



Adapter for XML 사용자 안내서

버전 3.3.x



Adapter for XML 사용자 안내서

버전 3.3.x

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 81 페이지의 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

이 문서 개정판은 새 개정판에서 별도로 명시하지 않는 한 커넥터 버전 3.3.x 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견을 보내시려면 ibmkspoe@kr.ibm.com으로 전자 우편을 보내십시오. 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.

목차

이 책의 정보	v
이 책의 사용자	v
이 책을 읽기 전에	v
관련 서적	v
인쇄 규칙	vi
이 릴리스의 새로운 기능	vii
버전 3.3.x.	vii
이전 릴리스	vii
제 1 장 개요	1
커넥터 구성요소	1
커넥터 작동 방법	4
제 2 장 커넥터 설치 및 구성.	9
호환성	9
전제조건	10
XML 어댑터 설치	10
커넥터 구성	10
Data Handler에 대한 최상위 Meta Object 구성	15
공통 구성 작업	16
Data Handler 지정.	16
다중 커넥터 인스턴스 작성	17
커넥터 시작	18
커넥터 중지	20
제 3 장 커넥터에 대한 Business Object 개발	21
커넥터 구현 계획	21
커넥터 Business Object 처리	22
커넥터 Business Object 구조	22
이벤트 공고에 대한 Business Object.	26
XML DTD 또는 스키마 문서 기반 Business Object	27
제 4 장 사용자 정의 프로토콜 핸들러 빌드.	29
프로토콜 핸들러 프레임워크	29
프로토콜 핸들러 클래스 작성	31
프로토콜 핸들러 프레임워크 메소드	31
사용자 정의 프로토콜 핸들러 샘플 코드.	33
부록 A. 커넥터의 표준 구성 등록 정보	35
신규 및 삭제된 등록 정보	35
표준 커넥터 등록 정보 구성.	35
표준 등록 정보 요약	37
표준 구성 등록 정보	40

부록 B. Connector Configurator	53
Connector Configurator 개요	53
Connector Configurator 시작	54
System Manager에서 Configurator 실행	55
커넥터 특정 등록 정보 템플릿 작성	55
새 구성 파일 작성	58
기존 파일 사용	60
구성 파일 완료	61
구성 파일 등록 정보 설정	61
구성 파일 저장	68
구성 파일 변경	69
구성 완료	69
국제화된 환경에서 Connector Configurator 사용	69
부록 C. XML 어댑터 샘플 시나리오의 개요	71
WebSphere MQ Integrator Broker 연결로 XML 샘플 시나리오 설치	72
설치 전 참고 및 가정사항	72
샘플 시나리오 설치	72
서비스 호출 요청 시나리오 실행	74
폴 시나리오 실행	75
WebSphere InterChange Server 연결로 XML 샘플 시나리오 설치	76
설치 전 참고 및 가정사항	76
샘플 시나리오 설치	77
서비스 호출 요청 시나리오 실행	78
폴 시나리오 실행	79
주의사항	81
프로그래밍 인터페이스 정보	82
상표 및 서비스표	83
색인	85

이 책의 정보

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

이 책에서는 IBM WebSphere Business Integration Adapter for XML의 설치, 구성, 비즈니스 오브젝트 개발 및 문제점 해결에 대해 설명합니다.

이 책의 사용자

이 책은 WebSphere Business Integration System의 일부로 커넥터를 구현하는 WebSphere 컨설턴트 및 고객을 위한 것입니다. 이 책의 사용자는 다음 영역에 대해 잘 알고 있어야 합니다.

- 커넥터 개발
- Business Object 개발
- HTTP 및 HTTPS 기반 응용프로그램 구조

이 책을 읽기 전에

WebSphere Business Integration Adapter 시스템, Business Object 개발 및 Data Handler에 익숙해야 합니다. XML 마크업 언어 및 스키마 언어, DTD(Document Type Definition) 또는 XSDL(스키마 문서의 경우) 중 하나에 익숙해야 합니다.

관련 서적

이 제품과 함께 사용할 수 있는 전체 문서 세트에는 모든 WebSphere Business Integration Adapter 설치에 공통되는 사양 및 구성요소에 대한 설명과 특정 구성요소에 대한 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 경우에는 다음 웹 사이트를 참조하십시오.

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

InterChange Server와 함께 어댑터를 사용하는 경우에는 다음 웹 사이트를 참조하십시오.

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

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

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

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

WebSphere Application Server에 대한 자세한 정보는 다음 웹 사이트를 참조하십시오.

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

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

인쇄 규칙

이 책에서는 다음과 같은 규칙을 사용합니다.

courier 글꼴	명령어, 파일 이름, 입력하는 정보 또는 시스템이 화면에 인쇄하는 정보와 같은 리터럴 값을 표시합니다.
굵은체	처음 나타나는 새 용어를 표시합니다.
가울임	변수 이름이나 상호 참조를 표시합니다.
파란색 텍스트	매뉴얼을 온라인으로 볼 경우에만 볼 수 있는 파란색 텍스트는 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 바로 가려면 파란색 텍스트를 누르십시오.
{ }	구문 행에서, 사용자가 하나만을 선택해야 하는 세트는 중괄호 안에 표시됩니다.
[]	구문 행에서, 선택적 매개변수는 대괄호 안에 표시됩니다.
...	구문 행에서, 생략 부호는 이전 매개변수의 반복을 표시합니다. 예를 들어, option[,...]은 쉼표로 구분되는 여러 개의 옵션을 입력할 수 있습니다.
< >	이름 지정 규칙에서 꺾쇠괄호는 <server_name><connector_name> tmp.log에서처럼 이름의 각 요소를 묶어 이를 서로 구별합니다.
/, \	이 책에서 백슬래시(\)는 디렉토리 경로의 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체하십시오. 모든 WebSphere Business Integration System 제품의 경로 이름은 시스템에 제품을 설치한 디렉토리 및 관련이 있습니다.
%text% 및 \$text	퍼센트(%) 내의 텍스트는 Windows text 시스템 변수나 사용자 변수의 값을 표시합니다. UNIX 환경에서 이에 해당하는 표기는 \$text로, 이는 text UNIX 환경 변수의 값을 표시합니다.
ProductDir	제품 설치 디렉토리를 표시합니다.

이 릴리스의 새로운 기능

버전 3.3.x

Adapter for XML은 일반 유지보수 수정사항과 함께 갱신되었습니다.

3.3 버전으로 시작할 때 Adapter for XML은 더 이상 Microsoft Windows NT에서 지원되지 않습니다.

이 안내서 어댑터 설치 정보가 이동되었습니다. 해당 정보의 새 위치는 10 페이지의 『XML 어댑터 설치』를 참조하십시오.

이전 릴리스

다음은 이전 버전의 변경사항입니다.

버전 3.2.x

이 사용자 안내서의 새 부록인 71 페이지의 부록 C 『XML 어댑터 샘플 시나리오의 개요』에는 XML 어댑터를 테스트하기 위한 프로시저가 들어 있습니다.

어댑터는 이제 WebSphere Application Server를 통합 브로커로 사용할 수 있습니다. 자세한 정보는 9 페이지의 『호환성』을 참조하십시오.

어댑터는 이제 다음 플랫폼에서 실행됩니다.

- Solaris 7, 8
- AIX 5.x

버전 3.1.x

2003년 3월에 갱신되었습니다. 전체 시스템을 설명하거나 구성요소 또는 도구의 이름을 수정하기 위해 더 이상 “CrossWorlds” 이름을 사용하지 않습니다. 그 외는 이전과 거의 동일합니다. 예를 들어 “CrossWorlds System Manager”의 경우 이제 “System Manager”로, “CrossWorlds InterChange Server”는 이제 “WebSphere InterChange Server”로 사용됩니다.

버전 3.0.x

IBM WebSphere Business Integration Adapter for XML과 함께 제공된 커넥터는 국제화되어 있습니다. 자세한 정보는 8 페이지의 『로케일 종속 데이터 처리』 및 35 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

버전 2.5.x

IBM WebSphere Business Integration Adapter for XML에는 Connector for XML 이 있습니다. 이 어댑터는 ICS(InterChange Server) 및 WebSphere MQIntegrator 통합 브로커로 작동합니다. 통합 브로커는 이기종의 응용프로그램 세트 통합을 수행하는 응용프로그램으로, 데이터 라우팅을 포함하는 서비스를 제공합니다.

어댑터에는 다음이 포함됩니다.

- XML 특정 응용프로그램 구성요소
- \connectors\XML\samples 디렉토리에 포함된 샘플 Business Object
- 다음으로 구성된 IBM WebSphere Adapter Framework
 - 커넥터 프레임워크
 - 개발 도구(Business Object Designer 및 Connector Configurator 포함)
 - API(CDK, JCDK 및 CDK 포함)

이 매뉴얼에서는 두 가지 통합 브로커, 즉 InterChange Server(ICS) 및 WebSphere MQ Integrator Broker에서 이 어댑터를 사용하는 방법에 대한 정보를 제공합니다.

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

제 1 장 개요

이 장에서는 IBM WebSphere Business Integration Adapter for XML의 커넥터 구성요소에 대해 설명합니다. 커넥터는 통합 브로커가 HTTP 및 HTTPS 프로토콜을 사용하여 URL을 통해 Business Object를 교환할 수 있게 합니다. URL은 웹 서버상의 원격 응용프로그램이거나 Servlet과 같은 대상일 수 있습니다. 커넥터는 XML 버전 1.0을 지원합니다.

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 언급되어 있습니다.

커넥터와 통합 브로커의 관계에 대한 자세한 정보는 *IBM WebSphere InterChange Server System Administration Guide* 또는 *IBM WebSphere Business Integration Implementation Guide for WebSphere MQ Integrator Broker*를 참조하십시오.

주: XML 환경에서 작업할 때, 제품과 함께 제공되는 커넥터를 사용하거나 사용자 정의 모듈을 빌드할 수 있습니다. 어느 것을 사용할 것인지 결정하는 지침은 21 페이지의 『커넥터 구현 계획』을 참조하십시오.

이 장에는 다음 절이 포함되어 있습니다.

- 『커넥터 구성요소』
- 4 페이지의 『커넥터 작동 방법』

커넥터 구성요소

어댑터는 Java로 작성되며 다음 세 가지 구성요소로 구성됩니다.

- 커넥터
- XML Data Handler
- 프로토콜 핸들러(HTTP 및 HTTPS)

커넥터는 XML Data Handler와 상호작용합니다. Data Handler에 대해서는 *Data Handler* 안내서를 참조하십시오.

그림 1은 커넥터 구성요소의 구조를 보여줍니다. 커넥터의 모듈 형태 설계를 이용하면 사용자 정의 구성요소를 설계하여 제품 제공 기능을 대체할 수 있습니다.

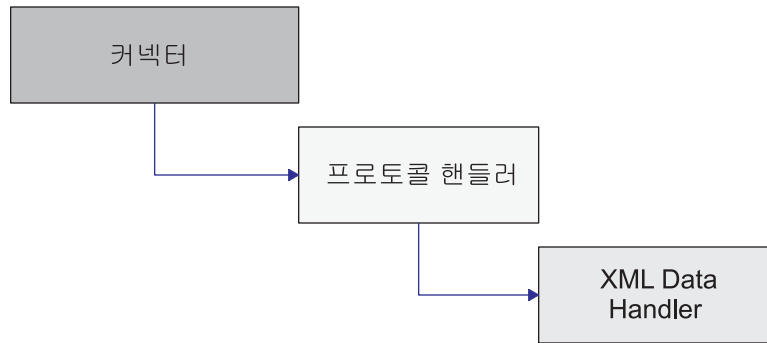


그림 1. 커넥터 구조

커넥터

커넥터는 통합 브로커와 프로토콜 핸들러 사이에 Business Object를 전달합니다. 커넥터는 다음을 수행합니다.

- 통합 브로커로부터 Business Object 요청을 수신합니다.
- 프로토콜 핸들러 프레임워크를 호출하고 Business Object의 URL 문자열을 전달하여 적절한 프로토콜 핸들러 인스턴스를 호출합니다.
- Business Object 요청을 프로토콜 핸들러에 전달합니다.
- 프로토콜 핸들러에서 Business Object 응답이나 성공/실패 리턴 코드를 수신합니다. 커넥터가 동기식 프로토콜 핸들러를 사용할 경우, Business Object 응답을 수신합니다. 커넥터가 비동기식 프로토콜 핸들러를 사용할 경우, 리턴 코드를 기초로 성공 또는 실패를 보고합니다.

커넥터에서 사용되는 1차 메소드는 `init()`, `doVerbFor()` 및 `pollForEvents()`입니다. `init()` 메소드는 통합 브로커의 저장소에서 모든 구성값을 읽고, 프록시 이름(HTTP 및 HTTPS)과 관련 포트를 설정한 후 프로토콜 핸들러(`JavaProtocolHandlerPkgs`) 및 XML Data Handler(`JavaDataHandlerPkgs`)의 Java 클래스 패키지 이름과 Data Handler 및 프로토콜 핸들러의 등록 정보 값을 읽습니다.

`doVerbFor()` 메소드는 Business Object 요청/응답 조작을 처리합니다. 커넥터가 통합 브로커에서 최상위 레벨 Business Object를 수신할 때, `doVerbFor()` 메소드는 요청 Business Object와 대상 URL을 추출합니다. 그러면 `doVerbFor()` 메소드가 적절한 프로토콜 핸들러 인스턴스를 작성합니다.

커넥터가 대상 URL에서 응답을 수신할 때, doVerbFor() 메소드는 최상위 레벨 Business Object의 하위 오브젝트로 응답 Business Object를 채우고 결과를 통합 브로커에 리턴합니다. 커넥터에서 모든 오류는 예외로 전파되어 커넥터를 통해 처리됩니다. 이때 BON_FAIL이 리턴되고 리턴 상태 설명자가 설정됩니다.

pollForEvents() 메소드는 이벤트 광고에 사용됩니다. 커넥터에는 Business Object를 사용하여 URL에서 이벤트를 확인할 수 있는 기능이 있습니다. 이벤트 광고에 대한 자세한 정보는 7 페이지의 『이벤트 광고』를 참조하십시오.

커넥터는 최상위 레벨 Data Handler Meta Object의 이름을 포함하도록 정적 등록 정보를 설정합니다. 그 이름이 DataHandlerConfigMO 커넥터 구성 등록 정보에 지정되어 있기 때문입니다.

프로토콜 핸들러(HTTP 및 HTTPS)

프로토콜 핸들러는 커넥터가 HTTP 및 HTTPS 프로토콜을 사용하여 URL을 통해 통신할 수 있도록 합니다. 프로토콜 핸들러는 Java URLConnection 클래스를 확장하는 추상 기반 클래스입니다. 이 클래스에는 HTTP 및 HTTPS와 같은 특정 프로토콜에 대한 지원을 제공하도록 구현할 수 있는 추상 메소드가 있습니다. 프로토콜 핸들러의 인스턴스는 커넥터에 의해 호출되는 프로토콜 핸들러 프레임워크에 의해 작성됩니다.

WebSphere Business Integration Adapter for XML에는 비동기 및 동기식 프로토콜 핸들러가 포함되어 있습니다. 동기식 프로토콜 핸들러는 동기식 응답에서 Business Object를 리턴합니다. 비동기식 프로토콜 핸들러는 응답 Business Object를 예상하지 않습니다. 이는 단지 POST 조작에서의 리턴 코드를 기반으로 성공 또는 실패 메시지만 리턴합니다. 비동기식 프로토콜 핸들러는 이벤트 광고를 지원하지 않습니다.

주: 프로토콜 핸들러 프레임워크를 사용하면 FTP와 같은 다른 프로토콜에 대한 지원을 추가할 수 있습니다. 프로토콜 핸들러 프레임워크는 CWURLConnection이라고 하는 추상 기반 클래스입니다.

프로토콜 핸들러 프레임워크는 프로토콜 핸들러 인스턴스를 전달하고, 커넥터는 Business Object를 작성된 인스턴스에 전달합니다. 프로토콜 핸들러는 Business Object로부터 내용 유형(text/plain 또는 text/xml)을 추출하고 이를 사용하여 XML Data Handler 인스턴스를 작성합니다.

프로토콜 핸들러가 createHandler() 메소드를 호출할 때, 핸들러는 내용 유형으로 전달됩니다. Data Handler create 메소드는 슬래시(/) 문자를 마침표(.)로 바꾸고 영숫자가 아닌 모든 문자를 밑줄(_)로 바꿔서 내용 유형을 전달합니다. 그런 다음 내용 유형에 대해 구문 분석된 문자열과 일치하는 Data Handler 최상위 레벨 Meta Object에서

속성을 찾습니다. 일치하는 사항을 찾지 못하는 경우, 메소드는 클래스 이름을 다음과 같이 빌드합니다.

```
com.crossworlds.DataHandlers1
```

프로토콜 핸들러는 다음 조작을 수행합니다.

- 커넥터에서 Business Object를 수신하고 이를 XML Data Handler에 전달합니다. 프로토콜 핸들러는 MIMEType 속성을 구문 분석하여 작성할 Data Handler 인스턴스를 판별합니다.
- XML Data Handler에서 XML 스트림을 수신한 후 적절한 URL로 전달합니다. XML 스트림은 요청 Business Object를 나타냅니다.

Data Handler가 XML 문자열을 구문 분석할 경우, 프로토콜 핸들러는 이를 URL로 전달하기 전에 XML 문자열을 XML 스트림으로 변환합니다.

- 동기식일 경우, URL에서 응답 스트림을 수신한 후 이를 다시 WebSphere Business Integration Adapter Business Object로 변환하는 XML Data Handler로 전달합니다.
- 비동기식일 경우, 요청 조작에서의 리턴 코드를 기초로 성공 또는 실패를 URL에 보고합니다.
- 응답 Business Object를 다시 커넥터에 전송합니다.

커넥터 구현이 추가 프로토콜을 지원해야 할 경우, 사용자 정의 프로토콜 핸들러를 빌드해야 합니다. 사용자 정의 프로토콜 핸들러의 작성에 대한 정보는 29 페이지의 제 4 장 『사용자 정의 프로토콜 핸들러 빌드』를 참조하십시오.

커넥터 작동 방법

다음 절에서는 커넥터가 Business Object를 처리하는 방법, Meta Object가 구성에 사용되는 방법, 그리고 커넥터가 이벤트 공고를 처리하는 방법에 대해 설명합니다.

Business Object 처리

커넥터는 URL과의 사이에 데이터를 전달하기 위해 요청/응답 조작을 사용합니다. 커넥터는 통합 브로커로부터 Business Object 요청을 수신하고 요청을 XML 스트림으로 변환합니다. 요청 스트림은 POST 메소드를 사용하여 URL에 전달되고, 유사한 내용이 있거나 없을 수 있는 응답 스트림이 리턴됩니다. 응답 스트림은 응답 Business Object로 변환된 후 원래 최상위 레벨 Business Object와 함께 통합 브로커에 리턴됩니다. Business Object 요청 유형은 Business Object 응답 유형과 다를 수 있습니다.

완전한 요청 응답 주기는 그림 2에 설명되어 있습니다.

1.

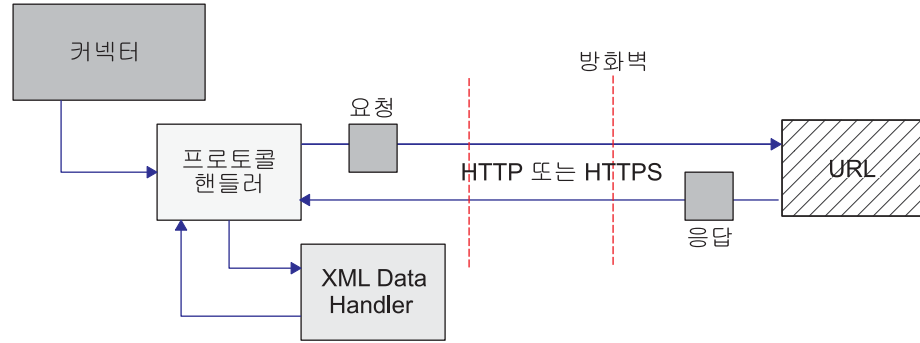


그림 2. Business Object 이벤트 처리

요청

커넥터가 통합 브로커로부터 Business Object 요청을 수신할 때, 커넥터는 적절한 프로토콜을 사용하여 전달될 수 있는 요청 스트림으로 변환해야 합니다. 프로토콜 핸들러 및 XML Data Handler는 요청 Business Object를 변환하여 URL에 전송하는 데 사용됩니다. 그림 3은 요청 프로세스를 보여줍니다.

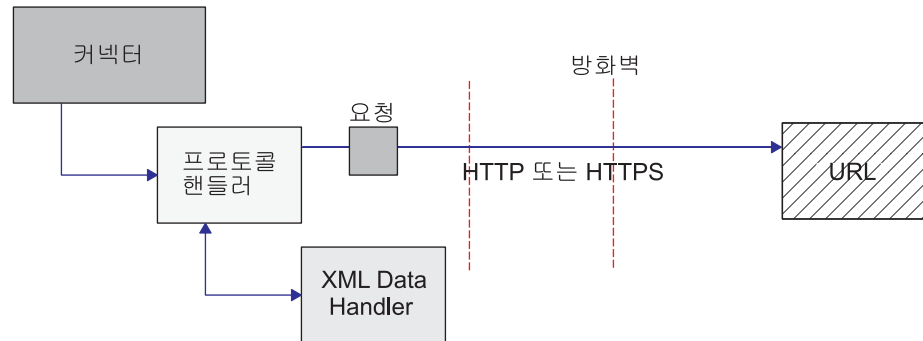


그림 3. 요청 처리

특히, 커넥터가 통합 브로커에서 최상위 레벨 Business Object를 수신하는 경우, 다음 프로세스 흐름이 발생합니다.

1. 커넥터는 getAttrValue("URL")를 호출하고 URL을 검색합니다. 또한 getAttrValue("MimeType") 및 getAttrValue("BOPrefix")를 호출하여 Business Object로부터 MimeType 및 BOPrefix 속성값을 검색합니다.
2. 커넥터는 최상위 레벨 Business Object에서 요청 Business Object를 추출합니다.
3. 커넥터는 최상위 레벨 Business Object의 URL 필드에 지정된 프로토콜과, 지정된 프로토콜 핸들러 패키지 이름을 기초로 적절한 프로토콜 핸들러(HTTP 또는 HTTPS)를 호출합니다.

4. 프로토콜 핸들러는 최상위 레벨 Business Object의 MimeType 및 BOPrefix 속성을 기초로 적절한 Data Handler를 호출합니다(최상위 레벨 Meta Object에서 구성된 대로).
5. Data Handler는 Business Object를 요청 스트림으로 변환한 후 다시 프로토콜 핸들러로 전달합니다.
6. 프로토콜 핸들러는 요청 스트림을 최상위 레벨 Business Object에 지정된 대상 URL로 전송하거나 리턴 코드를 전달합니다.

응답

동기식 프로토콜 핸들러를 사용할 경우, 응답 Business Object가 URL에서 리턴될 때 응답 스트림 양식으로 리턴됩니다. 비동기식 프로토콜 핸들러를 사용할 경우, 리턴 코드만 전달됩니다. 응답 처리는 응답 스트림을 다시 Business Object로 변환해야 하는 것을 제외하고는 요청 처리와 유사합니다.

주: 응답 스트림이 요청 스트림과 동일한 Business Object 유형에 의해 항상 표시될 수 있는 것은 아닙니다.

그림 4는 커넥터에 리턴하는 응답 Business Object의 프로세스 흐름을 보여줍니다.

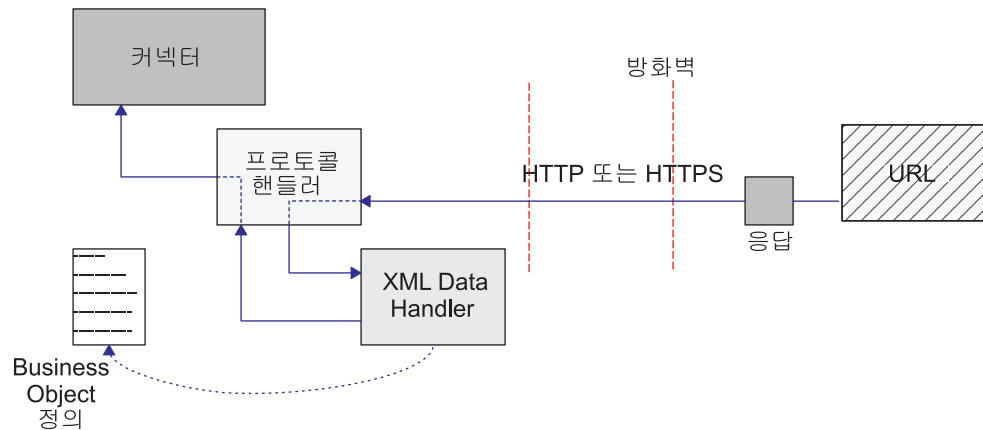


그림 4. URL에서 리턴하는 데이터 프로세스 흐름

특히, 프로토콜 핸들러가 URL로부터 응답 스트림을 수신할 때 MIME 유형이 text/xml 일 경우 다음과 같은 프로세스 흐름이 발생합니다.

1. 프로토콜 핸들러는 getContentTyp() 메소드를 호출하여 사용할 Data Handler를 판별하기 위해 MIME 유형값을 검색합니다.
2. 프로토콜 핸들러는 DataHandler 클래스를 호출하여 XML Data Handler 인스턴스를 작성합니다.

응답 스트림의 데이터 형식은 원래의 요청 Business Object에서 사용된 데이터 형식과는 다를 수 있습니다.

3. 프로토콜 핸들러는 응답 스트림을 문자열로 변환하고, 이를 XML Data Handler에 전달합니다.
4. XML Data Handler는 메시지 내용을 기초로 Business Object 이름을 확보하고 요청 스트림의 데이터(XML 문서)를 Business Object로 추출합니다.
5. XML Data Handler는 완료한 요청 Business Object를 프로토콜 핸들러에 전달합니다.
6. 프로토콜 핸들러는 요청 Business Object를 커넥터에 전달하고, 커넥터는 이를 원래의 최상위 레벨 Business Object에 추가합니다.
7. 커넥터는 요청 Business Object를 포함하는 원래의 최상위 레벨 Business Object를 다시 통합 브로커에 전달합니다.

이벤트 공고

이벤트 공고의 경우, 커넥터는 Business Object를 사용하여 URL에서 이벤트를 검색합니다. 커넥터는 응답 XML 문서로 리턴하는 요청 XML 문서를 전송하여 URL을 폴링합니다. 응답에는 커넥터가 이벤트로서 통합 브로커에 전달하는 하위 Business Object가 있습니다. 각각의 하위 Business Object는 단일 이벤트로 처리됩니다. 비동기식 프로토콜 핸들러는 이벤트 공고를 지원하지 않습니다.

주: 이벤트 처리에 대한 폴링은 응답 Business Object에서 이벤트 오브젝트를 추출하여 통합 브로커에 전송하기 위한 추가 단계가 있는 점을 제외하고 Business Object 요청 처리와 같습니다.

이벤트 공고 Business Object는 XML Business Object의 요청 및 응답 Business Object와 동일한 Business Object 처리 조작을 따릅니다. 등록하지 않은 모든 이벤트는 WebSphere Business Integration Adapter 표준 Business Object 덤프 형식의 파일로 아카이브됩니다.

이벤트 공고를 사용 가능하게 하려면, 이벤트 공고 Business Object를 정의하고 이러한 Business Object를 처리하도록 URL(예: 웹 Servlet 또는 cgi-bin 스크립트)을 설정해야 합니다. 커넥터는 POST 메소드를 사용하여 XML 이벤트 요청 문서를 스트림으로 URL에 전송합니다. URL은 STDIN에서 스트림으로 XML 문서를 읽고 하나 이상의 이벤트 오브젝트를 포함하는 XML 문서를 스트림으로 STDOUT에 기록해야 합니다.

그림 5는 이벤트 공고의 기본 프로세스를 보여줍니다.

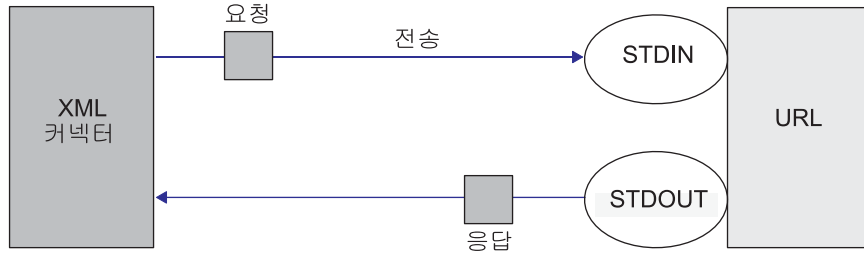


그림 5. 이벤트 광고 프로세스

Business Object 정의에 대한 자세한 정보는 21 페이지의 제 3 장 『커넥터에 대한 Business Object 개발』을 참조하십시오

로케일 종속 데이터 처리

2바이트 문자 세트를 지원하고 지정된 언어로 메시지 텍스트를 전달할 수 있도록 커넥터는 국제화되어 있습니다. 커넥터가 데이터를 하나의 문자 세트를 사용하는 위치에서 다른 문자 세트를 사용하는 위치로 전송하는 경우, 이는 문자 변환을 수행하여 데이터의 의미를 보존합니다. JVM(Java Virtual Machine) 내의 Java 런타임 환경은 Unicode 문자 코드 세트의 데이터를 표현합니다. Unicode에는 가장 잘 알려진 문자 코드 세트의 문자에 대한 인코딩이 들어 있습니다(1바이트 및 다중 바이트 모두). WebSphere Business Integration System의 대부분의 구성요소는 Java로 기록됩니다. 따라서 대부분의 통합 구성요소 사이에서 데이터가 전송되는 경우에는 문자 변환에 대한 필요성이 없습니다. 적합한 언어로, 그리고 적합한 국가나 지역에 대해 오류와 정보 메시지를 기록하려면 환경에 대해 Locale 표준 구성 등록 정보를 구성하십시오. 구성 등록 정보에 대한 자세한 정보는 35 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

제 2 장 커넥터 설치 및 구성

이 장에서는 커넥터 설치 및 구성 프로세스에 대해 설명합니다. 이 장에는 다음 절이 포함되어 있습니다.

- 『호환성』
- v 페이지의 『이 책을 읽기 전에』
- 10 페이지의 『XML 어댑터 설치』
- 10 페이지의 『커넥터 구성』
- 15 페이지의 『Data Handler에 대한 최상위 Meta Object 구성』
- 16 페이지의 『공통 구성 작업』
- 16 페이지의 『Data Handler 지정』
- 17 페이지의 『다중 커넥터 인스턴스 작성』
- 18 페이지의 『커넥터 시작』
- 20 페이지의 『커넥터 중지』

호환성

어댑터에 사용되는 어댑터 프레임워크는 어댑터가 통신하는 통합 브로커(단수 또는 복수) 버전과 호환 가능해야 합니다. Aadapter for XML 버전 3.3은 다음 어댑터 프레임워크 버전 및 통합 브로커에서 지원됩니다.

어댑터 프레임워크: WebSphere Business Integration Adapter Framework, 버전 2.1, 2.2, 2.3.x 및 2.4.

통합 브로커:

- WebSphere InterChange Server, 버전 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

예외사항은 릴리스 정보를 참조하십시오.

주: 통합 브로커 설치 지시사항 및 전제조건은 다음 문서를 참조하십시오. WebSphere ICS(InterChange Server)는 UNIX용 시스템 설치 안내서 또는 Windows용 시스템 설치 안내서를 참조하십시오.

메시지 브로커(WebSphere MQ Integrator Broker, WebSphere MQ Integrator 및 WebSphere Business Integration Message Broker)는 WebSphere 메시지 브로커로 어댑터 구현 및 메시지 브로커의 설치 문서를 참조하십시오. 이 중 일부는 웹 사이트 <http://www.ibm.com/software/integration/mqfamily/library/manualsa/>에서 찾을 수 있습니다.

WebSphere Application Server는 WebSphere Application Server로 어댑터 구현 및 <http://www.ibm.com/software/webservers/appserv/library.html>에 있는 문서를 참조하십시오.

전제조건

커넥터를 사용하려면 다음 환경을 갖추고 있어야 합니다.

- JDK 1.2 이상
- JSSE(Java Secure Socket Extension) 1.0

WebSphere Business Integration Adapter는 국제적으로 공용된 버전을 제공합니다. 국내에서 사용되는 암호화를 사용하여 다운로드한 후 이를 `\connector\xml\dependencies` 디렉토리에 추가할 수 있습니다.

- 대상 URL에 대한 액세스

XML 어댑터 설치

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

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

커넥터 구성

통합 브로커로 WebSphere MQ Integrator Broker를 사용 중이면, Connector Configurator에서 커넥터 등록 정보를 구성합니다. WebSphere ICS를 통합 브로커로 사용하는 경우, System Manager에서 액세스하는 Connector Configurator에서 커넥터 등록 정보를 구성하십시오.

Data Handler 구성

XML Data Handler에 사용되는 Meta Object를 구성하십시오. Meta Object 구성에 대한 정보는 15 페이지의 『Data Handler에 대한 최상위 Meta Object 구성』을 참조하십시오.

표준 커넥터 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이러한 등록 정보에 대한 설명은 35 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

중요: 이 커넥터가 모든 통합 브로커를 지원하므로 모든 브로커에 대한 구성 등록 정보가 이와 관련됩니다.

표 1에서는 부록에 있는 구성 등록 정보에 대해 이 커넥터에 고유한 정보를 제공합니다.

표 1. 이 커넥터에 고유한 등록 정보

등록 정보	참고
CharacterEncoding	이 커넥터는 이 등록 정보를 사용하지 않습니다.
Locale	이 커넥터가 국제화되어 있으므로 이 등록 정보의 값을 변경할 수 있습니다.

커넥터를 실행하기 전에 ApplicationName 구성 등록 정보의 값을 제공해야 합니다. 또한 커넥터를 실행하기 전에 최소한 다음의 표준 커넥터 구성 등록 정보를 설정해야 합니다.

- AgentTraceLevel
- ApplicationName
- ControllerStoreAndForwardMode
- ControllerTraceLevel
- DeliveryTransport

커넥터 특정 등록 정보

커넥터 특정 구성 등록 정보는 런타임 시 커넥터에 필요한 정보를 제공합니다. 커넥터 특정 등록 정보는 또한 커넥터를 다시 코드화하거나 다시 빌드하지 않고 커넥터에서 정적 정보 또는 논리를 변경하는 방법을 제공합니다.

표 2에는 커넥터의 커넥터 특정 구성 등록 정보가 되어 있습니다. 등록 정보 설명에 대해서는 다음에 오는 절을 참조하십시오.

표 2. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수 여부
ArchiveDirectory	아카이브 디렉토리 이름	\connectors\xml\archive	
DataHandlerConfigMO	Data Handler Meta Object 이름	MO_DataHandler_Default	예
HttpProxyHost	http 호스트 이름		
HttpProxyPort	http 프록시 포트	80	
HttpsDebug	등록 정보를 13 페이지의 표 3에 있는 값 중에서 하나로 설정하십시오.		
HttpsProxyHost	https 호스트 이름		
HttpsProxyPort	https 프록시 포트	443	
JavaProtocolHandlerPkgs	프로토콜 핸들러 이름	com.crossworlds.connectors.utils.ProtocolHandlers	예
MaxNumRetries	양수	10	
PollingBusinessObjects	Business Object 이름		
ReturnBusObjResponse	true 또는 false	true	
SecurityProvider	SSL 구현	com.sun.net.ssl.internal.ssl.Provider.sun.security.provider.Sun	예
UseCaches	true 또는 false	false	
UseDefaults	true 또는 false	false	
UseDigitalSignature	true 또는 false	false	

ArchiveDirectory

아카이브된 이벤트를 포함하는 디렉토리. 각 이벤트는 Business Object 이름과 Verb로 식별할 수 있습니다. 기본적으로, Create Verb는 Business Object 이름에 추가됩니다. 기본값은 \connectors\xml\archive입니다.

DataHandlerConfigMO

XML Connector가 Data Handler 지원을 판별하기 위해 사용하는 최상위 레벨 Meta Object의 이름. 이 Meta Object에는 XML Data Handler가 구성 등록 정보를 설정하기 위해 사용하는 하위 Meta Object 이름이 있습니다. 이 등록 정보는 또한 특정 내용 유형에 대한 인스턴스를 생성할 DataHandler 클래스를 판별하기 위해 DataHandler 기반 클래스에서 사용됩니다. 기본값은 MO_DataHandler_Default입니다. 자세한 정보는 15 페이지의 『Data Handler에 대한 최상위 Meta Object 구성』을 참조하십시오.

HttpProxyHost

HTTP의 프록시로 작동하는 호스트의 이름. 이 등록 정보는 네트워크가 HTTP 프로토콜을 사용하는 프록시 서버를 사용할 경우에만 필요합니다.

HttpProxyPort

HTTP를 연결하기 위해 사용되는 프록시 포트 번호. 이 등록 정보는 네트워크가 HTTP 프로토콜을 사용하는 프록시 서버를 사용할 경우에만 필요합니다. 기본값 포트 번호는 80입니다.

HttpsDebug

HTTPS 세션에 대해 생성되는 디버깅 정보를 판별하는 설정. 표 3에는 HTTPS 프로토콜 핸들러의 HTTPS 디버그 값이 나열되어 있습니다.

표 3. *HttpsDebug* 값

이름	의미
all	모든 디버깅을 켭니다.
data	각 데이터 교환 메시지의 16진 덤프. 데이터 교환 디버깅 범위를 넓히기 위해 사용할 수 있습니다.
handshake	각 데이터 교환 메시지를 인쇄합니다. SSL과 함께 사용할 수 있습니다.
keygen	키 생성 데이터를 인쇄합니다. SSL과 함께 사용할 수 있습니다.
plaintext	레코드 일반 텍스트의 16진 덤프. 레코드 디버깅을 확장하기 위해 사용할 수 있습니다.
record	레코드별 추적을 사용 가능하게 합니다. SSL과 함께 사용할 수 있습니다.
session	세션 활동을 인쇄합니다. SSL과 함께 사용할 수 있습니다.
ssl	SSL 디버깅만 켭니다.
verbose	상세한 데이터 교환 메시지를 인쇄합니다. 레코드 디버깅을 확장하기 위해 사용할 수 있습니다.

HttpsProxyHost

HTTPS 프록시 시스템 이름. 이 등록 정보는 네트워크가 HTTPS 프로토콜을 사용하는 프록시 서버를 사용할 경우에만 필요합니다.

HttpsProxyPort

HTTPS를 연결하기 위해 사용되는 프록시 포트 번호. 이 등록 정보는 네트워크가 HTTPS 프로토콜을 사용하는 프록시 서버를 사용할 경우에만 필요합니다.

JavaProtocolHandlerPkgs

이 속성이 존재할 경우, 이 속성은 기본 Java 핸들러 대신 프로토콜 핸들러로 사용하는 패키지를 지정합니다. 이 클래스는 Java의 프로토콜 핸들러 프레임워크를 준수해야 합니다. 예를 들어, 프로토콜 핸들러 `com.mycompany.http`(HTTP의 경우)를 사용하려면, 이 필드를 `com.mycompany`로 설정하십시오. 또한 해당되는 클래스의 `.jar` 파일이 클래스 경로에 있는지 확인하십시오.

Java 프로토콜 핸들러에 대한 자세한 정보는 다음 웹 사이트의 지침서를 참조하십시오.
<http://developer.java.sun.com/developer/onlineTraining/protocolhandlers/>

이 값의 여러 패키지를 수직선(“|”)으로 구분하여 지정할 수도 있습니다(예: `com.crossworlds.Protocol Handlers|com.mycompany`).

WebSphere Business Integration Adapter는 두 가지의 패키지를 제공합니다.

- `com.crossworlds.connectors.utils.ProtocolHandlers`(동기식 프로토콜 핸들러)

- `com.crossworlds.connectors.utils.ProtocolHandlers.async`(비동기식 프로토콜 핸들러)

기본값은 `com.crossworlds.connectors.utils.ProtocolHandlers`입니다.

MaxNumRetries

비동기식 프로토콜 핸들러가 대상 URL에서 응답을 수신하지 못한 후에 시도할 재시도 횟수를 지정합니다. 이 등록 정보는 비동기식 프로토콜 핸들러에서만 사용됩니다. 값을 지정하지 않을 경우, 이 등록 정보의 기본값은 0입니다. 기본값은 0입니다.

PollingBusinessObjects

이벤트 광고에 사용되는 Business Object. 쉽표로 여러 개의 항목을 구분하십시오(예: XMLPoll_Cust, XMLPoll_Order). 각 Business Object는 커넥터에서 지원해야 합니다. 이 등록 정보는 커넥터가 이벤트 광고에 대해 설정된 경우에 필요합니다.

ReturnBusObjResponse

커넥터가 프로토콜 핸들러에서 Business Object가 리턴될 것으로 예상하는지 여부를 판별합니다. 값을 true로 설정하면, 커넥터는 Business Object를 예상합니다. 값을 false로 설정할 경우, 커넥터는 Business Object를 예상하지 않습니다. 단지 성공 또는 실패 응답만 예상합니다. 기본값은 true입니다.

주: 비동기식 프로토콜 핸들러를 구성할 경우, 비동기식 프로토콜 핸들러가 Business Object 응답을 예상하지 않으므로 값을 false로 설정해야 합니다.

SecurityProvider

SSL 데이터 교환 중 HTTPS에서 사용됩니다. 쉽표로 구분되는 이 속성의 값들은 HTTPS URL에 연결할 때 사용할 SSL 구현을 판별합니다. 값을 설정하지 않으면, HTTPS 연결이 작동하지 않습니다. 기본값은 `com.sun.net.ssl.internal.ssl.Provider.sun.security.provider.Sun`입니다.

UseCaches

이 속성을 false로 설정하면, 커넥터는 XML 문서의 캐시되지 않은 버전을 검색하려고 합니다. 이는 요청에 불과하므로, 커넥터가 확실하게 시행할 수는 없습니다. 캐시된 XML 문서만 검색하려면 이 값을 true로 설정하십시오.

UseDefaults

작성 조작에서 UseDefaults를 true로 설정하면, 커넥터는 isRequired Business Object 속성마다 올바른 값이나 기본값이 제공되는지 확인합니다. 값이 제공되면 작성 조작이 성공하고, 그렇지 않으면 실패합니다. 매개변수를 false로 설정하면, 커넥터는 올바른 값에 대해서만 확인하므로, 올바른 값을 제공하지 않으면 작성 조작은 실패합니다. 기본값은 false입니다.

UseDigitalSignature

디지털 서명 길이(2진 0)가 HTTP 또는 HTTPS 프로토콜을 사용하여 전송되는 메시지 끝에 추가되는지 여부를 지정합니다. 커넥터가 디지털 서명을 지원할 경우, 이 등록 정보를 true로 설정하십시오. 기본값은 false입니다.

주: 제품과 함께 제공되는 커넥터는 디지털 서명을 지원하지 않습니다.

Data Handler에 대한 최상위 Meta Object 구성

Meta Object는 구성 정보를 포함하는 Business Object입니다. Data Handler Meta Object에는 Data Handler를 구성하기 위한 정보가 있습니다. 커넥터는 XML Data Handler 인스턴스를 작성하기 위해 Data Handler Meta Object의 정보를 사용합니다.

커넥터를 실행하기 전에, 커넥터가 MIME 유형을 기초로 사용할 Data Handler를 지정하는 Data Handler Meta Object를 설정해야 합니다. 커넥터가 Business Object 요청을 수신할 때, 적절한 Data Handler의 인스턴스를 동적으로 작성하기 위해 Meta Object에서 정보를 사용합니다.

커넥터는 커넥터 구성 등록 정보 DataHandlerConfigMO에서 최상위 레벨 Data Handler Meta Object의 이름을 확보합니다. 최상위 레벨 Meta Object는 임의 개수의 하위 오브젝트를 포함할 수 있는 계층 구조의 Business Object입니다. 각 하위 오브젝트는 특정 Data Handler 인스턴스를 표시하는 플랫폼 오브젝트입니다. 하위 Meta Object에는 Data Handler 인스턴스가 작업을 수행할 수 있도록 구성값을 제공하는 속성이 있습니다. Data Handler 유형마다 다른 구성 등록 정보가 필요하므로, 특정 핸들러를 지원하는 하위 Meta Object는 다른 속성을 가집니다.

XML Connector는 XML Data Handler를 사용하여 Business Object와 XML 문서 사이에 변환합니다. 커넥터에 맞는 XML Data Handler를 구성하려면 다음을 수행하십시오.

- 최상위 레벨 Data Handler Meta Object를 설정하여 커넥터가 지원하는 각 MIME 유형의 속성을 확보하십시오. 속성 이름은 MIME 유형의 이름이 됩니다. 속성은 Data Handler 인스턴스에 대해 하위 Meta Object를 표시합니다.

XML Connector의 경우, 최상위 레벨 Meta Object에 text/xml MIME 유형의 속성이 있는지 확인하십시오. 이 속성은 또한 XML Data Handler에 대한 하위 Meta Object 이름도 나열해야 합니다.

- 각 하위 Meta Object의 기본 속성 값을 설정하십시오. WebSphere Business Integration Adapter Data Handler에 대한 구성 등록 정보는 *Data Handler 안내서*에서 설명합니다.

XML Data Handler에 대한 하위 Meta Object에서 적절한 기본 속성 값을 설정하십시오.

개별 Data Handler에 대한 자세한 Meta Object 설정 정보는 *Data Handler 안내서*를 참조하십시오.

주: 커넥터가 Data Handler 인스턴스를 생성하도록 하려면, Data Handler 최상위 레벨 Meta Object가 커넥터에서 지원하는 오브젝트 목록의 일부여야 합니다.

공통 구성 작업

이 장에서는 커넥터에 대해 공통되는 구성 작업에 대해 설명합니다.

이벤트 공고 설정

커넥터에 대해 이벤트 공고를 사용 가능하도록 하려면 다음 단계를 수행하십시오.

1. 하위 요청 및 응답 Business Object를 포함하는 최상위 레벨 Business Object를 작성하십시오.
2. 요청 및 응답 Business Object의 구조를 처리하려면 URL을 구성하십시오. Business Object 정의에 대한 자세한 정보는 21 페이지의 제 3 장 『커넥터에 대한 Business Object 개발』을 참조하십시오.
3. 이벤트 공고 Business Object를 정의하고 나면, WebSphere MQ Integrator Broker가 통합 브로커인 경우에는 Connector Configurator를 사용하고, InterChange Server가 통합 브로커인 경우는 System Manager에서 액세스하는 Connector Designer를 사용하여 PollingBusinessObjects 및 ArchiveDirectory 구성 등록 정보를 설정하십시오.

Data Handler 지정

XML Connector에서 사용할 Data Handler를 지정하려면 다음 단계를 사용하십시오.

1. 커넥터가 지원할 데이터 형식의 유형을 판별하십시오.

기본적으로, 커넥터는 text.xml MIME 유형에 대해 XML Data Handler를 사용합니다. Business Object와 일부 다른 MIME 유형 사이에 변환할 경우, MIME 유형이 최상위 레벨 Data Handler Meta Object에 있는 속성 (MO_DataHandler_Default)인지 확인하십시오. 제공된 형식 유형을 변환하기 위해 단 하나의 Data Handler만 사용할 수 있습니다.

2. 커넥터가 사용할 Data Handler를 판별하십시오.

최상위 레벨 Data Handler Meta Object는 MIME 유형을 하위 Data Handler Meta Object와 연관짓습니다. 하위 Data Handler Meta Object는 인스턴스가 생성되는 Data Handler를 판별합니다.

3. Data Handler Meta Object를 수정하려면 Business Object Designer를 사용하십시오.

주: 통합 브로커가 Interchange Server인 경우, System Manager 내에서 Business Object Designer를 실행할 수 있습니다.

4. Connector Configurator 또는 System Manager Data Handler 최상위 레벨 Meta Object를 커넥터에 대해 지원되는 오브젝트 목록에 추가하십시오. 커넥터가 최상위 레벨 Data Handler Meta Object에 등록하지 않은 경우, 커넥터는 시작할 때 Meta Object를 로드하지 않습니다.
5. 커넥터 DataHandlerConfigMO 구성 등록 정보에서 최상위 레벨 Data Handler Meta Object의 이름을 지정하십시오. 제품과 함께 제공되는 기본값은 MO_DataHandler_Default Meta Object입니다.

Data Handler Meta Object에 대한 자세한 정보는 *Data Handler* 안내서를 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\repository`의 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 17 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 4에 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.

표 4. 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 Message Broker(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 message broker 또는 WebSphere Application Server에서는 -c 옵션 다음에 커넥터 구성 파일의 이름이 포함되어야 합니다. ICS의 경우 -c는 선택적입니다.

- System Manager를 시작할 때 실행되는 Adaptor Monitor(WebSphere Business Integration Adapters 제품 전용)에서 다음을 수행할 수 있습니다.

이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.

- System Monitor(WebSphere InterChange Server 제품 전용)에서 다음을 수행할 수 있습니다.

이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.

- Windows 시스템에서 Windows 서비스로 시작하도록 커넥터를 구성할 수 있습니다. 이 경우, Windows 시스템이 시동하거나(자동 서비스인 경우) Windows 서비스 창을 통해 서비스를 시작할 때(수동 서비스인 경우) 커넥터가 시작됩니다.

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

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

커넥터 중지

커넥터를 중지하는 방법은 다음과 같이 커넥터를 시작한 방법에 따라 다릅니다.

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

```
connector_manager_connName -stop
```

여기서 *connName*은 커넥터의 이름입니다.

- System Manager를 시작할 때 실행되는 Adapt Monitor(WebSphere Business Integration Adapters 제품 전용)에서 다음을 수행할 수 있습니다.

이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.

- System Monitor(WebSphere InterChange Server 제품 전용)에서 다음을 수행할 수 있습니다.

이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.

- Windows 시스템에서 Windows 서비스로 시작하도록 커넥터를 구성할 수 있습니다. 이 경우, Windows 시스템이 종료되면 커넥터가 중지됩니다.

제 3 장 커넥터에 대한 Business Object 개발

이 장에서는 커넥터에 의해 전달되는 최상위 레벨 Business Object 구조에 대해 설명합니다. 또한 필수 속성에 대해 설명하고 최상위 레벨 Business Object를 커넥터가 처리하는 방법에 대해 설명합니다. 이 장에는 다음 절이 포함되어 있습니다.

- 『커넥터 구현 계획』
- 22 페이지의 『커넥터 Business Object 구조』
- 26 페이지의 『이벤트 공고에 대한 Business Object』
- 27 페이지의 『XML DTD 또는 스키마 문서 기반 Business Object』

커넥터 구현 계획

커넥터는 모듈 형태로 설계되어 있으므로, 커넥터 무결성을 조정하지 않고도 구성요소를 바꾸거나 추가할 수 있습니다. 커넥터 및 해당 구성요소의 구성을 시작하기 전에, 개발해야 할 시스템을 먼저 분석하십시오.

다음 정보를 사용하여 커넥터 구성요소가 제품과 함께 제공된 대로 변경하지 않고 사용할 수 있는지 판별하십시오. 커넥터 구성요소의 기능이 사용자 요구에 맞지 않으면, 이를 사용자 정의 구성요소로 바꿀 수 있습니다. 예를 들어, 응용프로그램이 XML 이외의 다른 데이터 유형을 예상할 경우 사용자 정의 Data Handler를 구현해야 할 필요가 있습니다.

표 5를 참조하여 커넥터 구성요소가 제공된 대로 사용할 수 있는지, 또는 사용자 정의 구성요소를 작성해야 하는지 판별하십시오.

표 5. WebSphere Business Integration Adapter 구성요소 사용 또는 사용자 정의 구성요소 빌드 여부

WebSphere Business Integration

Adapter 제공 구성요소 사용	조건이 모두 해당되는 경우	해당되지 않는 경우 다음 빌드
동기식 프로토콜 핸들러(HTTP/HTTPS)	<ul style="list-style-type: none">• HTTP 또는 HTTPS 프로토콜 사용• 사용자/암호 교환이 필요하지 않음• 인증 세부사항이 필요하지 않음• URL로부터의 응답 Business Object가 필요함	사용자 정의 프로토콜 핸들러 빌드. 29 페이지의 제 4 장 『사용자 정의 프로토콜 핸들러 빌드』 참조.
비동기식 프로토콜 핸들러(HTTP/HTTPS)	<ul style="list-style-type: none">• HTTP 또는 HTTPS 프로토콜 사용• 사용자/암호 교환이 필요하지 않음• 인증 세부사항이 필요하지 않음• URL로부터의 성공 또는 실패 리턴 코드만 필요함(응답 Business Object가 필요하지 않음)	사용자 정의 프로토콜 핸들러 빌드. 29 페이지의 제 4 장 『사용자 정의 프로토콜 핸들러 빌드』 참조.

WebSphere Business Integration

Adapter 제공 구성요소 사용	조건이 모두 해당되는 경우	해당되지 않는 경우 다음 빌드
XML Data Handler	<ul style="list-style-type: none"> 데이터 형식은 XML 1.0 Data Handler 안내서 참조 	사용자 정의 Data Handler 빌드. Data Handler 안내서 참조
Name Resolver(XML Data Handler)	Business Object 이름은 XML 문서의 루트 요소 이름 및 XML Data Handler(구성 가능)의 하위 Meta Object에 있는 BOPrefix 속성을 사용하여 결정됩니다.	사용자 정의 Name Resolver 빌드. Data Handler 안내서 참조
Name Resolver(XML Data Handler)	<ul style="list-style-type: none"> 외부 엔터티 무시 외부 엔터티에 대해 로컬 파일 시스템 검색 	사용자 정의 Entity Resolver 빌드. Data Handler 안내서 참조
SAX Parser(XML Data Handler)	<ul style="list-style-type: none"> 데이터 형식은 XML 	사용자 정의 구문 분석기 빌드

커넥터 Business Object 처리

커넥터는 통합 브로커와 프로토콜 핸들러 사이에 Business Object를 전달합니다. 요청 Business Object를 프로토콜 핸들러에 전송하고 프로토콜 핸들러로부터 응답 Business Object를 수신합니다. 그러나 Business Object의 어떤 데이터도 처리하지 않습니다.

통합 브로커가 Business Object를 커넥터에 전달할 때, 커넥터는 다음 조작을 수행합니다.

1. 최상위 레벨 Business Object에서 요청 Business Object를 추출합니다. 커넥터는 요청 Business Object가 CxIgnore 또는 CxBlank 값을 가지고 있지 않은 첫 번째 하위 Business Object로 예상합니다.
2. 요청 Business Object를 프로토콜 핸들러에 전송합니다.
3. 프로토콜 핸들러가 응답 Business Object를 리턴할 때, 커넥터는 응답 Business Object를 최상위 레벨 Business Object에 추가하고 완전한 최상위 레벨 Business Object를 통합 브로커에 리턴합니다.

커넥터 Business Object 구조

커넥터에는 계층 구조 Business Object가 필요합니다. 최상위 레벨 Business Object에는 값이 대상 URL 문자열, 데이터의 MIME 유형, Business Object 접두부, 그리고 요청 및 응답 Business Object인 속성이 있습니다.

그림 6은 IBM WebSphere Business Integration Adapter for XML에 대한 최상위 레벨 Business Object의 필수 기본 구조를 보여줍니다.

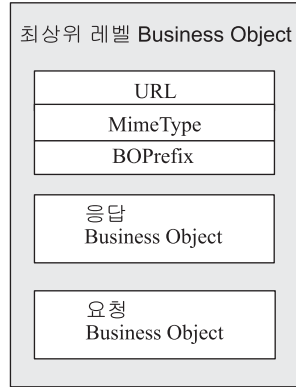


그림 6. 최상위 레벨 Business Object의 기본 구조

예를 들어, 두 개의 Business Object인 XMLApp_CustCreateRequest와 XMLApp_CustCreateResponse를 작성할 경우, 커넥터에 대한 최상위 레벨 Business Object 정의는 다음과 같을 수 있습니다.

```
XMLApp_CustCreate
    URL      String
    MimeType String
    BOPrefix String
    Response XMLApp_CustCreateResponse
    Request  XMLApp_CustCreateRequest
```

요청 및 응답 Business Object를 작성하려면 Business Object Designer를 사용하십시오. 최상위 레벨 Business Object 정의를 작성하고 필수 속성과 요청 및 응답 Business Object를 추가하십시오. 그런 다음 최상위 레벨 Business Object를 지원하도록 커넥터를 구성하십시오.

최상위 레벨 Business Object에 대한 필수 속성

최상위 레벨 Business Object는 URL 문자열, MIME 유형, BOPrefix, 요청 Business Object 및 응답 Business Object에 대한 최소한 하나의 속성을 가지고 있어야 합니다. 각 속성은 IsRequired = True로 표시해야 합니다.

표 6에서는 최상위 레벨 Business Object의 필수 속성에 대해 설명합니다. 자세한 정보는 뒤에 나오는 절들을 참조하십시오.

표 6. 최상위 레벨 XML Business Object에 대한 필수 속성

속성	유형	설명
URL	String	대상 URL.
MimeType	String	트랜잭션에 사용할 MIME 유형.
BOPrefix	String	XML Data Handler의 인스턴스를 작성하기 위해 MIME 유형과 함께 사용됩니다.
응답	Business Object	응답 메시지를 나타내는 Business Object. 25 페이지의 『요청 및 응답 Business Object』를 참조하십시오.
요청	Business Object	요청 메시지를 나타내는 Business Object. 최상위 레벨 Business Object에서 응답 Business Object에 대한 속성 다음에 이 속성을 놓으십시오.

주: 커넥터는 최소한 하나의 속성을 키 속성으로 설정해야 합니다. 그러나 이 커넥터는 어떠한 속성도 키로 설정할 필요가 없습니다.

URL

URL 문자열은 Business Object에서 데이터의 목적지와, 데이터를 전달하기 위해 사용할 프로토콜을 정의합니다. 문자열에는 프로토콜(예: HTTP 또는 HTTPS)을 포함하여 전체 목적지가 포함되므로, 프로토콜을 지정하는 별도의 속성은 필요하지 않습니다.

URL 문자열은 대상 URL에 대한 연결을 열기 위해 커넥터에 의해 사용됩니다. 연결이 열리면, 커넥터는 URL 문자열을 사용하여 적절한 프로토콜 핸들러 인스턴스를 작성합니다.

예를 들어, 문자열 `http://www.ibm.com`은 HTTP 프로토콜이 사용되고 HTTP 프로토콜 핸들러 인스턴스가 작성되도록 지정합니다.

MIME 유형

MIME 유형은 URL에 전달하는 데이터의 형식과 내용 유형을 정의합니다. 커넥터는 MIME 유형을 사용하여 적절한 Data Handler를 호출합니다. Meta Object는 MIME 유형/BOPrefix 조합에 대한 Data Handler 인스턴스를 식별합니다. 이 Data Handler 구현이 단지 하나의 MIME 유형만 처리하는 경우, 하위 Meta Object의 BOPrefix 속성은 선택적입니다. 최상위 레벨 Business Object에서는 이 속성이 필수입니다.

커넥터는 MIME 유형의 기본값을 `text/xml`로 예상하지만, 다른 MIME 유형을 사용하도록 커넥터를 구성할 수 있습니다.

BOPrefix

커넥터는 MimeType 속성의 BOPrefix를 사용하여 적합한 Data Handler 인스턴스를 호출합니다. 이 속성은 Business Object 이름의 고유함을 보증하기 위해 필요합니다. 예를 들어, AppA_PO 및 AppB_PO 응용프로그램에 대한 두 개의 구매 주문 Business Object를 가질 수 있습니다.

주: 최상위 레벨 Business Object의 BOPrefix 속성은 XML Data Handler에 대한 하위 Meta Object에 있는 BOPrefix 속성과 다릅니다. XML Data Handler 하위 Meta Object에 대한 자세한 정보는 *Data Handler* 안내서를 참조하십시오.

URL에서 XML 스트림이 리턴되면, XML Data Handler가 XML 스트림의 루트 요소 이름을 Business Object 정의 BOPrefix_name에 맵핑합니다. 루트 요소 이름의 값은 항상 BOPrefix 값 다음에 표시됩니다.

예를 들어 XML 문서의 루트 요소가 <Customer>이고 BOPrefix=AppA인 경우, BOPrefix_Name은 AppA_Customer입니다.

요청 및 응답 Business Object

요청 및 응답 Business Object에는 대상 URL에 전달할 실제 데이터가 있습니다. 커넥터가 최상위 레벨 Business Object를 수신할 때, 요청 Business Object만 채워집니다. 응답 Business Object는 대상 URL에서 리턴되는 데이터로 채워집니다.

최상위 레벨 Business Object에서 요청 및 응답 Business Object를 정의할 경우 다음 지침에 유의하십시오.

- 다음이 해당될 경우 요청 Business Object 이전에 응답 Business Object를 위치시키십시오.
 - 요청 및 응답 Business Object가 같은 유형의 Business Object입니다.
 - Business Object가 협업 요청에 사용됩니다(WebSphere ICS가 통합 브로커인 경우에만 적용).
 - 요청 Business Object의 데이터는 보존해야 합니다(URL로부터의 응답으로 겹쳐 쓰지 않도록).
- 최상위 레벨 Business Object에서 응답 Business Object의 속성값을 CxIgnore 또는 CxBlank로 설정하십시오. 커넥터는 첫 번째의 널이 아닌 속성값을 프로토콜 핸들러에 전달합니다.
- 요청을 나타내는 Business Object가 응답을 나타내는 Business Object와 같은 경우, 요청 및 응답 속성의 유형은 같아야 합니다.
- 요청 및 응답 Business Object는 다를 수 있습니다. 예를 들어, 사용자 정의 구매 주문 Business Object를 전송하고 주문 상태 Business Object를 수신할 수 있습니다.
- 최상위 레벨 Business Object에 리턴할 각각의 응답 XML 문서를 지원하도록 여러 응답 Business Object를 정의할 수 있습니다. 여러 개의 응답 Business Object는 커넥터가 다른 유형의 XML 문서(다른 Business Object 유형에 해당하는)가 웹 서버에서 리턴될 수 있는 가능성을 처리할 수 있게 합니다.

Data Handler 요구사항에 부합하는 Business Object

커넥터에 대한 최상위 레벨 래퍼 Business Object에 WebSphere Business Integration Adapter Business Object를 포함할 수는 있지만, 포함된 Business Object는 데이터의 변환을 위해 사용된 Data Handler의 요구사항에 부합하는 양식으로 데이터를 전달해야 합니다.

예를 들어, BySize Data Handler의 경우 Business Object 정의는 각 Business Object 속성에 대한 MaxLength 속성 등록 정보 값을 지정해야 합니다. XML Data Handler의 경우, Business Object 정의에는 Data Handler가 XML 문서를 생성할 수 있도록 하는 응용프로그램 특정 정보가 있어야 합니다.

그러므로 좋은 실습은 처리할 유형의 데이터마다 사용자 고유의 Business Object 정의를 작성하는 것입니다. Business Object 정의에서 응용프로그램에 필요한 데이터와 Data Handler에 필요한 정보만 제공하십시오. 그러면 최상위 레벨 커넥터 Business Object에 이러한 Business Object를 포함할 수 있습니다.

각 Data Handler에 고유한 정보는 *Data Handler* 안내서를 참조하십시오.

이벤트 공고에 대한 Business Object

이벤트 공고 Business Object와 요청 Business Object가 둘 다 MIME 유형, BOPrefix, 응답 Business Object, 요청 Business Object 및 URL에 대한 속성을 요구한다는 점에서 이벤트 공고 Business Object의 구조는 요청 Business Object의 구조와 유사합니다. Business Object 처리에서 유일한 차이점은 커넥터가 응답 Business Object의 내용을 처리하는 방법에 있습니다. 이벤트 공고의 경우, 커넥터는 응답 Business Object가 이벤트를 나타내는 하위 Business Object를 포함할 것으로 예상합니다.

이벤트 공고 Business Object를 정의할 때, 다음에 유의하십시오.

- 최상위 레벨 Business Object에는 요청과 응답 속성이 모두 필요합니다. 둘 모두는 필수이어야 하며 유형이 달라야 합니다.
- 응답 Business Object 이전에 요청 Business Object를 놓으십시오.
- 응답 Business Object는 같은 유형의 여러 하위 Business Object를 리턴할 수 있습니다. 예를 들어, 고객 이벤트만 리턴하도록 응답 Business Object를 설계할 수 있습니다.
- 응답 Business Object는 다른 유형의 여러 하위 Business Object를 리턴할 수 있습니다. 예를 들어, 주문과 고객 이벤트를 리턴하도록 응답 Business Object를 설계할 수 있습니다.
- 등록되지 않은 모든 하위 Business Object는 아카이브 디렉토리에 아카이브됩니다.
- Business Object에서는 Business Object 정의의 응용프로그램 특정 정보 열에 있는 기본 Verb와 함께 지원되는 Verb 열에 'DefaultVerbName' Verb를 추가해야

합니다. 기본 Verb는 등록을 적절하게 확인할 수 있도록 이벤트 공고에 사용됩니다. 통합 브로커에 전송하는 Business Object마다 Verb를 설정해야 합니다.

그림 7은 Business Object 정의에 'DefaultVerbName' 넣기를 보여줍니다.

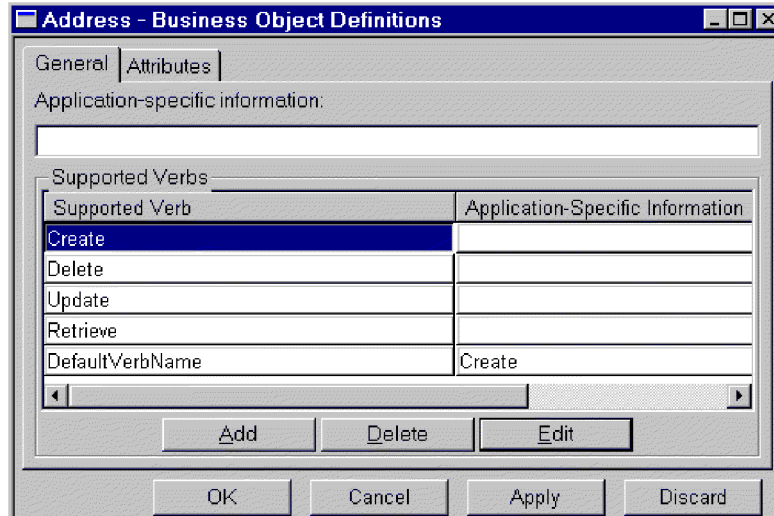


그림 7. Business Object 정의에 'DefaultVerbName' 넣기

XML DTD 또는 스키마 문서 기반 Business Object

XML DTD 또는 스키마 문서를 기반으로 요청 및 응답 Business Object를 작성할 경우, 처리할 각 유형의 XML 문서의 Business Object 정의를 작성해야 합니다. Business Object 정의는 XML 문서의 DTD 또는 스키마 문서에 들어 있는 구조 정보를 포함합니다. 예를 들어, 하나의 요청 스트림(단일 DTD 또는 스키마 문서)이 있지만 네 개의 가능한 응답 스트림 유형(네개의 개별 DTD 또는 스키마 문서)이 있을 경우, 5개의 Business Object 정의를 정의해야 합니다. 반면에, 요청 및 응답 스트림에서 같은 스키마를 사용하면, 하나의 Business Object 정의만 필요합니다. XML ODA(Object Discovery Agent)를 사용하여 DTD나 스키마 문서를 기반으로 Business Object 정의를 생성할 수 있습니다.

주: XML ODA가 DTD 또는 스키마를 읽으면 FIXED 속성을 무시합니다. 이 속성값은 xml 인스턴스에서 선택적이고 값이 언제나 고정되기 때문입니다. 이러한 FIXED 값이 BO에서 작성되고 읽은 xml 인스턴스에 있도록 하려면, FIXED 속성을 BO 속성으로 수동으로 추가해야 합니다. 런타임 시 이 값이 변경되지 않는지 확인해야 합니다.

XML ODA를 사용하거나 수동으로 XML 문서의 Business Object 정의를 정의하는 방법에 대한 정보는 *Data Handler* 안내서를 참조하십시오.

제 4 장 사용자 정의 프로토콜 핸들러 빌드

이 장에서는 프로토콜 핸들러 프레임워크와 이를 사용하여 사용자 정의 프로토콜 핸들러를 빌드하는 방법에 대해 설명합니다. 이 장에는 다음 절이 포함되어 있습니다.

- 『프로토콜 핸들러 프레임워크』
- 31 페이지의 『프로토콜 핸들러 클래스 작성』
- 31 페이지의 『프로토콜 핸들러 프레임워크 메소드』
- 33 페이지의 『사용자 정의 프로토콜 핸들러 샘플 코드』

프로토콜 핸들러 프레임워크

WebSphere Business Integration Adapter Protocol Handler Framework는 개발자가 여러 유형의 프로토콜에 대해 프로토콜 핸들러를 일정한 방식으로 작성할 수 있게 합니다. 프로토콜 핸들러 프레임워크에는 CWURLConnection이라는 클래스가 있고, 이 클래스에는 사용자 정의 프로토콜 핸들러를 작성하기 위해 구현해야 하는 추상 메소드가 있습니다. 프레임워크는 `com.crossworlds.protocolhandler` 패키지의 일부입니다.

프로토콜 핸들러 프레임워크 클래스

모든 사용자 정의 프로토콜 핸들러는 최소한 두 개의 클래스를 가지고 있어야 합니다.

- Handler
- `cw_protocolconnection`(HTTP 프로토콜의 경우 `cw_httpconnection`)

`connection` 클래스는 CWURLConnection 클래스를 확장합니다.

그림 8은 `com.crossworlds.connectors.utils.protocolhandler` 기본 클래스의 계층 구조를 보여줍니다.

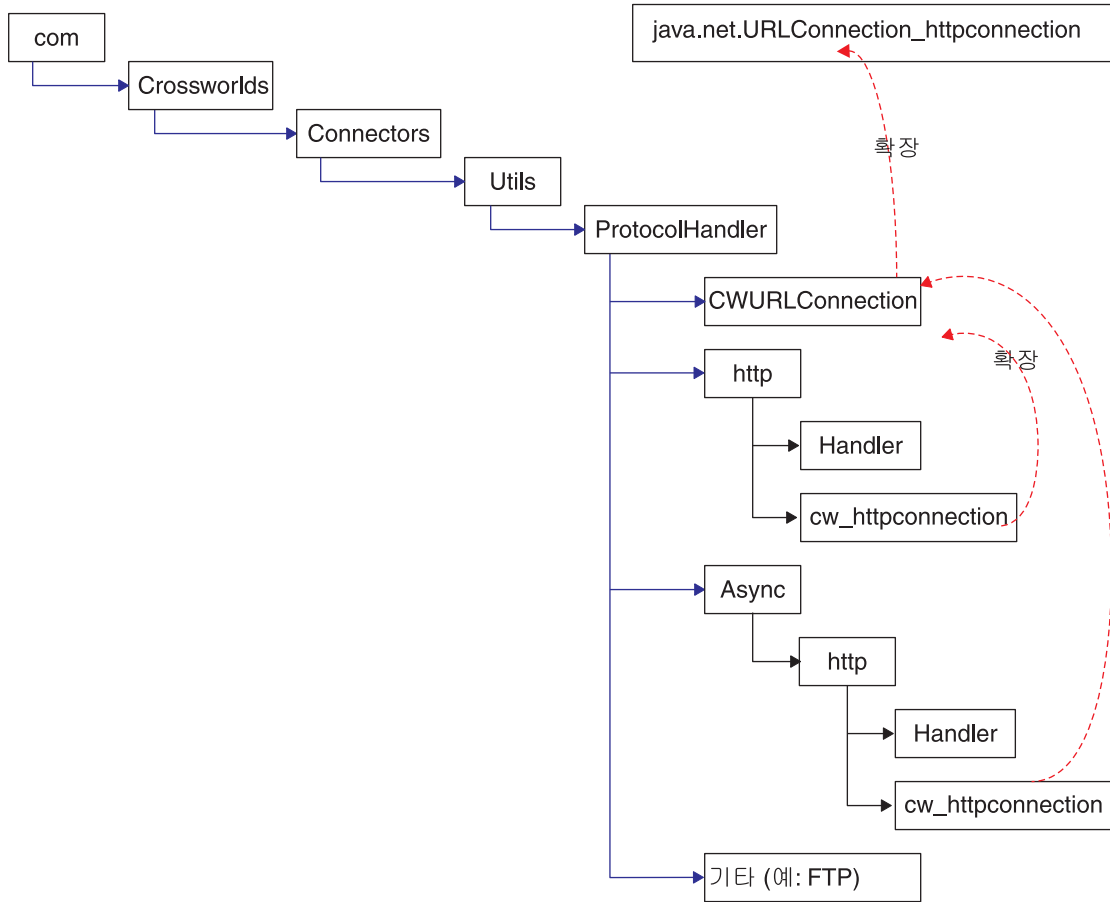


그림 8. 프로토콜 핸들러 클래스 계층 구조

프로토콜 핸들러 프레임워크를 사용하여 사용자 정의 프로토콜 핸들러를 개발하려면 다음을 수행하십시오.

- ProtocolNameConnection 클래스를 작성하십시오. 여기서 ProtocolName은 프로토콜의 이름입니다.
- connection 클래스에서 최소한 하나의 getContent() 메소드 구현을 제공하십시오.
- 핸들러 클래스를 작성하십시오.

핸들러 클래스 요약

```
Public URLConnection openConnection(URL url); throws IOException
```

연결 클래스 요약

```
public String getContent (object input, String mimeType,
String BOPrefix) throws IOException
public String getContentType()
public synchronized void connect() throws IOException
```

프로토콜 핸들러 클래스 작성

커넥터를 설치할 때, 프로토콜 핸들러에 대한 스텝 코드와 makefile이 설치됩니다. 스텝 파일에는 구현해야 하는 모든 메소드를 나열하며 빈 클래스를 정의하는 Java 코드가 있습니다. 사용자 정의 프로토콜 핸들러를 생성하기 위한 템플릿으로 스텝 파일을 사용할 수 있습니다.

새 프로토콜 핸들러를 구현하려면 다음을 수행하십시오.

1. stubProtocolHandler.java 파일을 수정(및 이름 바꾸기)하십시오.
2. 소스 파일의 이름을 포함하도록 makefile을 편집하십시오.
3. makeProtocolHandler.bat 파일을 실행하여 클래스를 컴파일하십시오. makefile은 클래스만 컴파일합니다. 클래스를 .jar 파일에 추가하지 않습니다.
4. 새 클래스를 .jar 파일에 추가하십시오. 다음 명령을 사용하십시오.

```
jar cvf MyProtocolHandler.jar <classes>.
```

여기서

- MyProtocolHandler.jar은 프로토콜 핸들러 .jar 파일입니다. 이 파일은 커넥터 시작 일괄처리 파일인 start_xml.bat(UNIX의 경우 start_xml.sh)가 상주하는 클래스 경로에 있어야 합니다.
 - <classes>는 프로토콜 핸들러의 모든 클래스입니다. 모든 클래스를 나열하고 각 항목을 공백으로 구분하십시오.
5. 커넥터가 새 클래스를 선정할 수 있는지 확인하십시오. 새 .jar 파일이 클래스 경로에 포함되도록 start_xml.bat(UNIX의 경우 start_xml.sh)를 편집하십시오.

프로토콜 핸들러 프레임워크 메소드

다음 절에서는 새 프로토콜 핸들러를 설계하거나 기존의 프로토콜 핸들러를 수정할 때 사용하는 프로토콜 핸들러 프레임워크 메소드에 대해 설명합니다.

getContent ()

getContent() 메소드는 Business Object 처리에 사용됩니다. 다음을 수행합니다.

- MimeType 및 BOPrefix Business Object 속성을 사용하여 작성할 적절한 Data Handler 인스턴스를 판별합니다.
- Business Object를 변환에 적절한 Data Handler로 전송한 후 파일을 URL로 전송합니다.
- 대상 URL로부터 응답 스트림을 수신하고 Data Handler 인스턴스를 호출하여 스트림을 WebSphere Business Integration Adapter Business Object로 변환합니다.
- 원래의 호출자(예: 커넥터)에게 Business Object를 다시 리턴합니다.

구문

```
public abstract Object getContent (Object input, String mimeType,  
String BOprefix) throws IOException
```

매개변수

input Business Object 인터페이스(전송할 Business Object)를 지정합니다.
mimeType Data Handler에 전달하는 데이터의 MIME 유형을 지정합니다.
BOprefix Data Handler에 전달하는 데이터의 BOPrefix 유형을 지정합니다.

리턴값

Business Object 인터페이스를 리턴합니다.

WebSphere Business Integration Adapter에서 제공하는 프로토콜 핸들러 호출

다음의 코드 예는 WebSphere Business Integration Adapter에서 제공하는 프로토콜 핸들러를 호출하는 방법을 보여줍니다.

```
try  
{  
// set the system property, so that Java knows where to look for  
// the protocol handlers. You only need to do it once.  
Properties prop = System.getProperties();  
prop.put("java.protocol.handler.pkgs",  
"com.crossworlds.connectors.utils.ProtocolHandlers");  
  
URL url = new URL("http://www.crossworlds.com");  
CWURLConnection uc = (CWURLConnection) url.openConnection();  
BusinessObjectInterface respBO = (BusinessObjectInterface)  
uc.getContent (input, mime, prefix);  
  
}  
catch (Exception XX)  
{  
//flag error  
}
```

사용자 정의 프로토콜 핸들러 샘플 코드

다음의 샘플 코드는 사용자 정의 프로토콜 핸들러를 개발할 때 지침으로 사용할 수 있습니다.

```
/**
 * This package hierarchy is used to write the Protocol Handler.
 * [ProtocolName] should be substituted with the name of the protocol
 * for which the handler is being written.
 * For example com.crossworlds.connectors.utils.ProtocolHandlers.ftp
 * or com.crossworlds.connectors.utils.ProtocolHandlers.http
 */
package com.crossworlds.connectors.utils.ProtocolHandlers.[ProtocolName];

import CxCommon.BusinessObjectInterface;
import com.crossworlds.connectors.utils.ProtocolHandlers.CWURLConnection;
import com.crossworlds.DataHandlers.DataHandler;

import AppSide_Connector.JavaConnectorUtil;

import java.net.*;
import java.io.*;

/**
 * The handler class creates a ProtocolNameConnection class instance
 * It is invoked indirectly via Java's URL getContent() mechanism.
 *
 * how to use it:
 * System.setProperty ("java.protocol.handler.pkgs",
 * "com.crossworlds.ProtocolHandler");
 * URL url = new URL ("the URL");
 * CWURLConnection uc = (CWURLConnection) url.openConnection ();
 * /
public class Handler
{
    // this will return the appropriate URLConnection
    // But the constructor takes only one argument - the URL. As this
    // is called by Java Networking Framework.
    public URLConnection openConnection(URL url) throws IOException
    {
        // you can pass in any parameters here.
        return new MyURLConnection (url);
    }
}

class MyURLConnection extends CWURLConnection {

    /**
     * This is instantiated by URL.openConnection()
     */
    public MyURLConnection(URL url)
    {
        // store this URL some where
    }

    /**
     * This method returns the content type of the data
     */
    public String getContentType()
    {
        // here is where you have to determine the content Type (aka
        // Mimetype) of URL streams
    }

    /**
     * This method is used to create a connection

```

```

    */
public synchronized void connect() throws IOException
{
    // you might call super().connect as it suffices most of the
    // time.
    // If it is custom protocol, do the handshaking stuff here
}

/**
 * getContent () : The getContent method used by CrossWorlds.
 * This method takes in 3 parameters
 * - input Object,
 * - content type for the data &
 * - Business Object Prefix to * be used to create the Business
 * Object name
 * It returns an appropriate Object back to the caller. This
 * method interacts with the DataHandler using the exposed APIs
 * for the DataHandler.
 */
public Object getContent(Object input, String mimeType, String B0prefix)
    throws IOException
{
    // log a message
    JavaConnectorUtil.logMessage ("logging a message", JavaConnectorUtil.XRD_INFO);

    // write a trace
    if (JavaConnectorUtil.isTraceEnabled (JavaConnectorUtil.LEVEL3))
        JavaConnectorUtil.traceWrite (JavaConnectorUtil.LEVEL3, "Level 3 trace msg");

    // get a datahandler
    DataHandler dh = DataHandler.createHandler (null, mimeType, B0prefix);

    InputStream in = dh.getStreamFromBO ( (BusinessObjectInterface) input, null);

    // Send this to URL
    - read data from Input Stream
    - write to URL
    - repeat until input stream is drained.

    // Now read the response
    String replyString = // some how read the reply from URL
    String outputType = // get the mime of reply some how

    // remember to get a fresh DH, as the incoming data may be of
    // different mime type than was originally received by the
    // protocol handler
    DataHandler dh2 = DataHandler.createHandler(null, outputType, B0prefix);

    BusinessObjectInterface replyBO = dh2.getBO (replyString, outputType);

    return replyBO; // DONE!
}
}

```

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 7은 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 7. 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 7. 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 7. 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 7. 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 `true`로 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 `false`로 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 켜져서 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 42 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.

- 53 페이지의 『Connector Configurator 개요』
- 54 페이지의 『Connector Configurator 시작』
- 55 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 58 페이지의 『새 구성 파일 작성』
- 61 페이지의 『구성 파일 등록 정보 설정』
- 69 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(55 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, 56 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(61 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 56 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 템플

리트에 나타납니다.

등록 정보가 종속되도록 지정하고 종속 조건을 설정하려면, 다음을 수행하십시오.

1. 사용 가능한 등록 정보 화면에서 종속될 등록 정보를 선택하십시오.
2. 등록 정보 선택 필드에서 드롭 다운 메뉴를 사용하여 조건 값을 보유할 등록 정보를 선택하십시오.
3. 조건 연산자 필드에서 다음 중 하나를 선택하십시오.

== (같음)

!= (같지 않음)

> (초과)

< (미만)

>= (이상)

<= (이하)

4. 조건 값 필드에 종속 등록 정보를 템플릿에 포함하기 위해 필요한 값을 입력하십시오.
5. 사용 가능한 등록 정보 화면에서 종속 등록 정보가 강조표시된 상태에서 화살표를 눌러서 종속 등록 정보 화면으로 이동하십시오.
6. 완료 버튼을 누르십시오. Connector Configurator가 Connector Configurator를 설치한 \bin 디렉토리의 \data\app에 XML 문서로 입력한 정보를 저장합니다.

새 구성 파일 작성

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

브로커를 선택하려면 다음을 수행하십시오.

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

다음 작업도 수행할 수 있습니다.

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

커넥터 특정 템플릿에서 구성 파일 작성

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

1. 파일 > 새로 작성 > 커넥터 구성을 누르십시오.
2. 다음 필드가 있는 새 커넥터 대화 상자가 나타납니다.

- 이름

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

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

- 시스템 연결

ICS, WebSphere Message Brokers 또는 WAS를 누르십시오.

- 커넥터 특정 등록 정보 템플릿

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

사용하려는 템플릿을 선택하고 확인을 누르십시오.

3. 구성 중인 커넥터의 구성 화면이 표시됩니다. 제목 표시줄이 통합 브로커와 커넥터 이름을 표시합니다. 모든 필드값을 입력하여 지금 정의를 완료하거나, 파일을 저장했다가 나중에 필드를 완성할 수 있습니다.
4. 파일을 저장하려면 파일 > 저장 > 파일에 또는 파일 > 저장 > 프로젝트에를 누르십시오. 프로젝트를 저장하려면 System Manager가 실행 중이어야 합니다. 파일로 저장하는 경우 파일 커넥터 저장 대화 상자가 나타납니다. 파일 유형으로 *.cfg를 선택하고 파일 이름 필드에서 이름의 철자 및 대소문자가 올바른지 확인한 후, 파일을 저장할 디렉토리로 이동하고 저장을 누르십시오. Connector Configurator의 메시지 패널에 있는 상태 화면은 구성 파일이 성공적으로 작성되었음을 표시합니다.

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

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

기존 파일 사용

다음 중 하나 이상의 형식으로 기존 파일을 사용할 수 있습니다.

- 커넥터 정의 파일.
이 파일은 특정 커넥터의 등록 정보와 적용 가능한 기본값을 나열한 텍스트 파일입니다. 일부 커넥터는 전달 패키지의 \repository 디렉토리에 이러한 파일을 포함합니다. (이 파일의 확장자는 보통 .txt입니다. 예를 들어, XML 커넥터의 경우에는 CN_XML.txt입니다.)
- ICS 저장소 파일.
커넥터의 이전 ICS 구현에서 사용된 정의는 해당 커넥터의 구성에 사용된 저장소 파일에서 사용할 수도 있습니다. 이러한 파일의 확장자는 일반적으로 .in 또는 .out입니다.
- 커넥터의 이전 구성 파일.
이러한 파일의 확장자는 일반적으로 *.cfg입니다.

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 > 파일에서를 누르십시오.
2. 파일 커넥터 열기 대화 상자에서 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 > 프로젝트에서를 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 64 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 메시징을 사용하는 커넥터의 경우 적용 가능)

주: 데이터를 Business Object로 변환하는 Data Handler의 구성에서 JMS 메시징을 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. 값은 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 **아니오**를 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 **예**를 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 **추가**를 누르십시오. 하위 등록 정보를 추가하려면 상위 행 번호를 마우스 오른쪽 단추로 누른 다음 **하위 추가**를 누르십시오.
2. 등록 정보 또는 하위 등록 정보의 값을 입력하십시오.
3. 등록 정보를 암호화하려면 **암호화** 상자를 선택하십시오.
4. 『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

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

지원되는 Business Object 정의 지정

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

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

ICS가 브로커인 경우

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

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

1. **Business Object** 이름 목록의 빈 필드를 누르십시오. System Manager 프로젝트에 존재하는 모든 Business Object 정의를 보여주는 드롭 다운 목록이 표시됩니다.
2. Business Object를 눌러 추가하십시오.
3. Business Object에 대한 에이전트 지원(아래에 설명됨)을 설정하십시오.

4. Connector Configurator 창의 파일 메뉴에서 프로젝트에 저장을 누르십시오. 추가된 Business Object 정의에 지정된 지원을 포함하여 변경된 커넥터 정의가 System Manager의 프로젝트에 저장됩니다.

지원되는 목록에서 Business Object를 삭제하려면 다음을 수행하십시오.

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

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

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

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

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

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

대부분의 커넥터의 경우 최상의 노력만이 가능한 선택사항입니다.

트랜잭션 레벨 변경 사항을 적용하려면 서버를 다시 시작해야 합니다.

WebSphere Message Broker가 브로커인 경우

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

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

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

WAS가 브로커인 경우

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

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

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

연관된 맵(ICCS에만 해당)

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

Business Object 목록에는 에이전트가 지원하는 응용프로그램 특정 Business Object와 제어가 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 맵에서는 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 템플릿으로 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>
  </ValidValues>
  <DefaultValue>en_US</DefaultValue>
</Property>
```

부록 C. XML 어댑터 샘플 시나리오의 개요

기업이 IBM WebSphere Business Integration Adapter for XML을 사용하여 웹 서버에서 XML 문서를 읽고 게시하려는 경우를 고려하십시오. 다음은 XML 어댑터 기능의 기본 사항을 간단하게 보여주기 위해 설계된 두 가지 다른 샘플 시나리오입니다.

- WebSphere MQ Integrator Broker 연결로 XML 샘플 시나리오 설치

이 가상 시나리오에는 두 가지 데이터 교환 방향을 나타내는 다음 두 가지 통합이 포함됩니다.

- 첫 번째 통합은 WebSphere MQ Integrator Broker에서 WebSphere MQ 대기열로 전송된 "XML_REQUEST_Order" 이벤트로 시작됩니다. XML 어댑터는 대기열의 이벤트를 승인하고 XML DataHandler를 호출하여 이벤트를 XML 문서로 변환시킵니다. XML 문서는 웹 서버에 게시됩니다. 웹 서버는 자체 XML이 있는 XML 어댑터에 응답합니다.
- 두 번째 통합에서는 XML 어댑터가 XML 문서의 URL을 폴합니다. XML 어댑터가 문서를 읽으면 XML DataHandler를 통해 전송합니다. DataHandler는 WebSphere MQ Integrator Broker가 읽을 WebSphere MQ 대기열로 전송할 응답 이벤트를 리턴합니다.

- WebSphere InterChange Server 연결로 XML 샘플 시나리오 설치

이 가상 시나리오에는 두 가지 데이터 교환 방향을 나타내는 다음 두 가지 통합이 포함됩니다.

- 첫 번째 통합은 "Port_To_XML" 협업 오브젝트를 통해 PortConnector에서 XML 어댑터로 전송된 "XML_REQUEST_Order" 오브젝트로 시작됩니다. XML 어댑터는 XML DataHandler를 호출하여 요청 Business Object를 XML 문서로 변환시킵니다. XML은 웹 서버에 게시됩니다. 웹 서버는 XML 어댑터에 응답하고, XML 어댑터는 WebSphere ICS에 다시 전송한 응답 Business Object로 응답 XML을 변환시킵니다.
- 두 번째 통합에서는 XML 어댑터가 XML 문서의 URL을 폴합니다. XML 어댑터가 문서를 읽으면 XML DataHandler를 통해 전송하여, WebSphere InterChange Server로 전송될 응답 Business Object로 변환시킵니다. 그런 다음 두 협업, "XML_To_Port_Customer" 또는 "XML_To_Port_Manifest" 중 하나를 통해 포트 커넥터로 이벤트가 전송됩니다.

WebSphere MQ Integrator Broker 연결로 XML 샘플 시나리오 설치

주: 이 예에서는 실제 WebSphere MQ Integrator Broker를 사용하지 않습니다. XML 어댑터가 WebSphere MQ 대기열에서 읽고 쓸 뿐입니다. Portconnector를 실행하는 Visual Test Connector를 사용하여 WebSphere MQ Integrator Broker를 시험합니다.

샘플 시나리오를 설치 및 확인하기 위한 단계는 다음과 같습니다.

- 설치 전 참고 및 가정사항
- 샘플 시나리오 설치
- 서비스 호출 요청 시나리오 실행
- 폴 시나리오 실행
- 요약

설치 전 참고 및 가정사항

1. IBM WebSphere Business Integration Adapter를 설치하고 사용했습니다.
2. IBM WebSphere MQ를 설치하고 사용했습니다.
3. IBM WebSphere Business Integration Adapter for XML을 설치했습니다.
4. Webserver가 Java Servlet을 처리하도록 설정했습니다.
5. 이 문서에서 말하는 %CROSSWORLDS%는 현재 IBM WebSphere Business Integration Adapters 설치가 들어 있는 폴더를 의미합니다.
6. 모든 환경 변수 및 파일 분리는 Windows NT/2000 형식으로 지정합니다. UNIX 플랫폼에서 실행 중인 경우, 적절히 변경하십시오(예를 들어, %CROSSWORLDS%\connectors는 \${CROSSWORLDS}/connectors가 됩니다).

샘플 시나리오 설치

1. WebSphere MQ 설정

먼저 채널 초기자 및 리스너를 실행하여 WebSphere MQ 대기열 관리자를 작성 및 시작하십시오. 그런 다음 이름이 다음과 같은 대기열을 작성하십시오.

- ADMININQUEUE
- ADMINOUTQUEUE
- DELIVERYQUEUE
- FAULTQUEUE
- REQUESTQUEUE

2. XML Connector 및 포트 커넥터 CFG 파일 구성

%CROSSWORLDS%\connectors\XML\samples\

WebSphereMQIntegratorBroker\XMLConnector.cfg 파일을 여십시오.

파일의 다음 등록 정보를 사용자의 특정 설정에 맞게 수정해야 합니다. 자세한 정보는 "IBM WebSphere Business Integration Adapter for XML 안내서"를 참조하십시오. 등록 정보가 아직 존재하지 않는 경로 또는 파일 이름을 표시하는 경우, 다음을 작성해야 합니다.

- QueueManager
- RepositoryDirectory
- ArchiveDirectory

%CROSSWORLDS%\connectors\XML\samples\

WebSphereMQIntegratorBroker\PortConnector.cfg 파일을 여십시오.

파일의 다음 등록 정보를 사용자의 특정 설정에 맞게 수정해야 합니다. 등록 정보가 아직 존재하지 않는 경로 또는 파일 이름을 표시하는 경우, 다음을 작성해야 합니다.

- QueueManager
- RepositoryDirectory

3. XML Connector CFG 파일 지정

- *NT/2000*의 경우

XML CONNECTOR에 대한 단축 아이콘의 등록 정보를 여십시오. 대상의 마지막 인수로 "-c" + <XMLConnector.cfg 파일의 전체 경로 및 파일 이름>을 추가하십시오.

예:

```
c%CROSSWORLDS%\connectors\XML\samples\WebSphereMQIntegratorBroker\XMLConnector.cfg.
```

- *UNIX*의 경우

\${CROSSWORLDS}/bin/connector_manager_XML 파일을 여십시오.

AGENTCONFIG_FILE 등록 정보 값을 "-c" + <XMLConnector.cfg 파일의 전체 경로 및 파일 이름>으로 설정하십시오.

예:

```
AGENTCONFIG_FILE=c${CROSSWORLDS}/connectors/XML/samples/WebSphereMQIntegratorBroker/XMLConnector.cfg.
```

4. 사용자 환경에서 사용할 수 있도록 Servlet 수정

다음과 같이 Servlet을 변경해야 합니다.

- PollXMLOrder.java의 경우

소스 파일의 41행에서, outFileName 문자열 값을 Servlet이 수신 XML 메시지를 로그해야 하는 로컬 시스템의 파일 이름으로 변경해야 합니다. 소스 파일의 56행에서, FileInputStream 구성자로 전달된 값은 전달된 SamplePollingInput.xml 파일에 대한 로컬 시스템의 정확한 파일 이름이어야 합니다.

- MirrorXMLOrder.java의 경우

변경하지 않아도 됩니다.

5. XML 폴 Business Object 구성

Business Object Designer를 시작하십시오. 파일 -> 파일에서 열기를 선택하고, "XML_POLL_Order.xsd" 파일을 선택하십시오. "XML_POLL_Order"로 레이블 표시된 Business Object가 열립니다. 이 Business Object의 첫 번째 속성 이름은 "URL"이어야 합니다. URL 속성의 기본값을 XML 어댑터가 XML 응답을 청취하는 위치로 변경하십시오(예: PollXMLOrder.java). XML_POLL_Order.xsd 파일을 수정된 Business Object로 바꿔 Business Object를 저장하십시오.

6. 웹 서버 구성

전달된 Servlet을 컴파일하십시오. 웹 서버가 생성된 클래스 파일을 선택하여 실행하려면 올바른 디렉토리로 이동시켜야 합니다. 또한 Servlet을 등록하기 위한 추가 단계를 수행해야 합니다. 정확한 요구사항은 전적으로 사용 중인 웹 서버에 따라 다릅니다.

서비스 호출 요청 시나리오 실행

1. 시작

- XML 어댑터를 시작하십시오.
- WebServer를 시작하십시오.
- Visual Test Connector의 한 인스턴스를 시작하십시오.

2. 포트 커넥터 시험

Visual Test Connector를 사용하여 "PortConnector"의 프로파일을 정의하십시오. Test Connector의 메뉴에서 파일 -> 프로파일 열기를 선택하십시오. 추가 단추를 누르고 %CROSSWORLDS%\connectors\XML\samples\WebSphereMQIntegratorBroker\PortConnector.cfg를 커넥터 구성 파일로 지정하십시오. 다음 두 창에서 확인을 누르고 에이전트를 연결하십시오.

3. 테스트 데이터 로드

"PortConnector"를 시험하는 Test Connector를 사용하여 메뉴에서 편집 -> BO 로드를 선택하십시오. 다음 파일을 로드하십시오.

%CROSSWORLDS%\connectors\XML\samples\
WebSphereMQIntegratorBroker\sampleOrderData.bo

4. URL 설정

Test Connector에 로드된 테스트 데이터를 여십시오. XML 어댑터가 해당 XML 요청을 게시해야 하는 위치로 URL 속성값을 수정하십시오.

5. 테스트 데이터 전송

"PortConnector"를 시험하는 Test Connector를 사용하여 로드된 테스트 BO를 누르십시오. 메뉴에서 요청 -> 전송을 선택하십시오.

6. 처리 완료 확인

처리가 완료되었는지 확인하려면, XML 어댑터가 이벤트를 받아 XML 문서로 변환하고 웹 서버에 게시했으며, 응답을 받아 구문 분석했는지 확인하십시오.

폴 시나리오 실행

1. 시작

- XML 어댑터
- WebServer
- Visual Test Connector의 한 인스턴스

2. 포트 커넥터 시험

Visual Test Connector를 사용하여 "PortConnector"의 프로파일을 정의하십시오. Test Connector의 메뉴에서 파일 -> 프로파일 열기를 선택하십시오. 추가 단추를 누르고 %CROSSWORLDS%\connectors\XML\samples\
WebSphereMQIntegratorBroker\PortConnector.cfg를 커넥터 구성 파일로 지정하십시오. 다음 두 창에서 확인을 누르고 에이전트를 연결하십시오.

3. 샘플 데이터 폴

- **NT/2000 사용**

PollFrequency가 이미 키로 설정되어 있습니다. XML 어댑터를 시작한 명령 창에서 문자 'p'를 입력하고 Enter 키를 누르십시오. XML 어댑터를 중지하십시오. XMLConnector.cfg 파일을 열고, PollFrequency의 숫자(예: 30000(폴/밀리세컨드))를 변경한 후 파일을 저장하십시오. 커넥터를 다시 시작하고, 커넥터가 폴할 때까지 30초 대기하십시오.

- **UNIX 사용**

XML 어댑터를 중지하십시오. XMLConnector.cfg 파일을 열고, PollFrequency의 숫자(예: 30000(폴/밀리세컨드))를 변경한 후 파일을 저장하십시오. 커넥터를 다시 시작하고, 커넥터가 폴할 때까지 30초 대기하십시오.

4. 포트 커넥터를 사용하여 요청 승인

XML 어댑터가 XML 문서를 받아, WebSphere MQ 대기열에 배치할 메시지로 변환시킵니다. Test Connector를 사용하여 요청을 승인하고 응답 성공으로 응답하십시오.

5. 처리 완료 확인

처리가 완료되었는지 확인하려면, Test Connector 및 Archive Directory에 있는 승인된 요청의 데이터가 샘플에 제공된 SamplePollingInput.xml 파일의 이벤트와 일치하는지 확인하십시오.

요약:

위의 모든 단계를 수행한 경우, XML 어댑터 및 XML DataHandler를 사용하여 WebSphere MQ Integrator Broker와 웹 서버 사이에 XML 문서를 교환하는 작업 샘플 시나리오가 존재해야 합니다.

WebSphere InterChange Server 연결로 XML 샘플 시나리오 설치

주: 이 샘플에서는 폴이 다음 세 가지 Business Object를 리턴합니다.

- XML_Order_Customer
- XML_Order_Manifest
- XML_Order_Receipt

이들 중 두 가지 Business Object(Customer 및 Manifest)만 등록을 제공하는 협업을 갖고 있으므로, 세 번째 Business Object(Receipt)는 XML 어댑터에 의해 지정된 위치에 아카이브됩니다.

샘플 시나리오를 설치 및 확인하기 위한 단계는 다음과 같습니다.

- 설치 전 참고 및 가정사항
- 샘플 시나리오 설치
- 서비스 호출 요청 시나리오 실행
- 폴 시나리오 실행
- 요약

설치 전 참고 및 가정사항

1. IBM WebSphere Business Integration Adapter를 설치하고 사용했습니다.
2. IBM WebSphere InterChange Server를 설치하고 사용했습니다.
3. IBM WebSphere Business Integration Adapter for XML을 설치했습니다.
4. Webserver가 Java Servlet을 처리하도록 설정했습니다.

5. 이 문서에서 말하는 %CROSSWORLDS%는 현재 IBM WebSphere Business Integration Adapters 설치가 들어 있는 폴더를 의미합니다.
6. 모든 환경 변수 및 파일 분리자는 Windows NT/2000 형식으로 지정합니다. UNIX 플랫폼에서 실행 중인 경우, 적절히 변경하십시오(예를 들어, %CROSSWORLDS%\connectors는 \${CROSSWORLDS}/connectors가 됩니다).

샘플 시나리오 설치

1. 저장소에 Business Object 로드

IBM WebSphere ICS Interchange Server를 시작하고, WebSphere Business Integration System Manager를 사용하여 Business Object Designer의 "파일" 메뉴에서 "파일에서 열기" 메뉴 항목을 선택하십시오.

%CROSSWORLDS%\connectors\XML\Samples\WebSphereICS 폴더에 있는 "Sample_XML_Order_Objects.in"으로 레이블 표시된 저장소 파일을 로드하십시오. Business Object가 로드되었는지 확인하십시오. 총 12개가 있어야 합니다.

2. 저장소에 커넥터 로드

WebSphere Business Integration System Manager를 사용하여 Connector Designer의 "파일" 메뉴에서 "파일에서 열기" 메뉴 항목을 선택하십시오.

%CROSSWORLDS%\connectors\XML\Samples\WebSphereICS 폴더에 있는 "Sample_XML_Order_Connectors.in"으로 레이블 표시된 저장소 파일을 로드하십시오. XMLConnector 및 PortConnector 정의가 로드되었는지 확인하십시오.

3. XML Connector 구성

WebSphere Business Integration System Manager를 사용하여 XML CONNECTOR 정의를 두 번 눌러 Connector Designer를 실행하십시오. 다음 응용프로그램 구성 등록 정보 값을 사용자의 특정 파일 구조에 맞게 수정해야 합니다. 다음 경로 및/또는 파일이 아직 존재하지 않는 경우 이를 작성해야 합니다.

- ArchiveDirectory

4. 저장소에 협업 템플릿 및 오브젝트 로드

WebSphere Business Integration System Manager를 사용하여 "파일" 메뉴에서 "파일에서 열기" 메뉴 항목을 선택하십시오.

%CROSSWORLDS%\connectors\XML\samples\WebSphereICS 폴더에 있는 "Sample_XML_Order_Collaborations.in"으로 레이블 표시된 저장소 파일을 로드하십시오. 세 가지 템플릿 정의 및 세 가지 협업 오브젝트가 로드되었는지 확인하십시오.

5. 협업 템플릿 컴파일

WebSphere Business Integration System Manager를 사용하여 협업 템플릿으로 레이블 표시된 폴더를 마우스 오른쪽 단추로 누르고 드롭 다운 목록에서 모두 컴파일을 선택하십시오.

6. 사용자 환경에서 사용할 수 있도록 Servlet 수정

다음과 같이 Servlet을 변경해야 합니다.

- PollXMLOrder.java의 경우

소스 파일의 41행에서 outFileName 문자열 값을 Servlet이 수신 XML 메시지를 기록해야 하는 로컬 시스템의 파일 이름으로 변경해야 합니다. 소스 파일의 56행에서 FileInputStream 구성자로 전달된 값은 전달된 SamplePollingInput.xml 파일에 대한 로컬 시스템의 정확한 파일 이름이어야 합니다.

- MirrorXMLOrder.java의 경우

변경하지 않아도 됩니다.

7. XML 폴 Business Object 구성

WebSphere Business Integration System Manager에서 "XML_POLL_Order"로 레이블 표시된 Business Object를 여십시오. 이 Business Object의 첫 번째 속성 이름은 "URL"이어야 합니다. URL 속성의 기본값을 XML 어댑터가 XML 응답을 청취하는 위치로 변경하십시오(예: PollXMLOrder.java). Business Object를 서버에 저장하십시오.

8. 웹 서버 구성

전달된 Servlet을 컴파일하십시오. 웹 서버가 생성된 클래스 파일을 선택하여 실행하려면 올바른 디렉토리로 이동시켜야 합니다. 또한 Servlet을 등록하기 위한 추가 단계를 수행해야 합니다. 정확한 요구사항은 전적으로 사용 중인 웹 서버에 따라 다릅니다.

9. WebSphere InterChange Server 다시 시작

모든 변경사항을 적용하려면 Interchange Server를 다시 시동하십시오. WebSphere Business Integration System Manager에서 시스템 보기를 사용하여 모든 협업 오브젝트 및 커넥터 제어기의 상태가 녹색인지 확인하십시오.

서비스 호출 요청 시나리오 실행

1. 시작

- WebSphere Interchange Server(아직 실행하지 않는 경우)
- XML 어댑터
- 웹 서버
- Visual Test Connector의 한 인스턴스

2. 포트 커넥터 시험

Test Connector를 사용하여 "PortConnector"의 프로파일을 정의하십시오. Test Connector 메뉴에서 파일 -> 에이전트 연결을 선택하여 에이전트 시험을 시작하십시오.

3. 테스트 데이터 로드

"PortConnector"를 시험하는 Test Connector를 사용하여 메뉴에서 편집 -> BO 로드를 선택하십시오. 다음 파일을 로드하십시오.

```
%CROSSWORLDS%\connectors\XML\samples\WebSphereICS\sampleOrderData.bo
```

4. URL 설정

Test Connector에 로드된 테스트 데이터를 여십시오. XML 어댑터가 해당 XML 요청을 게시해야 하는 위치로 URL 속성값을 수정하십시오.

5. 테스트 데이터 전송

"PortConnector"를 시험하는 Test Connector를 사용하여 로드된 테스트 BO를 누르십시오. 메뉴에서 요청 -> 전송을 선택하십시오.

6. 처리 완료 확인

처리가 완료되었는지 확인하려면, XML 어댑터가 이벤트를 받아 BO를 XML 문서로 변환하고 웹 서버에 게시했으며, 응답을 받아 구문 분석한 후 다시 협업에 전송했는지 확인하십시오.

폴 시나리오 실행

1. 시작

- WebSphere Interchange Server(아직 실행하지 않는 경우)
- XML 어댑터
- 웹 서버
- Visual Test Connector의 한 인스턴스를 시작하십시오.

2. 포트 커넥터 시험

Test Connector를 사용하여 "PortConnector"의 프로파일을 정의하십시오. Test Connector의 메뉴에서 파일 -> 에이전트 연결을 선택하여 에이전트 시험을 시작하십시오.

3. 샘플 데이터 폴

- *NT/2000*의 경우

PollFrequency가 이미 키로 설정되어 있습니다. XML 어댑터를 시작한 명령 창에서 문자 'p'를 입력하고 Enter 키를 누르십시오.

- *UNIX*의 경우

XML 어댑터를 중지하십시오. XML 어댑터 제어기를 열고, PollFrequency의 숫자(예: 30000(폴/밀리세컨드))를 변경하십시오. 커넥터를 다시 시작하고, 커넥터가 폴할 때까지 30초 대기하십시오.

4. 포트 커넥터를 사용하여 요청 승인

XMLConnector가 XML 문서를 받아, CrossWorlds Business Object로 변환시킨 후 WebSphere InterChange Server로 전달합니다. WebSphere ICS는 이 이벤트에 등록된 두 가지 다른 협업으로 이벤트를 전달합니다. 협업은 PortConnector로 이벤트를 전달합니다. Test Connector를 사용하여 요청을 승인하고 두 이벤트에 대한 응답 성공으로 응답하십시오.

5. 처리 완료 확인

처리가 완료되었는지 확인하려면, Test Connector 및 Archive Directory에 있는 승인된 요청의 데이터가 샘플에 제공된 SamplePollingInput.xml 파일의 이벤트와 일치하는지 확인하십시오.

요약:

위의 모든 단계를 수행한 경우, XML 어댑터 및 XML DataHandler를 사용하여 IBM WebSphere ICS와 웹 서버 사이에 XML 문서를 교환하는 작업 샘플 시나리오가 존재해야 합니다.

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

색인

[다]

디버깅 13
all 13
data 13
handshake 13
keygen 13
plaintext 13
record 13
session 13
ssl 13
verbose 13

[사]

사용자 정의 구성요소 21
스키마 문서 27

[아]

요청/응답 2, 4, 5, 6, 25
Business Object 25
응답 Business Object 3, 6
이벤트 공고 26
개요 7
Business Object 26
PollForEvents() 메소드 3

[카]

커넥터 에이전트 2
및 Business Object 22
응답 Business Object 3
조작 2
Meta Object 3
클래스 이름
com.crossworlds.DataHandlers.modified_content_type 3
URLConnection 3

[파]

프로토콜 핸들러 4
사용자 정의(샘플 코드) 33
클래스 개발 31

프로토콜 핸들러 클래스 31
프로토콜 핸들러 프레임워크 29, 30
메소드 31, 32
CWURLConnection 3
프로토콜 핸들러 프레임워크 메소드
public abstract Object getContent() 31

A

all
디버깅 13

B

BOPrefix 24
Business Object
요청 5
필수 속성 23

C

createHandler() 메소드 3
CWURLConnection 29

D

data
디버깅 13
Data Handler 프레임워크 3
createHandler() 메소드 3
doVerbFor() 메소드 2
DTD 27

G

getAttrValue() 5

H

handshake
디버깅 13
HTTP/HTTPS 3
프록시 이름 2

I

init() 메소드 2

J

Java 클래스 패키지

JavaProtocolHandlerPkgs 2

JavaProtocolHandlerPkgs 2

K

keygen

디버깅 13

M

Meta Object

modified_content_type 3

modified_content_type_BOPrefix 3

MimeType 4, 24

modified_content_type_BOPrefix 3

P

plaintext

디버깅 13

PollForEvents() 메소드 2, 3

public abstract Object getContent() 메소드 31

R

record

디버깅 13

S

session

디버깅 13

ssl

디버깅 13

V

verbose

디버깅 13

X

XML Connector

관련 책 v

구성요소 1

구조 2

릴리스 정보 v

및 프로토콜 핸들러 2

사용자 정의 구성요소의 필요성 판별 21

조작 4

Business Object

처리 4

XML ODA(Object Discovery Agent) 27

Business Object 구조 22, 27

BOPrefix 24

MIME 유형 24

Business Object 정의 21, 27

XML Connector 에이전트

메소드 2

XML Connector 에이전트 메소드

doVerbFor() 2

init() 2

pollForEvents() 2

XML Connector의 구조 2

XML Data Handler 4

XML Data Handler 패키지

JavaDataHandlerPkgs 2

IBM 한글 지원에 관한 설문

IBM WebSphere Business Integration Adapters
Adapter for XML 사용자 안내서
버전 3.3.x



FAX : (02) 3787-0123

보내 주시는 의견은 더 나은 고객 지원 체제를 위한 귀중한
자료가 됩니다. 독자 여러분의 좋은 의견을 기다립니다.

성 명		직위/담당업무	
회 사 명		부 서 명	
주 소			
전화번호		팩스번호	
전자우편 주소			
사용중인 시스템	<input type="checkbox"/> 중대형 서버 <input type="checkbox"/> UNIX 서버 <input type="checkbox"/> PC 및 PC 서버		

- IBM에서 제공하는 한글 책자와 영문 책자 중 어느 것을 더 좋아하십니까?
그 이유는 무엇입니까?
 한글 책자 영문 책자
(이유: _____)
- 본 책자와 해당 소프트웨어에서 사용된 한글 용어에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 본 책자와 해당 소프트웨어에서 번역 품질에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 본 책자의 인쇄 상태에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 한글 소프트웨어 및 책자가 지원되는 분야에 대해 귀하는 어떻게 생각하십니까?
 한글 책자를 늘려야 함 현재 수준으로 만족
 그다지 필요성을 느끼지 않음
- IBM은 인쇄물 형식(hardcopy)과 화면 형식(softcopy)의 두 종류로 책자를 제공합니다.
어느 형식을 더 좋아하십니까?
 인쇄물 형식(hardcopy) 화면 형식(softcopy) 둘 다

☞ IBM 한글 지원 서비스에 대해 기타 제안사항이 있으시면 적어주십시오.

☺ 설문에 답해 주셔서 감사합니다.
귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해
드리기 위해 최선을 다하겠습니다.

IBM