즈 메시지 전달 시스템에 액세스하기 위한 개방된 표준 API입니다. 이것은 비즈니스 응용프로그램에서 비즈니스 데이터 및 이벤트를 전송 및 수신할 수 있도록 설계되었습니다.

커넥터는 응용프로그램 특정 구성요소와 커넥터 프레임워크로 구성됩니다. 응용프로그램 특정 구성요소에는 특정 응용프로그램에 맞게 조정된 코드가 있습니다. 코드가 모든 커넥터에 공통되는 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이에 매개체로 작동합니다. 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이에 다음과 같은 서비스를 제공합니다.

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

이 책에는 응용프로그램 특정 구성요소와 커넥터 프레임워크에 대한 정보가 있습니다. 두 구성요소 모두 커넥터로 언급됩니다.

주: 모든 WebSphere Business Integration Adapters는 통합 브로커와 함께 작동합니다. Connector for JMS는 다음과 함께 작동합니다.

- InterChange Server 통합 브로커 - *Technical Introduction to IBM WebSphere InterChange Server*에서 설명합니다.
- WebSphere MQ 메시지 브로커 - *Implementing Adapters with WebSphere Message Brokers*에서 설명합니다.
- WAS(WebSphere Application Server) 통합 브로커 - *Implementing Adapters with WebSphere Application Server*에서 설명합니다.

Adapter for JMS 환경

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

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

브로커 호환성

어댑터에 사용되는 어댑터 프레임워크는 어댑터가 통신하고 있는 통합 브로커의 버전과 호환되어야 합니다. Adapter for JMS 2.5 버전은 다음 어댑터 프레임워크 및 통합 브로커에서 지원됩니다.

- 어댑터 프레임워크: WebSphere Business Integration Adapter Framework, 버전
 - 2.2.0
 - 2.3.0
 - 2.3.1
 - 2.4.0
- 통합 브로커:
 - WebSphere InterChange Server, 버전:
 - 4.11
 - 4.2
 - 4.2.1
 - 4.2.x
 - WebSphere MQ Integrator, 버전 2.1.0
 - WebSphere MQ Integrator Broker, 버전 2.1.0
 - WebSphere Business Integration Message Broker, 버전 5.0
 - WebSphere Studio Application Developer Integration Edition 버전 5.0.1이 있는 WebSphere Application Server Enterprise, 버전 5.0.2

예외는 릴리스 정보를 참조하십시오.

어댑터 표준

어댑터는 JMS 1.0.2 표준에 쓰여집니다. 현재 지원을 불가능하게 하는 알려진 문제점이 없지만 다른 버전의 표준 지원 여부를 확인할 수는 없습니다.

어댑터는 JMS 표준에 정의된 PTP(Point-to-Point) 메시징 및 Pub/Sub(Publish-and-Subscribe) 메시징 인터페이스를 지원합니다. 일반적으로, 이런 양식은 각각 대기열 기반 메시징 및 항목 기반 메시징으로 알려져 있기도 합니다. 어댑터의 단일 인스턴스는 동시에 하나의 메시징 양식만 지원합니다(예를 들어, 구성에서 항목 및 대기열이 함께 있을 수 없음). 그러나 두 가지 메시징 양식은 PTP 또는 Pub/Sub용으로 구성된 인스턴스를 통해 병렬로 어댑터의 다중 인스턴스를 실행하여 지원될 수 있습니다.

어댑터 플랫폼

어댑터는 다음 플랫폼에서 지원됩니다.

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

어댑터 종속성

어댑터는 데이터베이스를 사용하지 않거나 데이터베이스에 종속되지 않습니다. JMS 제공자 및 JNDI 제공자에 필요한 모든 클라이언트 라이브러리는 어댑터 클래스 경로에 포함되어야 합니다. 이런 라이브러리는 제공자별로 다릅니다.

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. 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 정보용 메시지를 기록하려면 사용자 환경에 대한 Locale 표준 구성 등록 정보를 구성하십시오. 구성 등록 정보에 대한 자세한 정보는 51 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

Adapter for JMS 용어

- JMS 제공자 JMS를 구현하는 메시징 시스템.

- 메시지 엔터프라이즈 응용프로그램에서 사용되는 비즈니스 데이터를 포함하는 요청 및 이벤트.
- **PTP** Point-to-Point 양식 또는 대기열 기반 메시징.
- **Pub/Sub** Publish-and-Subscribe 양식 또는 항목 기반 메시징.
- **JMS** 목적지는 메시지의 소스 또는 메시지 대상을 표시합니다. PTP 메시징에서 목적지는 대기열입니다. Pub/Sub에서 목적지는 항목입니다. 이 용어는 주어진 상황에 대기열 또는 항목을 적용할 수 있을 때 설명 및 실제 등록 정보 이름 모두의 스펙에서 광범위하게 사용됩니다.
- **ASI** 응용프로그램 특정 정보. 비즈니스 및 Meta Object에서 세미콜론으로 구분된 이름=값 쌍으로 나타나는 메타 데이터.

Connector for JMS 구조

메시지는 이 어댑터의 컨텍스트에서 엔터프라이즈 응용프로그램에서 사용되는 비즈니스 데이터를 포함하는 요청 및 이벤트입니다. MOM(Message Oriented Middleware) 제품을 사용하면 엔터프라이즈 응용프로그램이 다른 엔터프라이즈 응용프로그램(서) 비동기식으로 메시지를 전송 및 수신할 수 있습니다. JMS(Java Message Service) API는 Java 프로그램이 이런 메시징 시스템과 통신하는 방법을 표준화하기 위해 만들어졌습니다. 과거에 메시징 클라이언트는 종종 특정 단일 MOM 시스템에 대해 작업하기 위해 쓰여졌습니다. 일반적으로 어댑터와 같은 JMS 클라이언트는 JMS 지원을 제공하는 메시징 시스템을 이용할 수 있습니다. WBI adapter for JMS를 사용하면 JMS 표준을 지원하는 증가하는 수의 엔터프라이즈 메시징 시스템을 통합할 수 있습니다.

메시지 처리

어댑터는 두 가지 기본 조작을 지원합니다.

1. JMS 목적지에서 메시지 검색
2. JMS 목적지로 메시지 전달

어댑터는 JMS 제공자(예: WebSphere MQ)와의 연결을 수립한 후 JMS API를 사용하여 다음을 수행합니다.

- JMS 목적지에서 기존 메시지 폴링 및 검색
- 브로커에서 요청한 새 메시지 생성 및 전달

이런 두 가지 조작은 『이벤트 메시지 처리』 및 11 페이지의 『요청 메시지 처리』에 자세히 설명되어 있습니다.

이벤트 메시지 처리

커넥터는 하나 이상의 JMS 목적지로 전달된 새 메시지를 주기적으로 확인합니다. 각각의 폴 주기 동안 커넥터는 다음을 수행합니다.

1. JMS API를 사용하여 대기 메시지를 검색합니다.
2. 구성된 Data Handler를 호출하여 메시지 내용을 Business Object로 변환합니다.
3. 등록 비즈니스 프로세스에서 처리하기 위해 이벤트 Business Object를 구성된 통합 브로커에 전달하거나 공개합니다.

이런 단계는 그림 1에 나와 있고 다음에 자세히 설명되어 있습니다.

- 『이벤트 발견』
- 6 페이지의 『이벤트 상태 및 복구』
- 8 페이지의 『이벤트 검색』

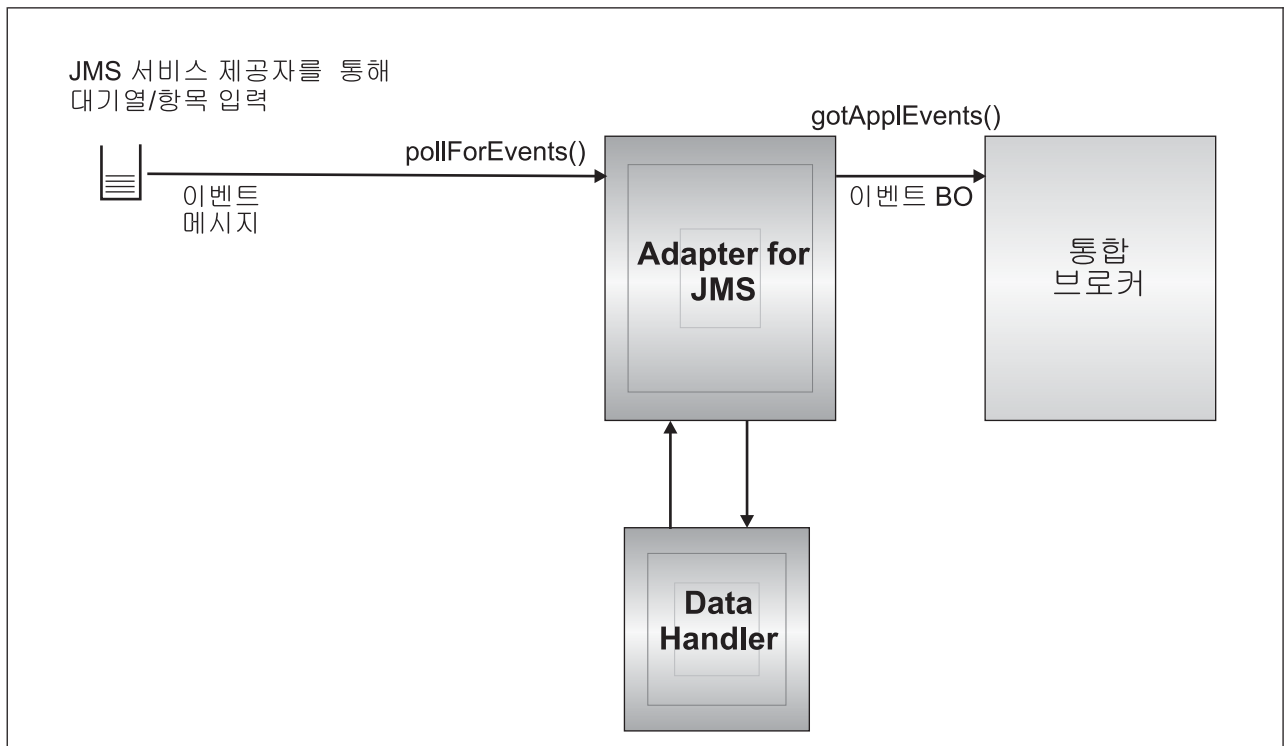


그림 1. 이벤트 메시지 플로우

이벤트 발견

각 이벤트 폴링 주기 동안 커넥터는 커넥터 등록 정보 `InputDestination`에 지정된 목적지에서 비블로킹 메시지 읽기를 수행합니다(커넥터 등록 정보에 대한 자세한 정보는 21 페이지의 『커넥터 등록 정보 구성』 참조). 커넥터는 메시지를 검색한 후 이를 브로커에 공개합니다.

커넥터는 `pollForEvents()` 메소드를 사용하여 메시지에 대해 정기적으로 폴링합니다. 각 폴 주기 동안 메시지 검색은 커넥터 등록 정보 `PollQuantity`에 지정된 최대 수로 제

한됩니다. 지정된 최대값에 도달하기 전에 사용 가능한 모든 메시지를 검색한 경우 커넥터는 더 이상 메시지를 기다리지 않지만 대신 폴 주기에서 즉시 리턴합니다.

커넥터 등록 정보 `InputDestination`에 다중 목적지가 지정된 경우 커넥터는 지정된 각 목적지를 라운드 로빈 방식으로 폴링합니다. 각 목적지에서 최대 `PollQuantity` 수의 메시지를 검색하고 브로커에 공개합니다. `PollQuantity`에 지정된 최대값에 도달하기 전에 모든 목적지를 비운 경우 커넥터는 폴 주기에서 즉시 리턴합니다.

예를 들어, 다음과 같은 시나리오에서

- 커넥터는 `PollQuantity` 값 2와 입력 대기열 A, B 및 C로 구성됩니다.
- 대기열 A에는 2개의 메시지가 포함되어 있고 대기열 B에는 1개의 메시지가 포함되어 있으며 대기열 C에는 5개의 메시지가 포함되어 있습니다.

어댑터는 단일 폴 주기에서 다음 순서로 메시지를 검색할 수 있습니다.

1. 대기열 A에서 다음 메시지(남아 있는 1개의 메시지는 그대로 둠)
2. 대기열 B에서 다음 메시지(현재 비어 있음)
3. 대기열 C에서 다음 메시지(남아 있는 4개의 메시지는 그대로 둠)
4. 대기열 A에서 다음 메시지(현재 비어 있음)
5. 커넥터가 대기열 B를 검사하지만 대기열 B는 여전히 비어 있습니다.
6. 대기열 C에서 다음 메시지(남아 있는 3개의 메시지는 그대로 둠)

현재 최대값(`PollQuantity`에서 설정된)인 2개의 메시지가 각 대기열에서 폴링되었으므로 어댑터는 폴링 주기에서 리턴합니다.

이벤트 상태 및 복구

이벤트 메시지 검색은 트랜잭션의 일부입니다. 트랜잭션을 커밋하기 전에 커넥터가 예상치 않게 종료된 경우 트랜잭션이 롤백되고 원래 메시지가 복원됩니다. 커넥터 프레임워크가 현재 분산 트랜잭션을 지원하지 않으므로 커넥터가 이벤트를 브로커에 공개할 수 있지만 예상치 않게 종료되거나 브로커에서 승인을 수신하기 전에 통신이 끊어집니다. 이런 경우에 브로커가 이벤트를 수신했는지 여부는 커넥터에 사용 가능한 정보를 통해 판별할 수 없습니다. 이벤트 메시지 유실을 피할 수 있는 방법은 커넥터가 브로커에서 이벤트 수신을 확인하는 응답을 수신할 때까지 커넥터가 트랜잭션을 커밋하지 않는 것입니다. 커넥터가 이벤트를 공개한 시간과 승인을 수신한 시간 사이에 실패가 발생한 경우 트랜잭션이 자동으로 롤백되고 원래 메시지가 복원됩니다. 브로커가 메시지를 처리했는지 여부를 알 수 없으므로 이런 이벤트를 `In-doubt` 이벤트라고 합니다.

재시작 직후, 커넥터는 입력 목적지에서 메시지를 처리하기 시작하고 `In-doubt` 이벤트를 다시 제출합니다. 이 전략으로 이벤트가 손실될 위험이 없어진다 하더라도 동일한 이벤트가 두 번 공개되는 것을 막을 수는 없습니다.

진행 중인 목적지 사용(『진행 중인 목적지를 통한 복구』 참조) 또는 보증된 이벤트 전달(『보증된 이벤트 전달을 통한 복구』 참조)을 통해 중복 이벤트 전달의 위험을 줄이거나 제거할 수 있습니다.

진행 중인 목적지를 통한 복구: In-doubt 이벤트가 처리되는 방법을 제어하기 위해 커넥터 등록 정보 InProgressDestination을 지정하여 별도의 임시 목적지를 작성할 수 있습니다.

주: 진행 중인 목적지에 대한 복구는 Pub/Sub 양식 메시징에 지원되지 않습니다.

이벤트를 브로커에 공개하기 전에, 커넥터는 이벤트 메시지를 입력 목적지에서 진행 중인 목적지로 이동시킬 수 있습니다. 일단 브로커에서 승인을 수신하면, 커넥터는 진행 중인 목적지에서 그 메시지를 제거합니다. 이것은 처리되지 않은 In-doubt 메시지를 분리시킵니다. 시작 직후, 커넥터가 진행 중인 목적지에서 메시지를 찾을 경우 커넥터는 이런 메시지가 예상치 않게 종료된 커넥터의 이전 인스턴스에서 남겨진 것이라고 생각할 수 있습니다. 그런 메시지를 처리하기 위해 커넥터에 대해 여러 조치를 지정할 수 있습니다(중복 이벤트 공고를 승인할 수 없는 경우). 다음과 같은 네 가지 커넥터 구성 등록 정보 옵션 중 하나를 지정하여 이를 수행할 수 있습니다.

- 시작 시 실패 커넥터가 초기화 도중 진행 중인 목적지에서 메시지를 찾을 경우, 커넥터는 오류를 기록하고 즉시 종료됩니다. 그 다음 사용자 또는 시스템 관리자가 메시지를 조사하고 적절한 조치(이런 메시지를 완전히 삭제하거나 다른 위치로 이동시킴)를 취합니다.
- 재처리 커넥터가 초기화 도중 진행 중인 목적지에서 메시지를 찾을 경우, 커넥터는 후속 폴링 동안 이런 메시지를 처리합니다. 진행 중인 대기열의 모든 메시지가 처리되면, 커넥터는 입력 대기열에서 메시지를 처리하기 시작합니다.
- 무시 커넥터가 초기화 도중 진행 중인 목적지에서 메시지를 찾을 경우, 커넥터가 이 메시지를 무시하지만 종료되지는 않습니다.
- 오류 로그 오류 오그 옵션을 사용하면 커넥터가 초기화 도중 진행 중인 목적지에서 메시지를 찾을 경우, 오류를 기록하지만 종료되지는 않습니다.

자세한 정보는 25 페이지의 『InDoubtEvents』를 참조하십시오.

보증된 이벤트 전달을 통한 복구: 보증된 이벤트 전달 기능을 통해 커넥터 프레임워크는 이벤트가 유실되지 않으며 두 번 전송되지 않도록 보장할 수 있습니다. 커넥터 프레임워크는 컨테이너 관리 이벤트(CME) 및 중복 이벤트 제거(DEE)의 두 메커니즘을 통해 보증된 이벤트 전달을 지원합니다.

컨테이너 관리 이벤트(CME): 커넥터가 PTP 양식 메시징용으로 구성되어 있으면 CME를 사용할 수 있으며 Pub/Sub 양식 메시징용으로 구성되어 있으면 커넥터는 CME를 지원하지 않습니다. 이런 보증된 이벤트 전달 방법이 어떻게 작동하는지에 대한 자세한 정보는 *Connector Development Guide for Java*를 참조하십시오.

ContainerManagedEvents 커넥터 등록 정보에 대한 자세한 정보는 58 페이지의 『ContainerManagedEvents』를 참조하십시오.

중복 이벤트 제거(DEE): DEE는 JMS 어댑터에 보증된 이벤트 전달을 구현하는 데 권장되는 접근 방법입니다. DEE는 Pub/Sub 양식 메시징에도 지원되는 유일한 접근 방법입니다.

커넥터는 DEE를 통해 브로커에 공개하는 각 이벤트에 고유 ID를 포함시킵니다. 프레임워크는 커넥터가 동일한 이벤트 ID를 연속적으로 제출하지 않는 지를 확인합니다. 이것이 발생한 경우 프레임워크는 커넥터가 동일한 이벤트를 두 번 공개하는 중이고 두 번째 제출을 버린 것으로 간주합니다. PTP 양식 메시징의 경우 DEE는 진행 중인 목적지(서) 메시지를 복사하는 데에 관련되는 실질적인 오버헤드를 감소시킵니다.

이 커넥터는 Business Object를 브로커에 공개할 때 모든 이벤트의 메시지 ID를 포함합니다. 커넥터가 통신 실패 또는 예상치 않은 종료로 인해 이벤트를 브로커에 성공적으로 게시하는 데 실패한 경우 이전에 설명된 대로 원래 메시지가 입력 대기열에 롤백됩니다. 다시 시작하자마자, 커넥터는 In-doubt 메시지를 포함하여 대기열에서 이벤트를 다시 제출하기 시작합니다. DEE를 사용할 수 있는 경우 이전에 성공적으로 브로커에 도달한 In-doubt 메시지는 버려집니다. 이것은 각 메시지가 브로커에 한 번만 게시되는 것을 보장합니다.

DEE를 사용하는 경우, 커넥터가 오프라인인 동안에 목적지에서 메시지 순서를 조작하는 것은 피해야 합니다. DEE는 어댑터에서 검색한 마지막 메시지 ID만을 기록합니다. 예를 들어, 어댑터가 다시 시작할 수 있기 전에 더 높은 우선순위를 가진 새 메시지가 마지막 In-doubt 메시지를 푸시다운했을 경우에 DEE가 실패합니다.

DEE와 DEE를 사용 가능하게 하는 데 대한 정보는 *Connector Development Guide for Java*를 참조하십시오. DuplicateEventElimination 커넥터 등록 정보에 대한 자세한 정보는 60 페이지의 『DuplicateEventElimination』을 참조하십시오.

이벤트 검색

각 검색은 커넥터에 의해 일반적인 이벤트 처리를 수행합니다. 검색이 대상 응용프로그램에 적합한 형식으로 변환되고 지정된 통합 브로커에 성공적으로 전달된 경우에 수신 이벤트가 발견되고 종료될 때 검색이 시작됩니다. 커넥터는 모든 이벤트를 비동기적으로(“fire and forget”) 브로커에게 전달합니다.

다음 섹션에서 이벤트 검색에 대해 논의합니다.

- 9 페이지의 『메타 데이터 및 Meta Object』
- 9 페이지의 『Business Object 맵핑』
- 10 페이지의 『메시지 머리글 맵핑 이해』
- 11 페이지의 『아카이브』
- 11 페이지의 『오류 복구』

메타 데이터 및 Meta Object: 커넥터가 메시지를 Business Object로 변환하거나 그 반대의 경우를 성공적으로 수행하려면 메타 데이터로 알려진 추가 정보가 필요합니다. 메타 데이터는 오브젝트, 메시지 또는 응용프로그램에서 데이터가 표시되거나 처리되는 방법에 대해 설명합니다. 메타 데이터에는 커넥터가 목적지 XYZ에서 메시지를 검색하는 경우 어떤 Business Object가 작성되는지 또는 Verb가 Create인 Customer 유형의 요청 Business Object를 순차적으로 정렬하기 위해 어떤 Data Handler가 사용되어야 하는지와 같은 세부사항이 포함됩니다.

속성, 등록 정보, Verb 및 응용프로그램 특정 정보가 Business Object 정의에 대한 메타 데이터를 구성합니다. 또한, 목적지, 데이터 형식, Data Handler 등에 대한 메타 데이터를 포함하는 하나 또는 여러 개의 Meta Object를 지정할 수 있습니다.

정적 및 동적 유형의 Meta Object가 있습니다. 구현 중에 정적 Meta Object를 작성할 수 있습니다. 여기에는 커넥터가 지원해야 하는 각 Business Object 유형에 대한 메타 데이터를 제공하는 속성이 포함됩니다. 정적 Meta Object는 커넥터 특정 등록 정보에 지정되며 초기화 도중 커넥터에서 읽혀집니다. Meta Object 등록 정보 개요와 이 등록 정보가 메시지 전환에 어떤 영향을 미치는지에 대한 정보는 『Business Object 맵핑』 및 10 페이지의 『메시지 머릿글 맵핑 이해』를 참조하십시오.

Meta Object의 두 번째 유형은 동적 Meta Object입니다. 이 Meta Object를 사용하면 요청 처리 중에 요청 기반으로 Business Object를 처리하기 위해 어댑터에서 사용되는 메타 데이터를 변경할 수 있습니다. 이벤트 처리 동안, 동적 Meta Object는 이벤트에 대한 전송 특정 정보(예를 들어, 메시지 ID, 우선순위 등)를 보유하는 컨네이터 역할을 하여 다운스트림 비즈니스 프로세스가 비즈니스 논리에 그 정보를 사용할 수 있도록 합니다. 동적 Meta Object는 이벤트(또는 요청) 최상위 레벨 오브젝트에 정의되어 있는 특별히 표시된 하위 오브젝트로 표시됩니다.

동일한 구현에 하나 또는 두 가지 유형의 Meta Object를 사용하도록 선택할 수 있습니다. 동적 Meta Object에 제공된 값은 일반적으로 정적 Meta Object에 제공된 값에 우선합니다. 메타 데이터에 대한 자세한 정보는 *Connector Development Guide for Java*를 참조하십시오. 정적 및 동적 Meta Object 구성에 대한 자세한 정보는 29 페이지의 『Meta Object 구성』을 참조하십시오.

Business Object 맵핑: 메시지 검색 직후, 커넥터는 메시지가 맵핑해야 하는 Business Object를 식별하려고 시도합니다.

기본적으로, 커넥터를 통해 커넥터 등록 정보에 구성된 Data Handler가 Business Object 유형을 판별할 수 있습니다. 커넥터는 메시지 본문을 Data Handler에게 전달하고 Data Handler가 리턴하는 Business Object를 브로커에게 공개합니다. Data Handler가 적절한 Business Object를 판별할 수 없는 경우 커넥터는 이벤트를 실패 처리합니다.

정적 Meta Object가 ConfigurationMetaObject 커넥터 구성 등록 정보에 지정된 경우, 커넥터는 입력 형식 또는 입력 목적지에 관해 메시지를 일치시키는 규칙을 찾기 위해

이 오브젝트를 검색합니다. Meta Object에 지정된 규칙에서 입력 형식 및 입력 목적지 모두를 지정하면 커넥터는 메시지가 두 등록 정보 모두와 일치하는 경우에만 이 규칙을 준수합니다. 두 등록 정보 중 하나가 누락된 경우 커넥터는 지정된 등록 정보만을 사용합니다.

예를 들어, 입력 형식이 Cust_In인 입력 목적지 MyInputDest로부터의 메시지는 다음 정적 Meta Object 규칙에 일치됩니다.

1. InputFormat=Cust_In;InputDestination=MyInputDest
2. InputFormat=Cust_In
3. InputDestination=MyInputDest

이벤트 메시지를 단일 규칙에 일치시킬 수 있는 경우, 커넥터는 이 Business Object의 새 인스턴스를 작성하고 이 인스턴스를 메시지 본문과 함께 규칙에서 지정된 Data Handler에게 전달하여 Business Object를 받아 쓰게 합니다. 규칙에 Data Handler가 지정되지 않은 경우, 커넥터는 커넥터 구성 등록 정보에 지정된 기본 Data Handler를 사용합니다.

어댑터가 이벤트 메시지를 다중 규칙에 일치시키거나 어떤 규칙에도 일치시키지 않는 경우, 커넥터는 메시지 본문만을 커넥터 구성 등록 정보에 지정된 Data Handler에게 전달하여 Data Handler가 Business Object 유형을 판별할 수 있도록 합니다.

메시지 머리글 맵핑 이해: 이벤트 메시지를 Business Object로 변환하기 위해 커넥터는 맵핑을 통해 Business Object에 대한 메타 데이터를 메시지에 대한 메타 데이터와 비교합니다. 9 페이지의 『메타 데이터 및 Meta Object』에 설명된 대로 Business Object에 대한 메타 데이터는 Business Object 정의(하위 동적 Meta Object와 응용프로그램 특정 정보), 커넥터 등록 정보 및 정적 Meta Object에 있습니다. 메시지 메타 데이터가 메시지 머리글에 포함됩니다.

전송 특정 메시지 머리글 정보와 메시지 전송에 대한 자세한 정보 및 더 많은 제어를 얻기 위해 Business Object 정의의 하위인 동적 Meta Object에 속성을 추가할 수 있습니다. 이런 속성을 추가하면 메시지 머리글에서 읽고 선택적으로 메시지 머리글에 쓸 수 있으며 이로 인해 메시지 메타 데이터를 수정할 수 있습니다. 이런 수정에는 JMS 등록 정보 변경, 요청별로 목적지 제어(어댑터 등록 정보에 지정된 기본 목적지를 사용하지 않고), 메시지 CorrelationID를 다시 대상으로 지정 등의 작업이 포함됩니다. Business Object 정의의 하위인 동적 Meta Object에 이런 등록 정보를 지정할 때, 커넥터는 메시지 머리글에서 대응 파트를 확인한 후 메시지 머리글의 내용에 따라 동적 Meta Object를 채웁니다. 지원되는 동적 Meta Object 속성을 하나 또는 모두 정의할 수 있습니다. 커넥터는 이에 따라 Meta Object를 채웁니다. 읽거나 쓸 수 있는 메시지 머리글 등록 정보 목록을 포함한 자세한 정보는 37 페이지의 『폴링 중에 동적 하위 Meta Object 채우기』를 참조하십시오.

아카이브: ArchiveDestination 커넥터 특정 등록 정보를 지정한 경우, 커넥터는 성공적으로 처리된 메시지 모두의 사본을 이 목적지에 가져다 놓습니다. ArchiveDestination 이 정의되어 있지 않으면 성공적으로 처리된 메시지는 버려집니다. 자세한 정보는 22 페이지의 『커넥터 특정 등록 정보 구성』을 참조하십시오.

오류 복구: 입력 목적지에서 읽는 중에 오류가 발생하면 커넥터는 즉시 APPRESPONSETIMEOUT 상수 값을 브로커에 리턴하여 커넥터를 종료하고 가능한 커넥터를 다시 시작하게 합니다. 일반적으로, 복구할 수 없는 오류의 원인은 JMS 제공자와의 연결이 끊어졌거나 JMS 제공자에 의해 보고되는 내부 오류입니다. 내부 오류는 커넥터가 인식하지 못하거나 인식하지만 복구할 수 없는 것으로 간주하는 오류(예 : 트랜잭션 실패)를 말합니다.

인바운드 메시지를 이벤트 Business Object로 변환하는 경우(예를 들어, Dtheta Handler 가 올바르게 읽은 메시지 형식을 보고하는 경우), 커넥터는 그 이벤트를 실패 처리하고 그 이유에 대해하는 적절한 오류 메시지를 로그합니다. ErrorDestination 커넥터 등록 정보가 정의되고 올바른 경우, 커넥터는 이 오류 목적지에 실패한 메시지의 사본을 가져다 놓습니다. 그렇지 않은 경우, 그 메시지는 버려집니다.

커넥터가 이벤트 Business Object를 공개한 후에 브로커에서 오류를 보고하는 경우, 커넥터는 이벤트를 실패 처리하고 브로커에서 보고한 오류 메시지를 로그합니다. ErrorDestination 커넥터 등록 정보가 정의되고 올바른 경우, 커넥터는 이 목적지에 실패한 메시지의 사본을 가져다 놓습니다. 그렇지 않은 경우, 그 메시지는 버려집니다.

메시지의 Business Object를 판별하기 불가능한 경우 또는 메시지를 브로커에 공개하고 브로커에서 그 메시지가 지원되지 않는다고 보고한 경우, 커넥터는 그 메시지가 등록되지 않은 것으로 간주합니다. UnsubscribedDestination 커넥터 등록 정보가 정의되고 올바른 경우, 커넥터는 이 목적지에 등록되지 않은 메시지의 사본을 가져다 놓습니다. 그렇지 않은 경우, 그 메시지는 버려집니다.

요청 메시지 처리

Business Object 요청이 커넥터에 전송될 때 커넥터는 대상 목적지에 새 메시지를 작성합니다. 메시지 머릿글이 요청 Meta Objects에서 지정된 사용자 정의 값과 커넥터 등록 정보에서 지정된 기본 매개변수의 조합으로 채워집니다. 메시지의 본문은 구성된 Data Handler를 통해 요청 Business Object를 전달하여 생성된 결과 내용으로 채워집니다.

그림 2에서는 메시지 요청 통신을 보여줍니다. doVerbFor() 메소드가 브로커에서 Business Object를 수신할 때, 커넥터는 Business Object를 Data Handler에 전달합니다. Data Handler는 Business Object를 적절한 메시지로 변환하고 커넥터는 이를 메시지로 목적지에 발행합니다.

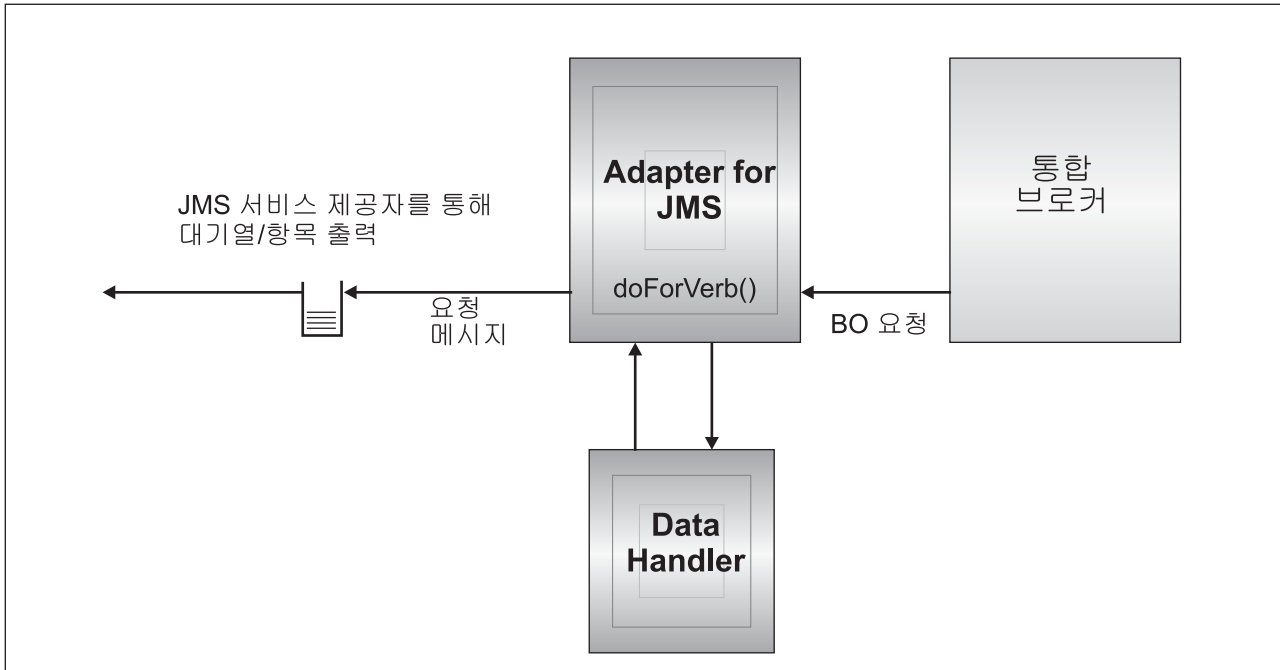


그림 2. 요청 플로우

요청 처리 중에 커넥터가 취할 수 있는 조치에는 두 가지 유형이 있습니다. 첫 번째로 아래에서 비동기 처리로 설명된 대로, 커넥터는 메시지를 대상 목적지에 가져다 놓고 성공적으로 리턴시킵니다. 일반적으로 이것을 ‘fire-and-forget’이라고 부릅니다. 두 번째로 아래에서 동기 처리로 설명된 대로, 커넥터는 메시지를 대상 목적지에 가져다 놓지만 대상 응용프로그램에서 리턴되는 응답을 기다립니다.

처리 모드는 ResponseTimeout 숫자 등록 정보에 의해 판별됩니다. 이 등록 정보는 Business Object 요청에 대한 동적 또는 정적 Meta Object에서 지정됩니다. 이 등록 정보가 정의되지 않았거나 그 값이 -1인 경우, 커넥터는 요청을 비동기식으로 전달합니다. 이 등록 정보가 0 또는 그 이상인 경우, 어댑터는 요청을 동기식으로 처리하여 최소한 다수의 밀리초 동안 대상 응용프로그램에서 응답 메시지가 리턴되기를 기다립니다. 그림 2에 설명된 요청 처리는 다음에서 자세히 설명됩니다.

- 『Verb 지원』
- 13 페이지의 『비동기 처리』
- 14 페이지의 『동기 처리』

Verb 지원

커넥터는 요청 Business Object에 정의된 Verb에 의미가 있는 값을 두지 않습니다. 지정된 Verb에 관계없이 동일한 조치를 수행합니다. 다시 말해서, JMS 목적지에 메시지를 가져다 놓습니다.

비동기 처리

비동기 처리에서, 커넥터는 요청 Business Object를 메시지로 변환하고 그 메시지를 대상 목적지에 가져다 놓은 후 브로커에게 즉시 리턴합니다. 요청의 성공 또는 실패는 전적으로 해당 메시지를 JMS 목적지에 가져다 놓는 커넥터의 성능을 기반으로 합니다. 이 전달의 성공이 대상 응용프로그램이 그 메시지를 수신했거나 수신할 것이라는 것을 의미하지는 않음에 유의하십시오. 메시징 시스템의 비동기 특성으로 인해 메시지는 대상 응용프로그램에 이를 처리하거나 폐기시킬 때까지(그렇게 구성된 경우) JMS 제공자에 무기한으로 남아 있을 수 있습니다.

커넥터는 우선 구성된 Data Handler를 사용하여 요청 Business Object를 텍스트로 순차적으로 정렬합니다. 커넥터는 다음의 환경 설정을 위해 지정된 Data Handler를 사용합니다.

1. 동적 Meta Object
2. 정적 Meta Object
3. 커넥터 구성 등록 정보

커넥터는 순차적으로 정렬된 Business Object 데이터를 메시지 본문으로 포함하고 있는 새 메시지를 작성합니다. 다음 테이블에 설명된 대로 메시지 머리글을 채웁니다. 등록 정보가 동적 또는 정적 Meta Object에 지정될 수 있는 모든 경우에 동적 Meta Object에 지정된 값이 정적 Meta Object에 지정된 값보다 우선합니다. Meta Object를 지정할 수 있는 등록 정보 목록과 설명은 29 페이지의 『Meta Object 구성』을 참조하십시오.

표 1. 비동기 요청 처리 중 JMS 메시지 머리글 채우기

Meta Object 등록 정보	등록 정보가 정의되지 않은 경우의 기본 조치	등록 정보가 정의된 경우 취하는 조치
OutputFormat	커넥터가 메시지 형식을 지정하지 않습니다.	커넥터가 메시지 형식에 이 값을 지정합니다.
CorrelationID	커넥터는 메시지 머리글에서 이 값을 공백으로 남겨둡니다.	커넥터는 요청 메시지 머리글의 Correlation ID에 이 값을 지정합니다.
ReplyToDestination	커넥터는 메시지 머리글에서 이 값을 공백으로 남겨둡니다.	커넥터는 요청 메시지 머리글의 응답 목적지에 이 값을 지정합니다.
Priority	커넥터는 JMS 제공자가 기본 우선순위를 사용할 수 있도록 합니다.	커넥터는 이 값을 사용하여 숫자 메시지 우선순위를 설정합니다.
JMSProperties	없음	커넥터는 지정된 JMS 등록 정보를 메시지 머리글의 JMS 등록 정보로 맵핑합니다.

Meta Object의 다음 속성은 메시지가 전달되는 방법을 판별합니다.

표 2. 목적지로의 비동기 전달

Meta Object 등록 정보	등록 정보가 정의되지 않은 경우의 기본 조치	등록 정보가 정의된 경우 취하는 조치
OutputDestination	값은 필수입니다.	메시지의 대상 목적지

표 2. 목적지로의 비동기 전달 (계속)

DeliveryMode	커넥터는 JMS 제공자가 메시지 지속성을 지시할 수 있도록 합니다.	커넥터는 사용자가 표시하는 대로 지속적으로/비지속적으로 메시지를 씁니다.
--------------	---------------------------------------	--

출력(대상) 목적지로 요청 메시지를 성공적으로 전달하는 커넥터의 성능에 따라 다음 코드 중 하나가 브로커에 리턴됩니다.

표 3. 비동기 리턴 코드

커넥터 조치	브로커로의 리턴 코드
성공적으로 메시지를 대상 목적지에 전달합니다.	SUCCEED
부적절하거나 미완료된 메타 데이터, Data Handler의 실패 또는 일반 처리 문제점과 같이 복구할 수 있는 오류로 인해 전달에 실패했습니다.	FAIL
연결 실패와 같이 JMS 제공자에서 보고되는 복구할 수 없는 오류로 인해 전달에 실패했습니다.	APPRESPONSETIMEOUT

동기 처리

동기 처리에서, 커넥터는 요청을 대상 목적지에 전달한 후 잠시 목적지에서 응답 메시지를 기다립니다. 요청 메시지 작성은 비동기 처리에서 설명된 것과 동일합니다. 그러나 커넥터는 Meta Object에서 다음 추가 속성을 확인하기도 합니다.

표 4. 동기 Meta Object 등록 정보

Meta Object 등록 정보	등록 정보가 정의되지 않은 경우의 기본 조치	등록 정보가 정의된 경우 취하는 조치
ResponseTimeout	값은 필수입니다.	어댑터가 리턴될 응답 메시지를 기다려야 하는 최소 시간 (밀리초)
TimeoutFatal	ResponseTimeout에 지정된 시간까지 응답을 수신하지 않으면 커넥터는 브로커에 APPRESPONSETIMEOUT을 리턴합니다. 일반적으로, 이로 인해 커넥터가 종료됩니다.	응답을 수신하지 않으면 커넥터는 요청을 실패 처리하지만 (브로커에 FAIL 리턴) 종료되지는 않습니다.

대상 목적지에 메시지를 전달하는 것은 다음을 제외하고 비동기 처리에서 설명한 것과 동일합니다.

표 5. 목적지로의 동기 전달

Meta Object 등록 정보	등록 정보가 정의되지 않은 경우의 기본 조치	등록 정보가 정의된 경우 취하는 조치
ReplyToDestination	비동기의 경우와 동일합니다.	커넥터는 요청 메시지의 이 필드를 커넥터 특정 등록 정보 ReplyToDestination의 값으로 채웁니다.

커넥터는 최소한 ResponseTimeout Meta Object 속성에서 지정된 시간 동안 ReplyToDestination에서 지정된 대상 목적지로부터의 응답 메시지를 기다립니다. 그 시간 안에 응답이 리턴되지 않으면 커넥터는 시간 종료로 되고 오류를 보고합니다.

응답 기준: 커넥터는 응답 목적지의 처음 메시지가 올바른 응답 메시지라고 간주하지 않습니다. 대신, JMS 요청-응답 규칙을 따르고 요청의 메시지 ID와 일치하는 Correlation ID가 있는 첫 번째 메시지를 찾습니다. 다시 말해, 요청 메시지를 수신하는 응용프로그램은 Correlation ID가 요청 메시지 ID와 같은 응답 메시지를 작성해야 하고 요청 메시지에서 지정된 응답 목적지에 해당 메시지를 가져다 놓아야 합니다.

모든 응용프로그램이 요청 및 응답 메시지를 맵핑하기 위해 Correlation ID 사용 규칙을 따르는 것은 아닙니다. 그런 경우에, 커넥터는 응답 메시지 식별을 위한 사용자 정의 기준을 승인합니다.

동기 요청 처리에 대한 Business Object를 수신하자마자 커넥터는 Verb의 응용프로그램 특정 정보에 있는 response_selector= 이름-값 쌍의 존재를 확인합니다. 이런 이름-값 쌍이 존재하지 않으면 커넥터는 위에 설명된 Correlation ID를 사용하여 응답 메시지를 식별합니다.

응답 선택자 이름-값 쌍이 정의되면 커넥터는 그 값이 응답 메시지를 식별할 수 있는 JMS 메시지 선택자 문자열을 표시하는 것으로 간주합니다. 다음은 사용법 예입니다. JMS 메시지 선택자 구문에 대한 자세한 정보는 JMS API 스펙을 참조하십시오. JMS 메시지 선택자 구문은 커넥터에 의해 구문 분석되지 않음에 주의하십시오. 오히려, JMS 제공자가 이 구문을 이해합니다. 이 커넥터를 사용하면 JMS 제공자에서 메시지 필터링(데이터베이스 조회와 유사함) 수단으로 선택자를 사용할 수 있게 합니다.

예를 들어, 이름-값 쌍을 포함하는 Verb 응용프로그램 특정 정보는

```
response_selector=JMSType = 'xmlResponse'
```

응답 메시지가 선택자 문자열 JMSType = 'xmlResponse'에 일치해야 함을 커넥터에게 알립니다. 커넥터는 JMS 제공자에게 이 선택자를 제공하여 메시지의 JMS 유형 필드가 xmlResponse와 같은 응답 목적지에 처음 전달된 메시지를 리턴합니다.

모든 경우에, 메시지 선택자 문자열은 고유하게 하나의 응답만을 식별할 수 있어야 합니다. 다중 메시지가 응답 선택자의 기준을 충족하는 응답 목적지에 전달된 경우, 어댑터는 첫 번째 메시지만을 검색할 것입니다. 기준에 일치하는 잠재적인 다른 응답 메시지는 무시될 것입니다.

런타임 시 고유 메시지 선택자를 허용하기 위해 커넥터는 속성 값을 메시지 선택자 자체로 동적 대체를 할 수 있습니다. 이를 수행하려면 응답 선택자에 중괄호로 묶은 정수 형식("{1}")으로 위치 표시기를 지정해야 합니다. 그런 다음 콜론을 추가하고 대체에 사용할 속성 목록을 쉼표로 구분하여 붙여야 합니다. 위치 표시기에 있는 정수가 대체에 사용할 속성에 대한 색인 역할을 합니다.

예를 들어, 다음 메시지 선택자는

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

토큰 {1}이 MyDynamicMO 하위 오브젝트의 CorrelationID 속성 값으로 대체되었음을 커넥터에게 알립니다. CorrelationID 속성의 값이 123ABC인 경우, 커넥터는 메시지 선택자를 생성하고 사용합니다.

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

아래와 같이 여러 대체를 지정할 수도 있습니다.

```
response_selector=Name LIKE '{1}'AND Zip LIKE '{2}':PrimaryID,Address[4].AddressID
```

이 예에서는, 커넥터는 '{1}'을 최상위 레벨 Business Object의 PrimaryID 속성 값으로 대체하고 '{2}'를 하위 컨테이너 오브젝트 Address의 다섯 번째 위치(기본 0)에 있는 AddressID 값으로 대체합니다. 이런 접근 방법으로 응답 메시지 선택자에 있는 Business Object와 Meta Object의 아무 속성이나 참조할 수 있습니다.

메시지 선택자에서 리터럴 값 "{"를 지정하려면 "{"를 대신 사용하십시오. 예를 들어, 다음 선택자는

```
response_selector=PrimaryID LIKE {{1}}
```

어댑터에서 리터럴 값으로 인식됩니다.

```
PrimaryID LIKE {1}
```

커넥터는 이 경우의 '{1}' 값에 대한 대체를 수행하지 않습니다.

커넥터가 속성값에서 '{', '}', ':', 또는 ';' 등의 특수 문자를 발견하면, 조회 문자열로 직접 삽입됩니다. 이렇게 하면 조회 문자열에 응용프로그램 특정 정보 분리문자 역할도 하는 특수 문자를 포함할 수 있습니다. 예를 들어, 다음 선택자는

```
Response_selector=PrimaryID = '{1}':Foo
```

Foo 속성의 값이 {A:B};{C:D}인 경우 다음과 같은 리터럴 메시지 선택자로 변환될 수 있습니다.

```
PrimaryID = '{A:B};{C:D}'
```

응답 처리: 응답 메시지의 수신을 처리할 조치를 판별하기 위해 커넥터는 MessageResponseResultProperty 커넥터 등록 정보에서 지정된 JMS 결과 등록 정보를 확인합니다. 이 JMS 등록 정보 값에 따라 커넥터는 응답 메시지가 메시지 본문에 Business Object 또는 오류 메시지를 포함한다고 예상합니다(아래 테이블 참조). 모든 경우에, 커넥터는 브로커에 해당 리턴 코드를 리턴합니다. 예를 들어, JMS 결과 등록 정보가 메시지의 VALCHANGE와 같으면 커넥터는 VALCHANGE에 대해 아래 설명된 조치를 취하고 브로커 상수 VALCHANGE에 해당하는 숫자 값을 브로커에 리턴합니다.

표 6. 응답 메시지 처리

JMS 결과 등록 정보 값	커넥터 조치
SUCCESS	요청 Business Object에 변경사항을 작성하지 않고 단지 성공적으로 브로커에 리턴합니다.

표 6. 응답 메시지 처리 (계속)

VALCHANGE MULTIPLE_HITS	요청 Business Object를 응답 메시지 본문의 내용으로 다시 채웁니다. 응답 메시지 본문이 비어 있으면 응답 Business Object는 변경되지 않고 그대로 남아 있습니다. 응답 Business Object의 동적 Meta Object를 응답 메시지의 JMS 머릿글 필드로 다시 채웁니다.
FAIL FAIL_RETRIEVE_BY_CONTENT BO_DOES_NOT_EXIST UNABLE_TO_LOGIN VALDUPES	응답이 채워지면 커넥터는 이것을 오류 메시지로 간주하여 브로커에 리턴합니다. 응답 메시지 본문이 비어 있으면 커넥터는 일반 오류 메시지를 브로커에 리턴합니다.
APPRESPONSETIMEOUT	APPRESPONSETIMEOUT을 브로커에 리턴하면 그 결과 어댑터 에이전트가 종료된다는 것을 제외하고 위와 동일합니다.
정의되지 않았거나 인식되지 않은 값	커넥터가 요청을 실패 처리합니다.

오류 처리: 대상 목적지에(서) 요청 메시지를 읽거나 쓰는 중이나 응답 메시지를 확인하는 중에(적용 가능한 경우) 오류가 발생한 경우, 커넥터는 즉시 APPRESPONSETIMEOUT을 브로커에 리턴합니다. 이로 인해 어댑터가 종료되고 가능한 경우 다시 시작됩니다. 일반적으로 복구할 수 없는 오류의 원인은 JMS 제공자와의 연결이 끊어졌거나 JMS 제공자에 의해 보고되는 내부 오류입니다. 내부 오류는 커넥터가 인식하지 못하거나 인식하지만 복구할 수 없는 것으로 간주하는 오류(예: 트랜잭션 실패)를 말합니다.

Business Object를 메시지로 변환하거나 그 반대 작업 중 오류가 발생하는 경우(예를 들어, Data Handler가 올바르게 읽지 않은 메시지 형식을 보고하는 경우), 커넥터는 그 요청을 실패 처리하고 그 이유에 대해하는 적절한 오류 메시지를 로그합니다.

이벤트 실패 시나리오에 대한 자세한 정보는 47 페이지의 『오류 처리』를 참조하십시오.

제 2 장 어댑터 설치 및 구성

- 『설치 작업』
- 『어댑터 및 관련 파일 설치』
- 『설치된 파일 구조』
- 21 페이지의 『커넥터 등록 정보 구성』
- 28 페이지의 『메시지 양식 구성』
- 28 페이지의 『JNDI 구성』
- 29 페이지의 『Meta Object 구성』
- 40 페이지의 『시작 스크립트 구성』
- 40 페이지의 『다중 커넥터 인스턴스 작성』
- 42 페이지의 『커넥터 시작』
- 43 페이지의 『커넥터 중지』

이 장에서는 커넥터를 설치 및 구성하는 방법과, 커넥터로 작업하기 위해 메시지 플로우를 구성하는 방법에 대해 설명합니다.

설치 작업

Adapter for JMS를 설치하려면 다음 작업을 수행해야 합니다.

- 통합 브로커 설치 WebSphere Business Integration System 설치 및 통합 브로커 시작을 포함한 이 작업은 브로커 및 운영 체제의 설치 문서에서 설명됩니다.
- 어댑터 및 관련 파일 설치 이 작업은 소프트웨어 패키지에서 시스템으로 어댑터 파일을 설치하는 작업을 포함합니다. 『어댑터 및 관련 파일 설치』를 참조하십시오.

어댑터 및 관련 파일 설치

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

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

설치된 파일 구조

아래 섹션은 설치 후에 제품의 경로와 파일명을 설명합니다.

Windows 커넥터 파일 구조

설치 프로그램은 커넥터와 연관되는 표준 파일을 시스템에 복사합니다.

유틸리티는 커넥터 에이전트를 *ProductDir*\connectors\JMS 디렉토리에 설치하고, 커넥터 에이전트에 대한 바로 가기를 시작 메뉴에 추가합니다. *ProductDir*은 IBM WebSphere Business Integration Adapters 제품이 설치된 디렉토리를 나타냄에 유의하십시오. 환경 변수에는 기본값이 IBM\WebSphereAdapters인 *ProductDir* 디렉토리 경로가 포함됩니다.

표 7은 커넥터에서 사용되는 Windows 파일 구조에 대해 설명하며, 설치 프로그램을 통한 커넥터 설치를 선택할 때 자동으로 설치되는 파일을 표시합니다.

표 7. 커넥터에 설치된 Windows 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors\JMS\CWJMS.jar	JMS Connector가 사용하는 클래스를 포함합니다.
connectors\JMS\jms.jar	커넥터에서 요구하는 타사 라이브러리
connectors\JMS\start_JMS.bat	커넥터의 시작 스크립트(NT/2000)
connectors\messages\JMSConnector.txt	커넥터의 메시지 파일
repository\JMS\CN_JMS.txt	커넥터의 저장소 정의
connectors\JMS\Samples\JMSConnector.cfg	샘플 커넥터 구성 파일
connectors\JMS\Samples\PortConnector.cfg	샘플 포트 커넥터 구성 파일
connectors\JMS\Samples\Sample_JMS_Contact.xsd	
connectors\JMS\Samples\Sample_JMS_MO_Config.xsd	샘플 Meta Object
connectors\JMS\Samples\Sample_JMS_MO_DataHandler.xsd	샘플 Data Handler Meta Object
connectors\JMS\Samples\Sample_JMS_MO_DataHandler_DelimitedConfig.xsd	샘플 구분된 Data Handler Meta Object
connectors\JMS\Samples\Sample_JMS_DynMO.xsd	샘플 동적 Meta Object
connectors\JMS\Samples\JMSPROPERTYPAIRS.xsd	샘플 JMS 등록 정보

주: 모든 제품 경로 이름은 제품이 시스템에 설치된 디렉토리에 상대적입니다.

UNIX 커넥터 파일 구조

설치 프로그램은 커넥터와 연관되는 표준 파일을 시스템에 복사합니다.

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

표 8은 커넥터에서 사용되는 UNIX 파일 구조에 대해 설명하며, 설치 프로그램을 통한 커넥터 설치를 선택할 때 자동으로 설치되는 파일을 표시합니다.

표 8. 커넥터에 설치된 UNIX 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors/JMS/CWJMS.jar	JMS Connector가 사용하는 클래스를 포함합니다.
connectors/JMS/jms.jar	커넥터에서 요구하는 타사 라이브러리

표 8. 커넥터에 설치된 UNIX 파일 구조 (계속)

<i>ProductDir</i> 의 서브디렉토리	설명
connectors/JMS/start_JMS.sh	커넥터의 시스템 시작 스크립트입니다. 이 스크립트는 일반 커넥터 관리자 스크립트에서 호출됩니다. System Manager의 커넥터 구성 화면을 누르면, 설치 프로그램에서 이 커넥터 관리자 스크립트에 맞게 사용자 정의된 랩퍼를 작성합니다. 커넥터를 시작 및 중지하려면 사용자 정의 랩퍼를 사용하십시오.
connectors/messages/JMSConnector.txt	커넥터의 메시지 파일
repository/JMS/CN_JMS.txt	커넥터의 저장소 정의
connectors/JMS/Samples/JMSConnector.cfg	샘플 커넥터 구성 파일
connectors/JMS/Samples/PortConnector.cfg	샘플 포트 커넥터 구성 파일
connectors/JMS/Samples/Sample_JMS_Contact.xsd	샘플 Meta Object
connectors/JMS/Samples/Sample_JMS_MO_Config.xsd	샘플 Data Handler Meta Object
connectors/JMS/Samples/Sample_JMS_MO_DataHandler.xsd	샘플 구분된 Data Handler Meta Object
connectors/JMS/Samples/Sample_JMS_MO_DataHandler_DelimitedConfig.xsd	샘플 동적 Meta Object
connectors/JMS/Samples/Sample_JMS_DynMO.xsd	샘플 JMS 등록 정보
connectors/JMS/Samples/JMSPropertyPairs.xsd	

주: 모든 제품 경로 이름은 제품이 시스템에 설치된 디렉토리에 상대적입니다.

커넥터 구성

어댑터를 설치한 후에 커넥터를 구성해야 합니다. 이를 위해서는 다음 섹션에서 설명된 작업을 수행해야 합니다.

- 『커넥터 등록 정보 구성』
- 28 페이지의 『메시지 양식 구성』
- 28 페이지의 『JNDI 구성』
- 29 페이지의 『Meta Object 구성』
- 40 페이지의 『시작 스크립트 구성』

커넥터 등록 정보 구성

커넥터에는 다음 섹션에서 설명된 두 가지 유형의 구성 등록 정보가 있습니다.

- 22 페이지의 『표준 커넥터 등록 정보 구성』
- 22 페이지의 『커넥터 특정 등록 정보 구성』

어댑터를 실행하기 전에 이들 등록 정보의 값을 설정해야 합니다.

Connector Configurator를 사용하여 커넥터 등록 정보를 구성합니다.

- Connector Configurator 및 단계별 프로시저의 설명에 대해서는 69 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

- 표준 커넥터 등록 정보의 설명에 대해서는 『표준 커넥터 등록 정보 구성』 및 51 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.
- 커넥터 특정 등록 정보의 설명에 대해서는 『커넥터 특정 등록 정보 구성』을 참조하십시오.

표준 커넥터 등록 정보 구성

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이러한 등록 정보 설명에 대해서는 51 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오. 그 다음, 이런 등록 정보 설정 방법을 설명하는 단계별 프로시저는 69 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

주: Connector Configurator에서 구성 등록 정보를 설정할 때 BrokerType 등록 정보를 사용하여 브로커를 지정합니다. 이것이 설정된 후에는, 사용자 브로커와 관련된 등록 정보가 Connector Configurator 창에 나타납니다.

커넥터 특정 등록 정보 구성

커넥터 특정 구성 등록 정보는 런타임 시 커넥터 에이전트에 필요한 정보를 제공합니다. 또한 커넥터 특정 등록 정보는 에이전트를 다시 코드화하고 다시 빌드할 필요 없이 커넥터 에이전트 내의 논리나 정적 정보를 변경하는 방식을 제공합니다.

표 9는 커넥터의 커넥터 특정 구성 등록 정보를 나열한 것입니다. 등록 정보에 대한 설명은 뒤에 나오는 섹션을 참조하십시오.

표 9. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수 여부
ArchiveDestination	성공적으로 처리된 메시지 사본이 전송되는 목적지		아니오
ConfigurationMetaObject	구성 <i>Meta Object</i>		등록 정보 설명 참조
ConnectionFactoryName	<i>JNDI</i> 상점에 정의된 <i>JMS</i> 대기열 또는 항목 연결 생성기		예
CTX_InitialContextFactory	초기 <i>JNDI</i> 컨텍스트를 수립하기 위해 사용되는 생성기 클래스의 이름		예
CTX_ProviderURL	연결 생성기가 있는 <i>JNDI</i> 컨텍스트를 식별하는 <i>URL</i>		예
DataHandlerClassName	인스턴스를 생성할 <i>Data Handler</i> 클래스의 이름		등록 정보 설명 참조
DataHandlerConfigMO	<i>DataHandlerMimeType</i> 에 대한 구성 정보를 포함하는 <i>Data Handler Meta Object</i> 의 이름	MO_DataHandler_Default	등록 정보 설명 참조
DataHandlerMimeType	기본 <i>Data Handler</i> 선택 시 사용할 <i>MIME</i> 유형	text/delimited	등록 정보 설명 참조
DefaultVerb	수신 <i>Business Object</i> 내에 설정될 동사를 지정합니다.	Create	아니오
ErrorDestination	처리되지 않은 메시지의 목적지		아니오
InDoubtEvents	FailOnStartup Reprocess Ignore LogError	Reprocess	아니오
InProgressDestination	임시 저장영역 목적지		아니오
InputDestination	폴 목적지의 이름		아니오
LookupDestinationsUsingJNDI	true 또는 false	false	아니오
MessageFormatProperty	메시지 형식을 지정하는 등록 정보 이름	JMSType	아니오
MessageResponseResultProperty	요청 조작의 결과를 표시하는 응답 메시지의 등록 정보	WBI_Result	예, 동기 처리용.
PollQuantity	<i>InputDestination</i> 등록 정보에 지정된 각 목적지에서 검색할 메시지 수	1	아니오
ReplyToDestination	커넥터에서 요청 발행 시 응답 메시지가 전달되는 목적지		예, 동기 처리용.
UnsubscribedDestination	메시지가 인식되지 않거나 메시지가 맵핑되는 <i>Business Object</i> 가 지원되지 않는 경우 인바운드 메시지의 사본이 배치되는 목적지		아니오
UnsubscribeOnTerminate	<i>InputDestination</i> 에서 제거된 주제를 지정합니다.		아니오
UseDefaults	true 또는 false	false	아니오
UseDurableSubscriptions	true 또는 false	false	아니오

ArchiveDestination

성공적으로 처리된 메시지 사본이 전송되는 목적지입니다.

기본값은 CWLD_ARCHIVE입니다.

ConfigurationMetaObject

커넥터의 구성 정보를 포함하는 정적 Meta Object의 이름입니다.

기본값은 없습니다.

ConnectionFactoryName

커넥터가 JMS 제공자에 대한 연결을 수립하기 위해 검색하고 사용해야 할 JNDI 상점에 정의된 JMS 대기열 또는 항목 연결 생성기 오브젝트입니다. 이 이름을 찾을 때 커넥터는 CTX_InitialContextFactory 및 CTX_ProviderURL 등록 정보에 의해 수립된 초기 JNDI 컨텍스트를 사용합니다.

기본값 = 없음

CTX_InitialContextFactory

초기 JNDI 컨텍스트를 수립하기 위해 사용되는 생성기 클래스의 이름

기본값 = 없음

CTX_ProviderURL

연결 생성기가 있는 JNDI 컨텍스트를 식별하는 완전한 URL입니다. 이 값은 컨텍스트 인수에 전달됩니다.

기본값 = 없음

DataHandlerClassName

메시지를 Business Object로(부터) 변환할 때 사용하는 Data Handler 클래스입니다. DataHandlerConfigMO와 DataHandlerMimeType 모두 또는 DataHandlerClassName만 지정하십시오. 세 가지 등록 정보 모두를 지정하지는 마십시오.

주: 정적 또는 동적 Meta Object의 DataHandlerClassName 값은 이 커넥터 구성 등록 정보에 지정된 값에 우선합니다. Meta Object에 DataHandlerClassName 값을 제공하지 않으면 커넥터는 이 커넥터 구성 등록 정보에서 해당 값을 취합니다.

기본값 = 없음

DataHandlerConfigMO

DataHandlerMimeType 등록 정보에서 지정된 MIME 유형에 대한 구성 정보를 포함하는 Meta Object의 이름. Data Handler에 대한 구성 정보를 제공합니다.

DataHandlerConfigMO와 DataHandlerMimeType 또는 DataHandlerClassName만 지정하십시오. 세 가지 등록 정보 모두를 지정하지는 마십시오.

주: 정적 또는 동적 Meta Object의 DataHandlerConfigMO 값은 이 커넥터 구성 등록 정보에 지정된 값에 우선합니다. Meta Object에 DataHandlerConfigMO 값을 제공하지 않으면 커넥터는 이 커넥터 구성 등록 정보에서 해당 값을 취합니다.

기본값은 MO_DataHandler_Default입니다.

DataHandlerMimeType

특정 MIME 유형을 기초로 Data Handler를 요청할 수 있게 합니다.

DataHandlerConfigMO와 DataHandlerMimeType 또는 DataHandlerClassName만 지정하십시오. 세 가지 등록 정보 모두를 지정하지는 마십시오.

주: 정적 또는 동적 Meta Object의 DataHandlerMimeType 값은 이 커넥터 구성 등록 정보에 지정된 값에 우선합니다. Meta Object에 DataHandlerMimeType 값을 제공하지 않으면 커넥터는 이 커넥터 구성 등록 정보에서 해당 값을 취합니다.

기본값 = text/delimited

DefaultVerb

플링 시 Data Handler에서 동사를 설정하지 않은 경우 수신 Business Object 내에 설정될 동사를 지정합니다.

기본값 = Create

ErrorDestination

처리 중 커넥터에 오류가 발생할 때 인바운드 메시지의 사본이 전송되는 목적지

기본값은 CWLD_ERROR입니다.

InDoubtEvents

예기치 않은 커넥터 종료로 인해 완전히 처리되지 않은 진행 중인 이벤트를 처리하는 방법을 지정합니다. 초기화하는 중에 진행 중인 대기열에서 이벤트가 발견될 경우 취할 네 가지 조치 중 하나를 선택하십시오.

- FailOnStartup - 오류를 기록하고 즉시 종료합니다.
- Reprocess - 먼저 나머지 이벤트를 처리한 후 입력 대기열에서 메시지를 처리합니다.
- Ignore - 진행 중인 대기열에서 메시지를 버립니다.
- LogError - 오류를 기록하지만 종료하지는 않습니다.

기본값은 Reprocess입니다.

주: InProgressDestination 등록 정보를 구성하는 경우 이 등록 정보에 대한 값을 지정해야 합니다.

InProgressDestination

처리 중 메시지가 보유되는 임시 목적지입니다.

기본값 = 없음

InputDestination

새 메시지에 대해 커넥터에서 폴링될 목적지입니다. 커넥터는 세미콜론으로 구분되는 여러 개의 이름을 승인합니다. 예를 들어, 대기열 기반 구성의 세 가지 대기열인 MyQueueA, MyQueueB 및 MyQueueC를 폴링하기 위해 *InputQueue* 커넥터 구성 등록 정보의 값이 MyQueueA;MyQueueB;MyQueueC와 같게 됩니다.

InputDestination 등록 정보가 제공되지 않는 경우 커넥터는 폴링하지 않습니다.

기본값 = 없음

LookupDestinationsUsingJNDI

이 등록 정보가 true이면 커넥터는 JNDI 상점에 있는 모든 JMS 목적지 이름을 찾습니다. 이를 위해서는 지정된 목적지에 JNDI 상점에 정의되어야 합니다.

기본적으로, 커넥터는 이 단계를 건너뛰고 런타임 시 JMS 제공자가 이름을 적절한 목적지로 바꿀 수 있도록 합니다.

기본값 = false

MessageFormatProperty

메시지에 대한 입력 또는 출력 형식을 포함하는 JMS 메시지의 필드. 기본적으로, 커넥터는 메시지 형식에 대한 인바운드 메시지의 JMSType 필드를 확인하고 메시지 형식을 아웃바운드 메시지의 JMSType 필드에 씁니다.

기본값 = JMSType

MessageResponseResultProperty

동기 요청 처리에 필요하므로, 이 등록 정보는 커넥터가 요청의 결과를 판별하기 위해 확인해야 할 응답 JMS 메시지의 필드를 지정합니다. 이 등록 정보는 비동기 처리에 사용되지 않습니다.

기본값은 WBI_Result입니다.

PollQuantity

pollForEvents 주기 중 InputDestination 등록 정보에 지정된 각 목적지에서 검색할 수 있는 메시지의 최대 수

기본값은 1입니다.

ReplyToDestination

커넥터에서 요청 발행 시 응답 메시지가 전달되는 목적지. 기본적으로, 이것은 커넥터가 대상 응용프로그램과의 요청 메시지 교환을 조정하기 위해 사용하는 목적지입니다. 동기 처리용으로만 이 등록 정보를 지정하십시오.

기본값 = 없음

UnsubscribedDestination

메시지가 인식되지 않거나 메시지가 맵핑되는 Business Object가 지원되지 않는 경우 인바운드 메시지의 사본이 배치되는 목적지. 등록 정보가 정의되고 올바른 경우 커넥터는 이 목적지에 등록되지 않은 메시지의 사본을 가져다 놓으며, 그렇지 않은 경우 해당 메시지는 버려집니다.

기본값 = 없음

UnsubscribeOnTerminate

UserDurableSubscriptions가 true로 설정될 때만 적용 가능합니다. 항목을 커넥터 구성에서 제거하는 경우 영구 등록의 사용은 문제점을 생성합니다. 커넥터가 이런 등록을 다시 확인하지 않더라도 JMS 제공자는 계속해서 영구 등록에 대한 메시지를 저장합니다.

InputDestination에서 지정된 목록에서 항목을 제거할 때마다 이 등록 정보 값에 이런 제거된 항목(세미콜론으로 구분된)을 지정하십시오. 기존 영구 등록을 파기하려면 다음 단계를 따르십시오.

1. 더 이상 등록을 원치 않는 해당 주제 이름을 InputDestination에서 UnsubscribeOnTerminate로 이동시키십시오.
2. 커넥터를 시작한 후 중지하십시오(이렇게 하면 영구 등록이 파기됨).
3. UnsubscribeOnTerminate에 지정된 모든 주제를 지우십시오.

이 조치는 어떤 InputDestination 값도 변경하지 않습니다.

위 단계 수행에 실패해도 커넥터에 영향을 미치지 않지만 JMS 제공자가 불필요한 메시지를 저장하는 원인이 됩니다.

기본값 = 없음

UseDefaults

UseDefaults가 true로 설정된 경우, 커넥터는 isRequired로 표시된 각 Business Object 속성에 올바른 값 또는 기본값이 제공되었는지 여부를 확인합니다.

기본값 = false

UseDurableSubscriptions

이 등록 정보는 Pub/Sub 항목 양식 메시징에만 사용하십시오. 이 등록 정보가 true로 설정된 경우, 커넥터는 적용 가능한 목적지에 대한 영구 등록자로 사용됩니다. 더 높은 오버헤드를 발생시키면서도, 커넥터는 커넥터가 오프라인인 동안에도 등록하는 이런 항목에 대한 모든 메시지를 저장하도록 JMS 제공자에게 지시합니다. 다시 온라인 상태가 되었을 때 커넥터는 누락된 공개 메시지를 다시 처리합니다.

기본값 = false

메시지 양식 구성

어댑터는 JMS 표준에 의해 정의된 PTP(Point-to-Point) 메시징 및 Pub/Sub(Publish-and-Subscribe) 메시징 인터페이스 모듈을 지원합니다. 어댑터에서 사용되는 메시징 양식은 사용자가 커넥터 특정 등록 정보 `ConnectionFactoryName`에 지정한 관리 오브젝트 유형으로 판별됩니다. 이런 프로시저를 진행하기 전에 24 페이지의 『`ConnectionFactoryName`』을 참조하십시오.

- 『PTP 메시징 양식 구성』
- 『Pub/Sub 양식 구성』

PTP 메시징 양식 구성

어댑터의 인스턴스를 PTP 메시지 양식으로 구성하려면 다음을 수행하십시오.

1. Connector Configurator를 여십시오.
2. 커넥터 특정 등록 정보 탭을 누르십시오.
3. JNDI 저장소에 있는 JMS `QueueConnectionFactory`의 인스턴스에 맵핑되는 `ConnectionFactoryName`의 이름을 지정하십시오. 어댑터는 PTP 양식으로 작동되며 목적지를 식별하는 모든 커넥터 및 Meta Object 등록 정보(예: `OutputDestination` 등록 정보)가 대기열을 표시한다고 가정합니다.

Pub/Sub 양식 구성

어댑터의 인스턴스를 Pub/Sub 메시지 양식으로 구성하려면 다음을 수행하십시오.

1. Connector Configurator를 여십시오.
2. 커넥터 특정 등록 정보 탭을 누르십시오.
3. JNDI 저장소에 있는 JMS `TopicConnectionFactory`의 인스턴스에 맵핑되는 `ConnectionFactoryName`의 이름을 지정하십시오. 어댑터는 Pub/Sub 양식으로 작동되며 목적지를 식별하는 모든 커넥터 및 Meta Object 등록 정보(예: `OutputDestination` 등록 정보)가 항목을 표시한다고 가정합니다.

JNDI 구성

JMS 제공자로의 연결을 수립하기 위해 커넥터는 JMS 연결 생성기에 액세스해야 합니다. JMS는 그 생성기에 대한 인터페이스를 정의합니다. 그러나 개별 JMS 제공자 각각

은 자체 구현을 제공해야 합니다. 커넥터가 이 생성기 구현에 대한 참조를 가지게 되면 커넥터는 등록 정보 프로토콜에 대한 지식이나 또는 제공자의 식별 없이도 JMS 제공자와의 연결을 수립하고 이와 통신할 수 있습니다.

이식을 가능하게 하기 위해 커넥터는 연결 생성기가 JNDI 상점에 위치하기를 요구합니다. 구현 중에 사용자 또는 시스템 관리자가 연결 생성기를 작성 및 구성한 후 이를 사용자 정의된 이름의 JNDI 상점에 둡니다. 런타임 시 JMS 상점과의 연결을 수립하고 연결 생성기를 찾아본 후 JMS 제공자와의 연결을 수립하기 위해 이를 사용합니다.

일부 JMS 제공자는 연결 생성기 또는 작성한 기타 관리 JMS 오브젝트를 포함하는 자체 JNDI 구현을 제공합니다. 이 접근을 통해 매우 수월하게 JNDI 상점을 구성할 수 있습니다. 다른 JMS 제공자의 경우, 사용자는 외부 JNDI 제공자를 설치 및 구성해야 하고 연결 생성기를 작성하여 어댑터에서 사용 가능하게 해야 합니다. 자세한 정보는 JNDI 제공자 문서를 참조하십시오.

JNDI 환경 변수 및 구성에 대한 자세한 정보는 www.javasoft.com을 참조하십시오. MA88 패치로 JNDI를 구성하는 방법에 대한 정보는 『WebSphere MQ Java 클라이언트 라이브러리로 JNDI 구성』을 참조하십시오.

WebSphere MQ Java 클라이언트 라이브러리로 JNDI 구성

WebSphere MQ Java 클라이언트 라이브러리에서 JNDI를 구성하는 방법은 95 페이지의 『대기열 기반 메시징 구성』 및 96 페이지의 『항목 기반 메시징 구성』 항목을 참조하십시오.

Meta Object 구성

Connector for JMS는 두 가지 종류의 Meta Object를 인식하고 읽을 수 있습니다.

- 정적 커넥터 Meta Object
- 동적 하위 Meta Object

동적 하위 Meta Object의 속성값은 정적 Meta Object의 값과 중복되며 이를 대체합니다. 메타 데이터 및 정적 대 동적 Meta Object의 개요는 9 페이지의 『메타 데이터 및 Meta Object』를 참조하십시오.

구현에 가장 적합한 Meta Object를 결정할 때 다음을 고려하십시오.

- 정적 **Meta Object**
 - 여러 메시지에 대한 모든 메타 데이터가 고정되었고 구성 시 지정될 수 있는 경우 유용합니다.
 - Business Object 유형에 따라 값 지정이 제한됩니다. 예를 들어, 모든 고객 유형 오브젝트는 동일한 목적지에 전송되어야 합니다.
- 동적 **Meta Object**

- 메시지 머릿글에 있는 정보에 대한 비즈니스 프로세스 액세스를 제공합니다.
- 비즈니스 프로세스가 런타임 시 비즈니스 유형에 관계없이 메시지 처리를 변경할 수 있도록 합니다. 예를 들어, 동적 Meta Object는 어댑터에 전송된 모든 고객 유형 오브젝트에 다른 목적지를 지정할 수 있게 합니다.
- 지원되는 Business Object의 구조를 변경해야 합니다. 이를 위해서는 맵과 비즈니스 프로세스를 변경해야 할 수 있습니다.
- 고객 Data Handler를 변경해야 합니다.

Meta Object 등록 정보

표 10에서는 Meta Object에서 지원되는 완전한 등록 정보 목록을 제공합니다. Meta Object 구현 시에는 이런 등록 정보를 참조하십시오.

두 오브젝트 모두에서 모든 등록 정보를 사용할 수는 없습니다. 또한 메시지 머릿글에 (서) 모든 등록 정보를 쓰거나 읽을 수 있는 것은 아닙니다. 특정 등록 정보를 해석하거나 커넥터에서 사용하는 방법을 판별하려면 1 페이지의 제 1 장 『Adapter for JMS 개요』에서 이벤트 및 요청 처리에 해당하는 섹션을 참조하십시오.

표 10. JMS Meta Object 등록 정보

등록 정보 이름	정적 Meta Object에 서 정의할 수 있음	동적 Meta Object 에서 정의할 수 있음	설명
DataHandlerConfigMO	예	예	구성 정보를 제공하기 위해 Data Handler에 전달되는 Meta Object입니다. 정적 Meta Object에서 지정한 경우, 이는 DataHandlerConfigMO 커넥터 등록 정보에 지정된 값을 대체합니다. 여러 Business Object 유형을 처리하기 위해 여러 가지의 Data Handler가 필요할 때 이 정적 Meta Object 등록 정보를 사용하십시오. 데이터 형식이 실제 비즈니스 데이터의 영향을 받을 수도 있을 경우, 요청 처리를 위해 동적 하위 Meta Object를 사용하십시오. 지정한 Business Object는 커넥터 에이전트에서 지원해야 합니다. 22 페이지의 『커넥터 특정 등록 정보 구성』의 설명을 참조하십시오.
DataHandlerMimeType	예	예	특정 MIME 유형을 기초로 Data Handler를 요청할 수 있게 합니다. 정적 Meta Object에서 지정한 경우, 이는 DataHandlerMimeType 커넥터 등록 정보에 지정된 값을 대체합니다. 여러 Business Object 유형을 처리하기 위해 여러 가지의 Data Handler가 필요할 때 이 정적 Meta Object 등록 정보를 사용하십시오. 데이터 형식이 실제 비즈니스 데이터의 영향을 받을 수도 있을 경우, 요청 처리를 위해 동적 하위 Meta Object를 사용하십시오. DataHandlerConfigMO에 지정된 Business Object는 이 등록 정보의 값에 해당하는 속성을 가지고 있어야 합니다. 22 페이지의 『커넥터 특정 등록 정보 구성』의 설명을 참조하십시오.
DataHandlerClassName	예	예	22 페이지의 『커넥터 특정 등록 정보 구성』의 설명을 참조하십시오.

표 10. JMS Meta Object 등록 정보 (계속)

등록 정보 이름	정적 Meta Object에 서 정의할 수 있음	동적 Meta Object 에서 정의할 수 있음	설명
InputFormat	예	예	인바운드(이벤트) 메시지의 형식 또는 유형. 이 값은 메시지의 내용을 식별하는 데 도움을 주며 메시지를 생성한 응용프로그램에 의해 지정될 수 있습니다. 커넥터가 메시지의 형식을 정의하는 것으로 생각하는 필드는 커넥터 특정 등록 정보 MessageFormatProperty를 통해 사용자 정의될 수 있습니다.
OutputFormat	예	예	아웃바운드 메시지에 채워질 수 있는 형식. OutputFormat을 지정하지 않으면, 입력 형식이 사용 가능할 경우 이 형식이 사용됩니다.
InputDestination	예	예	이 등록 정보는 수신 메시지를 Business Object에 일치시키는 데에만 사용됩니다. 이에 비해 InputDestination 커넥터 특정 등록 정보는 어댑터가 폴링하는 목적지를 정의하며, 어댑터가 폴링할 목적지를 판별하는 데 사용하는 유일한 등록 정보입니다. MO에서 InputDestination 등록 정보 및 InputFormat 등록 정보는 특정 Business Object에 정해진 메시지를 매핑하는 어댑터의 기준 역할을 할 수 있습니다. 이 기능을 구현하기 위해 사용자는 다중 입력 목적지를 구성하는 커넥터 특정 등록 정보를 사용하며, 선택적으로 다른 Data Handler를 수신 메시지의 입력 형식에 따라 각 목적지에 매핑하게 됩니다. 기본 변환 등록 정보를 사용하여 이 등록 정보를 설정하지 마십시오. 해당 값이 사용됩니다.
OutputDestination	예	예	아웃바운드 메시지가 쓰여지는 목적지
ResponseTimeout	예	예	동기 요청 처리에서 응답을 기다릴 때 제한시간 전에 대기하는 시간 길이(밀리초)를 표시합니다. 정의되지 않거나 값이 0보다 작으면, 커넥터는 응답을 기다리지 않고 즉시 SUCCESS를 리턴합니다.
TimeoutFatal	예	예	응답이 수신되지 않는 경우 오류 메시지를 리턴하는 커넥터를 트리거하기 위해 동기 요청 처리에 사용됩니다. 이 등록 정보가 True일 경우 커넥터는 응답이 ResponseTimeout으로 지정된 시간 이내에 수신되지 않으면 브로커에 APPRESPONSETIMEOUT을 리턴합니다. 등록 정보가 정의되어 있지 않거나 False로 설정되어 있는 경우 응답 제한시간 시 커넥터는 요청에 실패하지만 종료되지는 않습니다. 기본 값 = False.
아래는 특히 JMS 메시지 머릿글에 매핑하는 필드입니다. 특정 설명, 값의 해석 및 그 외 정보는 JMS API 스펙을 참조하십시오. JMS 제공자가 일부 필드를 다르게 해석할 수 있으므로 다른 경우의 JMS 제공자 문서도 확인하십시오.			
ReplyToDestination		예	요청에 대한 응답 메시지가 전송되는 목적지
Type		예	메시지의 유형. 일반적으로 JMS 제공자에 따라 사용자 정의 가능합니다.
MessageID		예	메시지의 고유 ID(JMS 제공자 특정)
CorrelationID	예	예	이 응답을 초기화한 요청 메시지의 ID를 표시하기 위해 응답 메시지에서 사용됩니다.
Delivery Mode	예	예	메시지가 지속되는지 또는 MOM 시스템에 없는지 여부를 지정합니다. 가능한 값은 다음과 같습니다. 1 = 지속되지 않음 2 = 지속됨
Priority		예	JMS 제공자에 따라 다른 값을 사용할 수 있습니다. 메시지의 숫자 우선순위. 가능한 값은 0에서 9 사이의 수입니다(낮은 우선순위 -> 높은 우선순위).

표 10. JMS Meta Object 등록 정보 (계속)

등록 정보 이름	정적 Meta Object에 서 정의할 수 있음	동적 Meta Object 에서 정의할 수 있음	설명
Destination		예	MOM 시스템에서 메시지의 현재 또는 마지막(제거된 경우) 위치
Expiration		예	메시지의 지속 시간. 0으로 지정된 경우 만기는 0으로 설정됩니다. 0은 메시지가 만기되지 않는 JMS 제공자를 표시합니다.
Redelivered		예	JMS 제공자가 초기에 클라이언트에 메시지를 전달하려고 시도했지만 수신자가 확인되지 않았음을 표시합니다.
Timestamp		예	메시지가 JMS 제공자에게 전달된 시간
UserID		예	메시지를 전송한 사용자의 ID
AppID		예	메시지를 전송한 응용프로그램의 ID
DeliveryCount		예	전달 시도 횟수
GroupID		예	메시지 그룹의 ID
GroupSeq		예	그룹 ID에 지정된 메시지 그룹에서 이 메시지의 순서
JMSProperties		예	38 페이지의 『JMS 등록 정보』를 참조하십시오.

정적 Meta Object 구성

JMS 구성 정적 Meta Object에는 여러 Business Object에 대해 정의된 변환 등록 정보 목록이 있습니다. 샘플 정적 Meta Object를 보려면, Business Object Designer를 실행한 후 어댑터에 탑재된 connectors\JMS\Samples\Sample_JMS_MO_Config.xsd 샘플을 여십시오.

커넥터는 주어진 시간에 많아야 하나의 정적 Meta Object를 지원합니다. 커넥터 등록 정보 ConfigurationMetaObject에 대한 이름을 지정하여 정적 Meta Object를 구현합니다.

정적 Meta Object의 구조는 각 속성이 단일 Business Object와 Verb의 조합 및 해당 오브젝트 처리와 연관된 메타 데이터를 표시하는 것입니다. 각 속성의 이름은 밑줄로 구별된 Business Object 유형과 Verb의 이름이어야 합니다(예: Customer_Create). 속성 응용프로그램 특정 정보는 이 고유한 오브젝트-Verb 조합에 대해 지정하려는 메타 데이터 등록 정보를 표시하는 하나 이상의 세미콜론으로 구분된 이름-값 쌍으로 이루어 집니다.

표 11. 정적 Meta Object 구조

속성 이름	응용프로그램 특정 텍스트
<Business Object 유형>_<Verb>	등록 정보=값;등록 정보=값;...
<Business Object 유형>_<Verb>	등록 정보=값;등록 정보=값;...

예를 들어, 다음 Meta Object를 고려하십시오.

표 12. 샘플 정적 Meta Object 구조

속성 이름	응용프로그램 특정 정보
Customer_Create	OutputFormat=CUST;OutputDestination=QueueA
Customer_Update	OutputFormat=CUST;OutputDestination=QueueB

표 12. 샘플 정적 Meta Object 구조 (계속)

속성 이름	응용프로그램 특정 정보
Order_Create	OutputFormat=ORDER;OutputDestination=QueueC

이 샘플의 Meta Object는 Verb가 Create인 Customer 유형의 요청 Business Object를 수신하면 커넥터에게 알려 이를 CUST 형식의 메시지로 변환한 후 QueueA 목적지에 위치시킵니다. 대신 고객 오브젝트가 Update Verb를 가지는 경우, 메시지는 QueueB에 놓이게 될 것입니다. 오브젝트 유형이 Order이고 Create Verb를 가지면 커넥터는 ORDER 형식으로 변환되며 QueueC에 전달됩니다. 커넥터에 전달된 다른 Business Object는 등록되지 않은 것으로 취급될 것입니다.

선택적으로, 하나의 속성 이름을 Default로 지정할 수 있고 이것을 ASI의 하나 이상의 등록 정보에 지정할 수 있습니다. Meta Object에 포함된 모든 속성의 경우 기본 속성의 등록 정보는 특정 오브젝트-Verb 속성의 등록 정보와 결합됩니다. 이것은 보편적으로(오브젝트-Verb 조합에 관계없이) 적용할 하나 이상의 등록 정보가 있을 때 유용합니다. 다음 예에서 커넥터는 Customer_Create 및 Order_Create의 오브젝트-Verb 조합을 개별 메타 데이터 등록 정보 외에 OutputDestination=QueueA를 가진 것으로 간주될 수 있습니다.

표 13. 샘플 정적 Meta Object 구조

속성 이름	응용프로그램 특정 정보
Default	OutputDestination=QueueA
Customer_Update	OutputFormat=CUST
Order_Create	OutputFormat=ORDER

정적 Meta Object에 응용프로그램 특정 정보로 지정할 수 있는 등록 정보에 대한 설명은 30 페이지의 『Meta Object 등록 정보』에서 30 페이지의 표 10을 참조하십시오.

정적 Meta Object를 구현하려면 다음을 수행하십시오.

1. Business Object Designer를 실행하십시오. 자세한 정보는 *Business Object Development Guide*를 참조하십시오.
2. 샘플 Meta Object connectors\JMS\Samples\Sample_JMS_MO_Config.xsd를 여십시오. 그림 3 항목은 Business Object Designer의 샘플 정적 Meta Object를 보여 줍니다.

	Pos	Name	Type	Key	Foreign	Required	MaxLength	App Spec Info
1	1	Default	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	OutputQueue=CWLD_Output
2	2	Sample_JMS_Contact_Create	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	InputFormat=CON_CR
3	3	Sample_JMS_Contact_Update	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	InputFormat=CON_UP
4	4	Sample_JMS_Contact_Delete	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	InputFormat=CON_DEL
5	5	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
6	6			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	255	

그림 3. 샘플 정적 Meta Object

- 30 페이지의 표 10을 참조하여 요구사항을 반영하도록 속성 및 ASI를 편집한 후 Meta Object 파일을 저장하십시오.
- 이 Meta Object 파일의 이름을 ConfigurationMetaObject 연결 등록 정보의 값으로 지정하십시오.

Data Handler를 입력 목적지로 맵핑

정적 Meta Object의 응용프로그램 특정 정보에서 InputQueue 등록 정보를 사용하여 Data Handler를 입력 목적지와 연관시킬 수 있습니다. 이 기능은 다른 형식과 변환을 요구하는 여러 거래 업체와 작업할 때 유용합니다.

Data Handler를 입력 목적지에 맵핑하려면 다음을 수행하십시오.

- Connector Configurator를 실행하십시오. 자세한 정보는 69 페이지의 부록 B 『Connector Configurator』를 참조하십시오.
- 하나 이상의 입력 목적지를 구성하려면 커넥터 특정 등록 정보(26 페이지의 『InputDestination』 참조)를 사용하십시오. 다중 목적지 이름은 세미콜론으로 구분되어야 합니다.
- 각 입력 대기열의 경우, 응용프로그램 특정 정보의 MIME 유형 및 Data Handler 클래스 이름과 더불어 목적지(PTP 메시징 양식을 구현 중인 경우) 및 입력 목적지 이름을 지정하십시오.

예를 들어, 정적 Meta Object의 다음과 같은 속성은 Data Handler를 CompReceipts라고 이름 지정된 InputQueue와 연관시킵니다.

```
[Attribute]
Name = Customer_Create
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
```



```

AppSpecificInfo =
InputDestination=//queue.manager/CompReceipts;DataHandlerClassName=com.crossworlds.
DataHandlers.MQ.disposition_notification;DataHandlerMimeType=message/
disposition_notification
IsRequiredServerBound = false
[End]

```

동적 하위 Meta Object 구성

정적 Meta Object를 통해 필요한 메타 데이터를 지정하기 어렵거나 이를 실행할 수 없는 경우, 커넥터는 선택적으로 각 Business Object 인스턴스마다 런타임 시에서 전달된 메타 데이터를 승인할 수 있습니다.

동적 Meta Object를 사용하면 요청 처리 중에 요청 기반으로 Business Object를 처리하고 이벤트 처리 중에 이벤트 메시지에 대한 정보를 검색하기 위해 커넥터에서 사용되는 메타 데이터를 변경할 수 있습니다.

동적 Meta Object의 구조는 각 속성이 단일 Business Object 등록 정보와 값 조합(Meta Object 등록 정보 이름=Meta Object 등록 정보 값)을 표시하는 것입니다.

동적 Meta Object를 구현하려면 이를 최상위 레벨 오브젝트의 하위로 추가하고 최상위 레벨 오브젝트 ASI에 이름-값 쌍, cw_mo_conn=<MO 속성>을 포함시키십시오. 여기서 <MO 속성>은 동적 Meta Object를 표시하는 최상위 레벨 오브젝트의 속성 이름입니다. 예를 들면, 다음과 같습니다.

```

Customer (ASI = cw_mo_conn=MetaData)
  -- Id
  -- FirstName
  -- LastName
  -- ContactInfo
  -- MetaData
    -- OutputFormat = CUST
    -- OutputDestination = QueueA

```

위에 표시된 대로 채워진 요청을 수신하면 커넥터는 Customer 오브젝트를 CUST 형식의 메시지로 변환한 후 그 메시지를 QueueA 대기열에 놓습니다.

Business Object는 동일하거나 다른 동적 Meta Object를 사용하거나 동적 Meta Object를 사용하지 않을 수 있습니다.

주: 모든 표준 IBM WebSphere Data Handler는 cw_mo_ 태그를 인식하여 동적 Meta Object를 무시하도록 설계되었습니다. 어댑터 사용을 위해 사용자 정의 Data Handler를 개발할 때 동일한 작업을 수행해야 합니다.

커넥터는 커넥터에 전달되는 최상위 레벨 Business Object에 하위 요소로 추가된 동적 Meta Object로부터 변환 등록 정보를 인식하여 읽습니다. 동적 하위 Meta Object의 속성값은 커넥터를 구성하기 위해 사용되는 정적 Meta Object를 통해 지정할 수 있는 변환 등록 정보를 복제합니다.

동적 하위 Meta Object 등록 정보는 정적 Meta Object에서 발견되는 등록 정보를 대체하므로, 동적 하위 Meta Object를 지정할 경우에는 정적 Meta Object를 지정하는 커넥터 등록 정보를 포함하지 않아야 합니다. 따라서 정적 Meta Object와 독립적으로 동적 하위 Meta Object를 사용하고, 동적 하위 Meta Object와 독립적으로 정적 Meta Object를 사용할 수 있습니다.

동적 Meta Object에 응용프로그램 특정 정보로 지정할 수 있는 등록 정보에 대한 설명은 30 페이지의 『Meta Object 등록 정보』에서 30 페이지의 표 10을 참조하십시오.

동적 Meta Object를 구성하려면 다음을 수행하십시오.

1. Business Object Designer를 실행하십시오. 자세한 정보는 *Business Object Development Guide*를 참조하십시오.
2. 샘플 Meta Object connectors\JMS\Samples\Sample_JMS_DynMO.xsd를 여십시오. 그림 4 항목은 Business Object Designer의 샘플 동적 Meta Object를 보여 줍니다.

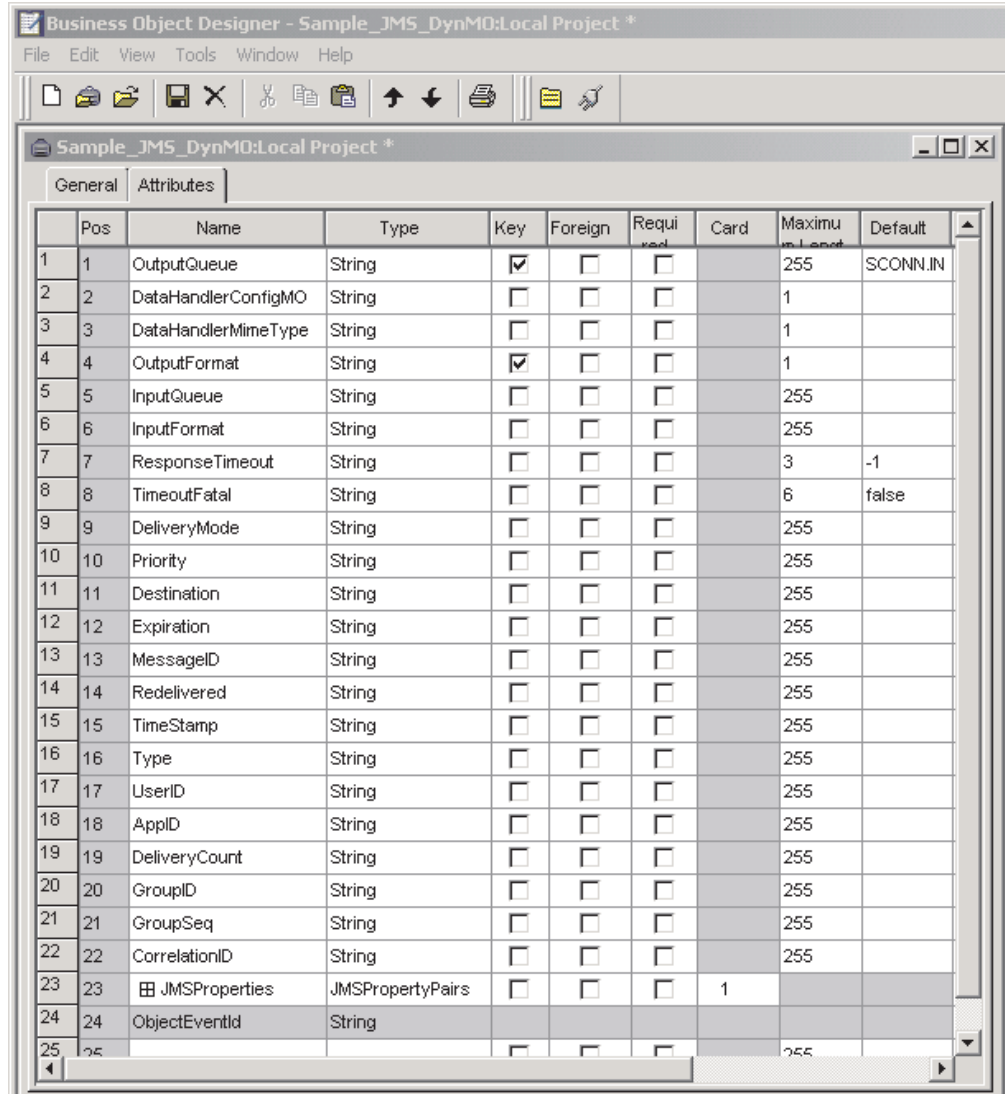


그림 4. 샘플 동적 Meta Object

- 이 Business Object에 대한 요구사항을 반영하도록 속성 및 등록 정보를 편집한 후 저장하십시오.
- 동적 Meta Object를 최상위 레벨 오브젝트의 하위로 추가하고 최상위 레벨 오브젝트 ASI에 이름-값 쌍, cw_mo_conn=<MO 속성>을 포함시키십시오. 여기서 <MO 속성>은 동적 Meta Object를 표시하는 최상위 레벨 오브젝트의 속성 이름입니다.

폴링 중에 동적 하위 Meta Object 채우기

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

38 페이지의 표 14는 폴링에 맞게 동적 하위 Meta Object를 구조화할 수 있는 방법을 보여줍니다.

표 14. 폴링용 JMS 동적 하위 Meta Object 구조

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

표 14에 표시된 대로 추가 속성, Input_Format과 Inputdestination을 동적 하위 Meta Object에 정의할 수 있습니다. InputDestination 속성에는 제공된 메시지가 검색되는 목적지의 이름이 포함되는 반면, Input_Format은 검색되는 메시지 형식으로 채워집니다. 이러한 등록 정보를 하위 Meta Object에 정의하지 않으면, 등록 정보는 채워지지 않습니다.

예제 시나리오:

- 커넥터는 대기열 MyInputQueue에서 형식 CUST_IN의 메시지를 검색합니다.
- 커넥터는 이 메시지를 고객 Business Object로 변환하고 응용프로그램 특정 텍스트를 확인하여 Meta Object가 정의되어 있는지 판별합니다.
- 정의되어 있으면, 커넥터는 이 Meta Object의 인스턴스를 작성하고 각각 InputDestination 및 InputFormat 속성을 채운 후 사용 가능한 협업에 Business Object를 공개합니다.

JMS 머리글 및 동적 하위 Meta Object 속성

메시지 전송에 대한 자세한 정보와 더 많은 제어를 얻기 위해 동적 Meta Object에 속성을 추가할 수 있습니다. 이 섹션에서는 이 속성과 이런 속성이 이벤트 공고와 요청 처리에 어떤 영향을 미치는지에 대해 설명합니다.

JMS 등록 정보: 동적 Meta Object에 있는 다른 속성과 달리, JMSProperties는 단일 카디널리티 하위 오브젝트를 정의해야 합니다. 이 하위 오브젝트에 있는 모든 속성은 다음과 같이 JMS 메시지 머리글의 변수 부분에서 읽고 쓰여지는 하나의 등록 정보를 정의해야 합니다.

1. 속성의 이름에는 시멘틱 값이 없습니다.
2. 속성의 유형은 JMS 등록 정보 유형에 관계없이 항상 String이어야 합니다.
3. 속성의 응용프로그램 특정 정보에는 속성이 맵핑하는 JMS 메시지 등록 정보의 이름과 형식을 정의하는 두 개의 이름-값 쌍이 있어야 합니다. 이름은 사용자 정의될 수 있습니다. 값 유형은 다음 중 하나이어야 합니다.
 - Boolean
 - String
 - Int

- Float
- Double
- Long
- Short
- Byte

아래에 있는 표는 JMSProperties 오브젝트에 있는 속성에 정의해야 하는 응용프로그램 특정 정보 등록 정보를 보여줍니다.

표 15. JMS 등록 정보 속성의 응용프로그램 특정 정보

속성	가능한 값	ASI	주석
이름	임의의 올바른 JMS 등록 정보 이름(올바른 값 = ASI에 정의된 유형과 호환 가능함)	name=<JMS 등록 정보 이름>;type=<JMS 등록 정보 유형>	어떤 벤더는 확장된 기능을 제공하기 위해 특정 등록 정보를 예약해 놓았습니다. 일반적으로 사용자가 이 벤더 특정 기능에 대한 액세스를 찾지 않으면, JMS로 시작하는 사용자 정의 등록 정보를 정의하지 않아야 합니다.
유형	String	type=<주석 참조>	이는 JMS 등록 정보의 유형입니다. JMS API는 JMS 메시지에 있는 값을 설정하기 위해 많은 메소드(setIntProperty, setLongProperty, setStringProperty 등)를 제공합니다. 여기에 지정되는 JMS 등록 정보의 유형은 이 메소드 중 어느 것이 메시지에서 등록 정보 값을 설정하기 위해 사용되는지를 말해줍니다.

아래 예에서 JMSProperties 하위 오브젝트는 Customer 오브젝트에서 메시지 머리글의 사용자 정의 필드에 액세스할 수 있도록 정의됩니다.

```
Customer (ASI = cw_mo_conn=MetaData)
|-- Id
|-- FirstName
|-- LastName
|-- ContactInfo
|-- MetaData
    |-- OutputFormat = CUST
    |-- OutputDestination = QueueA
    |-- JMSProperties
        |-- RoutingCode = 123 (ASI= name=RoutingCode;type=Int)
        |-- Dept = FD (ASI= name=RoutingDept;type=String)
```

다른 예에 대해하기 위해 그림 5에서는 동적 Meta Object에 있는 JMSProperties 속성과 JMS 메시지 머리글에 있는 네 개의 등록 정보(ID, GID, RESPONSE 및 RESPONSE_PERSIST)에 대한 정의를 표시합니다. 속성의 응용프로그램 특정 정보는 각각의 이름과 유형을 정의합니다. 예를 들어, 속성 ID는 String 유형의 JMS 등록 정보 ID에 매핑됩니다.

	Pos	Name	Type	Key	Reqd	Card	App Spec Info
1	1	JMSProperties	TeamCenter_JMS_Properties	<input type="checkbox"/>	<input type="checkbox"/>	1	
1.1	1.1	ID	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		name=ID,type=String
1.2	1.2	GID	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		name=GID,type=String
1.3	1.3	RESPONSE	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		name=RESPONSE,type=Boolean
1.4	1.4	RESP_PERSIST	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		name=RESPONSE_PERSIST,type=Boolean
1.5	1.5	ObjectEventId	String				
2	2	OutputFormat	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

그림 5. 동적 Meta Object의 JMS 등록 정보 속성

시작 스크립트 구성

커넥터는 운영 체제에 따라 JMS.bat 또는 JMS.sh 시작 스크립트와 수반됩니다. 커넥터 시작, 커넥터 중지 및 커넥터의 임시 시작 로그 파일에 대한 정보는 사용 중인 플랫폼용 시스템 설치 안내서의 시작 장을 참조하십시오.

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 40 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.

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 16에 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.

표 16. 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 [-cconfigFile ]
```

– 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 `-c`는 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』

커넥터와 함께 샘플 Business Object만 제공됩니다. 시스템 통합자, 컨설턴트 또는 고객이 Business Object를 빌드해야 합니다.

이 장에서는 Business Object를 위한 커넥터 요구사항에 대해 설명합니다. 이 정보는 새 Business Object를 구현하기 위한 안내서로 사용할 수 있습니다.

커넥터 Business Object 구조

커넥터를 설치한 후, Business Object를 작성해야 합니다. 구성된 Data Handler에 의해 포함된 Business Object가 아닌 다른 Business Object의 구조에 대해서는 요구사항이 없습니다. 커넥터가 처리하는 Business Object는 통합 브로커에서 허용되는 이름을 가질 수 있습니다. 이름 지정 규칙에 대한 자세한 설명은 통합 브로커의 이름 지정 규칙을 참조하십시오.

커넥터는 목적지에서 메시지를 검색하고 Business Object(Meta Object에 의해 정의된)를 메시지 내용으로 채우려고 합니다. 엄밀하게 말하면, 커넥터는 Business Object 구조를 제어하거나 영향을 주지 않습니다. 메시지는 커넥터의 Data Handler 요구사항과 Meta Object 정의의 기능입니다. 사실상, Business Object 레벨의 응용프로그램 텍스트는 없습니다. 오히려 Business Object를 검색하고 전달할 때 커넥터의 주요 역할은 오류에 대한 메시지와 Business Object 간의 프로세스를 모니터링하는 것입니다.

Business Object 작성

1. 브로커가 JMS 어댑터의 인스턴스로 구성될 때 통합 브로커에서 Business Object를 전송하는 응용프로그램을 식별하고 구성하십시오.
2. JMS 목적지로부터의 메시지를 대상 응용프로그램에서 처리하기 적합한 Business Object로 변환할 수 있는 Data Handler로 커넥터를 구성하십시오.
DataHandlerConfigMO 및 DataHandlerMimeType 커넥터 등록 정보를 지정하거나 DataHandlerClassName 등록 정보를 지정하여 이를 수행할 수 있습니다. 자세한 정보는 21 페이지의 『커넥터 등록 정보 구성』 항목을 참조하십시오. 선택적으로 정적 및 동적 Meta Object에서 규칙을 처리하는 특수 Data Handler를 지정할 수 있습니다. 자세한 정보는 30 페이지의 『Meta Object 등록 정보』 항목을 참조하십시오.
3. 응용프로그램 특정 Business Object를 작성하려면 Business Object Designer를 사용하십시오. 자세한 정보는 *Business Object Designer* 안내서를 참조하십시오.

4. 작성하는 Business Object를 지원되는 Business Object에 추가하십시오. Connector Configurator를 사용하여 JMS 어댑터에 대해 지원되는 **Business Object** 탭을 누르고 작성한 Business Object를 추가한 후 메시지 세트 **ID**를 지원되는 Business Object 각각에 고유한 값으로 설정하십시오. Connector Configurator 사용에 대한 자세한 정보는 80 페이지의 『지원되는 Business Object 정의 지정』을 참조하십시오.

제 4 장 문제점 해결

- 『오류 처리』
- 48 페이지의 『추적』
- 49 페이지의 『시작 문제점 수정』

이 장에서는 커넥터를 시작하거나 실행할 때 발생할 수 있는 문제점과 커넥터에서 오류 및 추적을 처리하는 방법에 대해 설명합니다.

오류 처리

커넥터에서 생성되는 모든 오류 메시지는 JMSConnector.txt라고 하는 메시지 파일에 저장됩니다. (파일 이름은 LogFileName 표준 커넥터 구성 등록 정보에 의해 판별됩니다.) 각 오류에는 오류 번호와 그 뒤에 나오는 오류 메시지가 있습니다.

메시지 번호
메시지 텍스트

커넥터는 다음 절에 설명된 대로 특정 오류를 처리합니다.

응용프로그램 제한시간

다음과 같은 경우, 오류 메시지인 ABON_APPRESPONSETIMEOUT이 리턴됩니다.

- 커넥터는 메시지를 검색하는 동안 JMS 서비스 제공자에 대한 연결을 설정할 수 없습니다.
- 커넥터는 정상적으로 Business Object를 메시지로 변환하지만 연결 손실로 인해 메시지를 전송 대기열에 전달할 수 없습니다.
- 커넥터는 메시지를 발행하지만 변환 등록 정보 TimeoutFatal이 True인 Business Object에 대해 응답을 기다리는 중에 제한시간에 도달합니다.
- 커넥터는 리턴 코드가 APP_RESPONSE_TIMEOUT 또는 UNABLE_TO_LOGIN인 응답 메시지를 수신합니다.

등록되지 않은 Business Object

커넥터가 등록되지 않은 Business Object와 연관된 메시지를 검색하거나, NO_SUBSCRIPTION_FOUND 코드가 getAppEvent() 메소드에 의해 리턴되는 경우, 커넥터는 UnsubscribedDestination 등록 정보에 의해 지정된 대기열로 메시지를 전달합니다.

주: UnsubscribedDestination이 정의되지 않은 경우, 등록되지 않은 메시지는 버려집니다.

활성화되지 않은 커넥터

gotAppEvent() 메소드가 CONNECTOR_NOT_ACTIVE 코드를 리턴할 때, 지정된 경우 pollForEvents() 메소드는 APP_RESPONSE_TIMEOUT 코드를 리턴하고 이벤트는 InProgress 목적지에 남게 됩니다.

Data Handler 변환

Data Handler가 메시지를 Business Object로 변환하는 데 실패하거나 Business Object에 고유한 처리 오류가 발생할 경우(JMS 제공자와 반대로), 메시지는 ErrorDestination에 의해 지정된 대기열로 전달됩니다. ErrorDestination가 정의되지 않은 경우, 오류로 인해 처리할 수 없는 메시지는 버려집니다.

Data Handler가 Business Object를 메시지로 변환하는 데 실패하면, BON_FAIL이 리턴됩니다.

추적

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

다음에 나오는 내용은 커넥터 추적 메시지에 대해 권장되는 내용입니다.

- 레벨 0 이 레벨은 커넥터 버전을 식별하는 추적 메시지에 사용됩니다.
- 레벨 1 처리되는 각 Business Object에 대한 주요 정보를 제공하거나 폴링 스레드가 입력 대기열에서 새 메시지를 발견할 때마다 기록하는 추적 메시지에 대해 이 레벨을 사용합니다.
- 레벨 2 Business Object가 gotAppEvent() 또는 executeCollaboration()에서 브로커에 게시될 때마다 기록하는 추적 메시지에 대해 이 레벨을 사용합니다.
- 레벨 3 메시지와 Business Object 간의 변환에 관한 정보를 제공하거나 출력 대기열로 메시지를 전달하는 데에 대한 정보를 제공하는 추적 메시지에 대해 이 레벨을 사용합니다.
- 레벨 4 커넥터가 기능을 입력하거나 종료하는 시기를 식별하는 추적 메시지에 대해 이 레벨을 사용합니다.
- 레벨 5 커넥터 초기화를 표시하거나, 응용프로그램에서 실행되는 명령문을 표시하거나, 메시지가 대기열에 놓이거나 대기열에서 나갈 때마다 표시하거나 또는 Business Object 덤프를 기록하는 추적 메시지에 대해 이 레벨을 사용합니다.

시작 문제점 수정

문제점

커넥터가 초기화 중 다음 메시지와 함께 예상치 않게 종료되었습니다.

Exception in thread "main" java.lang.

NoClassDefFoundError: javax/jms/JMSEException...

커넥터가 초기화 중 다음 메시지와 함께 예상치 않게 종료되었습니다.

Exception in thread "main" java.lang.

NoClassDefFoundError: javax/naming/Referenceable...

가능한 솔루션 / 설명

커넥터가 jms.jar 파일을 찾을 수 없습니다.

커넥터가 jndi.jar 파일을 찾을 수 없습니다.

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 17은 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 17. 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 17. 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 17. 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 17. 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 아닐 경우에는 CwXML: 저장소 디렉토리가 <REMOTE>인 경우에는 CwBO
WsifSynchronousRequest 제한시간	0 - 임의의 숫자 (밀리초)	0	구성요소 다시 시작	WAS만 해당됨
XMLNameSpaceFormat	short, long	short	에이전트 다시 시작	WebSphere MQ 메시징 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 58 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.

- 69 페이지의 『Connector Configurator 개요』
- 70 페이지의 『Connector Configurator 시작』
- 71 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 74 페이지의 『새 구성 파일 작성』
- 77 페이지의 『구성 파일 등록 정보 설정』
- 85 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(71 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, 72 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(76 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 72 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 - 커넥터 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 템플릿가 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 80 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. 78 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

52 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(ICS에만 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. ICS는 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>
```

부록 C. 자습서

- 『자습서 개요』
- 88 페이지의 『환경 설정』
- 90 페이지의 『시나리오 실행』
- 90 페이지의 『정적 Meta Object 시나리오 실행』
- 91 페이지의 『동적 Meta Object 시나리오 실행』

이 부록은 어댑터를 사용하여 JMS 메시지 대기열을 통해 통신하는 응용프로그램에서 Business Object를 전송 및 수신하는 방법을 표시합니다. 자습서의 시나리오는 어댑터 기능의 기본 특징을 표시하도록 설계되었습니다.

표기 규칙에 대한 안내는 이 문서의 서문을 참조하십시오.

자습서 개요

자습서는 정적 Meta Object를 사용하는 것과 동적 Meta Object를 사용하는 것의 두 가지 시나리오로 이루어져 있습니다. 두 시나리오 모두 ApplicationX를 포함하고 있어 작성, 갱신 또는 삭제 시 회사 담당자 정보를 교환할 수 있습니다. 작성한 Sample_JMS_Contact은 ApplicationX의 메시지에 정의된 필드에 일치됩니다. ApplicationX는 IBM WebSphere Business Integration 개발 킷에서 사용할 수 있는 한계가 정해진 Data Handler와 호환 가능한 형식으로 메시지를 전송하고 수신합니다.

자습서는 또한 WebSphere Adapters 설치에 포함되는 Port Connector 저장소를 사용합니다. Port Connector는 하위 코드가 없는 커넥터 정의로 구성되며 시뮬레이션 시나리오에 가장 적합합니다.

시작된 후 JMS 어댑터는 ApplicationX에 의해 입력 대기열에 게시된 담당자 메시지를 검색합니다. 어댑터는 한계가 정해진 Data Handler를 사용하여 이런 메시지를 Sample_JMS_Contact Business Object로 변환하고 이를 통합 브로커에 전달합니다. Test Connector(WBI 설치에 포함되는)를 사용하여 Port Connector를 시뮬레이션하고 JMS 어댑터에 의해 게시된 Business Object를 검색하며 속성을 조사할 수 있습니다. 데이터 변경 후에 데이터가 JMS 어댑터에 전송되고 메시지로 변환되고 어댑터의 출력 대기열(ApplicationX의 입력 대기열)로 전달된 통합 브로커에 메시지를 다시 전달합니다. 자습서에서, 어댑터는 WebSphere MQ Integrator Broker용으로 구성되었지만 자습서를 실행하기 위해 실제로 이 브로커를 설치할 필요는 없습니다.

이 자습서를 계속하기 전에 다음을 확인해야 합니다.

- IBM WebSphere 제품이 설치되어 있고 이 제품에 대한 지식이 있는지 여부.

- JMS 서비스 제공자가 설치되어 있는지 여부.
- WBI Adapter for JMS가 설치되어 있는지 여부.
- WebSphere MQ 5.3이 설치되어 있고 이에 대한 지식이 있는지 여부.

환경 설정

이 섹션에서는 지습서 작업을 위해 환경을 준비하는 방법을 설명합니다. 설명된 단계에서 *sample_folder*는 샘플이 있는 폴더를 참조합니다.

1. 대기열 작성 지습서에서는 JMS 서비스 제공자와 함께 정의된 여섯 개의 대기열을 필요로 합니다. 이 대기열을 정의하기 전에 JMS 제공자 문서를 확인하십시오. 다음 대기열을 정의하십시오(또는 JNDI 찾아보기를 통해 이 대기열을 사용 가능하게 하십시오).
 - CWLD_Input
 - CWLD_InProgress
 - CWLD_Error
 - CWLD_Archive
 - CWLD_Unsubscribed
 - CWLD_Output
2. 실행 중인 채널 초기자 및 리스너를 통해 **WebSphere MQ** 대기열 관리자를 작성 및 시작하십시오.
3. 다음과 같이 WMQI 브로커 구성을 위해 WebSphere MQ 어댑터 및 Port Connector에 필요한 대기열을 정의하십시오.
 - DEFINE QL('JMSConnector/ADMININQUEUE')
 - DEFINE QL('JMSConnector/ADMINOUTQUEUE')
 - DEFINE QL('JMSConnector/DELIVERYQUEUE')
 - DEFINE QL('JMSConnector/FAULTQUEUE')
 - DEFINE QL('JMSConnector/REQUESTQUEUE')
 - DEFINE QL('JMSConnector/RESPONSEQUEUE')
 - DEFINE QL('JMSConnector/SYNCHRONOUSREQUESTQUEUE')
 - DEFINE QL('JMSConnector/SYNCHRONOUSRESPONSEQUEUE')
 - DEFINE QL('PortConnector/ADMININQUEUE')
 - DEFINE QL('PortConnector/ADMINOUTQUEUE')
 - DEFINE QL('PortConnector/DELIVERYQUEUE')
 - DEFINE QL('PortConnector/FAULTQUEUE')
 - DEFINE QL('PortConnector/REQUESTQUEUE')
 - DEFINE QL('PortConnector/RESPONSEQUEUE')

- DEFINE QL('PortConnector/SYNCHRONOUSREQUESTQUEUE')
 - DEFINE QL('PortConnector/SYNCHRONOUSRESPONSEQUEUE')
4. 어댑터 구성 Connector Configurator를 사용하여 *sample_folder*\JMSConnector.cfg를 여십시오. Connector Configurator 사용에 대한 자세한 정보는 69 페이지의 부록 B 『Connector Configurator』를 참조하고 커넥터 특정 등록 정보에 대한 자세한 정보는 22 페이지의 『커넥터 특정 등록 정보 구성』을 참조하십시오. 다음 표준 등록 정보를 설정하십시오.
- Broker Type 이 등록 정보를 WMQI로 설정하십시오.
 - Repository Directory 이 등록 정보를 *sample_folder* 디렉토리로 설정하십시오.
- 다음 커넥터 특정 등록 정보를 설정하십시오.
- ConfigurationMetaObject 이 등록 정보를 Sample_JMS_MO_Config로 설정하십시오.
 - DataHandlerConfigMO 이 등록 정보를 Sample_JMS_MO_DataHandler로 설정하십시오.
 - DataHandlerMimeType 이 등록 정보를 text/delimited로 설정하십시오.
 - ErrorQueue 이 등록 정보를 CWLD_Error로 설정하십시오.
 - InProgressQueue 이 등록 정보를 CWLD_InProgress로 설정하십시오.
 - InputQueue 이 등록 정보를 CWLD_Input으로 설정하십시오.
 - UnsubscribedQueue 이 등록 정보를 CWLD_Unsubscribed로 설정하십시오.
5. **Port Connector** 구성 Connector Configurator를 사용하여 다음 표준 등록 정보를 설정하십시오.
- Broker Type 이 등록 정보를 WMQI로 설정하십시오.
 - Repository Directory 이 등록 정보를 *sample_folder* 디렉토리로 설정하십시오.
 - RequestQueue 이 등록 정보를 JMSConnector/DELIVERYQUEUE(JMS 어댑터에 대한 DeliveryQueue 등록 정보 값)로 설정하십시오.
 - DeliveryQueue 이 등록 정보를 JMSConnector/REQUESTQUEUE (JMS 어댑터에 대한 RequestQueue 등록 정보 값)로 설정하십시오.
6. **Business Object** 지원 Business Object를 사용하려면 어댑터에서 먼저 Business Object를 지원해야 합니다. Connector Configurator를 사용하여 JMS 어댑터에 대해 지원되는 **Business Object** 탭을 누르고 표 18에 표시된 Business Object를 추가한 후 메시지 세트 ID를 지원되는 Business Object 각각에 고유한 값으로 설정하십시오.

표 18. JMS 어댑터에 지원되는 샘플 Business Object

Business Object 이름	메시지 세트 ID
Sample_JMS_MO_Config	1
Sample_JMS_MO_DataHandler	2
Sample_JMS_Contact	3

Connector Configurator를 사용하여 *sample_folder*에서 제공된 Port Connector 정의, PortConnector.cfg를 열고 표 19에 표시된 지원되는 Business Object 및 메시지 세트 ID를 추가하십시오.

표 19. Port Connector에 지원되는 샘플 Business Object

Business Object 이름	메시지 세트 ID
Sample_JMS_Contact	1

7. 커넥터 시작 스크립트 작성 및 갱신

Windows:

- Adapter for JMS 단축 아이콘의 등록 정보를 여십시오.
- 대상의 마지막 인수에서 `-c <JMSConnector.cfg 파일의 전체 경로 및 파일 이름>`을 추가하십시오.

예:

```
-cProduct_Dir\connectors\JMS\samples\JMSConnector.cfg
```

UNIX:

- `Product_Dir/bin/connector_manager_JMS` 파일을 여십시오.
- AGENTCONFIG_FILE 등록 정보 값을 `-c <JMSConnector.cfg 파일의 전체 경로 및 파일 이름>`으로 설정하십시오.

예:

```
AGENTCONFIG_FILE=-cProduct_Dir/connectors/JMS/samples/JMSConnector.cfg
```

시나리오 실행

시나리오를 실행하기 전에 다음을 수행하십시오.

- 아직 실행 중이 아닌 경우, **Adapter for JMS**를 시작하십시오.
- 아직 실행 중이 아닌 경우, **Visual Test Connector**를 시작하십시오.

정적 Meta Object 시나리오 실행

이 자습서 파트에서는 정적 Meta Object를 사용하는 시나리오에 대해 설명합니다. 정적 Meta Object에 대한 자세한 설명은 32 페이지의 『정적 Meta Object 구성』을 참조하십시오.

- Port Connector** 시뮬레이트 Visual Test Connector를 사용하여 PortConnector의 프로파일을 정의하십시오.

- a. Visual Test Connector 메뉴에서 파일 -> 프로파일 작성/선택을 선택한 후 커넥터 프로파일 메뉴에서 파일 -> 새 프로파일을 선택하십시오.
 - b. Samples 디렉토리에서 Port Connector 구성 파일, PortConnector.cfg를 선택한 후 커넥터 이름 및 브로커 유형을 구성하고 확인을 누르십시오.
 - c. 작성한 프로파일을 선택하고 확인을 누르십시오.
 - d. Visual Test Connector 메뉴에서 파일 -> 연결을 선택하여 시뮬레이팅을 시작하십시오.
2. 요청 처리 테스트
- a. Test Connector를 통해 **BoType** 드롭 다운 상자에서 Business Object를 선택한 후 BOInstance의 작성을 선택하여 새 Business Object 인스턴스, Sample_JMS_Contact를 작성하십시오.
 - b. 원하는 경우 기본값을 변경하십시오. Verb를 **Create**로 설정하고 **BO** 전송을 눌러 메시지를 전송하십시오.
3. 메시지 전달 확인 CWLD_Output 대기열을 열어 JMS 어댑터에서 CON_CR 형식의 새 접속 메시지가 도착했는지 확인하십시오.
4. 이벤트 처리 테스트 메시지를 JMS 어댑터의 입력 대기열에 전송하십시오. 참고: 이 단계를 수행하려면 대기열에 메시지를 전송할 수 있는 유틸리티가 필요합니다. 그렇지 않으면, 용이한 접근을 구현하기 위해 어댑터가 자체 메시지를 폴링할 수 있도록 JMS 어댑터의 InputQueue 등록 정보를 CWLD_Output으로 설정할 수 있습니다. 입력 대기열에 메시지가 있으면 JMS 어댑터는 이 메시지를 폴링하고 Sample_JMS_Contact Business Object로 변환하려고 시도합니다. 어댑터가 메시지를 폴링할 수 있도록 하는 핵심 요소는 메시지 형식이 Sample_JMS_MO_Config Meta Object의 Sample_JMS_Contact Business Object와 연관된 값과 동일한지 확인하는 것입니다. 이 시나리오에서 해당 형식은 CON_CR입니다. 어댑터가 수신 메시지 형식을 CON_CR로 식별한 경우, Data Handler를 사용하여 메시지를 Verb가 Create인 Sample_JMS_Contact Business Object로 변환합니다. 그 후에 새로 작성된 Business Object가 Test Connector에 전달됩니다.
5. 메시지 전달 확인 위의 단계를 성공적으로 수행했다면 JMS 어댑터가 메시지를 검색하고 메시지를 Sample_JMS_Contact Business Object로 변환하고 Sample_JMS_Contact Business Object를 접속 메시지로 변환할 수 있게 하는 작업 시나리오를 수행해야 합니다.

동적 Meta Object 시나리오 실행

이 시나리오는 동적 Meta Object를 사용하여 JMS 서비스 제공자에서 정의된 다양한 대기열로 Business Object 경로를 재지정하는 방법에 대해 설명하고 있습니다. 동적 Meta Object에 대한 자세한 설명은 35 페이지의 『동적 하위 Meta Object 구성』을 참조하십시오. 아래 단계를 통해 Sample_JMS_Contact의 하위 Meta Object 속성을 작성할

수 있습니다. 특히, Sample_JMS_Contact Business Object를 여러 대기열로 재지정하기 위해 이 하위 Meta Object의 출력 대기열 값을 수정합니다.

하위 Meta Object 저장소, Sample_JMS_DynMO.xsd는 *sample_folder*에 있습니다.

1. 동적 **Meta Object** 속성 식별 Sample_JMS_Contact의 동적 Meta Object를 포함하는 속성을 식별하려면 우선 응용프로그램 특정 정보를 추가하고, 응용프로그램 특정 정보에 cw_mo_conn=DynMO를 추가해야 합니다. 이 작업을 통해 속성을 식별할 수 있습니다.
2. 속성 추가 Business Object Designer를 사용하여 다음을 수행하십시오.
 - a. *sample_folder*에서 Sample_JMS_DynMO.xsd 및 Sample_JMS_Contact.xsd를 여십시오.
 - b. Sample_JMS_Contact Object 창에서 DynMO라고 이름 지정된 Sample_JMS_DynMO 유형의 속성을 추가하십시오.
 - c. Sample_JMS_Contact 오브젝트를 두 번 누르십시오.
 - d. 속성 폴더를 선택하고 DynMO라고 이름 지정된 Sample_JMS_DynMO 유형의 속성을 추가하십시오.
3. 새 대상 대기열 정의 JMS 서비스 제공자에 REROUTE.IN 임시 대기열을 정의하십시오. 이것은 동적 Meta Object가 Sample_JMS_Contact Business Object 경로를 재지정하는 곳입니다.
4. 아직 실행 중이 아닌 경우, **Adapter for JMS**를 시작하십시오.
5. 아직 실행 중이 아닌 경우, **Visual Test Connector**를 시작하십시오.
6. **Port Connector** 시뮬레이트 Visual Test Connector를 사용하여 PortConnector의 프로파일을 정의하십시오.
 - a. Visual Test Connector 메뉴에서 파일 -> 프로파일 작성/선택을 선택한 후 커넥터 프로파일 메뉴에서 파일 -> 새 프로파일을 선택하십시오.
 - b. Samples 디렉토리에서 Port Connector 구성 파일, PortConnector.cfg를 선택한 후 커넥터 이름 및 브로커 유형을 구성하고 확인을 누르십시오.
 - c. 작성한 프로파일을 선택하고 확인을 누르십시오.
 - d. Visual Test Connector 메뉴에서 파일 -> 연결을 선택하여 시뮬레이팅을 시작하십시오.
7. 상위 **Business Object** 및 하위 **Meta Object**의 인스턴스 작성 Visual Test Connector를 사용하여 다음을 수행하십시오.
 - a. 원하는 경우 기본값을 변경하여 Sample_JMS_Contact Business Object의 새 인스턴스를 작성하십시오.
 - b. DynMO 속성을 오른쪽 마우스로 누르고 Sample_JMS_DynMO Business Object의 인스턴스를 작성하십시오.
8. 새 대상 대기열 설정

- a. DynMO 속성 옆에 있는 + 부호를 눌러 이 속성을 펼치십시오.
 - b. outputQueue라는 속성에서 대상 대기열 이름을 REROUTE.IN으로 입력하십시오.
9. **Business Object 전송 BO 전송**을 누르십시오.
10. 메시지 전달 확인 REROUTE.IN 대기열을 열어 JMS 어댑터에서 새 접속 메시지가 도착했는지 알아보십시오. JMS 어댑터에서 REROUTE.IN이라는 이름의 대기열에 새 메시지가 도착했으면 경로 재지정이 완료된 것입니다.

부록 D. 항목 기반 및 대기열 기반 메시징 구성

- 『대기열 기반 메시징 구성』
- 96 페이지의 『항목 기반 메시징 구성』

이 부록에서는 공통 JMS 제공자로 WebSphere MQ를 사용하는 Adapter for JMS 구성 방법을 보여줍니다. 자세한 정보는 *WebSphere MQ Using Java* 안내서를 참조하십시오.

주: WebSphere MQ를 JMS 제공자로 사용하고 있는 경우, 통합을 위해 WebSphere Business Integration Adapter for WebSphere MQ를 사용할 것을 강력히 권장합니다. 참조용으로만 제공된 아래 단계는 공통 JMS 제공자를 사용하여 JMS 어댑터를 구성하는 방법을 보여줍니다.

표기 규칙에 대한 안내는 이 문서의 서문을 참조하십시오.

대기열 기반 메시징 구성

1. WebSphere MQ 및 WebSphere MQ 클라이언트 라이브러리(JMS 지원 포함)를 설치하십시오.
2. fscontext.jar 및 providerutil.jar를 포함하여 모든 MQ 클라이언트 라이브러리가 시스템의 클래스 경로에 있는지 확인하십시오. 대신, jmsAdmin.bat 파일을 수정하여 Java 명령행 스크립트에 `-Djava.ext.dirs="<MQ 홈 디렉토리>/Java/lib`를 추가하여 모든 클라이언트 라이브러리 파일이 도구에서 사용 가능한지 확인할 수 있습니다. 도구에서 `ClassDefNotFoundsExceptions`가 보고되면 라이브러리가 누락된 것이므로 클래스 경로를 다시 확인해야 합니다.
3. `<MQ 홈 디렉토리>Java/bin/jmsAdmin.config`를 열고 다음 등록 정보를 설정하십시오.
 - `INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory`
 - `PROVIDER_URL=file://c:/temp`
 - `SECURITY_AUTHENTICATION=none`
4. 다음을 포함하는 MyJNDI.txt라는 이름의 파일을 작성하십시오.

```
DEFINE QCF(MyQCF) HOST(<호스트 이름>) +  
PORT(<MQ 리스너 포트 이름(예: 1414)>) +  
CHANNEL(<MQ 서버 연결 채널 이름(예: CHANNEL1)>) +
```

```
QMGR(<MQ 대기열 관리자 이름>) +  
TRAN(클라이언트)  
END
```

5. <MQ 홈 디렉토리>/java/bin/jmsAdmin.bat < MyJNDI.txt를 실행하여 오브젝트를 JNDI 이름에 바인드하십시오.

6. 표시된 대로 다음 JMS 커넥터 특정 등록 정보를 구성하십시오.

```
CTX_InitialContextFactory = com.sun.jndi.fscontext.RefFSContextFactory  
CTX_ProviderURL = file://c:/temp  
ConnectionFactoryName = MyQCF
```

항목 기반 메시징 구성

1. WebSphere MQ 및 WebSphere MQ 클라이언트 라이브러리(JMS 지원 포함)를 설치하십시오.
2. fscontext.jar 및 providerutil.jar를 포함하여 모든 MQ 클라이언트 라이브러리가 시스템의 클래스 경로에 있는지 확인하십시오. 대신, jmsAdmin.bat 파일을 수정하여 Java 명령행 스크립트에 -Djava.ext.dirs="<MQ 홈 디렉토리>/Java/lib를 추가하여 모든 클라이언트 라이브러리 파일이 도구에서 사용 가능한지 확인할 수 있습니다. 도구에서 ClassNotFoundException가 보고되면 라이브러리가 누락된 것이므로 클래스 경로를 다시 확인해야 합니다.
3. MQ에서 항목 기반(Pub/Sub) 메시징 지원을 사용 가능하게 하려면 IBM에서 적절한 WebSphere MQ MA0C SupportPac를 다운로드하여 설치하십시오. 예를 들어, ma0c_ntmq52 검색은 Windows MQ 5.2용 토픽 기반 메시징 패치를 찾습니다.
4. 디렉토리를 <MQ 홈 디렉토리>/Java/bin으로 변경한 후 runmqsc < MQJMS_PSQ.mqsc를 실행하십시오.
5. IVTSetup.bat를 실행하십시오. 이 프로세스가 오류 보고 없이 완료!를 표시해야 합니다.
6. <MQ 홈 디렉토리>Java/bin/jmsAdmin.config를 열고 다음 등록 정보를 설정하십시오.
 - INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory
 - PROVIDER_URL=file://c:/temp
 - SECURITY_AUTHENTICATION=none
7. 다음을 포함하는 MyJNDI.txt라는 이름의 파일을 작성하십시오.

```
DEFINE QCF(MyQCF) HOST(<호스트 이름>) +  
PORT(<MQ 리스너 포트 이름(예: 1414)>) +  
CHANNEL(<MQ 서버 연결 채널 이름(예: CHANNEL1)>) +
```

```
QMGR(<MQ 대기열 관리자 이름>) +  
TRAN(클라이언트)  
END
```

8. <MQ 홈 디렉토리>/java/bin/jmsAdmin.bat < MyJNDI.txt를 실행하여 오브젝트를 JNDI 이름에 바인드하십시오.
9. 표시된 대로 다음 JMS 커넥터 특정 등록 정보를 구성하십시오.

```
CTX_InitialContextFactory = com.sun.jndi.fscontext.RefFSContextFactory  
CTX_ProviderURL = file://c:/temp  
ConnectionFactoryName = MyQCF
```

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