

IBM WebSphere Adapter



# Adapter for JDBC 사용자 안내서

버전 6.0

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 반드시 87 페이지의 『주의사항』의 일반 정보를 읽으십시오.

**2005년 9월 29일**

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM WebSphere Adapter for JDBC(5724L77), 버전 6.0 및 모든 후속 릴리스와 수정판에 적용됩니다.

본 문서에 대한 의견이 있는 경우, [doc-comments@us.ibm.com](mailto:doc-comments@us.ibm.com)으로 보내주십시오. 귀하의 의견을 기다리겠습니다.

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 2005. All rights reserved.

---

# 목차

WebSphere Adapter for JDBC 버전 6.0 사용자 안내서	1
제품 개요	1
IBM WebSphere Adapter	1
대상	3
타스크 로드맵: IBM WebSphere Adapter for JDBC	4
엔터프라이즈 서비스 찾기	4
구조	5
로케일 및 세계화 지원	6
Business Object 개요	8
Business Object 구조	8
Business Object 속성 등록 정보	13
지원되는 조작	14
트랜잭션 관리	15
인바운드 지원	16
아웃바운드 조작을 위한 Business Object 처리	19
응용프로그램 특정 정보	26
어댑터 설치	34
어댑터 환경	34
Adapter for JDBC의 특정 설치 정보	34
설치된 파일 구조	34
어댑터 프로젝트 작성	36
어댑터를 위한 프로젝트 작성	36
벤더 라이브러리 추가	38
Business Object 생성	38
참조 바인딩 생성	53
어댑터 프로젝트 전개	53
서버에 어댑터 구성	55
문제점 해결	57
IBM 소프트웨어 지원 문의	57
로깅 사용	59
추적 사용	60
CEI(Common Event Infrastructure) 사용 기능	61
예제 파일 및 샘플 응용프로그램 사용	62
가져오기/내보내기 및 WSDL 파일 예제	62
샘플: 데이터베이스 응용프로그램 갱신	64
참조	77
구성 등록 정보	77
연결 등록 정보	82
오브젝트 선택 등록 정보	84
주의사항	87
프로그래밍 인터페이스 정보	89

상표 및 서비스표 . . . . . 89

---

# WebSphere Adapter for JDBC 버전 6.0 사용자 안내서

IBM WebSphere Adapter for JDBC는 J2EE 응용프로그램과 EIS(Enterprise Information System) 사이에 양방향 연결을 제공합니다.

---

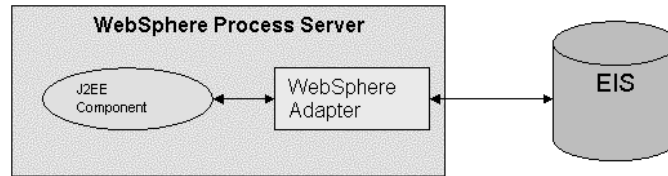
## 제품 개요

이 주제에서는 WebSphere Adapter for JDBC에 대한 기본 개념과 구조 및 환경 요구사항에 대해 소개합니다.

### IBM WebSphere Adapter

IBM WebSphere Adapter는 JCA(J2EE(Java 2 Enterprise Edition) Connector Architecture) 버전 1.5를 구현합니다. WebSphere Adapter는 자원 어댑터 또는 JCA 어댑터라고도 하며, WebSphere Process Server가 지원하는 J2EE 구성요소와 EIS(Enterprise Information System) 사이에서 관리를 통한 양방향 연결을 가능하게 합니다.

#### WebSphere Adapter



IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Adapter 포트폴리오는 J2EE(Java 2 Platform, Enterprise Edition) 표준을 기본으로 한 차세대 어댑터입니다. JCA는 J2EE 응용프로그램을 EIS에 통합하기 위한 표준 구조입니다. 이러한 각각의 시스템은 호출할 함수를 식별하고 입력 데이터를 지정하며 출력 데이터를 처리하는 데 필요한 원시 API를 제공합니다. JCA의 목적은 이러한 함수를 코딩하기 위한 독립된 API를 제공하고 데이터 공유를 지원하며 J2EE 응용프로그램을 기존 및 기타 EIS에 통합시키는 것입니다. JCA 표준은 응용프로그램 서버 내 J2EE 구성요소와 EIS 사이에 상호 작용을 관리하는 일련의 계약을 정의하는 방법으로 이를 수행합니다.

JCA 표준과 완전히 호환되는 WebSphere Adapter는 WebSphere Process Server에서 실행되도록 개발되었습니다. WebSphere Adapter는 다음을 수행합니다.

- WebSphere Process Server와 통합합니다.
- WebSphere Process Server에서 실행 중인 응용프로그램을 EIS에 연결합니다.
- 응용프로그램과 EIS 사이에서 데이터 교환을 가능하게 합니다.

각 WebSphere Adapter는 다음으로 구성됩니다.

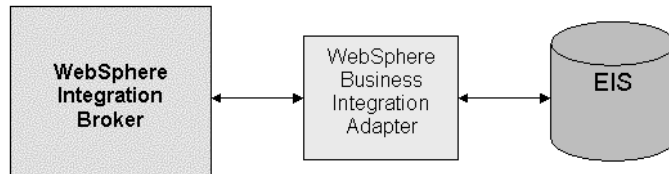
- WebSphere Process Server를 지원하는 JCA(J2EE Connector Architecture), 버전 1.5의 구현
- 엔터프라이즈 메타데이터 찾기 구성요소 — 이 구성요소를 엔터프라이즈 서비스 찾기 마법사와 함께 사용하여 EIS를 관찰하고 표준 EAR(Enterprise Application Archive) 파일로 컴파일되는 SCA(Service Component Architecture) 아티팩트와 Business Object를 생성할 수 있습니다.

WebSphere Adapter는 SDO(Service Data Object)를 사용하여 데이터 오브젝트를 표현합니다.

### WebSphere Adapter 및 WebSphere Business Integration Adapter

WebSphere Adapter와 달리 WebSphere Business Integration Adapter는 JCA와 호환되지 않습니다.

### WebSphere Business Integration Adapter



그림에 표시된 것과 같이 WebSphere Business Integration Adapter는 분산됩니다. 이 어댑터는 응용프로그램 서버의 외부에 상주합니다. 서버(통합 브로커)는 JMS(Java Messaging Service) 전송 계층을 통해 이러한 유형의 어댑터와 통신합니다.

그 밖에 WebSphere Adapter와 WebSphere Business Integration Adapter 사이의 차이점은 다음과 같습니다.

- 연결 관리 WebSphere Adapter는 표준 JCA 계약에 따라 시작, 중지 및 같은 수명 주기 작업을 관리하는 반면, WebSphere Business Integration Adapter는 WebSphere Adapter Framework에 따라 연결을 관리합니다.
- 이벤트 공고 WebSphere Adapter의 경우 인바운드 이벤트 공고입니다.
- 요청 처리 WebSphere Adapter에서는 아웃바운드 지원입니다.
- 오브젝트 정의 WebSphere Adapter에서는 엔터프라이즈 메타데이터 찾기 구성요소를 사용하여 EIS를 관찰하고 Business Object 및 기타 유용한 아티팩트를 개발할 수 있습니다. 이 엔터프라이즈 메타데이터 찾기 구성요소는 WebSphere Adapter의 일부입니다. 반면, WebSphere Business Integration Adapter에서는 별도의 ODA(Object Discovery Agent)를 사용하여 EIS를 관찰하고 Business Object 정의 스키마를 생성합니다.

## 대상

이 주제에서는 WebSphere Adapter 제품의 사용자를 정의하고 사용자에게 필요한 기술에 대해 상세히 설명합니다.

이 어댑터 사용자 안내서의 대상에는 응용프로그램 구성요소를 전체 솔루션으로 어셈블리하고 이 솔루션의 테스트 및 개발을 준비하는 일을 담당하는 데이터 및 응용프로그램 통합자가 포함됩니다. 이들 사용자에게는 다음과 같은 일반 기술이 필요합니다.

- 비즈니스 솔루션과 비즈니스 환경에 대한 이해
- 런타임 시 효율적인 협업을 수행할 수 있도록 응용프로그램 및 솔루션 구성요소에 대한 지식
- 이기종 관계형 데이터베이스, 대기열, 웹 서비스 등에 걸친 데이터베이스, 데이터 액세스 문제, 트랜잭션 모델 및 연결 등의 세부사항에 대한 이해
- 통합 도구의 익숙한 사용

응용프로그램 통합자에게는 세부 테스트 활동에 대한 책임이 있으며 다음과 같은 추가 기술이 요구됩니다.

- 테스트 및 개발에 필요한 필수 스크립트, 도구 및 템플릿 작성
- 통합 작업 영역과 통합 시스템 및 서브시스템 작성
- EJB(Enterprise Java Bean), 워크플로우 및 웹 페이지 같은 엔티티 사이의 상호 의존성 해결
- 응용프로그램 또는 솔루션의 유효성 검증

데이터 통합자에게는 응용프로그램 개발자가 광범위한 데이터 소스에 액세스할 수 있도록 해야 합니다. 다음과 같은 기술이 요구됩니다.

- 통합 기능 또는 지점간 게이트웨이 설치 및 구성
- 데이터베이스 액세스 논리를 효율적으로 사용하기 위한 프로시저 작성
- 외부 데이터 액세스 도구를 위한 데이터 모델 빌드
- 보안 수단 구현

### 관련 개념

64 페이지의 『샘플: 데이터베이스 응용프로그램 갱신』

IBM WebSphere Adapter for JDBC와 함께 제공되는 샘플 파일을 사용하여 데이터베이스 응용프로그램에서 특정 파일을 갱신하는 연습을 할 수 있습니다. 응용프로그램 통합자 및 데이터 통합자 역할을 위해 두 개의 단계별 시나리오가 제공됩니다. 역할에 따라 J2EE 응용프로그램 및 EIS(Enterprise Information System)와 통신하는 어댑터 구성, Business Object 생성 및 어댑터 전개를 연습할 수 있습니다.

## 타스크 로드맵: IBM WebSphere Adapter for JDBC

IBM WebSphere Adapter for JDBC를 사용하면 데이터베이스 레벨에서 J2EE 응용 프로그램과 데이터베이스 제공자의 EIS(Enterprise Information System) 사이에 Business Object를 쉽게 교환할 수 있습니다.

타스크	설명
어댑터 설치	Adapter for JDBC에 특정한 설치 고려사항을 설명하는 정보 및 설치 지시사항에 대한 링크
어댑터 프로젝트 작성	어댑터 프로젝트를 작성한 다음 Business Object와 서비스 구문을 생성하고 구성 등록 정보를 설정한 후 끝으로 어댑터 프로젝트를 응용프로그램 서버에 전개합니다.
어댑터 프로젝트 전개	프로젝트를 EAR 파일로 내보내고 이 파일을 응용 프로그램 서버에 설치합니다.
서버에 어댑터 구성	어댑터가 특정 데이터베이스 응용프로그램에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 재구성할 수 있습니다.
문제점 해결	이벤트 처리 상태를 추적하는 로그 파일을 설정하고 로그 파일에 캡처할 오류 또는 경고 레벨을 결정합니다.  IBM Software Support의 도움을 받고 TechNotes에 액세스하는 방법을 배웁니다.
샘플 데이터베이스 응용프로그램 갱신	Adapter for JDBC가 제공하는 샘플 파일을 사용하여 샘플 데이터베이스 응용프로그램의 전개, 구성 및 실행을 연습합니다. 공급하는 복잡도 레벨이 다른 두 개의 시나리오가 제공됩니다. 두 번째 시나리오에는 Business Object 생성이 포함됩니다.

## 엔터프라이즈 서비스 찾기

엔터프라이즈 서비스 찾기 마법사를 사용하여 EIS 엔티티에 대한 Business Object를 생성할 수 있습니다.

엔터프라이즈 서비스 찾기 마법사는 Business Object에 청사진을 제공합니다. 이 마법사를 사용하면 EIS의 메타데이터 정보를 찾아볼 수 있고 관심있는 아티팩트를 선택할 수 있으며 전개 가능한 서비스 오브젝트 및 설명을 생성할 수 있습니다. 메타데이터 트리 구조에서 Meta Object 노드를 선택하여 EIS 엔티티에 대한 Business Object를 생성할 수 있습니다. 메타데이터는 비즈니스 그래프와 Business Object로 구성되는 서비스 데이터 오브젝트로 변환됩니다.

엔터프라이즈 서비스 찾기 마법사를 사용하여 다음 조치를 수행할 수 있습니다.

- Business Object 생성
- Business Object에서 응용프로그램 특정 정보 설정
- 등록 정보에서 응용프로그램 특정 정보 설정



- 인바운드 및 아웃바운드 이벤트에 대한 서비스 설명 제공
- 인바운드 및 아웃바운드 이벤트에 대한 연결 설명 제공

## 구조

WebSphere Adapter for JDBC는 J2EE 응용프로그램과 EIS(Enterprise Information System) 사이에 양방향 연결을 제공하는 자원 어댑터입니다. 이러한 응용프로그램의 경우, Business Object 양식으로 되어 있는 데이터 교환이 데이터 레벨에 발생합니다.

Adapter for JDBC는 데이터베이스 제공업체인 EIS와 통신합니다. 데이터베이스에 대한 갱신은 다른 EIS(Enterprise Information System)에 적용되어야 하고 EIS의 데이터에 대한 변경사항은 데이터베이스에 적용되어야 합니다. 자원 어댑터는 JDBC 2.0 이상 스펙을 지원하는 JDBC 드라이버가 있는 데이터베이스에 빌드된 응용프로그램과 통합할 수 있습니다. 그 예로는 IBM DB2<sup>(R)</sup>, Oracle, Microsoft<sup>(TM)</sup> SQLServer, Sybase 및 Informix 데이터베이스를 들 수 있습니다.

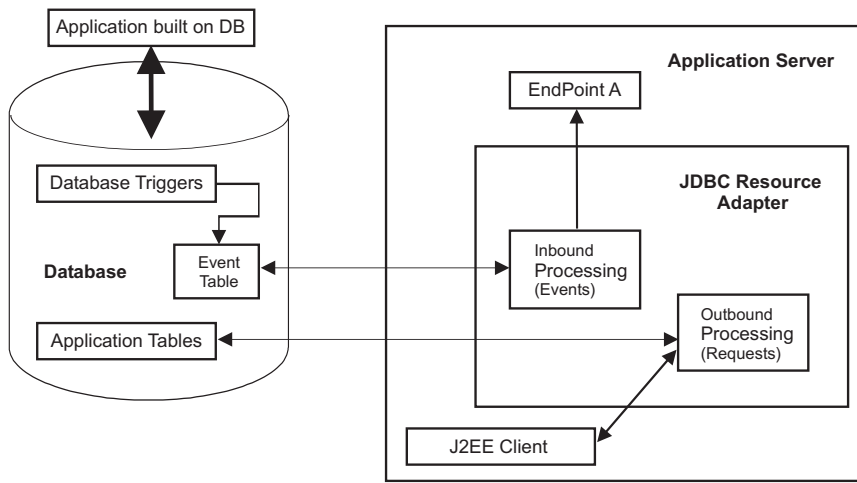
통합을 지원하기 위해 자원 어댑터는 EIS에서 수신한 요청을 처리하고 데이터베이스에 대한 갱신의 결과로 생성된 이벤트를 처리합니다. 어댑터는 이러한 이벤트를 응용프로그램 서버에 사전 정의된 여러 엔드포인트로 전송합니다. 엔드포인트는 J2EE 응용프로그램 또는 이벤트의 기타 클라이언트 사용자입니다. 데이터베이스의 테이블에 작성된 데이터 갱신은 이벤트 스토어에 게시된 이벤트 공고를 통해 Siebel, Peoplesoft 및 Oracle 같은 응용프로그램 서버에 연결된 다른 응용프로그램에 자동으로 전파됩니다. 어댑터는 Business Object에 지정된 대로 SQL 조회 또는 저장 프로시저를 사용하여 데이터베이스 테이블을 갱신합니다.

Adapter for JDBC는 JCA((J2EE(Java 2 Enterprise Edition) Connector Architecture)에서 인바운드 및 아웃바운드 지원을 제공하는 방법으로 JDBC API(Application Programming Interface)를 통해 액세스할 수 있는 데이터베이스 통합을 지원합니다. 아웃바운드 조작에서 Business Object는 Business Object에 지정된 조작(Create, Retrieve, Update, Delete 또는 RetrieveAll)에 따라 처리되는 요청으로서 어댑터에 전달됩니다. 요청은 어댑터에서 관리 중인 데이터베이스에 적용된 갱신이 있어야 할 다른 EIS 응용프로그램으로부터 수신됩니다. 이러한 요청의 처리 결과로 해당되는 데이터베이스 테이블의 행이 작성, 검색, 갱신 또는 삭제됩니다.

인바운드 조작의 경우, 데이터베이스의 적합한 테이블에서 데이터가 변경될 때 키 값 등의 관련 정보와 함께 적합한 이벤트가 이벤트 스토어로 삽입됩니다. 변경된 데이터를 캡처하기 위해 트리거가 각각의 테이블에 배치되거나 Oracle 데이터베이스에 제공된 Oracle Change Data Capture와 같은 다른 메소드가 사용됩니다. Adapter for JDBC는 이벤트 스토어를 폴링하여 이벤트 일괄처리를 검색합니다. 이러한 이벤트가 처리되고 각 이벤트는 JDBC 비즈니스 그래프를 구성하는 데 사용됩니다. 그런 다음 비즈니스 그래프는 특정 Business Object에 대한 등록이 있는 엔드포인트에 디스패치됩니다. 이미지 이후 지원만 인바운드 조작에 제공됩니다.

아래 『JDBC 어댑터 내 처리』에서는 인바운드 및 아웃바운드 조작을 보여 줍니다.

### JDBC 어댑터 내 처리



### 로케일 및 세계화 지원

이 어댑터는 1바이트 및 2바이트 문자 세트를 지원하고 지정된 언어로 메시지 텍스트를 전달할 수 있도록 세계화되어 있습니다.

이 어댑터는 아랍어와 히브리어에 대해 양방향 스크립트 데이터 처리를 지원합니다. 양방향 기능을 사용하려면 양방향 등록 정보를 구성해야 합니다. 이 사용자 안내서에서 양방향 등록 정보는 양방향 지원의 호출을 제어하는 등록 정보를 가리키는 용어입니다.

사용자의 EIS(Enterprise Information System)에서 Windows 표준 형식과 다른 양방향 형식을 사용하는 경우에는 양방향 지원이 되는 모든 등록 정보가 Windows 표준 형식에서 대상 EIS의 양방향 형식으로 변환됩니다. 또한 어댑터는 EIS의 데이터를 Windows 표준 형식으로 변환하여 WebSphere Process Server에 전달합니다.

JVM(Java Virtual Machine)의 Java™ 런타임 환경은 유니코드 문자 코드 세트의 데이터를 나타냅니다. 유니코드는 가장 많이 알려진 문자 코드 세트(단일 바이트 및 멀티 바이트)의 문자에 대한 인코딩을 포함합니다. WebSphere Business Integration 시스템의 구성요소는 대부분 Java로 작성되어 있습니다. 따라서 데이터가 대부분의 WebSphere Business Integration 시스템 구성요소들 사이에 전송되는 경우 문자 변환은 필요하지 않습니다.

어댑터는 실행 중인 시스템의 로케일을 사용하여 국가 또는 지역에 맞는 적합한 언어로 오류 및 정보 메시지를 로그합니다.

### WebSphere Process Server 양방향 언어 형식

WebSphere Process Server는 양방향 언어 형식인 ILYNN(implicit, left-to-right, on, off, nominal)을 사용하는데, 이는 Windows 양방향 언어 형식이기도 합니다. 기타 모든 언어 형식은 WebSphere Process Server로 가져오기 전에 변환 과정을 거쳐야 합니다.

적절한 양방향 언어 형식을 위해서는 다음의 다섯 가지 속성을 설정해야 합니다. 속성 및 설정은 『양방향 속성』 테이블에 나열되어 있습니다.

### 양방향 속성

문자 위치	목적	값	설명	기본 설정값
1	스키마 순서	I 또는 V	암시적(논리적) 또는 비주열	I
2	방향	L R CD	Left-to-Right Right-to-Left Contextual Left-to-Right Contextual Right-to-Left	L
3	대칭적 스와핑	Y 또는 N	대칭적 스와핑이 on 또는 off입니다.	Y
4	웨이핑	S N IM FB	텍스트가 웨이핑됩니다. 텍스트가 웨이핑되지 않습니다. 최초 웨이핑 중간 웨이핑 최종 웨이핑 격리 웨이핑	N
5	숫자 웨이핑	H, C 또는 N	힌두어(Hindi), 컨텍스트 (Contextual) 또는 명목 (Nominal)	N

어댑터는 데이터를 Logical-Left-to-Right로 변환하여 WebSphere Process Server 구성요소로 전송합니다.

주: 사용자 인터페이스(브라우저)의 로케일 설정이 양방향 언어 표시 및 편집 형식을 정의합니다. WebSphere Process Server 사용자 인터페이스는 로케일 특정 형식을 WebSphere Process Server 기본 형식으로 변환해야 합니다.

### 양방향 등록 정보 레벨

여러 가지 다른 레벨에서 양방향 등록 정보를 설정할 수 있습니다. 이들 등록 정보에 대한 자세한 내용과 엔터프라이즈 서비스 찾기 마법사를 사용한 등록 정보 설정 방법에 대해서는 어댑터 프로젝트 작성 및 어댑터 구성에 관한 섹션을 참조하십시오.

### 양방향 등록 정보 편집

WebSphere Integration Developer의 Business Object Designer에서 주석을 사용하여 Business Object 및 Business Object 속성에 대한 양방향 등록 정보를 편집할 수 있습니다. 주석은 Business Object에 저장됩니다(\*.xsd 파일). 자세한 내용은 WebSphere

Integration Developer 웹 사이트 <http://www.ibm.com/software/integration/wid>에 있는 Business Object Designer 문서를 참조하십시오.

또한 WebSphere Integration Developer의 어셈블리 편집기를 사용하여 정의가 완료된 특정한 양방향 등록 정보를 편집할 수도 있습니다. 런타임 시의 양방향 등록 정보 사용에 대한 자세한 내용은 양방향 지원에 관련된 일반 기술 서적 및 어댑터 기술지를 참조하십시오. 어셈블리 편집기에 대한 자세한 내용은 WebSphere Integration Developer 웹 사이트 <http://www.ibm.com/software/integration/wid>에 있는 어셈블리 편집기 문서를 참조하십시오.

---

## Business Object 개요

이 주제 세트에서는 Business Object의 구조, 속성 등록 정보, 지원되는 조작, 응용프로그램 특정 정보 및 이름 지정 규칙에 대해 설명합니다. 여기에는 트랜잭션 관리에 대한 정보도 들어 있습니다.

### Business Object 구조

각 Business Object는 데이터베이스 테이블 또는 보기에 해당하며 해당 오브젝트의 각 단순 속성은 해당 테이블 또는 보기의 열에 해당합니다.

단순 속성은 문자열, 정수 또는 날짜와 같은 단일 값을 나타냅니다. 따라서 같은 Business Object 내 속성은 서로 다른 데이터베이스 테이블에 저장될 수 없지만 다음 상황은 가능합니다.

- 해당 Business Object가 단순 속성을 갖는 것보다 데이터베이스 테이블에 더 많은 열이 있을 수 있습니다. 즉, 데이터베이스의 일부 열이 Business Object에 나타나지 않습니다. Business Object 처리에 필요한 열만 디자인에 포함되어야 합니다.
- 데이터베이스 테이블이 열을 갖는 것보다 Business Object에 더 많은 단순 속성이 있을 수 있습니다. 즉, Business Object의 일부 속성이 데이터베이스에 나타나지 않습니다. 데이터베이스에 표시되지 않는 속성은 응용프로그램 특정 정보가 없거나 기본값으로 설정되거나 또는 저장된 프로시저를 지정합니다.
- Business Object는 여러 데이터베이스 테이블을 이동시키는 보기를 나타낼 수 있습니다. 어댑터는 응용프로그램에 트리거된 이벤트(Create, Update 및 Delete)를 처리할 때 이러한 Business Object를 사용할 수 있습니다. 그러나 Business Object 요청을 처리할 때는 Retrieve 및 RetrieveAll 요청에 대해서만 Business Object를 사용할 수 있습니다.

주: Business Object가 SP(저장 프로시저)를 기본으로 하는 경우, 각 단순 속성(특별 SP 속성과 다름)에는 응용프로그램 특정 정보가 있을 수도 있고 없을 수도 있습니다. 자세한 내용은 『저장 프로시저 정의』를 참조하십시오.

Business Object는 평면일 수도 있고 계층 구조일 수도 있습니다. 평면 Business Object의 모든 속성은 단순하며 데이터베이스 테이블에 한 행을 표현합니다. 계층 구조 Business Object는 모든 레벨에서 모든 하위 Business Object를 포함하는 완전한 Business Object를 가리킵니다. 개별 Business Object는 포함하거나 포함할 수 있는 독립된 하위 Business Object와 하나의 Business Object를 가리킵니다. 각 Business Object는 여러 데이터베이스 테이블을 이동시키는 보기를 나타낼 수 있습니다. 최상위 레벨 Business Object는 상위 Business Object가 없는 계층 구조의 맨 위에 있는 개별 Business Object를 가리킵니다.

계층 구조 Business Object에는 하위 Business Object, 하위 Business Object 배열 또는 둘의 조합을 나타내는 속성이 있습니다. 다시 말해서 각 하위 Business Object에 하위 Business Object나 Business Object 배열 등을 포함할 수 있습니다. 상위 Business Object의 속성이 하나의 하위 Business Object를 나타낼 때 단일 카디널리티 관계가 발생합니다. 이 경우 속성은 하위 Business Object의 유형과 같습니다.

상위 Business Object의 속성이 하위 Business Object 배열을 나타낼 때 다중 카디널리티 관계가 발생합니다. 이 경우 속성은 하위 Business Object의 유형과 같습니다.

어댑터에서는 Business Object 사이에 다음 관계를 지원합니다.

- 단일 카디널리티 관계
- 단일 카디널리티 관계 및 소유권이 없는 데이터
- 다중 카디널리티 관계

각 카디널리티 유형에서 상위 Business Object와 하위 Business Object 사이의 관계는 해당 관계를 저장하는 Business Object의 키 속성에 대한 응용프로그램 특정 정보로 설명됩니다. 응용프로그램 특정 정보에 대한 자세한 내용은 『ForeignKey』를 참조하십시오.

#### 관련 참조

28 페이지의 『저장 프로시저 정의』

저장 프로시저는 Verb 레벨에 정의되어 있습니다. 각 저장 프로시저 정의는 StoredProcedureType, StoredProcedureName, ResultSet 및 Parameters라는 요소로 구성됩니다.

13 페이지의 『Business Object 속성 등록 정보』

Business Object 구조는 속성에 적용되는 여러 등록 정보를 정의합니다. 이 섹션의 『속성 등록 정보』 테이블에 어댑터가 이러한 등록 정보를 해석하고 그 설정을 설명하는 방법이 있습니다.

### Business Object의 단일 카디널리티 관계

이 주제에서는 이 어댑터가 지원하는 단일 카디널리티 관계에 대해 설명합니다.

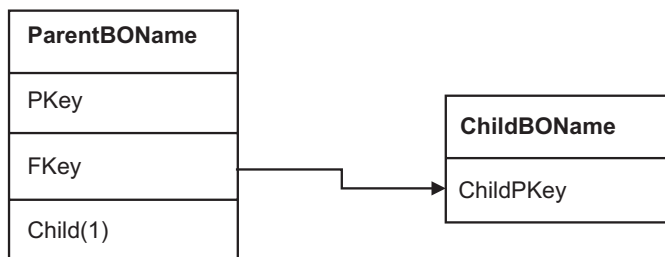
## 단일 카디널리티 관계

일반적으로 단일 카디널리티 하위 Business Object가 있는 Business Object에는 이러한 관계를 나타내는 속성이 두 개 이상 있습니다. 하나의 속성 유형은 하위 유형과 같습니다. 다른 속성은 단순 속성으로서 하위의 1차 키를 상위의 외부 키로서 포함합니다. 상위에는 하위에서 갖는 1차 키 속성 수만큼 외부 키 속성이 있습니다.

관계를 설정하는 외부 키가 상위에 저장되어 있기 때문에 각 상위마다 제공된 유형의 하위 Business Object를 하나씩만 가질 수 있습니다.

아래 『일반적인 단일 카디널리티 관계』에서는 일반적인 단일 카디널리티 관계를 보여줍니다. 이 예에서, ParentBOName 상자의 FKey는 하위 1차 키를 포함하는 단순 속성이 되고, ParentBOName 상자의 Child(1)는 하위 Business Object를 나타내는 속성이 됩니다.

### 일반 단일 카디널리티 관계



상위 Business Object는 단일 카디널리티 OWNERSHIP 하위와 단일 카디널리티 NOOWNERSHIP 하위를 가질 수 있습니다.

## 단일 카디널리티 관계 및 소유권이 없는 데이터

일반적으로 각 상위 Business Object는 포함된 하위 Business Object 안에 데이터를 소유합니다. 예를 들어, 각 고객 Business Object에 주소 Business Object가 하나씩 있는 경우 새 고객이 작성되면 고객 테이블과 주소 테이블 모두에 새 행이 삽입됩니다. 새 주소는 새 고객마다 다릅니다. 마찬가지로 고객 테이블에서 고객을 삭제할 때 고객의 주소도 주소 테이블에서 삭제됩니다.

그러나 여러 계층 구조 Business Object에 같은 데이터가 있지만, 아무도 소유하지 않는 경우가 발생할 수 있습니다. 예를 들어, 주소 Business Object에서 단일 카디널리티가 있는 StateProvince 찾아보기 테이블을 나타내는 StateProvince[1] 속성을 갖는다고 가정해 볼 수 있습니다. 해당 찾아보기 테이블은 잘 갱신되지 않고 주소 데이터와 관계 없이 유지보수되므로 주소 데이터의 작성 또는 수정은 찾아보기 테이블의 데이터에 영향을 주지 않습니다. 어댑터에서는 기존 상태 또는 지역 이름을 찾거나 실패하게 됩니다. 찾아보기 테이블에 값을 추가하거나 변경하지 않습니다.

여러 Business Object에 같은 단일 카디널리티 하위 Business Object가 있을 때는 각 상위 Business Object의 외부 키 속성이 관계를 NOOWNERSHIP로 지정해야 합니다. 응용프로그램 서버에서 어댑터에 Create, Delete 또는 Update Request를 사용하여 계층 구조 Business Object를 전송할 때는 어댑터에서 소유권이 없는 단일 카디널리티 하위를 무시합니다. 어댑터는 이러한 Business Object에서 Retrieve 조작만 수행합니다. 어댑터에서 단일 카디널리티 Business Object를 검색하지 못하면 오류를 리턴하고 처리를 중지합니다.

### 비정규화된 데이터 및 소유권이 없는 데이터

정적 찾아보기 테이블의 사용과 함께, 소유권이 없는 제약은 정규화된 데이터 및 비정규화된 데이터의 동기화라는 기능을 제공합니다.

**정규화된 데이터를 비정규화된 데이터로 동기화:** 관계를 NOOWNERSHIP으로 지정하면 정규화된 응용프로그램에서 비정규화된 응용프로그램으로 동기화할 때 데이터를 작성 또는 변경할 수 있습니다. 예를 들어, 정규화된 소스 응용프로그램이 A와 B라는 두 테이블에 데이터를 저장한다고 가정할 수 있습니다. 이때 엔티티 A마다 B 데이터를 반복하여 저장하도록 비정규화된 대상 응용프로그램에서 하나의 테이블에 모든 데이터를 저장한다고 가정해 볼 수 있습니다.

이 예의 경우, 소스 응용프로그램에서 대상 응용프로그램으로 테이블 B 데이터의 변경 사항을 동기화하려면 테이블 B 데이터가 변경될 때마다 테이블 A 이벤트를 트리거해야 합니다. 또한 테이블 B 데이터가 테이블 A에 반복적으로 저장되기 때문에 테이블 B에서 변경된 데이터를 포함하는 테이블 A의 각 행마다 Business Object를 전송해야 합니다.

**주:** 비정규화 테이블로 업데이트할 경우 각 레코드에 고유 키가 있어서 하나의 갱신 결과로 여러 키가 수정되지 않도록 해야 합니다. 고유 키가 존재하지 않으면 Adapter for JDBC는 여러 레코드가 갱신되었음을 나타내는 오류를 리턴합니다.

**비정규화된 데이터를 정규화된 데이터로 동기화:** 비정규화된 소스 응용프로그램에서 정규화된 대상 응용프로그램으로 데이터를 동기화할 때는 어댑터에서 정규화된 응용프로그램의 소유권이 없는 데이터를 작성, 삭제 또는 갱신하지 않습니다.

데이터를 정규화된 응용프로그램으로 동기화할 때는 어댑터에서 소유권이 없는 모든 단일 카디널리티 하위를 무시합니다. 이러한 하위 데이터를 작성, 제거 또는 수정하려면 데이터를 수동으로 처리해야 합니다.

### Business Object의 다중 카디널리티 관계

이 주제에서는 이 어댑터가 지원하는 다중 카디널리티 관계에 대해 설명합니다.

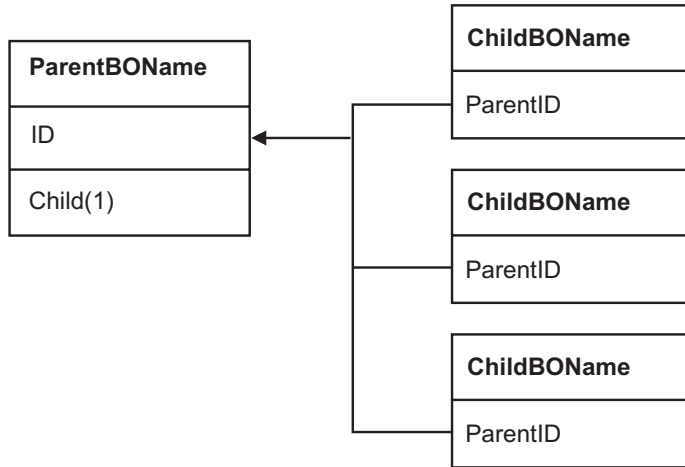
일반적으로 카디널리티 하위 Business Object의 배열을 가지는 Business Object에는 이러한 관계를 타나내는 속성이 하나만 있습니다. 이 속성의 유형은 하위 Business Object와 같은 유형의 배열이 됩니다. 두 개 이상의 하위가 있는 상위의 경우 관계를 설정하는 외부 키가 그 하위에 저장됩니다.

그러므로 각 하위에는 외부 키로서 상위의 1차 키를 포함하는 단순 속성이 하나 이상 있습니다. 하위에는 상위에서 갖는 1차 키 속성 수만큼 외부 키 속성이 있습니다.

관계를 설정하는 외부 키가 하위에 저장되어 있기 때문에 각 상위에는 하위가 0개 이상 있을 수 있습니다.

아래의 『다중 카디널리티 Business Object 관계』는 다중 카디널리티 관계를 보여 줍니다. 이 예에서, 세 개의 ChildBOName 상자에 있는 ParentID는 상위의 1차 키를 포함하는 단순 속성이고 ParentBOName 상자의 Child1은 하위 Business Object의 배열을 나타내는 속성입니다.

#### 다중 카디널리티 Business Object 관계

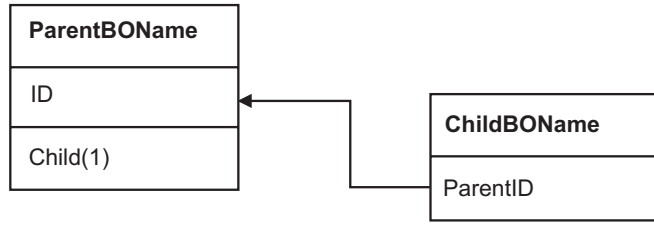


다중 카디널리티 관계는 N=1 관계가 될 수 있습니다. 일부 응용프로그램은 상위-하위 관계가 상위가 아닌 하위에 저장되도록 하위 엔티티를 한 개 저장합니다. 다시 말해서 하위에는 상위의 1차 키에 저장된 값과 동일한 외부 키가 포함됩니다.

응용프로그램에서는 하위 데이터가 상위와 별도로 존재하지 않고 그 상위를 통해서만 액세스될 수 있을 때 이 관계 유형을 사용합니다. 그러한 하위 데이터의 경우 하위와 그 외부 키 값이 작성되기 전에 상위와 그 1차 키 값이 존재해야 합니다. 아래 『N=1과의 다중 카디널리티 관계』는 이러한 관계 유형을 보여 줍니다.

#### N=1과의 다중 카디널리티 관계





## Business Object 속성 등록 정보

Business Object 구조는 속성에 적용되는 여러 등록 정보를 정의합니다. 이 섹션의 『속성 등록 정보』 테이블에 어댑터가 이러한 등록 정보를 해석하고 그 설정을 설명하는 방법이 있습니다.

### 속성 등록 정보

등록 정보	해석 및 설정
카디널리티	하위 Business Object 또는 하위 Business Object 배열을 나타내는 각 Business Object 속성은 각각 단일(1) 또는 다중(n) 카디널리티 값을 가집니다. 하위 Business Object를 나타내는 모든 속성은 또한 ContainedObjectVersion 등록 정보(하위의 버전 번호 지정) 및 관계 등록 정보(제약 값 지정)를 가집니다.
외부 키	카디널리티가 $n$ 인 하위 Business Object 배열을 검색할 때는 SELECT 명령문의 WHERE 절에 외부 키를 사용합니다. 주: 어댑터는 하위 Business Object를 외부 키로 나타내는 속성의 지정을 지원하지 않습니다. RetrieveAll Verb는 키 및 외부 키의 사용을 무효로 합니다.
키	각 Business Object에서 한 개 이상의 단순 속성을 키로서 지정해야 합니다. 주: 어댑터는 하위 Business Object 또는 하위 Business Object 배열을 키 속성으로 나타내는 속성의 지정을 지원하지 않습니다. 단순 속성에 대해 키 등록 정보가 true로 설정되는 경우 어댑터에서는 해당 속성을 Business Object 처리 중 생성하는 SELECT 명령문 및 UPDATE SQL 명령문의 WHERE 절에 추가합니다. RetrieveAll Verb는 키 및 외부 키의 사용을 무효로 합니다.
이름	하위 Business Object인 경우, Business Object의 이름 또는 단순 속성이면 속성의 고유한 이름입니다.

등록 정보	해석 및 설정
필수	속성에 값이 포함되어야 하는지 여부를 지정합니다. 카디널리티가 단일(1)인 컨테이너에 대해 이 등록 정보를 true로 설정하는 경우, 어댑터에서는 이 속성에 대한 하위 Business Object가 상위 Business Object에 있어야 합니다. Create, Update 및 Delete 조작을 위해 어댑터에 전달되는 Business Object에는 하위 Business Object도 있어야 합니다. 카디널리티는 단순 속성에 대해서는 단일(1)이고 컨테이너 속성에 대해서는 다중(n)이 됩니다. Business Object에 필수 속성에 대해 올바른 값 또는 기본값이 없는 경우 어댑터는 Create 조작을 실패하게 만듭니다. 또한 이 오브젝트에 대한 데이터베이스에서 검색 시 데이터를 사용할 수 없는 경우에도 실패합니다.
특별	
유형	단순 속성인 경우 속성의 유형(정수, 문자열, 날짜, 부울, 더블 또는 부동소수점 수 등) 또는 하위 Business Object인 경우 Business Object의 유형이 됩니다. 어댑터에서 지원되지 않는 속성 유형을 만나는 경우 어댑터는 값에 따옴표를 붙이고 해당 값을 문자 데이터로 취급합니다.

### 관련 개념

8 페이지의 『Business Object 구조』

각 Business Object는 데이터베이스 테이블 또는 보기에 해당하며 해당 오브젝트의 각 단순 속성은 해당 테이블 또는 보기의 열에 해당합니다.

### 지원되는 조작

어댑터는 인바운드 및 아웃바운드 조작을 수행합니다. 사후 이미지 및 델타 지원만 아웃바운드 조작에 제공됩니다. 어댑터는 인바운드 조작에 대해 사후 이미지 지원만 제공합니다. 지원되는 조작은 다음과 같습니다.

### 아웃바운드 조작

어댑터는 Business Object에서 전달한 정보의 양 및 목적에 관련된 두 가지 Business Object 양식을 지원합니다(사후 이미지 및 델타). 사후 이미지는 Business Object에 모든 변경 내용이 작성된 이후의 Business Object 상태입니다. 델타는 변경될 키 값과 데이터만 포함하는 Update 조작에 사용된 Business Object입니다. 어댑터에서는 사후 조작에 대해 사후 이미지와 델타 지원을 모두 제공합니다.

- Create
- Update
- Delete
- ApplyChanges
- Retrieve

- RetrieveAll

Verb는 사후 이미지 Business Object에 대해서만 정의할 수 있습니다. Verb는 Business Object의 상태를 반영하는 반면, 조작은 어댑터에서 수행할 조작을 반영합니다. 사후 이미지에 대해 지원되는 최상위 Verb는 다음과 같습니다.

- Create
- Update
- Delete
- UpdateWithDelete

## 인바운드 조작

어댑터는 인바운드 조작에 대해 사후 이미지 지원만 제공합니다. 어댑터에서 갱신에 대한 Business Object를 수신할 때는 해당 Business Object가 갱신 이후 원하는 데이터 상태를 나타내는 것으로 가정할 수 있습니다. 지원되는 조작은 다음과 같습니다.

- Create
- Update
- Delete

## 트랜잭션 관리

Adapter for JDBC는 로컬 및 XA 트랜잭션을 모두 지원합니다.

어댑터에서 트랜잭션은 백엔드 데이터베이스 또는 EIS(Enterprise Information System)와의 고립된 상호작용입니다. 트랜잭션은 데이터베이스에서 원자 단위로 수행되는 여러 조작으로 구성될 수 있습니다. 이러한 조작은 데이터베이스의 다른 클라이언트 응용프로그램에서 동시에 발생한 조작의 영향을 받지 않습니다.

Adapter for JDBC는 백엔드 데이터베이스가 트랜잭션을 지원하는 경우에만 트랜잭션을 지원합니다. 지원되는 트랜잭션 유형은 로컬 및 XA 트랜잭션입니다.

- 로컬 트랜잭션에서 주어진 클라이언트 응용프로그램은 데이터베이스에 대해 트랜잭션의 시작 및 끝을 정의합니다. 이것은 one-phase-commit 프로토콜을 사용합니다.
- XA 트랜잭션에서 트랜잭션은 여러 이기종 데이터베이스를 스캔합니다. 이것은 글로벌 또는 two-phase-commit 프로토콜을 사용합니다.

어댑터는 IBM DB2 및 Oracle 데이터베이스용 XA 트랜잭션을 지원합니다.

XA 트랜잭션에 대한 XADataSourceName 및 XADatabaseName 등록 정보를 사용하십시오. 이 등록 정보에 대한 세부 사항은 『참조』 섹션의 『연결 팩토리 등록 정보』에 있습니다.

주: DB2 데이터베이스를 사용 중이면 XADatabaseName 등록 정보를 구성해야 합니다.

『참조』 섹션에는 로컬 및 XA 트랜잭션에서 사용되는 등록 정보의 예제 값이 들어 있습니다.

#### 관련 참조

80 페이지의 『J2C 연결 팩토리 등록 정보』

J2C 연결 팩토리 등록 정보는 대상 EIS(Enterprise Information System) 인스턴스를 구성하는 데 사용됩니다. 이러한 등록 정보는 아웃바운드 처리에 영향을 주며 J2EE 커넥터 구조 스펙의 ManagedConnectionFactory 인터페이스에 해당합니다.

77 페이지의 『참조』

이 섹션에서는 구성, 연결 및 오브젝트 선택을 위한 등록 정보에 대해 설명합니다.

## 인바운드 지원

Adapter for JDBC는 비동기 이벤트 전달에 대한 인바운드 이벤트 관리를 지원합니다.

비동기 이벤트 전달은 이벤트 스토어라는 이벤트 테이블과 스테이징 테이블(이벤트 분배 테이블)을 사용하여 달성됩니다. 사용자 테이블이 변경되면 응용프로그램은 EIS의 이벤트 테이블에 입력합니다. 사용자 테이블 갱신에 해당하는 이벤트 테이블에 이벤트를 기록하는 사용자 테이블에 트리거를 넣으면 갱신됩니다.

스테이징 테이블은 WebSphere Application Server, 버전 6.0에 들어 있는 Cloudscape(TM)와 같은 XA 준수 데이터베이스의 파트로서 배치됩니다. EIS에서 이벤트가 검색될 때 그 이벤트에 대한 참조가 각 활성 엔드포인트에 대해 스테이징 테이블에 쓰여집니다. 스테이징 테이블의 각 이벤트는 응용프로그램 서버가 제어하는 고유 XA 트랜잭션의 파트로서 해당 엔드포인트에 전달됩니다. 엔드포인트가 이벤트를 소비하고 트랜잭션이 확약되면 스테이징 테이블의 이벤트 참조는 제거되며 엔드포인트만 남습니다.

아래 테이블은 이벤트 테이블을 설명합니다.

#### 이벤트 테이블

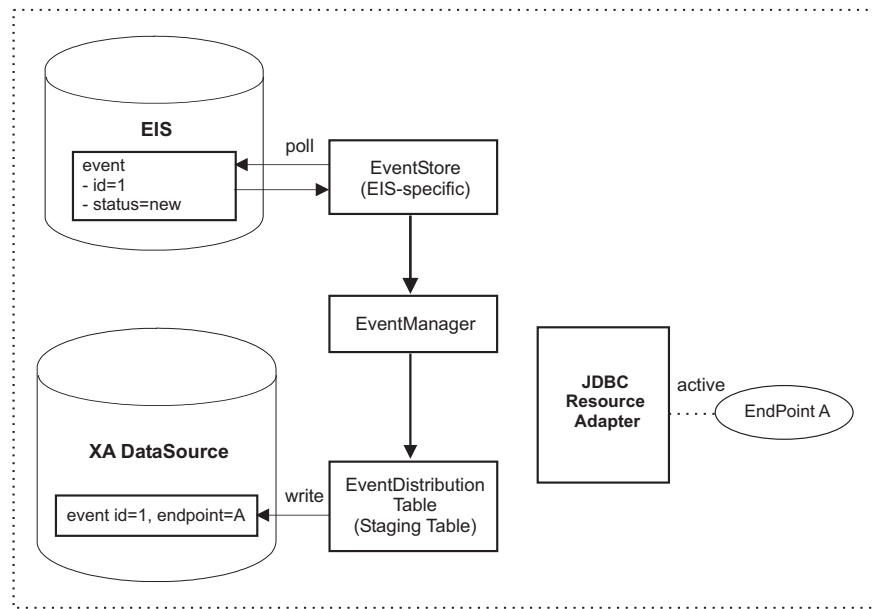
이름	유형	설명
event_id	숫자	테이블의 1차 키인 고유 이벤트 ID
object_key	문자열	Business Object에 대한 키를 포함하는 분리된 문자열(널(null)이 아님)
object_name	문자열	Business Object 이름(널(null)이 아님)
object_function	문자열	이벤트에 해당하는 조작(Delete, Create, Update 등(널(null)이 아님))
event_priority	숫자	널(null)이 아님
event_time	날짜	이벤트가 생성된 날짜/시간

이름	유형	설명
event_status	숫자	진행 중, 성공, 실패 등(널(null)이 아님)
event_comment	문자열	설명

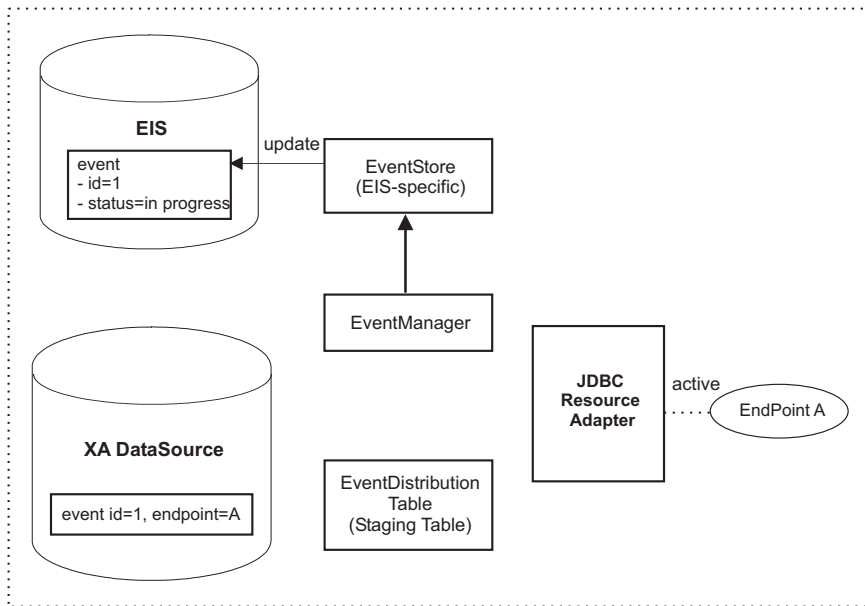
### Adapter for JDBC에서 이벤트 처리

Adapter for JDBC에 의한 이벤트 처리의 완료 플로우를 다음 네 개의 그림에 표시된 네 단계로 설명됩니다.

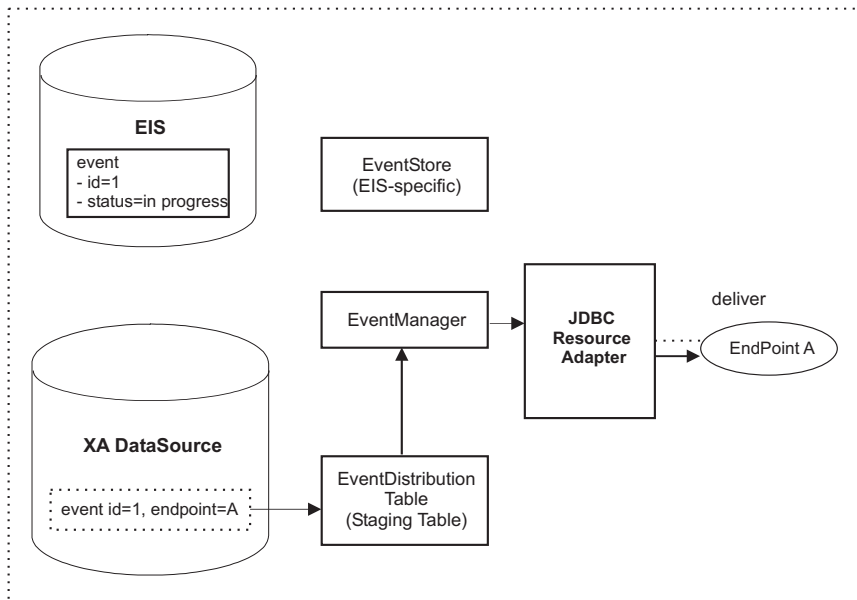
다음 그림은 1단계를 표시하며, 여기에서 EIS는 이벤트를 찾고 각 활성 엔드포인트에 대해 스테이징 테이블에 레코드가 추가됩니다.



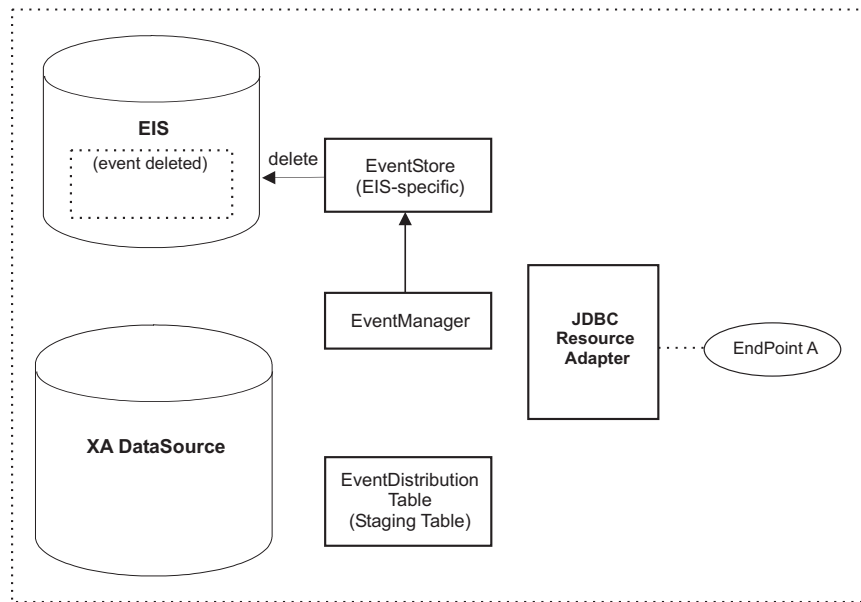
다음 그림은 2단계를 나타내며 여기에서 EIS의 이벤트는 진행 중으로 표시됩니다.



다음 그림은 3단계를 표시하며 여기에서는 엔드포인트가 요구하는 전달에 대한 안내로서 스테이징 테이블을 사용하여 모든 활성 엔드포인트에 이벤트가 발행됩니다. 동시에, 스테이징 테이블에서 각 엔드포인트 및 특정 레코드를 포함하는 각 XA 트랜잭션의 파트로서 스테이징 테이블에서 이벤트가 삭제됩니다. 실제로 엔드포인트는 이벤트의 수령 및 처리 기간에서 다른 것들과 분리됩니다.



4단계에서 모든 엔드포인트가 이벤트를 수신하면 스테이징 테이블에 남아 있는 참조가 부족하므로 원래 이벤트는 EIS에서 삭제됩니다. 1단계 - 4단계는 모든 이벤트에서 반복됩니다.



## 아웃바운드 조사를 위한 Business Object 처리

이 주제 세트에서는 Business Object를 작성, 검색, 갱신 또는 삭제할 때 어댑터에서 수행하는 단계를 설명합니다. 어댑터에서는 계층 구조 Business Object를 반복적으로 처리합니다. 즉, 개별 Business Object를 모두 처리할 때까지 각 하위 Business Object에 대해 같은 단계를 수행합니다.

## Business Object 비교

어댑터는 Business Object Verb 조사의 여러 위치에서 두 개의 Business Object를 비교하고 이들이 동일한지 확인합니다. 예를 들어, Update 조작 중 어댑터는 특정 Business Object가 Business Object의 배열에 존재하는지 여부를 확인합니다. 이를 확인하기 위해 Business Object를 배열 내 각 Business Object와 비교합니다. 두 Business Object가 같으려면 다음 두 조건이 충족되어야 합니다.

- 비교 중인 Business Object의 유형이 같아야 합니다. 예를 들어, 그 속성이 모두 같다고 해도 고객 Business Object와 담당자 Business Object는 같은 것으로 간주되지 않습니다.
- 두 Business Object의 해당되는 키 속성에는 모두 동일한 값이 있어야 합니다. 두 Business Object 모두에서 키 속성이 설정되지 않으면 어댑터는 이들을 동일한 것으로 간주합니다. 그러나 한 Business Object에서는 키 속성이 설정되고 다른 하나에서는 설정되지 않으면 Business Object는 동일하지 않습니다.

## Create 조작

계층 구조 Business Object를 제공했을 때 Create 조작에는 Business Object의 반복적인 이동이 필요하며 그에 따라 오브젝트를 이동시킬 때 각 테이블에 해당되는 행이 작성됩니다.

자세한 내용은 다음과 같습니다.

1. 소유권이 있는 단일 카디널리티 하위 Business Object를 데이터베이스에 반복적으로 삽입합니다. 즉, 어댑터는 하위와 그 하위에 포함된 하위와 모든 하위 Business Object를 작성합니다.

Business Object 정의에 속성이 단일 카디널리티가 있는 하위 Business Object를 나타내는 것과 속성이 비어 있음으로 지정된 경우, 어댑터는 그 속성을 무시합니다. 그러나 Business Object 정의에 속성이 하위를 나타내는 것이 필요한데 없는 경우에는 어댑터에서 오류를 리턴하고 처리를 중지합니다.

2. 소유권이 없는 단일 카디널리티 하위 Business Object의 존재를 검색하고 확인합니다. Retrieve 조작이 성공하지 않고 하위가 데이터베이스에 존재하지 않음을 나타내는 경우 어댑터에서는 오류를 리턴하고 처리를 중지합니다. Retrieve 조작이 성공할 때 어댑터는 하위 Business Object를 반복적으로 갱신합니다.

주: 하위 Business Object가 응용프로그램 데이터베이스에 존재할 때 이 접근법이 제대로 작동하게 하려면 하위 Business Object의 1차 키 속성이 Create 조작에서 바르게 교차 참조되어야 합니다. 하위 Business Object가 응용프로그램 데이터베이스에 존재하지 않는 경우에는 1차 키 속성이 설정되지 않아야 합니다.

3. 데이터베이스에 최상위 레벨 Business Object를 다음과 같이 삽입합니다.
  - a. 단일 카디널리티를 사용하여 표시된 해당 하위 Business Object의 1차 키 값으로 각 외부 키 값을 설정합니다. 하위 작성 중 데이터베이스 시퀀스나 카운터 또는 데이터베이스 자체에서 하위 Business Object의 값을 설정할 수 있기 때문에 이 단계에서는 어댑터에서 데이터베이스에 상위를 삽입하기 전에 상위의 외부 키 값이 올바른지 확인합니다.
  - b. 데이터베이스에서 자동으로 설정한 각 속성에 대해 새로운 고유 ID 값을 생성합니다. 데이터베이스 시퀀스 또는 카운터의 이름이 속성의 응용프로그램 특정 정보에 저장됩니다. 속성에 연관된 데이터베이스 시퀀스 또는 카운터가 있으면 어댑터에서 생성한 값이 응용프로그램 서버에서 전달한 값 대신 사용됩니다. 데이터베이스 시퀀스 또는 카운터를 지정하는 방법에 대한 자세한 내용은 『단순 속성에 대한 응용프로그램 특정 정보』의 UID=AUTO를 참조하십시오.
  - c. 데이터베이스에 최상위 레벨 Business Object를 삽입합니다.
4. 다중 카디널리티 하위 Business Object마다 다음과 같이 처리합니다.
  - a. 상위에서 해당되는 1차 키 속성의 값을 참조하도록 각 하위에 외부 키 값을 설정합니다. 상위 작성 중 상위의 1차 키 값이 생성되었을 수 있기 때문에 어댑터에서 데이터베이스에 하위를 삽입하기 전에 각 하위의 외부 키가 올바른지 확인합니다.
  - b. 각 카디널리티 하위 Business Object를 데이터베이스에 삽입합니다.



## Retrieve 조작

이 주제에서는 계층 구조 Business Object를 검색하기 위해 어댑터에서 수행하는 단계를 설명합니다.

어댑터에서는 다음과 같이 Retrieve 조작을 수행합니다.

1. 수신한 최상위 레벨의 Business Object에서 모든 하위 Business Object를 제거합니다. 다시 말해서 하위가 없는 최상위 레벨의 Business Object 사본을 만드는 것입니다.
2. 데이터베이스에서 최상위 레벨 Business Object를 검색합니다.
  - 검색에서 행을 하나 리턴하면 어댑터가 계속 처리를 진행합니다.
  - 검색에서 행을 리턴하지 않으면 최상위 레벨 Business Object가 데이터베이스에 존재하지 않는다는 뜻이므로 어댑터는 오류 RecordNotFoundException을 리턴합니다.
  - 검색에서 행을 두 개 이상 리턴하면 어댑터는 오류를 리턴합니다.
3. 모든 다중 카디널리티 하위 Business Object를 반복적으로 검색합니다.

주: 어댑터는 Business Object의 배열을 입력할 때 고유성을 강제하지 않습니다. 이러한 고유성을 확보하는 것이 데이터베이스의 역할입니다. 데이터베이스에서 중복 하위 Business Object를 리턴하면 어댑터에서는 중복 하위를 리턴합니다.

4. 하위 Business Object에 소유권이 있는지 여부에 관계 없이 단일 카디널리티 하위마다 반복적으로 검색합니다.

주: 모든 단일 카디널리티 하위 Business Object는 상위 Business Object가 처리되기 전에 Business Object에서 발생함을 기본으로 하여 처리됩니다. 하위 오브젝트 소유권과 비소유권 때문에 처리 순서가 결정되는 것은 아니지만 처리 유형은 그에 따라 결정됩니다.

## RetrieveAll 조작

이 조작을 사용하면 어댑터에서 데이터베이스에서 Business Object 배열을 검색할 수 있습니다.

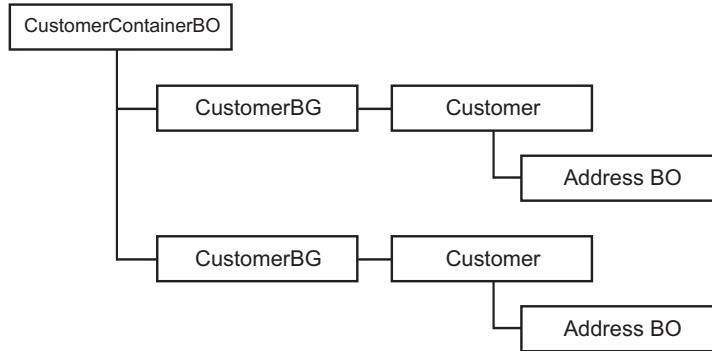
수신 Business Object에 채워진 키 및 키가 아닌 속성은 모두 선택 기준을 결정합니다. 어댑터에서는 선택된 속성에 따라 데이터베이스에서 최상위 레벨의 Business Object에 대해 여러 행을 검색할 수 있습니다. 수신 Business Object에 채워진 속성이 없으면 모든 행이 데이터베이스의 각각의 테이블에서 검색됩니다.

어댑터에서 다음 단계를 수행하여 Business Object 배열을 검색합니다.

1. 데이터베이스에서 검색된 행마다 어댑터에서는 최상위 레벨의 비즈니스 그래프를 구성하고, 검색된 행을 모두 사용하여 비즈니스 그래프의 컨테이너를 작성합니다. 컨테이너 비즈니스 그래프의 이름은 *BOName + ContainerBG*입니다.

2. 어댑터에서는 Retrieve 조작을 사용하여 컨테이너에서 각 비즈니스 그래프를 검색합니다.

다음 그림은 RetrieveAll 조작에 리턴된 Business Object의 구조를 설명합니다.



다음 오류는 RetrieveAll 조작의 결과로 나타납니다.

- 입력 오브젝트에 채워진 Business Object가 EIS에 존재하지 않는 경우, 어댑터가 오류 RecordNotFoundException을 리턴합니다.
- EIS의 히트 수가 상호 작용 스펙에 정의된 ResultSetLimit의 값을 초과하는 경우, 어댑터에서는 오류 MatchesExceededLimitException을 리턴합니다. 한계를 늘리거나 검색을 적합하게 다시 정의하기 위해 MatchCount 등록 정보는 어댑터에서 EIS에 보유한 실제 히트 수를 포함할 수 있습니다.

주: ResultSetLimit가 아주 큰 수로 설정되면 리턴되는 Business Object의 크기와 수에 따라 메모리 부족과 관련된 문제가 발생할 수 있습니다.

- EIS에서 복구할 수 없는 오류를 보고하는 경우, 어댑터에서는 오류 EISSystemException을 리턴합니다.

## Update 조작

Update 조작은 최상위 레벨의 수신 Business Object에 지정된 1차 키를 사용하여 데이터베이스에서 검색된 Business Object와 수신 Business Object를 비교하는 방법으로 수행됩니다.

계층 구조 Business Object를 갱신할 때 다음 단계를 수행하십시오.

1. 소스 Business Object의 1차 키 값을 사용하여 데이터베이스에서 해당하는 엔티티를 검색할 수 있습니다. 검색된 Business Object는 데이터베이스에서 현재 데이터 상태에 대한 정확한 표현입니다.

검색이 실패하면 최상위 레벨 Business Object가 데이터베이스에 존재하지 않는다는 뜻이므로 어댑터는 RecordNotFoundException과 갱신 실패를 리턴합니다.

검색이 성공하면 어댑터에서 검색된 Business Object를 소스 Business Object에 비교하여 데이터베이스에서 어떤 하위 Business Object에 변경이 필요한지 확인합니다.

다. 그러나 어댑터에서 소스 Business Object의 단순 속성의 값을 검색된 Business Object의 값과 비교하지 않습니다. 어댑터에서는 모든 키 아닌 단순 속성의 값을 갱신합니다.

최상위 레벨 Business Object의 단순 속성이 모두 키를 나타내는 경우에는 어댑터에서 해당 최상위 레벨 Business Object에 대한 갱신 조회를 생성할 수 없습니다. 이 경우 어댑터는 경고를 로그하고 2단계로 진행됩니다.

2. 최상위 레벨 Business Object의 모든 단일 카디널리티 하위를 반복적으로 갱신합니다.

Business Object 정의에 속성이 하위 Business Object를 나타냄이 필요한 경우 해당 하위는 소스 Business Object와 검색된 Business Object에 모두 존재해야 합니다. 그렇지 않으면 Update 조작이 실패하고 어댑터에서 오류를 리턴합니다.

어댑터는 다음 방법 중 하나로 소유권이 있는 단일 카디널리티 하위를 처리합니다.

- 하위가 소스 및 검색된 Business Object에 있는 경우, 어댑터에서는 데이터베이스에서 기존 하위를 갱신하는 대신, 기존 하위를 삭제하고 새 하위를 작성합니다.
- 하위가 소스 Business Object에 있지만 검색된 Business Object에는 없는 경우에는 데이터베이스에서 해당 하위를 반복적으로 작성합니다.
- 하위가 검색된 Business Object에 있지만 소스 Business Object에는 없는 경우에는 데이터베이스에서 해당 하위를 반복적으로 삭제합니다.

소유권이 없는 단일 카디널리티 하위의 경우, 어댑터에서는 소스 Business Object에 나타난 모든 하위를 데이터베이스에서 검색하도록 시도합니다. 하위를 검색한 경우, 어댑터는 하위 Business Object를 표시하지만 갱신하지는 않습니다. 이는 어댑터에서 소유권이 없는 단일 카디널리티 하위를 수정하지 않기 때문입니다.

3. 검색된 Business Object의 모든 단순 속성을 갱신하지만 소스 비즈니스의 해당 속성이 지정되지 않은 경우는 제외됩니다.

갱신 중인 Business Object가 고유해야 하므로 어댑터에서는 결과로 행이 하나만 처리되는지 확인합니다. 행이 두 개 이상 리턴되면 오류가 발생합니다.

4. 검색된 Business Object의 다중 카디널리티 하위마다 다음 방법 중 하나를 사용하여 처리합니다.

- 하위가 소스 및 검색된 Business Object의 배열에 있는 경우에는 어댑터가 데이터베이스에서 반복적으로 이를 갱신합니다.
- 하위가 소스 배열에 있지만 검색된 Business Object 배열에는 없는 경우에는 어댑터는 데이터베이스에서 해당 하위를 반복적으로 작성합니다.
- 검색된 Business Object의 배열에는 하위가 있지만 소스 배열에는 없는 경우, 상위에 있는 하위에 KeepRelationship이 있음을 나타내는 속성에 대한 응용프로

그럼 특정 정보가 true로 설정되지 않으면 어댑터가 데이터베이스에서 해당 하위를 반복적으로 삭제합니다. 설정된 경우에는 어댑터가 데이터베이스에서 하위를 삭제하지 않습니다.

## UpdateWithDelete 조작

이 주제에서는 Update 조작의 특별 양식에 대해 설명합니다.

이 조작을 사용하면 Update 조작보다 더 나은 성능을 제공할 수 있습니다. UpdateWithDelete에는 ChangeSummary가 필요합니다. ChangeSummary에는 Business Object 레벨의 작성 및 삭제가 포함되어야 합니다. 이렇게 하면 어댑터는 EIS에서 기존 엔티티를 검색이 필요한 오버헤드 및 비교를 수행할 필요 없이 조작을 수행할 수 있는데, 이는 ChangeSummary가 수행해야 할 사항을 나타내기 때문입니다.

ChangeSummary가 비어 있으면 어댑터에서 요청에 대한 조치를 수행하지 않습니다.

## DeltaUpdate 조작

InteractionSpec의 조작이 ApplyChanges이고 이 Verb가 비즈니스 그래프에 있지 않은 경우 어댑터는 DeltaUpdate 조작을 수행합니다. 어댑터는 입력 계층 구조에서 각 Business Object에 대한 조작을 식별하도록 ChangeSummary를 조사하고 식별된 조작을 수행합니다.

DeltaUpdate 조작은 Update 조작과는 다르며 그 내용은 다음과 같습니다.

- DeltaUpdate 조작에서는 갱신 전에 Retrieve 조작이 발생하지 않습니다.
- 수신 Business Object와 데이터베이스의 Business Object를 비교하지 않습니다.
- 모든 하위는 각 하위 오브젝트에서 Verb 세트를 기본으로 하여 처리됩니다. 하위에 Verb 세트가 없으면 어댑터에서 오류를 리턴합니다.

어댑터에서는 DeltaUpdate로 계층 구조 Business Object를 갱신할 때 다음 단계를 수행합니다. ChangeSummary의 오브젝트 변경만 처리합니다.

1. 상위 오브젝트의 모든 단일 카디널리티 하위를 반복적으로 처리합니다. 하위가 Business Object 스펙에서 필수로 표시되면 인바운드 오브젝트에 나타나야 합니다. 그렇지 않으면 DeltaUpdate 조작이 실패하고 어댑터에서 오류를 리턴합니다.
2. 단일 카디널리티 하위의 속성을 해당 하위 값으로 참조하도록 상위의 모든 외부 키 값을 설정합니다. 이는 단일 카디널리티 하위가 이전 단계 중 데이터베이스에 추가되어서 새 시퀀스 값이 생성되었기 때문입니다.
3. SQL Update 명령문 또는 저장 프로시저를 사용하여 처리 중인 현재 오브젝트를 갱신합니다. 개별 Business Object의 단순 속성이 모두 갱신됩니다. 어댑터에서는 갱신 명령문에 추가되어야 할 속성이 어떤 것인지 판별하기 위해 등록 정보 레벨의 변경 사항을 사용하지 않으며 모두 갱신됩니다. 갱신 중인 오브젝트가 고유해야 하므로 어댑터에서는 결과로 행이 하나만 처리되는지 확인합니다. 행이 두 개 이상 처리되면 오류가 리턴됩니다.

4. 상위 속성을 참조하는 현재 오브젝트의 모든 카디널리티 N 하위에서 모든 외부 키 값을 해당하는 상위 값으로 설정합니다. 보통 이러한 값은 데이터 맵핑 중 이미 상호 참조되지만 이 경우는 카디널리티 N 컨테이너의 새 하위에 대해서는 해당하지 않습니다. 이 단계에서는 하위가 갱신되기 전에 모든 카디널리티 N 하위의 외부 키 값이 올바른지 확인합니다.
5. 현재 오브젝트의 모든 카디널리티 N 컨테이너를 갱신합니다.

하위 오브젝트가 처리될 때, 각 하위의 Verb가 사용되고 적절한 조작이 수행됩니다. DeltaUpdate의 하위에서 허용된 조작은 Create, Delete 및 Update입니다.

- Create Verb가 하위에서 발견되면 해당 하위가 소유권 하위인 경우에 데이터베이스에 작성됩니다. 비소유권 하위는 데이터베이스에서 그 존재의 유효성을 검증하도록 검색됩니다.
- Delete Verb가 하위에서 발견되면 해당 하위는 삭제됩니다.
- Update Verb가 하위에서 발견되면 해당 하위는 데이터베이스에서 갱신됩니다.

### Delete 조작

Delete 조작은 수신 Business Object를 제거한 다음 데이터베이스에서 전체 Business Object를 검색하는 방법으로 수행됩니다. 그리고 나서 계층 구조의 각 Business Object에 반복적으로 적용됩니다.

Delete 조작은 오브젝트의 응용프로그램 특정 정보에 있는 StatusColumnName 값에 따라 실제 및 논리 Delete를 지원합니다. StatusColumnName 값이 정의되어 있으면 어댑터에서는 논리 Delete 조작을 수행합니다. StatusColumnName 값이 정의되어 있지 않으면 실제 Delete 조작이 수행됩니다.

#### 실제 삭제의 경우:

- 모든 다중 카디널리티 하위 Business Object를 반복하여 삭제합니다.
- 최상위 레벨의 Business Object를 삭제합니다.
- 소유권이 있는 모든 단일 카디널리티 하위 Business Object를 반복적으로 삭제합니다.

#### 논리 Delete의 경우:

- Business Object의 상태 속성을 Business Object의 응용프로그램 특정 정보에서 지정한 값으로 설정하도록 Update를 실행합니다. 어댑터에서는 하나의 데이터베이스 행만 갱신되고 그 외의 경우에는 오류가 리턴됨을 확인합니다.
- 소유권이 있는 모든 단일 카디널리티 하위 및 모든 다중 카디널리티 하위를 논리적으로 반복하여 삭제합니다. 소유권이 없는 단일 카디널리티 하위는 삭제하지 않습니다.

## ApplyChanges 조작

이것은 기본적으로 catch-all 조작으로서 Business Object에서 어댑터에 따라 처리될 Create, Update 또는 Delete 조작이 필요하도록 합니다.

최상위 레벨의 Verb가 Business Object에 존재하는 경우 해당 Business Object는 사후 이미지로서 처리됩니다. 최상위 레벨의 Verb가 Business Object에 존재하지 않는 경우에는 ChangeSummary가 처리됩니다.

## 응용프로그램 특정 정보

Business Object 정의의 응용프로그램 특정 정보는 어댑터에 Business Object를 처리하는 방법에 관한 응용프로그램 종속 지시사항을 제공합니다. 어댑터에서는 비즈니스의 속성 또는 Verb에서나 Business Object 자체에서 응용프로그램 특정 정보를 구문 분석하여 Create, Update, Retrieve 및 Delete 조작에 대한 조회를 생성합니다.

어댑터에서는 이 Business Object의 응용프로그램 특정 정보를 캐시에 저장하고 이 정보를 사용하여 모든 Verb에 대한 조회를 빌드합니다.

확장되거나 수정된 응용프로그램 특정 Business Object에서 Business Object 정의의 응용프로그램 특정 정보는 어댑터에서 예상하는 구문과 일치해야 합니다.

### 관련 태스크

47 페이지의 『데이터베이스 오브젝트 조회』

연결 등록 정보를 구성한 다음 데이터베이스 오브젝트 조회를 실행할 수 있습니다. 메타데이터 트리 구조를 찾아보면 EIS(Enterprise Information System)에서 오브젝트의 구조를 이해할 수 있고 서비스 설명에 필요한 오브젝트를 선택할 수 있습니다.

## Business Object 이름 지정 규칙

이 주제에서는 Business Object의 이름 지정 규칙을 설명합니다.

Business Object 이름은 고객 또는 주소와 같이 나타내는 구조를 반영해야 합니다. 이름은 대부분 EIS에서 주어진 이름에 따라 Enterprise Metadata Discovery의 메타데이터 가져오기 프로세스에서 추출됩니다. Business Object 이름은 공백이나 밑줄 등의 분리문자없이, 각 단어의 첫 문자는 대문자로 표시되는 카멜 문자로 변환되어야 합니다. 예를 들어, ORDER\_LINE\_ITEM은 OrderLineItem으로 변환됩니다.

상위 Business Object 그래프의 이름은 Business Object 뒤에 BG를 붙이면 됩니다. 예를 들어, Customer Business Object의 이름은 CustomerBG가 됩니다.

Business Object 이름에는 어댑터 또는 데이터베이스에 대한 기능 값이 없습니다.

Business Object는 데이터베이스 특정 메타데이터를 운반합니다. 응용프로그램 특정 및 일반이라는 두 가지 유형의 Business Object를 구별하는 데 도움이 되도록 JDBC 또

는 %AppName% 등의 접두부를 가질 수 있습니다. 이름의 나머지는 Business Object 가 나타내는 테이블 또는 저장 프로시저를 설명합니다. 예를 들어, Business Object 정의가 HR(Human Resources) 등의 데이터베이스 응용프로그램에 있는 Employee Table 을 위해 생성된 경우 Business Object 이름은 HREmployee가 됩니다.

### Business Object 레벨의 응용프로그램 특정 정보

Business Object 레벨의 응용프로그램 특정 정보를 사용하여 해당하는 데이터베이스 테이블의 이름을 지정하고 실제 또는 논리 Delete 조작을 수행하는데 필요한 정보를 제공할 수 있습니다.

Business Object 레벨에서 응용프로그램 특정 정보 형식은 jdbcasi.xsd 스키마 정의에서 정의된 xml로 구성됩니다.

- *TableName*은 Business Object와 연관된 데이터베이스 테이블을 식별합니다.
- *StatusColumnName*은 논리 Delete 조작을 수행하는 데 사용된 데이터베이스 열의 이름입니다.
- *StatusValue*는 비활성 또는 삭제된 Business Object를 나타내는 값입니다.

예를 들어, 고객 Business Object에 Business Object 응용프로그램 특정 정보에 대해 다음과 같이 지정된 값이 있다고 가정합니다.

```
<jdbcasi:TableName>customer</jdbcasi:TableName>
<jdbcasi>StatusColumnName>status</jdbcasi>StatusColumnName>
<jdbcasi::StatusValue>deleted</jdbcasi>StatusValue>
```

어댑터에서 고객을 삭제하라는 요청을 받는다고 가정합니다. 이러한 요청에 따라 어댑터는 다음 SQL 문을 실행하게 됩니다.

```
UPDATE customer set status = 'deleted' where pkey = . . . .
```

StatusColumnName이 포함되지 않거나 그에 대한 값이 지정되어 있지 않으면 어댑터는 데이터베이스에서 실제로 Business Object를 삭제합니다. 즉, Business Object에 응용프로그램 특정 정보의 StatusColumnName 매개변수가 포함되어 있으면, 어댑터는 논리 Delete 조작을 수행합니다. Business Object에 응용프로그램 특정 정보의 StatusColumnName 매개변수가 없으면 어댑터는 실제 Delete 조작을 수행합니다.

Update 및 Delete 조작은 모두 StatusColumnName 등록 정보의 값을 사용할 수 있습니다.

- 하위 데이터를 논리적으로 삭제하기 위해 어댑터에서는 StatusColumnName 매개변수를 사용하여 상태 열의 이름 및 상태 값의 텍스트를 얻을 수 있습니다. 자세한 내용은 『Update 조작』을 참조하십시오.
- Delete 조작을 수행할 때는 어댑터에서 StatusColumnName 등록 정보의 값을 사용하여 전체 Business Object를 실제로 삭제할 것인지, 논리적으로 삭제할 것인지 결정할 수 있습니다. StatusColumnName 매개변수에 값이 있으면 논리 Delete 조작

이 수행됩니다. StatusColumnName 매개변수에 값이 없으면 어댑터는 실제 Delete 조작을 수행합니다. 자세한 내용은 『Delete 조작』을 참조하십시오.

양방향 언어와 함께 사용할 수 있는 ASI의 매개변수는 TableName 및 StatusColumnName입니다. 이러한 매개변수의 형식은 BiDi.Metadata 등록 정보에 설정된 속성을 기본으로 하여 변환됩니다.

## Verb 응용프로그램 특정 정보

어댑터는 Business Object에 지정된 대로 SQL 문의 그룹인 저장 프로시저 또는 SQL 조화를 사용하여 데이터베이스 테이블을 갱신합니다. 이 섹션에는 저장 프로시저와 저장 프로시저 정의의 요소를 설명하며, 저장 프로시저 정의의 샘플이 포함되어 있습니다.

### 관련 태스크

39 페이지의 『오브젝트 선택 및 생성』

Business Object를 생성하려면 데이터베이스 오브젝트 노드를 선택하십시오. 그러면 엔터프라이즈 서비스 찾기 마법사는 선택된 노드의 오브젝트에 대해 Business Object를 생성합니다.

### 관련 참조

42 페이지의 『Business Object 속성 및 ASI』

## 저장 프로시저 개요:

저장 프로시저는 논리 단위를 양식화하고 특정 태스크를 수행하는 SQL 문의 그룹입니다. 저장 프로시저는 어댑터에 대한 조회 및 조작 세트를 캡슐화하여 데이터베이스 서버의 오브젝트에서 실행되게 합니다.

어댑터는 Select, Update, Retrieve, Delete 또는 RetrieveAll 조작에 대해 단순 SQL 문을 사용할 수 있습니다. SQL 문에 대한 열 이름은 속성의 AppSpecificInfo 등록 정보에서 파생됩니다. WHERE 절은 Business Object에 지정된 키 값을 사용하여 구성됩니다. 각 조회는 보기에 게시되지 않은 경우 하나의 테이블만 스캔합니다.

어댑터는 다음과 같은 상황에서 저장 프로시저를 호출합니다.

- Business Object를 처리하기 전에 준비 조작 프로세스를 수행합니다.
- Business Object를 처리한 후에 사후 조작 프로세스를 수행합니다.
- 단순 Create, Update, Delete, Retrieve 또는 RetrieveAll 문을 사용하는 대신 Business Object에서 조작 세트를 수행합니다.

계층 구조 Business Object를 처리할 때 어댑터에서는 저장 프로시저를 사용하여 최상위 레벨 Business Object 또는 하위 Business Object 중 하나를 처리할 수 있습니다. 그러나 각 Business Object 또는 Business Object의 배열에는 자체 저장 프로시저가 있어야 합니다.

## 저장 프로시저 정의:



저장 프로시저는 Verb 레벨에 정의되어 있습니다. 각 저장 프로시저 정의는 StoredProcedureType, StoredProcedureName, ResultSet 및 Parameters라는 요소로 구성됩니다.

**StoredProcedureType**은 사용될 저장 프로시저의 유형을 정의하며, 이 유형은 저장 프로시저가 호출될 때 결정됩니다. 값은 다음과 같습니다.

- BeforeCreateSP
- AfterCreateSP
- CreateSP
- BeforeUpdateSP
- AfterUpdateSP
- UpdateSP
- BeforeDeleteSP
- AfterDeleteSP
- DeleteSP
- BeforeRetrieveSP
- AfterRetrieveSP
- RetrieveSP
- BeforeRetrieveAllSP
- AfterRetrieveAllSP
- RetrieveAllSP

주: RetrieveAll과 연관된 저장 프로시저 유형은 최상위 레벨 Business Object에 적용됩니다.

**StoredProcedureName**은 적합한 StoredProcedureType에 연관되는 저장 프로시저의 이름입니다. 이것은 양방향 언어에 대해 사용 가능합니다.

**ResultSet**는 저장 프로시저가 결과를 리턴하는지 또는 리턴하지 않는지(true|false)를 판별합니다. 결과 세트가 리턴되면 결과 세트 행에 리턴된 값을 사용하여 현재 Business Object에 대해 N 카디널리티 하위가 작성됩니다.

**Parameters**는 입력 전용(IP), 출력 전용(OP) 및 입/출력(IO)의 조합이 될 수 있습니다. Oracle 저장 프로시저의 경우, 결과 세트는 출력 매개변수로서만 리턴될 수 있습니다. 이때 매개변수 목록의 값 중 하나는 결과 세트(RS)가 됩니다. 매개변수는 양방향 언어에 대해 사용 가능합니다.

다음은 저장 프로시저 정의의 샘플입니다.

```

<jdbcasi:JDBCBusinessObjectTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
  <jdbcasi:TableName>customer</jdbcasi:TableName><jdbcasi:Operation>
    <jdbcasi:Name>Retrieve</jdbcasi:Name>
    <jdbcasi:StoredProcedures>
<jdbcasi:StoredProcedureType>RetrieveSP</jdbcasi:StoredProcedureType>
<jdbcasi:StoredProcedureName>retrieve_cust</jdbcasi:StoredProcedureName>
<jdbcasi:ResultSet>false</jdbcasi:ResultSet>
  <jdbcasi:Parameters>
    <jdbcasi:Type>IP</jdbcasi:Type>
    <jdbcasi:PropertyName>primaryKey</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  <jdbcasi:Parameters>
    <jdbcasi:Type>OP</jdbcasi:Type>
    <jdbcasi:PropertyName>custCode</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  <jdbcasi:Parameters>
    <jdbcasi:Type>OP</jdbcasi:Type>
    <jdbcasi:PropertyName>firstName</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  <jdbcasi:Parameters>
    <jdbcasi:Type>OP</jdbcasi:Type>
    <jdbcasi:PropertyName>lastName</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  </jdbcasi:StoredProcedures>
  <jdbcasi:StoredProcedures>
<jdbcasi:StoredProcedureType>AfterRetrieveSP</jdbcasi:StoredProcedureType>
<jdbcasi:StoredProcedureName>retrieve_cust</jdbcasi:StoredProcedureName>
<jdbcasi:ResultSet>false</jdbcasi:ResultSet>
  <jdbcasi:Parameters>
    <jdbcasi:Type>IP</jdbcasi:Type>
    <jdbcasi:PropertyName>primaryKey</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  <jdbcasi:Parameters>
    <jdbcasi:Type>OP</jdbcasi:Type>
    <jdbcasi:PropertyName>custCode</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  <jdbcasi:Parameters>
    <jdbcasi:Type>OP</jdbcasi:Type>
    <jdbcasi:PropertyName>firstName</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  <jdbcasi:Parameters>
    <jdbcasi:Type>OP</jdbcasi:Type>
    <jdbcasi:PropertyName>lastName</jdbcasi:PropertyName>
  </jdbcasi:Parameters>
  </jdbcasi:StoredProcedures>
</jdbcasi:Operation>
</jdbcasi:JDBCBusinessObjectTypeMetadata>

```

ReturnDummyBOForSP 등록 정보에서는 결과 세트가 true로 지정되지만 비어 있을 때도 출력 매개변수를 리턴합니다. RetrieveSP의 경우, 결과 세트가 리턴됩니다. 결과 세트가 비어 있으면 Business Object가 작성되지 않고 프로시저 호출에서 리턴한 출력 매개변수를 검색할 방법이 없습니다. ReturnDummyBOForSP가 true이면 출력의 값이 있는 더미 Business Object와 해당 속성에 있는 입/출력(I/O) 매개변수가 리턴됩니다. 이 등록 정보의 기본값은 false입니다.

### 관련 개념

8 페이지의 『Business Object 구조』

각 Business Object는 데이터베이스 테이블 또는 보기에 해당하며 해당 오브젝트의 각 단순 속성은 해당 테이블 또는 보기의 열에 해당합니다.

## 속성 응용프로그램 특정 정보

이 주제에서는 속성에 대한 응용프로그램 특정 정보를 설명하고 그 설명과 함께 지원되는 매개변수를 나열합니다.

속성에 대한 응용프로그램 특정 정보는 속성이 단순 속성인지, 또는 하위 Business Object나 하위 Business Object 배열을 나타내는 속성인지 여부에 따라 달라집니다. 하위를 나타내는 속성에 대한 응용프로그램 특정 정보는 상위-하위 관계가 하위 또는 상위에 저장되는지 여부에 따라 달라집니다.

## 단순 속성에 대한 응용프로그램 특정 정보

단순 속성의 경우 응용프로그램 특정 정보 형식은 몇 개의 매개변수와 그 값으로 구성됩니다. 속성 응용프로그램 특정 정보의 형식은 다음 .xsd 파일의 예제 섹션에 표시되어 있습니다.

```
<jdbcasi:ColumnName>pkey</jdbcasi:ColumnName>
<jdbcasi:PrimaryKey>true</jdbcasi:PrimaryKey>
<jdbcasi:FixedChar>true</jdbcasi:FixedChar>
</jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
<simpleType>
  <restriction base="string">
    <maxLength value="10"/>
  </restriction>
</simpleType>
</element>
<element name="custCode" type="string">
<annotation>
<appinfo source="WBI">
<jdbcasi:JDBCAttributeTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
  <jdbcasi:ColumnName>ccode</jdbcasi:ColumnName>
  <jdbcasi:ForeignKey>custinfoObj/custCode</jdbcasi:ForeignKey>
</jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>
<element name="firstName" type="string">
<annotation>
<appinfo source="WBI">
<jdbcasi:JDBCAttributeTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
  <jdbcasi:ColumnName>fname</jdbcasi:ColumnName>
</jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>
<element name="lastName" type="string">
<annotation>
<appinfo source="WBI">
<jdbcasi:JDBCAttributeTypeMetadata
xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
  <jdbcasi:ColumnName>lname</jdbcasi:ColumnName>
</jdbcasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>
```

컨텍스트에서 처리할 단순 속성에 대한 필수 매개변수는 열 이름입니다. 예를 들어, 다음은 열 이름만 지정하는 형식이며 여기서 ccode는 고객 코드를 나타냅니다.

```
<jdbcasi:ColumnName>ccode</jdbcasi:ColumnName>
```

양방향 언어와 함께 사용할 수 있는 속성 ASI 매개변수는 ColumnName 및 ForeignKey입니다. 이러한 매개변수의 형식은 BiDi.Metadata 등록 정보에 설정된 속성을 기본으로 하여 변환됩니다. 양방향 등록 정보에 대한 자세한 내용은 IBM developerWorks<sup>(R)</sup> 웹 사이트에 있는 양방향 지원 일반 기술지 및 어댑터 기술지를 참조하십시오.

『속성 응용프로그램 특정 정보의 매개변수』 테이블에서는 각 매개변수와 그에 대한 설명을 제공합니다.

### 속성 응용프로그램 특정 정보의 매개변수

매개변수	설명
ByteArray	true인 경우 어댑터는 데이터베이스에 2진 데이터를 쓰고 읽으며 응용프로그램 서버에 문자열로 데이터를 전송합니다. 기본적으로 값은 false입니다. 자세한 내용은 『2진 데이터로 작업』을 참조하십시오.
ColumnName	이 매개변수의 값은 이 속성의 데이터베이스 열 이름입니다. 양방향 언어에 대해 사용 가능합니다.
FixedChar	이 매개변수는 테이블의 열 유형이 VARCHAR가 아닌 CHAR일 때 속성이 고정 길이인지 여부를 지정합니다. 예를 들어, CHAR 유형인 열에 특정 속성이 연결되는 경우 어댑터에서는 데이터베이스를 조회할 때 공백 있는 속성 값을 속성의 최대 길이로 채우게 됩니다. 기본적으로 값은 false입니다.  이 매개변수는 Business Object .xsd 파일에서 수동으로 갱신되어야 합니다. 이 파일을 텍스트 모드에서 편집할 수도 있고 WID의 Business Object 편집기를 사용할 수 있습니다. 갱신된 후 .xsd 파일에 유효성 검증 오류가 없는지 확인하십시오. 이 테이블 아래의 코드 예를 참조하십시오.
ForeignKey	이 등록 정보의 값은 상위/하위 관계가 상위 Business Object 또는 하위 Business Object에 저장되는지 여부에 따라 달라집니다.  상위에 저장되는 경우에는 하위에 외부 키 (ChildBOname/ChildPropertyName)로 사용될 하위 Business Object의 유형과 속성 이름을 모두 포함하도록 값을 설정하십시오.  하위에 저장되는 경우에는 상위에 외부 키로 사용될 속성 이름만 포함하도록 값을 설정하십시오.  속성이 외부 키가 아니면 응용프로그램 특정 정보에 이 매개변수를 포함하지 마십시오.  이것은 양방향 언어에 대해 사용 가능합니다.
KeepRelationship	true인 경우에는 이 매개변수에서 Update 조작 중 하위 Business Object가 삭제되지 않도록 합니다.

매개변수	설명
OrderBy	이 매개변수에 대한 값이 지정되고 속성이 하위 Business Object에 있는 경우 어댑터는 검색 조회의 ORDER BY 절에서 속성 값을 사용합니다. 어댑터는 오름차순(ASC) 또는 내림차순(DESC)으로 하위 Business Object를 검색할 수 있습니다. 응용프로그램 특정 정보에 이 매개변수를 포함하지 않으면 어댑터에서는 검색 순서를 지정할 때 이 속성을 사용하지 않습니다.
Ownership	이 매개변수는 상위가 하위 Business Object를 소유하는지를 지정합니다. true인 경우에는 하위 Business Object에 대한 Create, Update 및 Delete 조치가 허용됩니다. false이면 이러한 갱신 중 하나도 하위 Business Object에 적용될 수 없습니다. 상위가 작성될 때 하위의 존재에 대한 유효성 검증이 수행되어 데이터베이스에서 관계 무결성이 유지되는지 확인하게 됩니다.
PrimaryKey	값이 true이면 이 속성과 연관된 열이 데이터베이스에서 해당되는 테이블의 키가 됩니다.
UID(UniqueIdentifier)	어댑터에서는 이 매개변수를 사용하여 Business Object에 대해 고유 ID를 생성합니다. 시퀀스 및 ID 열( <i>UID=AUTO SequenceName</i> )의 생성을 지원합니다. 시퀀스는 DB2 및 Oracle 데이터베이스에 대해서만 정의될 수 있습니다. ID 열은 DB2 및 Microsoft SQL Server에 대해 정의될 수 있습니다. 속성에 고유 ID가 필요하지 않은 경우에는 응용프로그램 특정 정보에 이 매개변수를 포함하지 마십시오.

## Business Object .xsd 파일에서 FixedChar 매개변수의 예

```

<element name="primaryKey">
  <annotation>
    <appinfo source="WBI">
      <jdbcasi:JDBCAttributeTypeMetadata
        xmlns:jdbcasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/metadata">
        <jdbcasi:ColumnName>pkey</jdbcasi:ColumnName>
        <jdbcasi:PrimaryKey>true</jdbcasi:PrimaryKey>
        <jdbcasi:FixedChar>true</jdbcasi:FixedChar>
      </jdbcasi:JDBCAttributeTypeMetadata>
    </appinfo>
  </annotation>
  <simpleType>
    <restriction base="string">
      <maxLength value="10"/>
    </restriction>
  </simpleType>
</element>

```

### 관련 참조

42 페이지의 『Business Object 속성 및 ASI』

---

## 어댑터 설치

이 섹션에서는 Adapter for JDBC의 설치 요구사항 및 설치 파일 구조에 대해 설명합니다.

어댑터 설치 방법에 대한 정보는 *IBM WebSphere Adapter 설치*를 참조하십시오.

### 어댑터 환경

어댑터 환경의 하드웨어 및 소프트웨어 요구사항은 온라인에서 확인할 수 있습니다.

어댑터 요구사항을 찾으려면 <http://www.ibm.com/support/docview.wss?uid=swg27006249>에 있는 『IBM WebSphere Adapter 및 IBM WebSphere Business Integration Adapter: 하드웨어 및 소프트웨어 요구사항』을 참조하십시오. WebSphere 어댑터 목록에서 해당 어댑터를 선택하십시오.

### Adapter for JDBC의 특정 설치 정보

Adapter for JDBC에는 JDBC 드라이버 및 이벤트 스토어 설정을 고려한 설치 전제조건이 있습니다.

#### JDBC 드라이버 전제조건

JDBC 드라이버 jar 파일이 클래스 경로에 있도록 JDBC 드라이버를 설치해야 합니다.

JDBC 엔터프라이즈 서비스 찾기 마법사는 엔터프라이즈 서비스 찾기 마법사의 프로젝트 안에 있습니다. JDBC 드라이버 jar 파일은 이 도구에서 JDBC 프로젝트의 클래스 경로에 있어야 사용자가 JDBC 엔터프라이즈 서비스 찾기 프로세스를 실행할 수 있습니다.

#### 이벤트 스토어 설정

인바운드 처리가 수행되기 전에 데이터베이스에 이벤트 스토어를 설정해야 합니다. 샘플 스크립트가 제공되므로 다음과 같이 DB2, Oracle 또는 SQLServer 데이터베이스에 대한 이벤트 스토어를 설치할 수 있습니다.

- WBIA\_JDBC\_EventStore\_DB2.sql
- WBIA\_JDBC\_EventStore\_Oracle.sql
- WBIA\_JDBC\_EventStore\_MSSQL.sql

필요에 따라 사용자 테이블에 트리거를 설치해야 사용자 테이블에 대한 변경이 이벤트 스토어에 저장되는 이벤트를 자동으로 생성할 수 있습니다.

### 설치된 파일 구조

어댑터를 설치하면 설치된 파일 및 디렉토리를 볼 수 있는데 루트로 모두 설치 디렉토리가 있습니다.

예를 들어, 어댑터의 설치 디렉토리가 c:\WebSphereBI이면 CWYBC\_JDBC.rar 파일의 절대 경로는 다음과 같습니다. c:\WebSphereBI.\wadapter\wJDBC\wdeploy\wCWYBC\_JDBC.rar

어댑터 RAR 파일에는 어댑터와 엔터프라이즈 서비스 찾기 도구 파일이 있습니다.

UNIX<sup>(R)</sup> 및 Windows<sup>(TM)</sup> 플랫폼은 같은 설치 디렉토리 및 파일 구조를 공유하며 유일한 차이점은 디렉토리 경로의 표시입니다(UNIX에서는 슬래시 /, Windows에서는 백 슬래시 #).

### UNIX/Linux용 디렉토리 및 파일 구조

다음 테이블은 WebSphere Adapter for JDBC용 UNIX/Linux<sup>(TM)</sup> 디렉토리 및 파일을 설명합니다. 디렉토리와 파일은 범주로 그룹화됩니다.

파일 및 디렉토리 범주	디렉토리 및 파일
Adapter for JDBC RAR 파일	adapter/depoy/CWYBC_JDBC.rar
IBM Support Assistant 플러그인 Zip 파일	adapter/ISAPugin/com.ibm.esupport.client.SS6FE6 _RAJDBC.zip
ITLM(IBM Tivoli License Manager) 파일	adapter/5724L77E060000.sys
메시지 파일	adapter/messages/CWYBC_JDBC_messages.tar
Foundation Classes용 메시지 파일	adapter/messages/CWYBS_AdapterFoundation _messages.tar
ICU4J용 공지 파일	adapter/notices.txt
샘플 응용프로그램 EAR 파일	adapter/samples/Apps/JDBCApp.ear
IBM DB2용 샘플 스크립트	adapter/samples/scripts/scripts_db2.sql
Oracle용 샘플 스크립트	adapter/samples/scripts_oracle.sql

### Windows용 디렉토리 및 파일 구조

다음 테이블은 WebSphere Adapter for JDBC용 Windows 디렉토리 및 파일을 설명합니다. 디렉토리와 파일은 범주로 그룹화됩니다.

파일 및 디렉토리 범주	디렉토리 및 파일
Adapter for JDBC RAR 파일	adapter#wdeploy#wCWYBC_JDBC.rar
IBM Support Assistant 플러그인 Zip 파일	adapter#wISAPugin#wcom.ibm.esupport.client.SS6FE6 _RAJDBC.zip
ITLM(IBM Tivoli License Manager) 파일	adapter#w5724L77E060000.sys
메시지 파일	adapter#wmessages#wCWYBC_JDBC_messages.zip
Foundation Classes용 메시지 파일	adapter#wmessages#wCWYBS_AdapterFoundation _messages.zip
ICU4J용 공지 파일	adapter#wnotices.txt
샘플 응용프로그램 EAR 파일	adapter#wsamples#wApps#wJDBCApp.ear
IBM DB2용 샘플 스크립트	adapter#wsamples#wscripts#wscripts_db2.sql

파일 및 디렉토리 범주	디렉토리 및 파일
Oracle용 샘플 스크립트	adapter#samples#scripts_oracle.sql

## 어댑터 프로젝트 작성

WID(WebSphere Integrator Developer)에서 어댑터 프로젝트를 작성해야 합니다. 그런 다음 WID에서 Enterprise Service Discovery 마법사를 사용하여 Business Object와 서비스 구성을 생성하십시오. 구성 등록 정보를 설정하십시오. 끝으로 어댑터 프로젝트를 응용프로그램 서버에 전개하십시오.

어댑터를 전개하기 전에 다음 제품을 설치해야 합니다.

- WebSphere Integration Developer 버전 6.0
- WebSphere Process Server - WebSphere Process Server 관리 콘솔을 사용하여 응용프로그램 서버에 어댑터 프로젝트를 전개하고 등록 정보 값을 재구성합니다.

<http://www.ibm.com/software/integration/wps>에서 IBM WebSphere Process Server 설치 지시사항을 참조하십시오.

WebSphere Adapter는 Windows 또는 Linux 운영 체제의 컴퓨터에만 설치할 수 있습니다. 그런 다음 UNIX 기반 시스템으로 전개할 수 있습니다. 각 어댑터는 RAR(Resource Adapter Archive) 파일로 설치됩니다.

기본적으로 어댑터의 전개는 WebSphere Process Server의 다른 구성요소 전개와 같습니다. WebSphere Process Server의 구성요소 전개에 대한 자세한 내용은 <http://www.ibm.com/software/integration/wid>에 있는 WebSphere Integrator Developer 사용자 안내서를 참조하십시오.

어댑터 프로젝트를 작성하려면 다음 작업을 수행하십시오.

- 어댑터를 위한 프로젝트 작성
- 벤더 라이브러리 추가
- Business Object 및 서비스 구성 생성
- 구성 등록 정보 값을 설정하고 그 값과 아티팩트를 새 Business Integration 모듈에 저장
- 어댑터 프로젝트를 서버에 전개하고 응용프로그램 시작

### 어댑터를 위한 프로젝트 작성

어댑터 전개의 첫 번째 작업은 어댑터를 위한 J2EE 커넥터 프로젝트를 작성하는 것입니다. Adapter for JDBC용 RAR(Resource Adapter Archive) 파일을 WID(WebSphere Integrator Developer)로 가져와야 합니다. 그러면 WID의 작업 공간에 프로젝트가 설정됩니다.



## 1. WebSphere Integration Developer 시작

자세한 내용은 <http://www.ibm.com/software/integration.wid>에 있는 WebSphere Integration Developer 사용자 안내서를 참조하십시오.

## 2. RAR 파일 가져오기

WID에서 Business Integration 영역으로 이동하십시오. 파일 > 가져오기를 누르십시오. 선택 창에서 가져올 소스로 **RAR** 파일을 선택하고 다음을 누르십시오.

커넥터 가져오기 창에서 찾아보기 단추를 사용하여 RAR 파일의 위치를 선택하십시오. 커넥터 프로젝트 필드에 프로젝트 이름이 자동으로 표시되며 변경할 수 있습니다.

**EAR** 프로젝트에 모듈 추가 선택란을 취소하십시오.

완료 버튼을 눌러서 RAR 파일을 가져오십시오. 그러면 작업 공간에 J2EE 커넥터 프로젝트가 작성됩니다.

## 3. 커넥터 프로젝트에 JDBC 드라이버 추가

응용프로그램 서버에 전개할 EAR(Enterprise Application Archive) 파일의 파트가 되도록 하려면 커넥터 프로젝트에 JDBC 드라이버를 추가해야 합니다. 이 작업은 RAR 파일을 가져온 후에 할 수도 있고 응용프로그램 서버에 EAR을 설치한 후에 할 수도 있습니다.

RAR 파일을 가져온 후에는 작업 공간의 알맞은 폴더에 jar 파일을 추가합니다. 예를 들어, 다음 위치에 jar 파일을 추가할 수 있습니다.

```
C:#workspace#CWYBC_JDBC#connectorModule
```

응용프로그램 서버에 EAR을 설치한 후 커넥터 프로젝트에 JDBC 드라이버를 추가하는 경우에는, 응용프로그램을 설치한 다음 jar 파일을 WebSphere Process Server의 installedApps 디렉토리의 RAR 서브디렉토리에 추가합니다. 응용프로그램 서버에 응용프로그램을 설치하는 데 대한 자세한 내용은 『어댑터 프로젝트 전개』를 참조하십시오.

### 관련 태스크

53 페이지의 『어댑터 프로젝트 전개』

프로젝트 파일은 WID(WebSphere Integration Developer) 작업 공간의 J2EE Connector 프로젝트에 있습니다. 이 파일을 EAR(Enterprise Application Archive) 파일로서 로컬 파일 시스템으로 내보내야 합니다. 그런 다음, 프로젝트의 응용프로그램 서버에 EAR 파일을 업로드하고 설치하기 전에 JCA 커넥터 보안을 위한 인증 정보를 제공해야 합니다.

## 벤더 라이브러리 추가

WID(WebSphere Integration Developer)에서 프로젝트를 작성하였으면 JDBC 드라이버에 대한 참조를 프로젝트에 추가해야 합니다.

Java 빌드 경로에 JDBC 드라이버 추가

WID의 구성 창에서 커넥터 프로젝트를 마우스 오른쪽 단추로 누르십시오. 등록 정보를 선택하십시오.

외부 .jar 파일을 추가하려면 **Java 빌드 경로** 단추를 누르십시오. 라이브러리 탭을 선택하고 외부 **Jar** 추가를 누르십시오. 파일 시스템 창에서 **JDBC** 드라이버로 이동하여 .jar 파일을 선택하십시오.

## Business Object 생성

Business Object를 생성하려면 우선 연결 등록 정보를 설정해야 합니다. 그러면 데이터베이스 오브젝트에 대해 조회를 실행하여 서비스 설명에 필요한 오브젝트를 선택할 수 있습니다. 그런 다음 가져오기 및 내보내기 파일을 위해 선택 등록 정보의 값을 지정해야 합니다. 끝으로 구성 등록 정보를 설정하고 아티팩트 및 등록 정보 값을 새 Business Integration 모듈에 저장하십시오.

Business Object를 생성하는 작업을 시작하기 전에 메타데이터 가져오기, 시스템 성능 찾기 및 데이터 설명에 대한 섹션에서 프로세스의 세부 사항을 참조하십시오.

### 메타데이터 가져오기

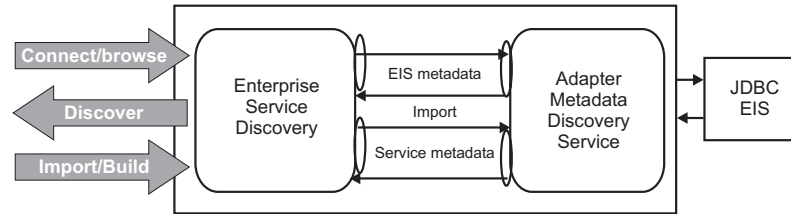
WebSphere Adapter for JDBC에서 엔터프라이즈 서비스 찾기 마법사를 통해 데이터베이스에서 오브젝트를 찾고, 그리고 선택된 오브젝트에서 Business Object를 생성하십시오. 엔터프라이즈 서비스 찾기 지원에는 어댑터가 SCA(Service Component Architecture) 구성요소로서 실행되도록 서비스 구문을 생성하는 것도 포함됩니다.

엔터프라이즈 서비스 찾기 마법사를 사용하여 데이터베이스의 오브젝트에서 Business Object를 생성할 수 있습니다. Business Object가 작성되는 오브젝트에는 테이블, 보기, 저장 프로시저 및 동의어/별명이 포함됩니다. 어댑터를 사용하면 데이터베이스에 모든 스키마 목록을 생성하여 오브젝트를 찾을 수 있습니다. 각 스키마 안에는 엔터프라이즈 서비스 찾기 마법사를 선택하고 요청하여 해당 Business Object를 생성할 수 있는 테이블, 보기, 저장 프로시저 및 동의어/별명 목록이 있습니다. 마법사는 Business Object에서 오브젝트의 메타데이터를 분석하고 속성을 생성합니다. 속성은 모든 데이터베이스 오브젝트에 대한 열 이름을 기본으로 하여 생성됩니다. 기능을 요약하기 위해 엔터프라이즈 서비스 찾기 마법사는 다음을 수행합니다.

- 데이터베이스 오브젝트에 해당하는 Business Object를 생성합니다.
- 데이터베이스 오브젝트의 등록 정보에 해당하는 Business Object에 등록 정보를 생성합니다.

- Business Object에서 응용프로그램 특정 정보를 설정합니다.
- 가져오기/내보내기 및 WSDL(Web Services Description Language) 파일을 생성하는 데 사용되는 서비스 설명(인바운드 및 아웃바운드)을 제공합니다.

### ESD 상위 레벨 다이어그램



### 시스템 성능 찾기

어댑터는 데이터베이스를 분석하여 스키마를 식별하여 데이터베이스에서 Business Object를 찾습니다. 그런 다음 데이터베이스에서 오브젝트를 모두 나열하고 트리 구조로 이를 표시합니다.

스키마는 트리에서 최상위 레벨 노드로 표시됩니다. 스키마 노드는 생성하도록 선택이 가능하지 않습니다.

각 스키마 아래에는 테이블, 보기, 저장 프로시저 및 동의어/별명이라는 레이블의 노드가 있습니다. 이러한 노드는 생성하도록 선택할 수 없습니다. 이러한 노드 아래 나열된 오브젝트는 테이블, 보기, 저장 프로시저 및 동의어/별명의 이름입니다. 이러한 노드는 생성하도록 선택 가능한 것으로 표시됩니다. 특정 스키마에 대해 테이블, 보기, 저장 프로시저 또는 동의어가 존재하지 않으면 아무 것도 나열되지 않습니다.

트리에 표시된 스키마 목록을 축소하려면 필터 등록 정보를 지정할 수 있습니다. 그렇지 않으면 스키마가 모두 표시됩니다. 『참조』 섹션의 『필터 및 노드 등록 정보』는 사용자가 제공해야 하는 등록 정보에 대해 설명합니다.

테이블, 보기, 저장 프로시저 또는 동의어/별명을 선택한 다음 필터 단추를 누르면 엔터프라이즈 서비스 찾기 마법사가 ObjectNameFilter에 대해 조회합니다. ObjectNameFilter 등록 정보를 사용하여 표시할 데이터베이스 오브젝트 목록을 필터링할 수 있습니다. 『참조』 섹션의 『필터 및 노드 등록 정보』는 ObjectNameFilter 등록 정보를 설명합니다.

#### 관련 참조

84 페이지의 『필터 및 노드 등록 정보』

Business Object를 찾는 동안 트리에 표시된 스키마 목록을 축소하려면 필터 등록 정보를 지정할 수 있습니다. 그렇지 않으면 스키마가 모두 표시됩니다. 표시할 데이터베이스 오브젝트 목록을 축소하려면 노드 등록 정보를 설정할 수 있습니다.

### 오브젝트 선택 및 생성:

Business Object를 생성하려면 데이터베이스 오브젝트 노드를 선택하십시오. 그러면 엔터프라이즈 서비스 찾기 마법사는 선택된 노드의 오브젝트에 대해 Business Object를 생성합니다.

여러 데이터베이스 오브젝트 노드를 선택할 수 있습니다. 필터 등록 정보를 지정할 경우 『Business Object ASI 추가』를 선택하면 엔터프라이즈 서비스 찾기 마법사는 StatusColumnName, StatusValue 및 저장 프로시저 연관에 대해 오브젝트 레벨의 매개변수를 조회합니다.

StatusColumnName의 경우 StatusColumnName을 설정하는 특정 오브젝트에 대해 실제 열 이름의 목록이 제공됩니다. StatusValue를 입력해야 합니다. 이러한 값은 Business Object 레벨의 응용프로그램 특정 정보(ASI)에서 설정됩니다.

저장 프로시저를 Business Object에 연관시키도록 선택할 수 있습니다. 지원되는 모든 저장 프로시저 유형의 목록이 제공됩니다. 각 저장 프로시저 유형에는 데이터베이스에서 사용할 수 있는 저장 프로시저의 목록이 있습니다. 특정 저장 프로시저 유형에는 한 개의 저장 프로시저가 지정될 수 있습니다.

지정된 저장 프로시저마다 해당 저장 프로시저의 입력 및 출력 매개변수 목록이 제공됩니다. 각 매개변수에는 Business Object의 속성 목록이 있습니다. 매개변수의 유형은 등록 정보의 설명에 나열되어 있습니다. 저장 프로시저 매개변수마다 하나의 Business Object 속성을 선택할 수 있습니다. 저장 프로시저가 지정되어 있는 모든 저장 프로시저 유형은 Business Object의 Verb ASI에 추가됩니다. 『Verb 응용프로그램 특정 정보』를 참조하십시오.

데이터베이스 오브젝트를 선택한 후 선택 등록 정보의 값을 설정해야 합니다. 엔터프라이즈 서비스 찾기 마법사는 선택 등록 정보를 조회합니다. 이 등록 정보에 대한 자세한 내용은 『참조』 섹션의 『선택 등록 정보』를 참조하십시오.

#### 관련 태스크

85 페이지의 『선택 등록 정보』

데이터베이스 오브젝트를 선택한 후 선택 등록 정보의 값을 설정해야 합니다.

#### 관련 참조

28 페이지의 『Verb 응용프로그램 특정 정보』

어댑터는 Business Object에 지정된 대로 SQL 문의 그룹인 저장 프로시저 또는 SQL 조회를 사용하여 데이터베이스 테이블을 갱신합니다. 이 섹션에는 저장 프로시저와 저장 프로시저 정의의 요소를 설명하며, 저장 프로시저 정의의 샘플이 포함되어 있습니다.

## 데이터 설명

데이터 설명은 어댑터가 SCA(Service Component Architecture) 구성요소로서 실행되도록 엔터프라이즈 서비스 찾기 프로세스가 생성하는 서비스 구문의 일부입니다.

데이터 설명에는 런타임에 클라이언트 응용프로그램과 어댑터 사이에 전달된 어댑터 Business Object의 구조 및 내용에 관한 정의가 포함됩니다. 데이터 설명을 사용하면 클라이언트 응용프로그램에서 요청에 따라 적합한 데이터 오브젝트를 작성하고 응답 시 리턴된 데이터 오브젝트를 해석할 수 있습니다. 데이터베이스 구성요소에서 생성한 데이터 설명은 XML 스키마로 나타냅니다.

- Business Object는 복합 유형 정의에 맵핑됩니다.
- Business Object의 속성은 요소 유형 정의에 맵핑됩니다.
- Business Object에 대한 응용프로그램 특정 정보는 복합 유형의 주석에 들어 있습니다.
- Business Object의 각 등록 정보에 대한 응용프로그램 특정 정보는 요소 유형의 주석에 들어 있습니다.

주: Business Object에 대한 응용프로그램 특정 등록 정보의 템플릿과 속성 레벨은 JDBC 어댑터를 위한 메타데이터 스키마에 정의되어 있습니다. 스키마 파일의 이름은 JDBCASI.xsd입니다. 생성된 스키마 파일에는 주석에 있는 템플릿에 대한 참조가 포함되어 있습니다.

### Business Object 스키마 및 ASI:

Business Object 스키마는 사용자가 선택한 데이터베이스 구성요소로부터 빌드됩니다. 각 구성요소는 최상위 레벨의 Business Object로 변환합니다. 엔터프라이즈 서비스 찾기 마법사는 Business Object의 이름을 *PrefixSchemaNameObjectName* 양식으로 생성합니다. 여기에서,

- *SchemaName*은 오브젝트가 속하는 스키마의 이름입니다.
- *ObjectName*은 테이블, 보기, 저장 프로시저 또는 동의어/별명의 이름입니다.
- *Prefix*는 Prefix라는 연결 등록 정보에서 지정된 값입니다. Prefix는 필수가 아니므로 지정하지 않으면 Business Object 이름에 접두부가 추가되지 않습니다.

Business Object 이름에서는 세계화 문자가 지원됩니다.

엔터프라이즈 서비스 찾기 마법사는 *TableName* 응용프로그램 특정 정보 속성을 *schemaname.tablename* 양식으로 값에 설정합니다. 이것은 Business Object 레벨의 응용프로그램 특정 정보를 『Business Object ASI』 테이블에 나열된 대로 설정합니다. 선택한 조작용 Business Object에 설정됩니다. 생성된 모든 Business Object는 테이블, 보기, 저장 프로시저 또는 동의어/별명 중 생성된 오브젝트 유형에 관계 없이 같은 구조를 가집니다. 모두 해당 열을 기반으로 한 속성을 가지며 *TableName ASI*는 오브젝트의 이름으로 설정됩니다.

## Business Object ASI

Business Object ASI	ESD가 설정	추가 정보
TableName	예	열의 실제 이름에 설정
StatusColumnName	예	오브젝트를 선택하는 동안 지정하십시오.
StatusValue	예	오브젝트를 선택하는 동안 지정하십시오.

작성된 모든 Business Object는 최상위 레벨입니다. 엔터프라이즈 서비스 찾기 마법사는 반복(하위) Business Object를 작성하지 않습니다. 엔터프라이즈 서비스 찾기 마법사는 모두가 최상위 레벨이므로 또한 모든 Business Object에 대한 비즈니스 그래프를 생성합니다. 비즈니스 그래프의 이름은 Business Object 이름과 『BG』입니다. 예를 들어, JDBCShema1Customer라는 이름의 Business Object는 JDBCShema1CustomerBG라는 이름의 비즈니스 그래프를 갖게 됩니다. Business Object의 조작 세트도 비즈니스 그래프에서 설정됩니다.

### Business Object 속성 및 ASI:

Business Object의 속성은 데이터베이스 오브젝트의 열 목록에서 빌드됩니다. 엔터프라이즈 서비스 찾기 마법사는 속성 이름을 열 이름으로 설정합니다. 속성 이름에는 세계화 문자가 지원됩니다. 어댑터는 속성 이름, 유형 및 응용프로그램 특정 정보를 추가합니다.

JDBC 메타데이터가 리턴한 유형은 『JDBC 메타데이터 열 및 Business Object 속성 유형』 테이블에서 설명하는 Business Object 속성 유형에 맵핑됩니다. 어댑터는 여기에서 설명하는 JDBC 유형만 지원합니다. 여기에서 설명되지 않은 유형의 열은 Business Object에 추가할 수 없습니다. 정보 레벨 메시지는 『yyyy 테이블의 xxxx라는 열은 지원되지 않는 유형이므로 Business Object에 추가되지 않습니다.』와 같이 상태를 나타내도록 생성됩니다.

### JDBC 메타데이터 열 및 Business Object 속성 유형

JDBC 메타데이터 열 유형	Business Object 속성 유형
BIT	BOOLEAN
CHAR LONGVARCHAR VARCHAR	STRING
INTEGER NUMERIC SMALLINT TINYINT BIGINT	INTEGER

JDBC 메타데이터 열 유형	Business Object 속성 유형
TIME TIMESTAMP DATE	DATE
DECIMAL	STRING
DOUBLE FLOAT	DOUBLE
REAL	FLOAT

『속성 정보』 테이블에서는 엔터프라이즈 서비스 찾기 마법사가 설정한 속성 정보를 나열하고 그들을 설정하는 방법에 대해 설명합니다.

### 속성 정보

속성 정보	ESD가 설정	추가 정보
카디널리티	아니오	
이름	예	속성 이름. 이것은 양방향 언어에 대해 사용 가능합니다.
MinOccurs/MaxOccurs	예	1차 키가 아닌 열이 널(null) 값이 될 수 없으면 그 속성은 필수이며, 그 속성에 대한 값은 최소 1로 설정되어야 합니다.
유형	예	『JDBC 메타데이터 열 및 Business Object 속성 유형』 테이블처럼 설정

엔터프라이즈 서비스 찾기 마법사는 Business Object의 속성 ASI를 『속성 ASI』 테이블에 표시된 것처럼 설정합니다. 속성 응용프로그램 특정 정보에 대한 자세한 내용은 『단순 속성에 대한 응용프로그램 특정 정보』를 참조하십시오.

### 속성 ASI

속성 ASI	ESD가 설정	추가 정보
ColumnName	예	열의 실제 이름에 설정. 이것은 양방향 언어에 대해 사용 가능합니다.
FixedChar	아니오	Business Object .xsd 파일에서 수동으로 갱신되어야 합니다. 텍스트 모드를 사용하거나 WID의 Business Object 편집기를 사용하여 파일을 편집하십시오. 파일 갱신 후, 유효성 검증 오류가 없는지 확인하십시오. 『단순 속성을 위한 응용프로그램 특정 정보』 섹션의 .xsd 파일에서 FixedChar의 예를 참조하십시오.
ForeignKey	아니오	

속성 ASI	ESD가 설정	추가 정보
OrderBy	아니오	
PrimaryKey	예	열이 1차 키이면 PrimaryKey가 true로 설정됩니다.
UID	아니오	

저장 프로시저를 Business Object에 추가하려면 Verb ASI가 『Verb ASI』 테이블에 지정된 것처럼 설정되어야 합니다. 올바른 저장 프로시저 유형에 대한 정보는 『Verb 응용프로그램 특정 정보』를 참조하십시오.

### Verb ASI

Verb ASI 또는 저장 프로시저 매개변수 요소	ESD가 설정	추가 정보
매개변수	예	저장 프로시저 매개변수를 설명합니다. 이것은 양방향 언어에 대해 사용 가능합니다.
PropertyName	예	선택한 Business Object 속성의 이름을 설정합니다. 이것은 양방향 언어에 대해 사용 가능합니다.
ResultSet	아니오	저장 프로시저가 ResultSet를 리턴하면 Business Object 정의에서 이 매개변수를 true로 설정해야 합니다.
StoredProcedure	예	저장 프로시저 이름에 설정. 이것은 양방향 언어에 대해 사용 가능합니다.
StoredProcedure Type	예	유형 목록에서 선택합니다.
유형	예	SP 매개변수의 유형에 설정 (IP/OP/IO).

### 계층 구조 Business Object를 빌드하려면

엔터프라이즈 서비스 찾기 마법사는 플랫폼 Business Object를 생성합니다. 이것은 여러 테이블 사이에 자동으로 관계를 빌드하는 데이터베이스에서 정의된 외부 키 제한조건을 사용하지 않습니다. 이것은 수동으로 링크되어야 합니다. 텍스트 모드에서 또는 Business Object 편집기를 사용하여 Business Object 정의를 갱신할 수 있습니다.

단일 및 다중 카디널리티 하위 Business Object에 대한 .xsd 정의 파일의 예가 여기에서 제공됩니다. custInfoObj 요소는 단일 카디널리티 하위 Business Object이며, addressObj는 다중 카디널리티 하위 Business Object입니다.

```
<element name="addressObj" minOccurs="1" type="Address:Address"
maxOccurs="unbounded">
  <annotation>
    <appinfo source="WBI">
      <pasi:JDBCAttributeTypeMetadata xmlns:pasi=
```



```

"urn:app:jdbc:asi">
    <pasi:Ownership>true</pasi:Ownership>
  </pasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>
<element name="custInfoObj" minOccurs="0" type=
"CustInfo:CustInfo" maxOccurs="1">
  <annotation>
  <appinfo source="WBI">
  <pasi:JDBCAttributeTypeMetadata xmlns:pasi=
"urn:app:jdbc:asi">
    <pasi:Ownership>false</pasi:Ownership>
  </pasi:JDBCAttributeTypeMetadata>
</appinfo>
</annotation>
</element>

```

## 관련 참조

31 페이지의 『속성 응용프로그램 특정 정보』

이 주제에서는 속성에 대한 응용프로그램 특정 정보를 설명하고 그 설명과 함께 지원되는 매개변수를 나열합니다.

28 페이지의 『Verb 응용프로그램 특정 정보』

어댑터는 Business Object에 지정된 대로 SQL 문의 그룹인 저장 프로시저 또는 SQL 조화를 사용하여 데이터베이스 테이블을 갱신합니다. 이 섹션에는 저장 프로시저와 저장 프로시저 정의의 요소를 설명하며, 저장 프로시저 정의의 샘플이 포함되어 있습니다.

## 연결 등록 정보 설정

어댑터 프로젝트를 작성한 후, JDBC를 위해 엔터프라이즈 서비스 찾기 마법사를 초기화하고 데이터베이스 인스턴스를 위해 연결 등록 정보 값을 설정해야 합니다.

### 1. 엔터프라이즈 서비스 찾기 초기화

WID에서 **Business Integration** 영역으로 이동하십시오. JDBC 커넥터 프로젝트가 강조 표시된 **Business Integration** 탭에서 해당 프레임에 마우스 오른쪽 단추로 누르십시오. 팝업 메뉴에서 새로 작성 > 엔터프라이즈 서비스 찾기를 선택하십시오.

엔터프라이즈 서비스 자원 어댑터 선택 창에서 어댑터 옵션을 선택하고 다음을 누르십시오. RAR 파일을 아직 가져오지 않은 경우에는 이 창에서 자원 어댑터 가져오기 단추를 사용하여 가져올 수 있습니다.

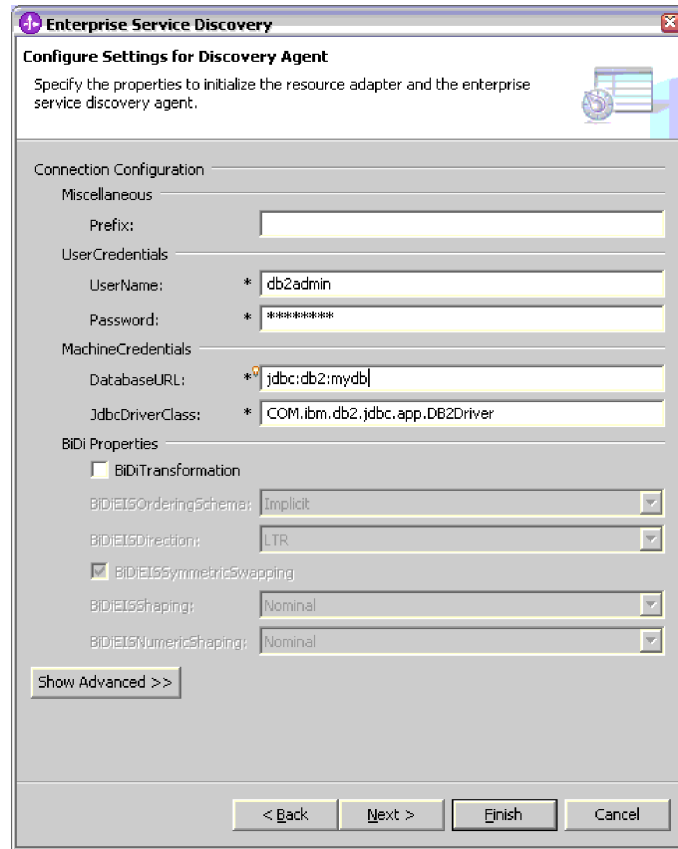
### 2. 연결 등록 정보 값 설정

찾기를 위한 대상 EIS 인스턴스에 연결하고 서비스 설명을 작성하는 데 사용되는 메타데이터 찾기 연결 등록 정보의 값을 설정해야 합니다. 양방향 스크립트 데이터 처리를 사용 가능하게 하려면 양방향 변환을 활성화하고 양방향 등록 정보의 값을 설정해야 합니다.

아래에 표시된 찾기 에이전트 설정 구성 창에서 연결 구성 등록 정보의 값을 입력 하십시오. 이러한 등록 정보에 대한 자세한 내용은 『참조』 섹션의 『메타데이터 찾기 연결 등록 정보』를 참조하십시오.

양방향 기능을 활성화하려면 양방향 변환 옆의 상자를 누르십시오. 그런 다음 양방 향 등록 정보 값을 설정하십시오. 이러한 등록 정보에 대한 자세한 내용은 『참조』 섹션의 『양방향 연결 등록 정보』를 참조하십시오.

### 찾기 에이전트 설정 구성 창



### 3. 로깅 옵션 선택

고급 옵션 표시를 누르면 로깅 옵션이 표시됩니다. 로깅 옵션은 엔터프라이즈 서비 스 찾기 프로세스의 로깅 및 추적 설정에만 사용됩니다. 그러나 로깅 및 추적 레벨 은 어댑터에 대해서는 동일합니다. 어댑터에 대한 로깅 및 추적 레벨의 자세한 내 용은 『로깅 사용』 및 『추적 사용』을 참조하십시오.

로깅 옵션 아래에서 로그 파일을 출력할 위치를 입력하거나 찾아보십시오. 엔터프라 이즈 서비스 찾기를 위한 로깅 및 추적 레벨을 선택하십시오. 다음을 누르십시오.

찾기 서비스는 연결 등록 정보를 사용하여 오브젝트 선택 및 탐색에서 표시되는 메타데 이터 트리를 준비합니다.

## 관련 태스크

### 59 페이지의 『로깅 사용』

WebSphere Adapter for JDBC는 이벤트 처리 상태를 판별하기 위해 살펴볼 수 있는 로그 파일을 유지합니다. 어댑터와 관련된 모든 이벤트와 오류는 로그 파일에 의해 각 로그 항목에 날짜, 시간 및 이벤트와 함께 추적됩니다. 어댑터는 오류 또는 경고 조건을 만날 때 오류 메시지를 로그하므로 로그 파일은 문제점 해결을 시작할 좋은 소스입니다.

### 60 페이지의 『추적 사용』

추적은 어댑터 로그 파일이 오류 또는 경고를 캡처하는 레벨을 결정합니다. 추적 레벨을 정의하여 어댑터 처리에 관한 메시지를 추적할 수 있습니다.

## 관련 참조

### 83 페이지의 『메타데이터 찾기 연결 등록 정보』

엔터프라이즈 서비스 찾기 프로세스에는 찾기 및 서비스 설명 작성을 위해 EIS(Enterprise Information System)에 연결하는 이러한 등록 정보가 필요합니다.

### 84 페이지의 『양방향 연결 등록 정보』

이러한 등록 정보를 사용하여 엔터프라이즈 서비스 찾기 마법사는 EIS(Enterprise Information System)로 전달된 데이터에 대해 알맞은 양방향 변환을 적용할 수 있습니다.

## 데이터베이스 오브젝트 조회

연결 등록 정보를 구성한 다음 데이터베이스 오브젝트 조회를 실행할 수 있습니다. 메타데이터 트리 구조를 찾아보면 EIS(Enterprise Information System)에서 오브젝트의 구조를 이해할 수 있고 서비스 설명에 필요한 오브젝트를 선택할 수 있습니다.

트리 구조로 표시되는 스키마, 노드 또는 오브젝트 목록을 축소하려면 조회를 실행하기 전에 필터 등록 정보를 지정할 수 있습니다.

### 1. 필터 등록 정보 지정

엔터프라이즈 서비스 찾기 창에서 **조회 편집** 단추를 누르십시오. **조회 필터 등록 정보** 팝업 창의 스키마 이름 필터 등록 정보 필드에 텍스트를 입력하십시오. 지정된 문자열로 시작하는 스키마가 표시됩니다. 사용할 스키마를 선택하십시오.

**유형** 등록 정보 필드에는 테이블, 보기, 저장 프로시저 및 동의어/별명 항목이 있습니다. 목록에서 노드를 추가하거나 제거할 수 있습니다.

**조회 필터 등록 정보** 창에서 **Business Object ASI** 추가를 선택할 수 있습니다. 그러면 오브젝트를 추가할 때마다 2단계의 메타데이터 조회 실행시 (**오브젝트 이름**) 구성 매개변수라는 창이 나타나서 응용프로그램 특정 정보를 입력할 수 있습니다.

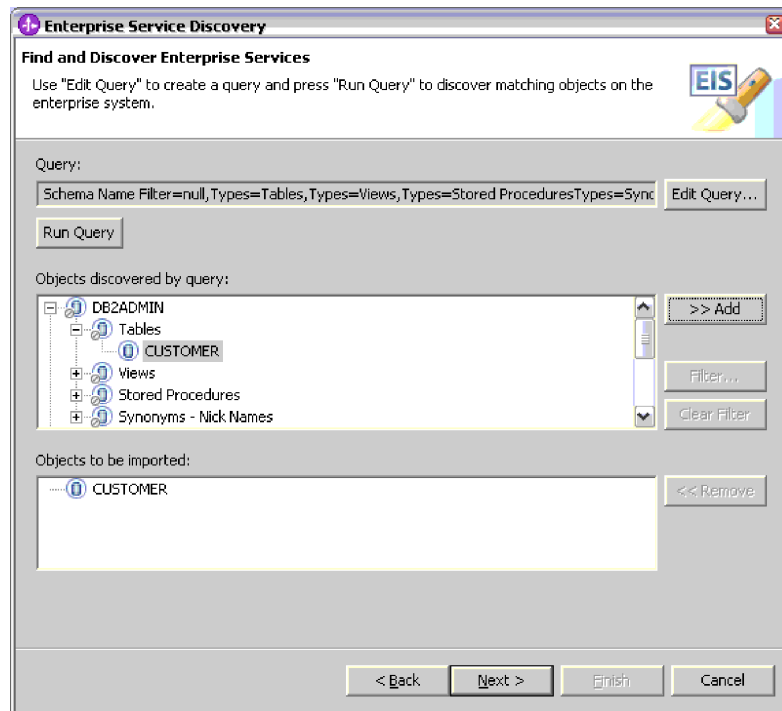
### 2. 메타데이터 조회 실행

아래에 표시된 엔터프라이즈 서비스 찾기 창에서 조회 실행 단추를 누르십시오. 오브젝트가 위쪽 분할창에 표시됩니다.

스키마를 필터링한 것과 유사한 방법으로 오브젝트도 필터링할 수 있습니다. 테이블, 보기, 저장 프로시저 또는 동의어/별명 노드를 선택하십시오. 필터 단추를 누르십시오. 엔터프라이즈 서비스 찾기 마법사는 오브젝트 이름 필터를 조회하여 해당 노드에 대해 표시할 데이터베이스 오브젝트 목록을 필터링합니다. 텍스트를 입력하면 지정된 문자열로 시작하는 데이터베이스 오브젝트만 표시됩니다.

오브젝트를 강조하고 추가 단추를 눌러서 포함될 오브젝트를 선택하십시오. 선택한 오브젝트가 아래쪽 분할창에 표시됩니다. 선택한 오브젝트를 제거하려면 해당 오브젝트를 강조한 다음 제거 단추를 누르십시오.

### 엔터프라이즈 서비스 찾기 창

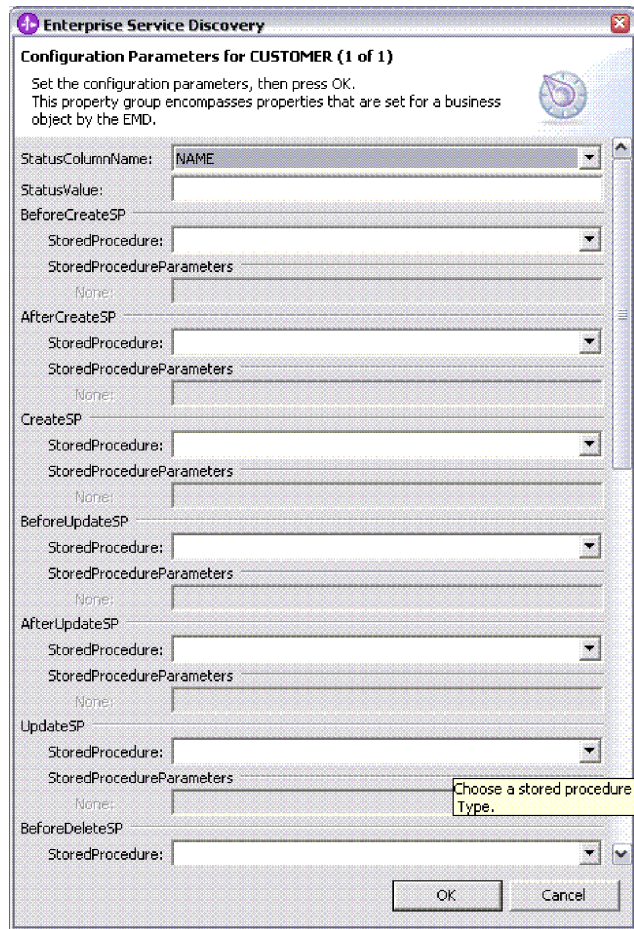


조회 필터 등록 정보 창에서 **Business Object ASI** 추가를 선택하면 오브젝트를 추가할 때마다 (오브젝트 이름) 구성 매개변수라는 창이 나타나서 응용프로그램 특정 정보를 입력할 수 있습니다. 이 창은 **CUSTOMER** 오브젝트의 Verb 응용프로그램 특정 정보 매개변수와 함께 아래에 표시됩니다. ASI 입력을 마치면 확인을 누르십시오.

추가할 오브젝트를 둘 이상 선택한 경우에는 첫 번째 오브젝트에 대한 창이 표시됩니다. ASI를 입력하고 확인을 누르면 다음 오브젝트에 대한 창이 나타납니다. 오브젝트 레벨 Verb에 대한 정보와 속성 응용프로그램 특정 정보는 『응용프로그램 특정 정보』를 참조하십시오.

완료하였으면 다음을 누르십시오.

## CUSTOMER 구성 매개변수 창



### 관련 태스크

70 페이지의 『시나리오 2의 전개 및 구성』

시나리오 2에서는 어댑터 연결 등록 정보를 설정하고 Business Object를 생성합니다. 프로젝트를 EAR 파일로 내보내고 응용프로그램 서버에 프로젝트를 전개하며 구성 등록 정보를 재설정합니다.

### 관련 참조

26 페이지의 『응용프로그램 특정 정보』

Business Object 정의의 응용프로그램 특정 정보는 어댑터에 Business Object를 처리하는 방법에 관한 응용프로그램 종속 지시사항을 제공합니다. 어댑터에서는 비즈니스의 속성 또는 Verb에서나 Business Object 자체에서 응용프로그램 특정 정보를 구문 분석하여 Create, Update, Retrieve 및 Delete 조작에 대한 조회를 생성합니다.

## 선택 등록 정보 설정

데이터베이스 오브젝트를 선택하였으면 가져오기 및 내보내기 파일을 위해 선택 등록 정보의 값을 지정해야 합니다.

선택 등록 정보에 대한 자세한 내용은 『참조』 섹션을 참조하십시오.

### 1. NameSpace 지정

NameSpace는 처음에 모든 Business Object의 기본 NameSpace로 설정됩니다. 이 기본값은 오브젝트 구성 창에 표시됩니다. 이것은 메타데이터 스키마 JDBCASI.xsd를 위한 NameSpace입니다.

NameSpace는 Business Object 스키마가 논리적으로 구별되도록 Business Object에 제공됩니다.

### 2. 서비스 유형 선택

서비스 유형으로 인바운드 또는 아웃바운드를 선택하십시오.

### 3. 조작 선택

오브젝트 구성 창의 조작 필드에는 선택된 서비스 유형에 대해 어댑터가 지원하는 조작이 나열됩니다. 조작 목록에 추가하려면 추가 단추를 누르십시오. 예를 들어, 조작을 제거했는데 다시 포함시켜야 할 수 있습니다. 추가 창에서, 조작 목록에서 선택한 다음 확인을 누르십시오. 완료하였으면 다음을 누르십시오.

지정된 조작은 생성 중인 모든 Business Object에 대해 설정됩니다.

### 4. MaxRecords 설정

RetrieveAll 조작에서 검색할 레코드의 최대 수를 입력하십시오.

### 5. BOLocation 지정

생성된 .xsd 파일을 저장할 위치의 경로를 입력하십시오.

#### 관련 태스크

85 페이지의 『선택 등록 정보』

데이터베이스 오브젝트를 선택한 후 선택 등록 정보의 값을 설정해야 합니다.

70 페이지의 『시나리오 2의 전개 및 구성』

시나리오 2에서는 어댑터 연결 등록 정보를 설정하고 Business Object를 생성합니다. 프로젝트를 EAR 파일로 내보내고 응용프로그램 서버에 프로젝트를 전개하며 구성 등록 정보를 재설정합니다.

## 어댑터 프로젝트 저장

선택 등록 정보를 지정한 후에는 어댑터가 특정 데이터베이스에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 구성하십시오. 여기에는 자원 어댑터, J2C 연결 팩토리,

J2C 활성화 스펙 및 양방향 등록 정보가 포함됩니다. 또한 모든 아티팩트 및 등록 정보 값이 저장될 수 있는 새 Business Integration 모듈을 작성해야 합니다.

### 1. 새 모듈 이름 지정

아래에 표시한 아티팩트 생성 창의 모듈 필드에 모듈 이름이 표시되지 않으면 새로 작성 단추를 누르십시오. 새 모듈 팝업 창에서 모듈 이름을 입력하고 완료를 누르십시오.

### 2. 서비스 설명용 폴더 지정

일반 아티팩트 창에서 서비스 설명을 저장할 새 모듈 내의 폴더를 지정하거나 찾습니다.

선택한 서비스 유형에 따라 가져오기 또는 내보내기 파일 이름이 이름 필드에 표시됩니다. 설명 필드에 주석을 추가할 수 있습니다.

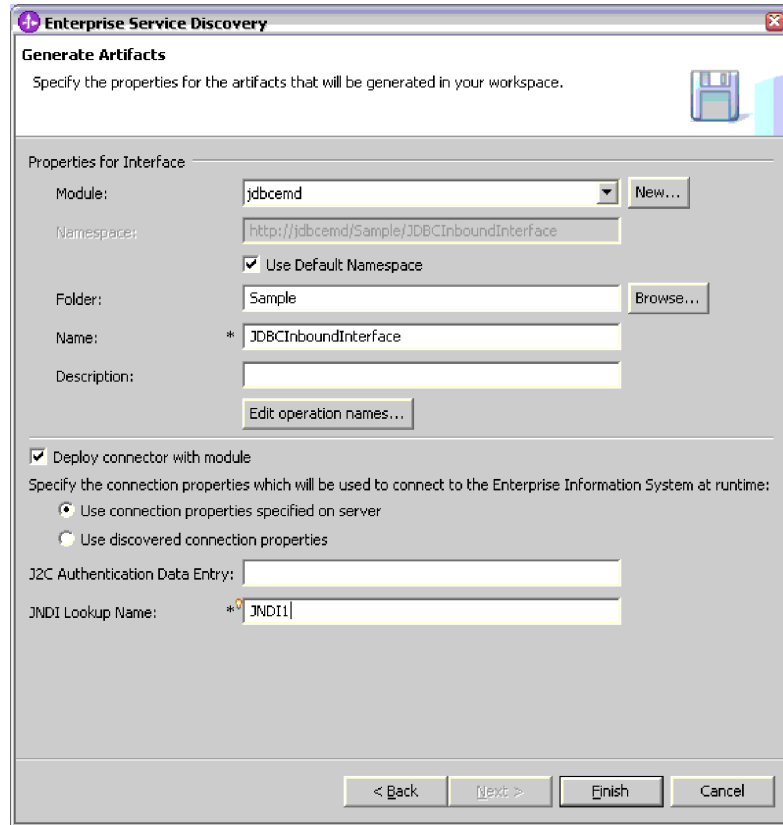
조작 편집 단추를 누르면 팝업 창에 작성 중인 Business Object에 대한 모든 조작의 기본 이름이 표시됩니다. 기본 이름은 Business Object 이름에 조작 이름이 결합되어 구성됩니다.

주: 모듈로 커넥터 전개 선택란을 체크하여 커넥터 프로젝트 RAR 파일이 응용프로그램 서버에 전개하는 EAR 파일에 포함되도록 해야 합니다.

### 3. 구성 등록 정보 값 설정

서버에서 지정된 연결 등록 정보 사용 옵션은 나중에 프로젝트 전개 후 새 Business Object를 작성해야 하는 경우 언제든지 사용할 수 있습니다. 이것은 응용프로그램 서버에 있는 등록 정보를 사용하려 한다는 것을 표시합니다. J2C 인증 데이터 항목 필드는 응용프로그램 서버에 이미 설정된 것이 있는 경우 인증 별명을 지정하는 데 사용됩니다. JNDI 찾아보기 이름은 응용프로그램 서버에서 사용할 연결 팩토리의 이름을 지정하는 데 사용됩니다.

아티팩트 생성 창



등록 정보 값을 설정하려면 **찾은 연결 등록 정보 사용**을 선택하십시오.

인바운드 처리인 경우 등록 정보 필드는 J2C 활성 스펙 및 자원 어댑터 등록 정보에 대해 표시됩니다. 아웃바운드 처리인 경우 등록 정보 필드는 J2C 연결 팩토리 및 자원 어댑터 등록 정보에 대해 표시됩니다. 양방향 변환을 활성화한 경우 양방향 등록 정보는 인바운드 또는 아웃바운드 처리에 대해 표시됩니다. 필수 등록 정보 필드는 별표로 표시됩니다. 이러한 구성 등록 정보에 대한 자세한 내용은 『참조』 섹션을 참조하십시오.

**중요사항:** 엔터프라이즈 서비스 찾기 마법사를 사용하여 J2C 활성 스펙 등록 정보를 지정하면 나중에 등록 정보 설정을 변경할 수 없습니다. 응용프로그램 서버에 새 응용프로그램으로서 프로젝트를 전개한 후에는 WebSphere Process Server 관리 콘솔을 사용하여 다른 구성 등록 정보를 변경할 수 있더라도 J2C 활성 스펙 등록 정보는 갱신할 수 없습니다. 나중에 등록 정보를 갱신하려 해도 어댑터는 갱신된 값을 인식하지 못합니다. 응용프로그램 서버에 프로젝트를 설치한 후에 J2C 활성 스펙 등록 정보를 설정하려면 지금은 이러한 등록 정보 값을 설정하지 마십시오.

#### 관련 참조

77 페이지의 『구성 등록 정보』

이 등록 정보 그룹에는 특정 데이터베이스 응용프로그램에 통신 채널을 설정하기 위해 어댑터에서 사용한 속성이 포함됩니다.



## 참조 바인딩 생성

참조 바인딩은 다른 WebSphere Business Integration SCA(Service Component Architecture) 구성 요소가 어댑터에 액세스하는 데 사용됩니다. 어댑터를 다른 서버 프로세스에 링크하듯이 프로젝트 모듈에서 어댑터에 대한 참조를 작성하십시오.

참조 바인딩 생성은 환경을 테스트할 때만 필요합니다. 프로덕션 환경에 어댑터를 전개할 때는 필요하지 않습니다.

### 1. SCA 구성요소 작성

WebSphere Integration Developer의 **Business Integration** 영역으로 이동하십시오. **Business Integration** 탭에서 **JDBCEND** 모듈을 마우스 오른쪽 단추로 누르고 함께 열기 > 어셈블리 편집기를 선택하십시오. 보기에 모듈의 가져오기 구성 요소와 함께 어셈블리 다이어그램이 표시됩니다.

새 구성요소를 작성하려면 어셈블리 다이어그램 창의 왼쪽(수직) 프레임에서 두 번째 아이콘(가져오기)을 누르십시오. 새 아이콘 메뉴가 표시됩니다.

새 아이콘 메뉴에서 맨 아래 아이콘을 누르십시오(독립형 참조라는 풍선 도움말이 있습니다). 커서가 배치 아이콘으로 바뀝니다.

팔레트를 눌러서 어셈블리 다이어그램 창에 새 구성요소를 추가하십시오.

### 2. 독립형 참조 작성

모듈의 가져오기 구성요소를 새 구성요소로 눌러서 끄십시오. 그러면 가져오기 구성 요소에서 새 구성요소로 선이 그려지며 선 추가 창이 표시됩니다.

선 추가 창에서 확인을 누르십시오. 다른 선 추가 창이 팝업되어 WSDL 인터페이스를 Java로 변환할 것인지 묻습니다. 아니오를 누르십시오.

새 독립형 참조 구성요소는 어셈블리 다이어그램 창에 모듈의 가져오기 구성요소로 연결하는 "선"을 표시합니다.

파일 > 저장을 눌러서 어셈블리 다이어그램을 저장하십시오. 자세한 내용은 <http://www.ibm.com/software/integration/wid>에 있는 WebSphere Integration Developer 사용자 안내서를 참조하십시오.

---

## 어댑터 프로젝트 전개

프로젝트 파일은 WID(WebSphere Integration Developer) 작업 공간의 J2EE Connector 프로젝트에 있습니다. 이 파일을 EAR(Enterprise Application Archive) 파일로서 로컬 파일 시스템으로 내보내야 합니다. 그런 다음, 프로젝트의 응용프로그램 서버에 EAR 파일을 업로드하고 설치하기 전에 JCA 커넥터 보안을 위한 인증 정보를 제공해야 합니다.

## 1. 프로젝트를 EAR 파일로 내보내기

WID에서 프로젝트를 EAR 파일로 내보내야 합니다. 선택 창의 내보내기 화면에 있는 목록에서 **EAR** 파일을 선택하고 다음을 누르십시오.

**EAR** 내보내기 창의 **EAR** 프로젝트 필드에서 Business Integration 모듈을 선택하십시오. 모듈 이름 **JDBCEND**에는 이제 접두어 **App**가 있습니다. EAR 파일을 작성할 위치를 입력하거나 찾아보십시오. 모든 선택란을 눌러서 EAR 파일에서 작성한 모든 것을 포함시키십시오. **완료**를 누르십시오.

## 2. 인증 정보 제공

WebSphere Process Server 관리 콘솔을 여십시오. 왼쪽 패널에서 **보안 > 글로벌 보안**을 누르십시오. 글로벌 보안 창의 구성 분할창에서 오른쪽의 인증 표제로 이동하십시오. **JAAS** 구성을 누르십시오. 그런 다음 **J2C** 인증 데이터 링크를 누르십시오.

글로벌 보안 > **J2EE Connector Architecture(J2C)** 인증 데이터 항목 창에서 새로 작성을 누르십시오. 별명 필드에 별명을 입력하십시오. 데이터베이스에 연결할 수 있는 사용자 ID와 암호를 입력하십시오. 확인을 누르십시오. 저장을 눌러서 인증 정보를 저장하십시오.

## 3. 응용프로그램 서버에 EAR 파일 설치

관리 콘솔을 사용하여 왼쪽 패널에서 **응용프로그램 > 새 응용프로그램 설치**를 누르십시오. **찾아보기** 단추를 누르고 EAR 파일을 선택한 후 다음을 누르십시오.

**7단계.** 자원에 자원 참조 매핑이 될 때까지 다음을 계속 누르십시오. 이전에 작성한 인증 별명을 선택하고 기본 메소드 사용 선택란을 체크한 다음 적용 단추를 누르십시오.

다음 창에서 설치 옵션 요약 분할창이 표시될 때까지 다음 또는 계속을 누르십시오. **완료**를 누르면 다음 메시지가 표시됩니다. **응용프로그램 name**이 설치되었습니다.

마스터 구성 저장을 눌러서 변경한 내용을 저장하십시오. 그런 다음 엔터프라이즈 응용프로그램 분할창에서 저장을 누르십시오.

### 관련 태스크

36 페이지의 『어댑터를 위한 프로젝트 작성』

어댑터 전개의 첫 번째 태스크는 어댑터를 위한 J2EE 커넥터 프로젝트를 작성하는 것입니다. Adapter for JDBC용 RAR(Resource Adapter Archive) 파일을 WID(WebSphere Integrator Developer)로 가져와야 합니다. 그러면 WID의 작업 공간에 프로젝트가 설정됩니다.

## 서버에 어댑터 구성

어댑터 프로젝트를 위한 프로젝트 EAR 파일이 응용프로그램 서버에 설치되면 어댑터가 특정 데이터베이스 응용프로그램에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 재구성할 수 있습니다. 그런 다음 구성된 어댑터 응용프로그램을 시작할 수 있습니다.

일반적으로 구성 등록 정보는 어댑터 프로젝트를 작성할 때 엔터프라이즈 서비스 찾기 마법사를 사용하여 설정됩니다. WebSphere Process Server 관리 콘솔을 사용하여 등록 정보를 재설정하거나 단순히 등록 정보가 엔터프라이즈 서비스 찾기 마법사에서 설정한 값에 따라 파플레이트되었는지 여부를 적어둘 수 있습니다.

### 1. 어댑터 프로젝트 열기

WebSphere Process Server 관리 콘솔의 구성 분할창에 사용자의 어댑터 프로젝트 이름이 표시됩니다. 관련 항목 표제 아래에서 커넥터 모듈을 선택하십시오. 프로젝트 RAR 파일 이름이 표시됩니다. 파일 이름 옆의 선택란을 누르십시오.

추가 등록 정보 아래에서 자원 어댑터를 선택하십시오.

### 2. 등록 정보 재구성

#### a. J2C 연결 팩토리 등록 정보 값 편집

관리 콘솔에서 어댑터 프로젝트는 일반 등록 정보 > 이름 아래에 표시됩니다. 창 오른쪽의 추가 등록 정보 아래에서 **J2C 연결 팩토리**를 누르십시오.

EJB 프로젝트에서 지정된 JNDI 이름과 함께 어댑터 프로젝트 이름이 표시됩니다. 어댑터의 팩토리 연결 옆의 선택란을 누르십시오.

J2C 연결 팩토리 등록 정보 및 값이 표시됩니다. 필요에 따라 등록 정보의 값을 편집하십시오. 이 등록 정보에 대한 자세한 내용은 『J2C 연결 팩토리 등록 정보』를 참조하십시오.

#### b. J2C 활성화 스펙 등록 정보 값 설정

아직 값이 설정되지 않은 경우 필요에 따라 2a 단계를 수행하여 J2C 활성화 스펙 등록 정보의 값을 설정하십시오.

**중요사항:** 엔터프라이즈 서비스 찾기 마법사를 사용하여 이미 J2C 활성화 스펙 등록 정보를 지정한 경우에는 등록 정보 설정을 지금 변경할 수 없습니다. 응용프로그램 서버에 새 응용프로그램으로 프로젝트를 전개한 후에는 WebSphere Process Server 관리 콘솔을 사용하여 J2C 활성화 스펙 등록 정보를 갱신할 수 없습니다. 설정을 갱신하려 해도 어댑터는 갱신된 값을 인식하지 못합니다. 응용프로그램 서버에 프로젝트를

설치하기 전에 J2C 활성화 스펙 등록 정보를 설정하지 않은 경우에는 이러한 등록 정보 값을 지금 설정할 수 있습니다.

이러한 등록 정보를 설정하려면 추가 등록 정보 아래에서 **J2C 활성화 스펙**을 누르십시오. 이 등록 정보에 대한 자세한 내용은 『J2C 활성화 스펙 등록 정보』를 참조하십시오.

### c. 자원 어댑터 등록 정보 값 편집

추가 등록 정보 아래에서 사용자 정의 등록 정보를 누르십시오. 자원 어댑터 등록 정보의 목록과 값이 표시됩니다. 필요에 따라 등록 정보 값을 편집하십시오. 『자원 어댑터 등록 정보』에서 등록 정보와 설명을 참조하십시오.

주: 사용자 정의 등록 정보 목록의 어댑터 레벨에서 **DatabaseVendor** 등록 정보를 찾으십시오. 이 등록 정보는 응용프로그램을 시작하는 데 필수입니다. 이 등록 정보가 비어 있으면 값을 입력하고 저장하십시오.

## 3. 어댑터 응용프로그램 시작

관리 콘솔의 왼쪽에 있는 패널에서 **응용프로그램 > 엔터프라이즈 응용프로그램**을 누르십시오. **엔터프라이즈 응용프로그램** 분할창에서 응용프로그램의 이름 옆에 있는 선택란을 누르고 **시작**을 누르십시오. 응용프로그램이 시작되었다는 메시지가 표시됩니다. 응용프로그램을 시작할 때 문제점이 발생하면 어댑터 로그 파일에서 오류 설명을 확인하십시오.

특정 메시지에 대한 정보는 IBM WebSphere Adapters Information Center의 『참조』 섹션에서 『메시지』를 참조하십시오. 로깅 및 추적에 대한 자세한 내용은 『문제점 해결』을 참조하십시오.

### 관련 태스크

57 페이지의 『문제점 해결』

이 섹션에서는 문제점을 보고해야 할 경우 IBM Software Support에 접속하는 방법에 대해 설명합니다. 또한 어댑터의 로깅 및 추적 방법을 설명합니다. 마지막으로 어댑터의 Common Event Infrastructure를 사용 가능하게 하는 방법에 대해 설명합니다.

### 관련 참조

80 페이지의 『J2C 연결 팩토리 등록 정보』

J2C 연결 팩토리 등록 정보는 대상 EIS(Enterprise Information System) 인스턴스를 구성하는 데 사용됩니다. 이러한 등록 정보는 아웃바운드 처리에 영향을 주며 J2EE 커넥터 구조 스펙의 ManagedConnectionFactory 인터페이스에 해당합니다.

81 페이지의 『J2C 활성화 스펙 등록 정보』

J2C 활성화 스펙 등록 정보는 메시지 엔드포인트를 활성화합니다. 이러한 등록 정보는 J2EE 커넥터 구조 스펙의 ActivationSpec 인터페이스에 해당합니다.

78 페이지의 『자원 어댑터 등록 정보』

이 구성 등록 정보는 자원 어댑터 레벨에서 정의됩니다.

---

## 문제점 해결

이 섹션에서는 문제점을 보고해야 할 경우 IBM Software Support에 접속하는 방법에 대해 설명합니다. 또한 어댑터의 로깅 및 추적 방법을 설명합니다. 마지막으로 어댑터의 Common Event Infrastructure를 사용 가능하게 하는 방법에 대해 설명합니다.

### 관련 태스크

55 페이지의 『서버에 어댑터 구성』

어댑터 프로젝트를 위한 프로젝트 EAR 파일이 응용프로그램 서버에 설치되면 어댑터가 특정 데이터베이스 응용프로그램에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 재구성할 수 있습니다. 그런 다음 구성된 어댑터 응용프로그램을 시작할 수 있습니다.

## IBM 소프트웨어 지원 문의

IBM 소프트웨어 지원은 제품 결함에 대한 지원을 제공합니다. 이 주제에 기술된 정보는 특별히 WebSphere Adapter에 대한 지원을 요청하는 방법에 대해 자세히 설명합니다.

IBM 소프트웨어 지원에 문의하려면 사용자의 회사가 IBM 소프트웨어 유지보수 계약을 보유하고 있어야 하며 사용자는 문제점을 IBM에 제출할 권한이 있어야 합니다. 필요한 소프트웨어 유지보수 계약의 유형은 현재 보유한 제품의 유형에 따라 달라집니다.

- IBM 분산 소프트웨어 제품(Windows® 또는 UNIX® 운영 체제에서 실행되는 DB2® 및 WebSphere® 제품을 비롯한 Tivoli®, Lotus® 및 Rational® 제품 등)의 경우 다음 방법 중 하나로 Passport Advantage®에 등록하십시오.
  - 온라인: Passport Advantage 웹 페이지로 이동한 후 등록 방법을 누르십시오.
  - 전화: 해당 국가의 번호는 웹에서 IBM 소프트웨어 지원 핸드북의 문의 페이지로 이동한 후 거주 지역명을 누르십시오.
- IBM eServer 소프트웨어 제품(zSeries®, pSeries® 및 iSeries™ 환경에서 실행되는 DB2 및 WebSphere 제품 등)의 경우에는 IBM 영업 담당자 또는 IBM 비즈니스 파트너에게 직접 문의하여 소프트웨어 유지보수 계약을 구매할 수 있습니다. eServer 소프트웨어 제품의 지원에 대한 자세한 내용은 IBM Technical Support Advantage 웹 페이지로 이동하십시오.

필요한 소프트웨어 유지보수 계약의 유형을 확실히 모르는 경우 미국에서는 1-800-IBMSERV(1-800-426-7378)로 전화하고, 기타 국가에서는 웹에서 IBM 소프트웨어 지원 핸드북의 문의 페이지로 이동한 후 사용자의 거주 지역명을 눌러 담당자의 전화번호를 확인하십시오.

1. 문제점을 파생시키는 비즈니스 영향을 파악하십시오. IBM에 문제점을 보고할 때에는 심각도 레벨을 제시해야 합니다. 따라서 보고하는 문제점의 비즈니스 영향을 이해하고 평가해야 합니다. 다음 기준을 사용하십시오.

심각도	설명
심각도 1	심각한 비즈니스 영향: 조작에 중대한 영향을 미치므로 프로그램을 사용할 수 없습니다. 이 경우에는 즉각적인 해결 방법이 필요합니다.
심각도 2	상당한 비즈니스 영향: 프로그램을 사용할 수는 있지만 상당히 제한적입니다.
심각도 3	약간의 비즈니스 영향: 프로그램은 사용할 수 있지만 사소한 기능(조작에 중요하지 않은)을 사용할 수 없습니다.
심각도 4	최소 비즈니스 영향: 조작에 별 영향을 주지 않거나 문제점을 해결할 수 있는 방법이 구현되었습니다.

2. 문제점을 설명하고 배경 정보를 수집하십시오. 가능한 구체적으로 IBM에 문제점을 설명하십시오. IBM 소프트웨어 지원 전문가가 문제점을 효과적으로 해결할 수 있도록 관련된 배경 정보를 모두 포함하십시오. 시간을 절약하려면 다음 질문에 대한 대답을 준비해 두십시오.

- 문제점이 발생했을 때 실행 중이던 소프트웨어 버전은 무엇입니까?
- 문제점 증상에 관련된 로그, 추적 및 메시지가 있습니까? IBM 소프트웨어 지원에서 이 정보에 대해 질문할 수 있습니다.
- 문제점을 다시 발생시킬 수 있습니까? 그렇다면 어떤 단계에서 장애가 발생했습니까?
- 시스템에 대한 변경사항이 있습니까(예: 하드웨어, 운영 체제, 네트워킹 소프트웨어 등)?
- 현재 이 문제점에 대한 해결 방법을 사용하고 있습니까? 그렇다면 문제점을 보고할 때 해결 방법에 대해서도 설명해 주십시오.

3. 문제점을 IBM 소프트웨어 지원에 제출하십시오. 다음 두 가지 방법 중 하나를 사용하여 문제점을 제출할 수 있습니다.

- **온라인:** IBM 소프트웨어 지원 사이트의 문제점 제출 및 추적 페이지로 이동하십시오. 적합한 문제점 제출 도구에 정보를 입력하십시오.
- **전화:** 해당 국가의 번호는 웹에서 IBM 소프트웨어 지원 핸드북의 문의 페이지로 이동한 후 거주 지역명을 누르십시오.

소프트웨어 결함이나 누락 또는 부정확한 문서에 대한 문제점을 제출하는 경우에는 IBM 소프트웨어 지원에서 APAR(Authorized Program Analysis Report)을 작성합니다. APAR은 문제점에 대해 상세히 설명합니다.

가능하다면 IBM 소프트웨어 지원에서는 APAR이 해결되고 수정사항이 전달될 때까지 사용자가 구현할 수 있는 해결 방법을 제공합니다. IBM에서는 매일 IBM 제품 지원 웹 페이지에 해결된 APAR을 게시하여 동일한 문제점을 겪고 있는 다른 사용자에게 도움을 줍니다.

## 로깅 사용

WebSphere Adapter for JDBC는 이벤트 처리 상태를 판별하기 위해 살펴볼 수 있는 로그 파일을 유지합니다. 어댑터와 관련된 모든 이벤트와 오류는 로그 파일에 의해 각 로그 항목에 날짜, 시간 및 이벤트와 함께 추적됩니다. 어댑터는 오류 또는 경고 조건을 만날 때 오류 메시지를 로그하므로 로그 파일은 문제점 해결을 시작할 좋은 소스입니다.

어댑터는 JDBC 드라이버를 통해 데이터베이스와 통신하며, SQL 조회 또는 저장 프로시저 호출의 결과는 SQLException이 될 수 있습니다. 이러한 결과는 캡처되고 로그되어 ResourceException 메시지를 생성합니다. 메시지는 요청 처리 동안에 발생한 오류 또는 이벤트 스토어 내의 오류에 따라 리턴됩니다. 이벤트 처리 중 발생한 오류는 이벤트 관리자가 처리합니다.

어댑터에 대한 로깅은 WebSphere Process Server 관리 콘솔을 통해 사용 가능합니다. 다음 단계를 수행하여 로깅을 사용할 수 있습니다.

### 1. 로깅 및 추적 시작

WebSphere Process Server 관리 콘솔을 시작하십시오. 관리 콘솔에서 문제점 해결 → 로그 및 추적을 선택하십시오.

### 2. 로그 세부사항 레벨 지정

구성요소를 눌러서 각 구성요소의 로그 세부사항 레벨을 지정하거나 그룹을 눌러서 사전 정의된 구성요소 그룹에 로그 세부사항을 지정하십시오.

### 3. 로깅 레벨 선택

"로깅 레벨" 테이블은 관리 콘솔을 통해 설정할 수 있는 여러 로깅 레벨을 설명합니다.

주: 세부사항 레벨 아래의 로그 이벤트를 보려면 진단 추적 서비스가 사용 가능해야 합니다. 세부사항 레벨 이상의 로그 이벤트는 SystemOut 로그, IBM Service 로그 또는 진단 추적 서비스에서 볼 수 있습니다.

### 로깅 레벨

레벨	표시기	설명
심각	F	타스크를 계속할 수 없습니다. 구성요소를 실행할 수 없습니다.
위험	E	타스크를 계속할 수 없습니다. 구성요소는 여전히 실행됩니다. 여기에는 심각한 오류가 예상되는 것들(즉, 자원이 거의 고갈되어 자원 보충을 제안하는 상황 보고 등)이 포함됩니다.

레벨	표시기	설명
경고	W	잠재적 오류 또는 예상되는 오류. 여기에는 진행 장애를 나타내는 조건(예: 잠재적 자원 누출)이 포함됩니다.
감사	A	서버 상태 또는 자원에 영향을 미치는 중요한 이벤트
정보	I	전체 TASK 진행에 대한 개요를 설명하는 일반 정보
구성	C	구성 변경 또는 상태
세부사항	D	서브TASK 진행을 자세히 설명하는 일반 정보

#### 4. 로깅 정보 저장

적용을 눌러서 변경한 내용을 저장하십시오.

##### 관련 TASK

45 페이지의 『연결 등록 정보 설정』

어댑터 프로젝트를 작성한 후, JDBC를 위해 엔터프라이즈 서비스 찾기 마법사를 초기화하고 데이터베이스 인스턴스를 위해 연결 등록 정보 값을 설정해야 합니다.

## 추적 사용

추적은 어댑터 로그 파일이 오류 또는 경고를 캡처하는 레벨을 결정합니다. 추적 레벨을 정의하여 어댑터 처리에 관한 메시지를 추적할 수 있습니다.

추적 레벨은 WebSphere Process Server 관리 콘솔에서 구성할 수 있습니다. 추적 레벨을 사용하고 설정하려면 다음 단계를 수행하십시오.

#### 1. 추적 시작

WebSphere Process Server 관리 콘솔을 시작하십시오. 문제점 해결 → 로그 및 추적을 선택하십시오.

#### 2. 추적 레벨 선택

"추적 레벨" 테이블은 관리 콘솔을 통해 설정할 수 있는 여러 추적 레벨을 설명합니다.

##### 추적 레벨

레벨	표시기	설명
1단계	1	일반 추적. EIS에 연결 구축, EIS의 이벤트를 Business Object로 변환(키 값만), Business Object 처리(키 값만) 등 어댑터가 수행하는 광범위한 조치를 포함합니다.



레벨	표시기	설명
2단계	2	EIS에 대한 다양한 API 호출 및 매개변수 또는 리턴 값 등 어댑터가 수행하는 논리에 대해 더 구체적인 정보를 제공하는 자세한 추적.
3단계	3	가장 자세한 레벨이며 메소드 / 항목 / 종료 / 리턴 값을 포함합니다. 전체 Business Object 덤프가 포함됩니다. 이 레벨에서는 문제점을 디버그하는 데 필요한 모든 자세한 정보가 제공됩니다.

### 3. 추적 정보 저장

적용을 눌러서 변경한 내용을 저장하십시오.

#### 관련 태스크

45 페이지의 『연결 등록 정보 설정』

어댑터 프로젝트를 작성한 후, JDBC를 위해 엔터프라이즈 서비스 찾기 마법사를 초기화하고 데이터베이스 인스턴스를 위해 연결 등록 정보 값을 설정해야 합니다.

## CEI(Common Event Infrastructure) 사용 가능

이 주제에서는 어댑터에 CEI(Common Event Infrastructure)를 사용 가능하게 하는 방법에 대해 설명합니다.

IBM WebSphere Adapter 이벤트 정의 파일을 CEI 카탈로그에 게시해야 이벤트 정의를 설정할 수 있습니다. 이를 수행하기 위한 지시사항은 WebSphere Process Server 웹 사이트 <http://www.ibm.com/software/integration/wps>에 있는 CEI 문서를 참조하십시오.

1. WebSphere 관리 콘솔을 시작하십시오. 문제점 해결> 로그 및 추적>으로 이동한 후 <사용자 서버 이름>을 선택하십시오.
2. 일반 등록 정보에 대한 다양한 옵션이 표시됩니다. 로그 세부사항 레벨 변경을 선택한 후 JCA 구성요소에 대해 **com.ibm.j2ca.\***를 선택하십시오. 다음은 각 어댑터 유형에 대한 서버 구성요소입니다: com.ibm.j2ca.flatfile.\*(WebSphere Adapter for Flat Files) com.ibm.j2ca.jdbc.\*(WebSphere Adapter for JDBC) com.ibm.j2ca.peoplesoft.\*(WebSphere Adapter for PeopleSoft) com.ibm.j2ca.sap.\*(WebSphere Adapter for SAP) com.ibm.j2ca.siebel.\*(WebSphere Adapter for Siebel)
3. 사용자의 어댑터와 일치하는 구성요소를 선택하십시오. 각 어댑터 구성요소에는 로깅과 CEI에 대해 각각 하나씩 두 개의 서버 구성요소가 있으며 다음과 같습니다. *subcomponent name.cei.adapter id subcomponent name.log.adapter id* 예를 들면 com.ibm.j2ca.siebel.cei.<AdapterID1>입니다. 전개된 어댑터의 각 인스턴스에 대해 시스템은 개별 ID를 표시합니다.

4. 사용 가능하게 하려는 CEI 어댑터 ID를 선택하십시오.
5. 드롭 다운 메뉴에서 다음 선택사항을 선택할 수 있습니다. off - CEI를 끕니다. fine- 이벤트 내용을 Empty로 설정하고 CEI를 켭니다. finer- 이벤트 내용을 Digest로 설정하고 CEI를 켭니다. finest- 이벤트 내용을 Full로 설정하고 CEI를 켭니다. all - finest와 동일합니다.

각 이벤트 내용 레벨(Empty, Digest 및 Full)의 의미와 Common Base Event 모델 및 CEI(Common Event Infrastructure)의 사용에 대한 자세한 내용은 WebSphere Process Server InfoCenter, <http://www.ibm.com/software/integration/wps>의 문서를 참조하십시오.

## 예제 파일 및 샘플 응용프로그램 사용

이 섹션에는 엔터프라이즈 서비스 찾기 프로세스의 SCA(Service Component Architecture) 입/출력 파일 및 해당 WSDL(Web Services Description Language) 파일과 같은 아티팩트의 예가 있습니다. 또한 이 섹션에는 어댑터를 사용하여 J2EE 응용 프로그램과 EIS(Enterprise Information System) 사이에 통신하는 방법에 익숙해지도록 도와주는 샘플 응용프로그램이 있습니다.

### 가져오기/내보내기 및 WSDL 파일 예제

SCA(Service Component Architecture) 가져오기 및 내보내기 파일과 해당 WSDL(Web Services Description Language) 파일의 예가 제공됩니다. 이러한 아티팩트는 엔터프라이즈 서비스 찾기 프로세스 동안에 생성됩니다.

#### 내보내기(인바운드) 파일 예제

```
<?xml version="1.0" encoding="UTF-8"?>
<scdl:Export xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:_="http://JDBCEMD/inbound"
xmlns:eis="http://www.ibm.com/xmlns/prod/websphere/scdl/eis/6.0.0"
xmlns:scdl="http://www.ibm.com/xmlns/prod/websphere/scdl/6.0.0"
xmlns:wSDL="http://www.ibm.com/xmlns/prod/websphere/scdl/wSDL/6.0.0"
name="inbound/JDBCInboundInterface">
  <interfaces>
    <interface xsi:type="wSDL:WSDLPortType" portType="_:JDBCInboundInterface"/>
  </interfaces>
  <esbBinding xsi:type="eis:EISExportBinding"
dataBindingType="com.ibm.j2ca.extension.emd.runtime.WBIDataBindingImpl">
    <resourceAdapter name="JDBCEMDApp.IBM JDBC Adapter"
type="com.ibm.j2ca.jdbc.JDBCResourceAdapter">
      <properties/>
    </resourceAdapter>
    <connection type="com.ibm.j2ca.jdbc.inbound.JDBCActivationSpec"
selectorType="com.ibm.j2ca.extension.emd.runtime.WBIFunctionSelectorImpl">
      <properties>
        <BONamespace>http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc</BONamespace>
        <jdbcDriverClass>COM.ibm.db2.jdbc.app.DB2Driver</jdbcDriverClass>
        <databaseURL>jdbc:db2:somedb<databaseURL>
        <password>abcdefg</password>
        <userName>db2admin</userName>
      </properties>
    </connection>
    <methodBinding method="createDb2adminCustomer"
nativeMethod="emitCreateAfterImageDb2adminCustomer"/>
    <methodBinding method="updateDb2adminCustomer"
nativeMethod="emitUpdateAfterImageDb2adminCustomer"/>
  </esbBinding>
</scdl:Export>
```

```

    <methodBinding method="deleteDb2adminCustomer"
    nativeMethod="emitDeleteAfterImageDb2adminCustomer"/>
  </esbBinding>
</scdl:Export>

```

## 가져오기(아웃바운드) 서비스 설명 예제

```

<?xml version="1.0" encoding="UTF-8"?>
<scdl:Import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:_="http://JDBCEMD/outbound"
xmlns:eis="http://www.ibm.com/xmlns/prod/websphere/scdl/eis/6.0.0"
xmlns:scdl="http://www.ibm.com/xmlns/prod/websphere/scdl/6.0.0"
xmlns:wsdl="http://www.ibm.com/xmlns/prod/websphere/scdl/wsdl/6.0.0"
name="outbound/JDBCOutboundInterface">
  <interfaces>
    <interface xsi:type="wsdl:WSDLPortType" portType="_:JDBCOutboundInterface"/>
  </interfaces>
  <esbBinding xsi:type="eis:EISImportBinding"
dataBindingType="com.ibm.j2ca.extension.emd.runtime.WBIDataBindingImpl">
    <resourceAdapter name="JDBCEMDApp.IBM JDBC Adapter"
type="com.ibm.j2ca.jdbc.JDBCResourceAdapter">
      <properties/>
    </resourceAdapter>
    <connection type="com.ibm.j2ca.jdbc.JDBCManagedConnectionFactory"
interactionType="com.ibm.j2ca.jdbc.JDBCInteractionSpec">
      <properties>
        <databaseURL>jdbc:db2:somedb</databaseURL>
        <jdbcDriverClass>COM.ibm.db2.jdbc.app.DB2Driver</jdbcDriverClass>
        <password>abcdefg</password>
        <userName>db2admin</userName>
      </properties>
    </connection>
    <methodBinding method="createDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Create</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="updateDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Update</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="deleteDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Delete</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="retrieveDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>Retrieve</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="retrievealldb2adminCustomer">
      <interaction>
        <properties>
          <functionName>RetrieveAll</functionName>
        </properties>
      </interaction>
    </methodBinding>
    <methodBinding method="applychangesDb2adminCustomer">
      <interaction>
        <properties>
          <functionName>ApplyChanges</functionName>
        </properties>
      </interaction>
    </methodBinding>
  </esbBinding>
</scdl:Import>

```

## WSDL 파일 예제

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="CUSTOMER"
  targetNamespace="http://test/j2c/jdbc/customer"
  xmlns:tns="http://test/j2c/jdbc/customer"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:datans="http://test/j2c/jdbc/customer">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace="http://test/j2c/jdbc/customer"
        schemaLocation="CUSTOMER.xsd">
      </xsd:import>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="CUSTOMERRequest">
    <wsdl:part name="request" element="datans:CUSTOMER"></wsdl:part>
  </wsdl:message>
  <wsdl:portType name="Customer">
    <wsdl:operation name="updateCustomer">
      <wsdl:input message="tns:CustomerRequest"></wsdl:input>
      <wsdl:output message="tns:CustomerRequest"></wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="createCustomer">
      <wsdl:input message="tns:CustomerRequest"></wsdl:input>
      <wsdl:output message="tns:CustomerRequest"></wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="retrieveCustomer">
      <wsdl:input message="tns:CustomerRequest"></wsdl:input>
      <wsdl:output message="tns:CustomerRequest"></wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

## 샘플: 데이터베이스 응용프로그램 갱신

IBM WebSphere Adapter for JDBC와 함께 제공되는 샘플 파일을 사용하여 데이터베이스 응용프로그램에서 특정 파일을 갱신하는 연습을 할 수 있습니다. 응용프로그램 통합자 및 데이터 통합자 역할을 위해 두 개의 단계별 시나리오가 제공됩니다. 역할에 따라 J2EE 응용프로그램 및 EIS(Enterprise Information System)와 통신하는 어댑터 구성, Business Object 생성 및 어댑터 전개를 연습할 수 있습니다.

### 샘플 응용프로그램 시나리오

시나리오	설명	대상
시나리오 1	<ul style="list-style-type: none"> <li>이미 생성된 아티팩트를 제공하고 어댑터가 Business Object를 처리하는 방법을 설명합니다. 이 시나리오에서는 아티팩트를 생성하는 데 엔터프라이즈 서비스 찾기 마법사를 사용하지 않아도 됩니다.</li> <li>응용프로그램 구성요소를 솔루션으로 어셈블하고 이 솔루션의 테스트 및 전개를 준비할 책임자를 위한 것입니다.</li> </ul>	응용프로그램 통합자

시나리오	설명	대상
시나리오 2	<ul style="list-style-type: none"> <li>엔터프라이즈 서비스 찾기 마법사를 사용하여 데이터 베이스 구성요소를 찾고 어댑터가 처리하는 Business Object를 개발하는 방법을 설명합니다.</li> <li>응용프로그램 구성자와 똑같은 책임이 있으며 응용프로그램 개발자를 위해 데이터 원본의 범위에 액세스 할 수 있도록 하는 책임도 있는 사람을 위한 것입니다.</li> </ul>	데이터 통합자

응용프로그램 통합자 및 데이터 통합자에 대한 자세한 내용은 『제품 개요』의 『대상』 주제를 참조하십시오.

### 관련 개념

3 페이지의 『대상』

이 주제에서는 WebSphere Adapter 제품의 사용자를 정의하고 사용자에게 필요한 기술에 대해 상세히 설명합니다.

### 시나리오 1의 프로젝트 파일 생성

시나리오 1의 샘플에서는 필요한 파일들이 모두 제공되므로 생성할 필요가 없습니다. 실제 작업 환경에서는 누군가가 엔터프라이즈 서비스 찾기 마법사를 사용하여 아티팩트를 생성하고 어댑터를 구성해야 합니다.

시나리오를 사용하기 전에 어댑터를 설치하고 『샘플』 폴더에서 샘플 패키지를 추출해야 합니다.

어댑터 설치에 대한 정보는 *IBM WebSphere Adapter* 설치를 참조하십시오.

### 프로젝트 파일 생성

『jdbc/samples/Apps』 폴더로 이동하여 JDBCApp.ear을 찾으십시오.

이 프로젝트의 EAR 파일에는 다음과 같은 파일들이 들어 있습니다.

- 기본으로 로컬 호스트 응용프로그램 서버에 전개된 어댑터의 구성된 인스턴스: CWYBC\_JDBC.rar
- SCA(Service Component Architecture) 모듈 및 다양한 SCA 아티팩트:
  - JDBCInboundInterface.export
  - JDBCInboundInterface.wsdl
  - JDBCOutboundInterface.import
  - JDBCOutboundInterface.wsdl
- WSDL 파일: CustomerInterface.wsdl
- Business object:

- InboundRtasserCustomer.xsd
- InboundRtasserAddress.xsd
- InboundRtasserCustInfo.xsd
- OutboundRtasserCustomer.xsd
- OutboundRtasserAddress.xsd
- OutboundRtasserCustInfo.xsd
- InboundRtasserCustomerBG.xsd
- InboundRtasserAddressBG.xsd
- InboundRtasserCustInfoBG.xsd
- OutboundRtasserCustomerBG.xsd
- OutboundRtasserAddressBG.xsd
- OutboundRtasserCustInfoBG.xsd
- InboundRtasserCustomerContainer.xsd
- InboundRtasserAddressContainer.xsd
- InboundRtasserCustInfoContainer.xsd
- OutboundRtasserCustomerContainer.xsd
- OutboundRtasserAddressContainer.xsd
- OutboundRtasserCustInfoContainer.xsd
- 데이터베이스 테이블을 작성하는 SQL 스크립트:
  - CreateCustomerTable.sql
  - CreateAddressTable.sql
  - CreateCustInfoTable.sql

다음, 타스크 전개 및 구성을 수행합니다.

### 시나리오 2의 프로젝트 파일 생성

시나리오 2에서는 꼭 필요한 파일만 샘플에 들어 있으므로 엔터프라이즈 서비스 찾기 마법사를 사용하여 포괄적인 아티팩트 세트를 생성하는 연습을 할 수 있습니다.

시나리오를 사용하기 전에 어댑터를 설치하고 『샘플』 폴더에서 샘플 패키지를 추출해야 합니다.

어댑터 설치에 대한 정보는 IBM WebSphere Adapter 설치를 참조하십시오.

프로젝트 파일 생성

『jdbc/samples/Apps』 폴더로 이동하여 JDBCApp.ear을 찾으십시오.

이 프로젝트의 EAR 파일에는 다음과 같은 파일들이 들어 있습니다. 이 파일들은 엔터프라이즈 서비스 찾기 마법사를 사용하여 아티팩트를 생성하고 전개 및 사용을 위해 어댑터를 구성할 때 이 시나리오에서 작성할 아티팩트의 예입니다. 시나리오를 완료하면 출력된 것을 이 파일들과 비교해 볼 수 있습니다. 이 파일에는 다음과 같은 샘플 가져오기 파일, 내보내기 파일, WSDL 파일 및 Business Object 파일이 포함됩니다.

- JDBCOutboundInterface.import
- JDBCInbound Interface.export
- JDBCOutboundInterface.wsdl
- JDBCInboundInterface.wsdl
- InboundRtasserCustomer.xsd
- InboundRtasserAddress.xsd
- InboundRtasserCustInfo.xsd
- OutboundRtasserCustomer.xsd
- OutboundRtasserAddress.xsd
- OutboundRtasser.CustInfo.xsd
- InboundRtasserCustomerBG.xsd
- InboundRtasserAddressBG.xsd
- InboundRtasserCustInfoBG.xsd
- OutboundRtasserCustomerBG.xsd
- OutboundRtasserAddressBG.xsd
- OutboundRtasserCustInfoBG.xsd
- InboundRtasserCustomerContainer.xsd
- InboundRtasserAddressContainer.xsd
- InboundRtasserCustInfoContainer.xsd
- OutboundRtasserCustomerContainer.xsd
- OutboundRtasserAddressContainer.xsd
- OutboundrtasserCustInfoContainer.xsd

Business Object에 대한 자세한 내용은『Business Object 개요』를 참조하십시오.

다음, 타스크 전개 및 구성을 수행합니다.

### 시나리오 1의 전개 및 구성

시나리오 1에 필요한 모든 아티팩트가 샘플에서 제공되므로 엔터프라이즈 서비스 찾기 마법사를 사용하여 작성할 필요가 없습니다.

샘플은 이미 구성된 어댑터의 인스턴스를 제공합니다. 다음 프로시저에서 Business Object 응용프로그램 특정 정보(ASI)를 환경에 맞게 갱신해야 합니다. 그리고 구성 등록 정보를 EIS(Enterprise Information System)의 데이터베이스 인스턴스에 일치하도록 변경해야 합니다.

#### 1. WebSphere Integration Developer 시작

자세한 내용은 <http://www.ibm.com/software/integration/wid>에 있는 WebSphere Integration Developer 사용자 안내서를 참조하십시오.

#### 2. Business Integration 모듈 작성

WID에서 Business Integration 영역으로 이동하십시오. Business Integration 분할창을 마우스 오른쪽 단추로 누르십시오. 새로 작성 > 모듈을 누르십시오. 새 모듈 창에 모듈 이름을 위한 JDBC를 입력하십시오. 완료를 누르십시오.

#### 3. RAR(Resource Adapter Archive) 파일 가져오기

파일 > 가져오기를 누르십시오. 선택 창에서 가져올 소스로 RAR 파일을 선택하고 다음을 누르십시오. 이 작업은 CWYBC\_JDBC.rar을 WID로 가져옵니다.

커넥터 가져오기 창의 커넥터 파일 필드에서, 이전 단계에서 작성한 모듈의 경로와 이름(예: `install dir \wjdbc\deploy\CWYBC_JDBC.rar`)을 찾아보십시오.

EAR 프로젝트에 모듈 추가 선택란이 체크되었는지 확인하십시오. EAR 프로젝트 필드에서 JDBCApp를 선택하십시오. 완료를 누르십시오.

Business Integration 영역의 Business Integration 분할창에서 JDBC 프로젝트를 마우스 오른쪽 단추로 누르고 종속성 편집기 열기를 선택하십시오.

회살표를 눌러서 J2EE 섹션을 펼치십시오. 추가 단추를 누르십시오. J2EE 프로젝트 선택 창에서 CWBC\_JDBC를 선택하고 확인을 누르십시오.

메뉴에서 파일 > 저장을 누르십시오.

#### 4. Java 빌드 경로에 JDBC 드라이버 추가

JDBC 드라이버에 대한 참조를 프로젝트에 추가해야 합니다.

WID의 Java 영역에서 CWBC\_JDBC Connector Project를 마우스 오른쪽 단추로 누르십시오. 등록 정보를 선택하십시오.

외부 .jar 파일을 추가하려면 Java 빌드 경로 단추를 누르십시오. 라이브러리 탭을 선택하고 외부 Jar 추가를 누르십시오. 파일 시스템 창에서 JDBC 드라이버로 이동하여 .jar 파일을 선택하십시오.

확인을 눌러서 변경한 내용을 저장하십시오.

#### 5. Business Integration 모듈에 Business Object 추가



JDBCApp.ear을 압축 해제해야 합니다.

그런 다음 .ear 파일에서 .jar 파일을 추출해야 합니다. JDBCAPP.ear에서 #workspace#JDBC로 JDBC.jar을 추출하십시오.

6. Business Object 응용프로그램 특정 정보 변경
  - a. Business Integration 영역에서 **JDBC** 모듈을 마우스 오른쪽 단추로 누르고 새로 고치기를 선택하십시오.
  - b. 데이터 유형 옆의 **+**를 누르십시오. Business Object **InboundRtasserCustomer.xsd**를 마우스 오른쪽 단추로 누르십시오.
  - c. 함께 열기 > **Business Object** 편집기를 누르십시오. 등록 정보 탭을 선택하십시오. 응용프로그램 정보 탭을 선택하십시오.
  - d. **ASI** 등록 정보 분할창에서, CUSTOMER 테이블이 작성된 데이터베이스의 스키마 이름을 사용하여 jdbcasi:TableName의 값을 **RTASSER.CUSTOMER**에서 **schema.CUSTOMER**로 변경하십시오. 메뉴에서 파일 > 저장을 누르십시오.

다음 Business Object에 대해 6a단계에서 6d단계까지 반복하십시오.

- InboundRtasserAddress
- InboundRtasserCustInfo
- OutboundRtasserCustomer
- OutboundRtasserAddress
- OutboundRtasserCustInfo

## 7. 인증 정보 제공

WebSphere Process Server 관리 콘솔을 사용하여 인증 별명을 작성해야 합니다.

관리 콘솔을 여십시오. 왼쪽 패널에서 보안 > 글로벌 보안을 누르십시오. 글로벌 보안 창의 구성 분할창에서 오른쪽의 인증 표제로 이동하십시오. **JAAS** 구성을 누르십시오. 그런 다음 **J2C** 인증 데이터 링크를 누르십시오.

글로벌 보안 > **J2EE Connector Architecture(J2C)** 인증 데이터 항목 창에서 새로 작성을 누르십시오. 별명 필드에 별명을 입력하십시오. 데이터베이스에 연결할 수 있는 사용자 ID와 암호를 입력하십시오. 확인을 누르십시오. 그런 다음 저장을 눌러서 인증 정보를 저장하십시오.

## 8. 구성 등록 정보 값 변경

데이터베이스 인스턴스에 일치하도록 구성 등록 정보 값을 변경해야 합니다. **WID**의 Business Integration 영역의 Business Integration 분할창에서 **JDBC** 모듈을 누르십시오. **Inbound/JDBCInboundInterface**를 두 번 누르십시오. 어셈블리 편집기가 열립니다.

등록 정보 탭에서 바인딩 탭을 누른 다음 연결 탭을 누르십시오.

**ActivationSpecProperties**를 펼치십시오. 데이터베이스 구성에 따라 J2C 활성 스펙에 대해 UserName, Password, DatabaseURL 및 JDBCDriverClass 등록 정보를 알맞은 값으로 설정하십시오.

연결 탭에서 인증 등록 정보를 펼치십시오. **J2C** 인증 데이터 항목 필드에 이전 단계에서 작성한 인증 별명을 입력하십시오. 메뉴에서 파일 > 저장을 누르십시오.

Outbound/JDBCOutboundInterface 오브젝트에 대해 이 프로시저를 반복해야 합니다. J2C 연결 팩토리(관리 연결 팩토리)의 UserName, Password, DatabaseURL 및 JDBCDriverClass 등록 정보 값을 갱신하십시오.

## 9. 응용프로그램 서버에 프로젝트 파일 전개

WebSphere Integration Developer에서 WebSphere Process Server로 프로젝트의 EAR 파일을 내보내야 합니다.

### a. EAR 파일 내보내기

WID 메뉴에서 파일 > 가져오기를 누르십시오. EAR 파일을 선택하고 다음을 누르십시오. **EAR** 내보내기 창에서 **EAR** 프로젝트 필드로 이동한 다음 **JDBCApp**를 입력하십시오. 대상 경로 및 파일 이름을 설정하십시오. 선택란 세 개를 모두 체크하고 완료를 누르십시오.

### b. 응용프로그램 서버에 EAR 파일 설치

관리 콘솔을 사용하여 응용프로그램 > 새 응용프로그램 설치를 누르십시오. 찾아보기 단추를 누르고 EAR 파일을 선택한 후 다음을 누르십시오.

**7단계.** 자원에 자원 참조 매핑이 될 때까지 다음을 계속 누르십시오. 이전에 작성한 인증 별명을 선택하고 선택란을 누른 다음 적용 단추를 누르십시오.

다음 화면에서 설치 옵션 요약 창이 표시될 때까지 다음 또는 계속을 누르십시오. 그런 다음 완료를 누르십시오.

응용프로그램 **SampleApp**가 설치되었습니다라는 메시지가 표시됩니다. 마스터 구성 저장을 눌러서 변경한 내용을 저장하십시오. 그런 다음 저장을 누르십시오.

이제 샘플 응용프로그램을 실행할 수 있습니다.

## 시나리오 2의 전개 및 구성

시나리오 2에서는 어댑터 연결 등록 정보를 설정하고 Business Object를 생성합니다. 프로젝트를 EAR 파일로 내보내고 응용프로그램 서버에 프로젝트를 전개하며 구성 등록 정보를 재설정합니다.

### 1. WebSphere Integration Developer 시작

자세한 내용은 <http://www.ibm.com/software/integration/wid>에 있는 WebSphere Integration Developer 사용자 안내서를 참조하십시오.

## 2. Business Integration 모듈 작성

WID에서 Business Integration 영역으로 이동하십시오. Business Integration 분할창을 마우스 오른쪽 단추로 누르십시오. 새로 작성 > 모듈을 누르십시오. 새 모듈 창에 모듈 이름을 위한 **JDBC**를 입력하십시오. 완료를 누르십시오.

## 3. 어댑터 .RAR 파일 가져오기

WID(WebSphere Integrator Developer)에서 파일 > 가져오기를 눌러서 어댑터 .RAR 파일을 가져오십시오. 가져올 RAR 파일은 jdbc/deploy/CWYBC\_JDBC.rar입니다.

선택 창에서 가져올 소스로 **RAR** 파일을 선택하십시오.

커넥터 가져오기 창의 커넥터 파일 필드에서, 이전 단계에서 작성한 모듈의 경로와 이름(예: *install dir/jdbc/deploy/CWYBC\_JDBC.rar*)을 찾아보십시오.

**EAR** 프로젝트에 모듈 추가 선택란이 체크되었는지 확인하십시오. **EAR** 프로젝트 필드에서 **JDBCApp**를 선택하십시오. 완료를 누르십시오.

## 4. Java 빌드 경로에 JDBC 드라이버 추가

JDBC 드라이버에 대한 참조를 프로젝트에 추가해야 합니다.

WID의 **Java** 영역에서 **CWBC\_JDBC Connector Project**를 마우스 오른쪽 단추로 누르십시오. 등록 정보를 선택하십시오.

외부 .jar 파일을 추가하려면 **Java** 빌드 경로 단추를 누르십시오. 라이브러리 탭을 선택하고 외부 **Jar** 추가를 누르십시오. 파일 시스템 창에서 **JDBC** 드라이버로 이동하여 .jar 파일을 선택하십시오.

확인을 눌러서 변경한 내용을 저장하십시오.

## 5. 인증 정보 제공

WebSphere Process Server 관리 콘솔을 여십시오. 왼쪽 패널에서 보안 > 글로벌 보안을 누르십시오. 글로벌 보안 창의 구성 분할창에서 오른쪽의 인증 표제로 이동하십시오. **JAAS** 구성을 누르십시오. 그런 다음 **J2C** 인증 데이터 링크를 누르십시오.

글로벌 보안 > **J2EE Connector Architecture(J2C)** 인증 데이터 항목 창에서 새로 작성을 누르십시오. 별명 필드에 별명을 입력하십시오. 데이터베이스에 연결할 수 있는 사용자 ID와 암호를 입력하십시오. 확인을 누르십시오. 저장을 눌러서 인증 정보를 저장하십시오.

## 6. 어댑터 연결 등록 정보 설정

- a. WID에서, JDBC 커넥터 프로젝트가 있는 **Business Integration** 탭에서 해당 프레임에 마우스 오른쪽 단추로 누르십시오. 팝업 메뉴에서 새로 작성 > 엔터프라이즈 서비스 찾기를 선택하십시오.
- b. 엔터프라이즈 서비스 자원 어댑터 선택 창에서 **JDBC EMD Adapter**를 선택하고 다음을 누르십시오.
- c. 찾기 에이전트 설정 구성 창의 연결 구성 등록 정보 필드에 데이터베이스 정보를 입력하십시오. 이 샘플에서는 네 가지 연결 등록 정보 설정만 필수입니다. 이러한 등록 정보에 대한 자세한 내용은 『참조』 섹션의 『연결 등록 정보』를 참조하십시오. 다음 테이블은 찾기 에이전트 초기화를 위한 예제 값을 표시합니다.

### 메타데이터 찾기 연결 등록 정보

등록 정보	예제 값
UserName	db2admin
암호	XXXXXXXXXX
DatabaseURL	jdbc:db2:mytestdb
JDBCDriverClass	COM.ibm.db2.jdbc.app.DB2Driver

다음을 누르십시오.

## 7. 데이터베이스 오브젝트 조회

연결 등록 정보를 구성한 다음 데이터베이스 오브젝트 조회를 실행할 수 있습니다. 메타데이터 트리 구조를 찾아보면 EIS(Enterprise Information System)에서 오브젝트의 구조를 이해할 수 있고 서비스 설명에 필요한 오브젝트를 선택할 수 있습니다.

여기에서 필터 등록 정보를 지정하여 트리 구조에 표시될 스키마 목록을 축소할 수 있습니다. 필터 등록 정보를 지정하는 프로시저는 『데이터베이스 오브젝트 조회』를 참조하십시오.

엔터프라이즈 서비스 찾기 창에서 조회 실행 단추를 누르십시오. 메타데이터 오브젝트가 위쪽 분할창에 표시됩니다.

조회할 오브젝트를 강조하고 추가를 누르십시오. 선택한 오브젝트가 아래쪽 분할창에 표시됩니다. 제거를 누르면 선택된 오브젝트가 제거됩니다.

오브젝트 선택을 완료하였으면 다음을 누르십시오.

## 8. 선택 등록 정보 지정

데이터베이스 오브젝트를 선택하였으면 가져오기 및 내보내기 파일을 위해 선택 등록 정보의 값을 지정해야 합니다.

Namespace 등록 정보는 처음에 모든 Business Object의 기본값으로 설정됩니다.

서비스 유형(인바운드 또는 아웃바운드) 및 해당 조작 선택에 대해 『선택 등록 정보 설정』에서 설명합니다. 선택 등록 정보를 지정한 후 다음을 누르십시오.

#### 9. 어댑터 프로젝트 저장

모듈 이름 **JDBC**는 아티팩트 생성 창에 자동으로 표시됩니다. 선택란 설정을 변경하지 마십시오. 새로 작성을 눌러서 아티팩트를 저장할 새 Business Integration 모듈을 작성하십시오.

새 모듈 창에 모듈 이름이 표시됩니다. 기본 폴더 이름이 선택됩니다. 여기에 서비스 설명이 저장됩니다.

**J2C** 인증 데이터 항목 필드에 이전에 작성한 인증 별명을 입력하십시오.

완료를 누르십시오.

출력을 샘플 파일과 비교할 수 있습니다. Business Integration 영역의 Business Integration 분할창에서 **JDBC** 모듈 옆의 +를 눌러서 펼치십시오. 데이터 유형 폴더 옆의 +를 누르십시오. 모든 Business Object의 목록이 표시됩니다.

#### 10. 응용프로그램 서버에 프로젝트 파일 전개

프로젝트 파일은 WebSphere Integration Developer 작업 공간의 J2EE Connector 프로젝트에 있습니다. 이 파일을 로컬 파일 시스템으로 내보내야 합니다. 그런 다음, 응용프로그램 서버에 EAR 파일을 업로드하고 설치하기 전에 JCA 커넥터 보안을 위한 인증 정보를 제공해야 합니다.

##### a. EAR 파일 내보내기

WebSphere Integration Developer 선택 창의 내보내기 화면에서 EAR 파일 **JDBCApp**를 선택하고 다음을 누르십시오.

**EAR** 내보내기 창에서 Business Integration 모듈 및 EAR 파일을 작성할 위치를 선택해야 합니다. **EAR** 프로젝트 필드에서 모듈을 선택하십시오. EAR 파일을 작성할 위치를 입력하거나 찾아보십시오. 선택란을 모두 체크하고 완료를 누르십시오.

##### b. 응용프로그램 서버에 EAR 파일 설치

관리 콘솔을 사용하여 왼쪽 분할창에서 응용프로그램 > 새 응용프로그램 설치를 누르십시오. 찾아보기 단추를 누르고 EAR 파일을 선택한 후 다음을 누르십시오.

7단계. 자원에 자원 참조 맵핑이 될 때까지 다음을 계속 누르십시오. 이전에 작성한 인증 별명을 선택하고 기본 메소드 사용 선택란을 체크한 다음 적용 단추를 누르십시오.

다음 화면에서 설치 옵션 요약 창이 표시될 때까지 다음 또는 계속을 누르십시오. 그런 다음 완료를 누르십시오.

응용프로그램 **SampleApp**가 설치되었습니다라는 메시지가 표시됩니다. 마스터 구성 저장을 눌러서 변경한 내용을 저장하십시오. 그런 다음 엔터프라이즈 응용프로그램 분할창에서 저장을 누르십시오.

## 11. 구성 등록 정보 값 변경

UserName, Password, DatabaseURL 및 JDBCDriverClass를 위한 어댑터 등록 정보는 이미 기본값으로 설정되었으며, 관리 콘솔을 사용하여 데이터베이스 인스턴스에 일치하도록 재구성해야 합니다.

인바운드 처리인 경우, J2C 활성화 스펙 등록 정보를 수정해야 합니다. 아웃바운드 처리인 경우, J2C 연결 팩토리 등록 정보를 변경해야 합니다. 샘플에서는 다른 등록 정보 값을 추가로 재구성할 필요가 없습니다. 자원 어댑터 구성 등록 정보를 원하는 설정으로 변경할 수도 있습니다. 이러한 등록 정보의 자세한 내용은 『참조』 섹션의 『구성』 등록 정보를 참조하십시오.

### a. 어댑터 프로젝트 파일 열기

관리 콘솔의 엔터프라이즈 응용프로그램 창에 있는 구성 탭에서 어댑터 프로젝트 이름을 누르십시오. 관련 항목 표제 아래에서 커넥터 모듈을 선택하십시오. 프로젝트 RAR 파일 이름이 표시됩니다. 파일 이름 옆의 선택란을 누르십시오.

구성 탭의 추가 등록 정보 표제 아래에서 자원 어댑터를 선택하십시오.

### b. 등록 정보 재구성

어댑터 프로젝트는 일반 등록 정보 분할창의 이름 아래에 표시됩니다. 추가 등록 정보 아래에서 등록 정보 범주 중 하나(J2C 활성화 스펙, J2C 연결 팩토리 또는 사용자 정의 등록 정보)를 눌러서 네 가지 등록 정보 값을 편집하십시오. 사용자 정의 등록 정보는 자원 어댑터 등록 정보입니다.

이제 샘플 응용프로그램을 실행할 수 있습니다.

### 관련 태스크

47 페이지의 『데이터베이스 오브젝트 조회』

연결 등록 정보를 구성한 다음 데이터베이스 오브젝트 조회를 실행할 수 있습니다.

메타데이터 트리 구조를 찾아보면 EIS(Enterprise Information System)에서 오브젝트의 구조를 이해할 수 있고 서비스 설명에 필요한 오브젝트를 선택할 수 있습니다.

50 페이지의 『선택 등록 정보 설정』

데이터베이스 오브젝트를 선택하였으면 가져오기 및 내보내기 파일을 위해 선택 등록 정보의 값을 지정해야 합니다.

관련 참조

82 페이지의 『연결 등록 정보』

연결 등록 정보는 엔터프라이즈 서비스 찾기 마법사가 대상 EIS(Enterprise Information System) 인스턴스에 연결하는 데 사용됩니다.

77 페이지의 『구성 등록 정보』

이 등록 정보 그룹에는 특정 데이터베이스 응용프로그램에 통신 채널을 설정하기 위해 어댑터에서 사용한 속성이 포함됩니다.

### 샘플 응용프로그램 실행

이 타스크에서는 어댑터 샘플 응용프로그램을 실행하여 인바운드 조작 중 이벤트의 처리를 테스트합니다. 이벤트 출력 값의 정확성을 확인할 수 있습니다.

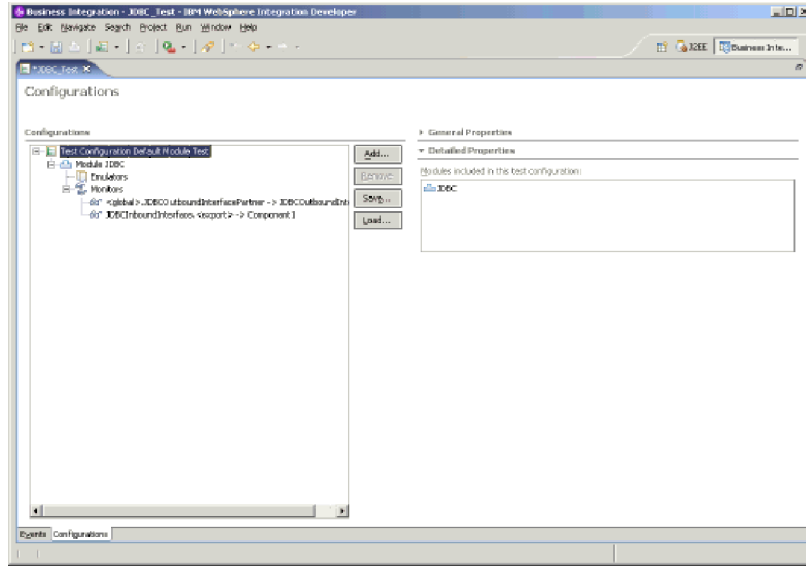
WID(WebSphere Integration Developer) Test Client를 사용하여 샘플을 실행합니다.

#### 1. Business Integration 모듈 및 서버 선택

WID의 Business Integration 영역의 Business Integration 탭에서 **JDBC** 모듈을 선택하십시오. 테스트 > 연결을 마우스 오른쪽 단추로 누르십시오.

WID Test Client는 이벤트 창을 표시합니다. 오른쪽의 기본 모듈 테스트는 구성 필드에 표시됩니다. 모듈 필드는 **JDBC**를 표시합니다.

왼쪽 아래에 있는 구성 탭을 누르십시오. 구성 창이 표시됩니다. 모니터 옆의 **+**를 누르면 내보낼 **JDBCInboundInterface <export> Component1** 모니터가 있는지 알 수 있습니다. 시나리오 1을 완료하였으면 이 모니터 파일이 생성되었습니다. 시나리오 2를 완료하였으면 이 파일은 9단계, 『어댑터 프로젝트 저장』에서 작성되었습니다.



왼쪽 아래에 있는 이벤트 탭을 누른 다음 계속을 누르십시오. 전개 위치 선택 팝업 창이 표시됩니다. 응용프로그램을 실행할 응용프로그램 서버를 선택해야 합니다. **WebSphere Process Server v6.0**을 선택하고 완료를 누르십시오. 통합 테스트 클라이언트 시작 팝업 창이 표시됩니다.

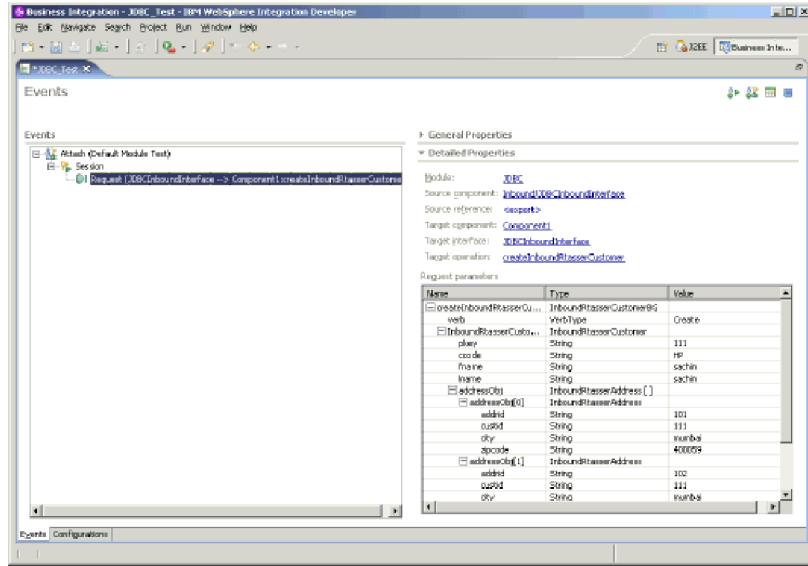
## 2. 이벤트 처리 테스트

이벤트 처리를 테스트하기 위해 샘플 응용프로그램 이벤트 테이블에 이벤트를 수동으로 삽입할 수 있습니다. 다음은 SQL 문을 사용하여 이벤트 스토어에 이벤트를 삽입하는 예입니다.

```
INSERT INTO wbia_jdbc_eventstore (object_key, object_name, object_function,
event_priority, event_status)
VALUES ('11234', 'InboundRtasserCustomerBG', 'Create', 1, 0)
```

모니터되는 구성요소가 이벤트를 수신하면 이벤트 창에 항목이 표시됩니다. 아래의 이벤트 창 예를 참조하십시오.





## 참조

이 섹션에서는 구성, 연결 및 오브젝트 선택을 위한 등록 정보에 대해 설명합니다.

### 관련 개념

15 페이지의 『트랜잭션 관리』

Adapter for JDBC는 로컬 및 XA 트랜잭션을 모두 지원합니다.

## 구성 등록 정보

이 등록 정보 그룹에는 특정 데이터베이스 응용프로그램에 통신 채널을 설정하기 위해 어댑터에서 사용한 속성이 포함됩니다.

등록 정보 유형은 다음과 같습니다.

- 자원 어댑터
- J2C 연결 팩토리
- J2C 활성화 스펙

### 관련 태스크

50 페이지의 『어댑터 프로젝트 저장』

선택 등록 정보를 지정한 후에는 어댑터가 특정 데이터베이스에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 구성하십시오. 여기에는 자원 어댑터, J2C 연결 팩토리, J2C 활성화 스펙 및 양방향 등록 정보가 포함됩니다. 또한 모든 아티팩트 및 등록 정보 값이 저장될 수 있는 새 Business Integration 모듈을 작성해야 합니다.

70 페이지의 『시나리오 2의 전개 및 구성』

시나리오 2에서는 어댑터 연결 등록 정보를 설정하고 Business Object를 생성합니다. 프로젝트를 EAR 파일로 내보내고 응용프로그램 서버에 프로젝트를 전개하며 구성 등록 정보를 재설정합니다.

## 자원 어댑터 등록 정보

이 구성 등록 정보는 자원 어댑터 레벨에서 정의됩니다.

『자원 어댑터 등록 정보』 테이블에서 이러한 구성 등록 정보에 대해 설명합니다.

### 자원 어댑터 등록 정보

등록 정보	유형	세계화	설명
BONamespace	문자열	아니오	Business Object 정의를 위한 이름 공간은 이 어댑터가 사용합니다. 이 값은 메타데이터 찾기 프로세스 동안에 사용자가 제공하는 값입니다. 이 등록 정보는 필수입니다.
DatabaseVendor	문자열	아니오	특별 처리를 위해 어댑터에서 사용하는 RDBMS가 무엇인지 지정합니다. IBM DB2, Oracle 또는 Microsoft SQL Server 데이터베이스를 사용하는 경우에는 값을 DB2, Oracle 또는 MSSQLServer로 설정하십시오. 다른 데이터베이스를 사용할 때는 해당 데이터베이스의 이름으로 값을 설정하십시오. 기본이 아닌 데이터베이스를 사용할 때는 적합한 드라이버가 로드되어 있는지 확인하십시오. 이 등록 정보가 기타로 설정되는 경우에는 어댑터에서 드라이버를 찾아 사용할 데이터베이스를 결정합니다. 어댑터가 제대로 처리되기 위해서는 값이 있어야 합니다.
LogFileName	문자열	아니오	로그 파일의 전체 경로. 이 등록 정보는 필수입니다.
LogNumberOfFiles	정수	아니오	사용할 로그 파일의 수. 한 로그 파일이 최대 크기가 되면 다른 로그 파일이 시작됩니다. 값을 지정하지 않으면 이 값은 1로 설정됩니다.

등록 정보	유형	세계화	설명
LogMaxFileSize	정수	아니오	로그 파일의 크기(KB). 값을 지정하지 않으면 파일에 최대 크기가 없습니다.
PingQuery	문자열	아니오	데이터베이스에 대한 올바른 연결을 테스트하는 데 사용되는 SQL 조회
QueryTimeOut	정수	예	모든 SQL 문에 대한 QueryTimeOut를 지정된 수(초)로 설정합니다. 값이 지정되지 않으면 조회에 제한시간이 설정되지 않습니다. 조회가 지정한 시간보다 오래 걸리면 데이터베이스에서 캡처된 SQL 예외를 만들어냅니다. 연관된 메시지는 로그 파일에 로그됩니다.
ReturnDummyBO ForSP	부울	아니오	결과 세트가 비어 있을 때 출력 매개변수를 리턴하는 데 사용됩니다. RetrieveSP의 경우, 결과 세트가 리턴됩니다. 결과 세트가 비어 있으면 Business Object가 작성되지 않고 프로시저 호출에서 리턴한 출력 매개변수를 검색할 수 없습니다. 이 등록 정보 값이 true이면 출력의 값이 있는 더미 Business Object와 해당 속성에 제공된 입/출력(I/O) 매개변수가 리턴됩니다. 기본값은 false입니다.
TraceFileName	문자열	아니오	추적 파일의 전체 경로. 이 등록 정보는 필수입니다.
TraceNumberOfFiles	정수	아니오	사용할 추적 파일의 수. 한 추적 파일이 최대 크기가 되면 다른 추적 파일이 시작됩니다. 값을 지정하지 않으면 이 값은 1로 설정됩니다.
TraceFileSizeMax	정수	아니오	추적 파일의 크기(KB). 값을 지정하지 않으면 파일에 최대 크기가 없습니다.

## 관련 태스크

55 페이지의 『서버에 어댑터 구성』

어댑터 프로젝트를 위한 프로젝트 EAR 파일이 응용프로그램 서버에 설치되면 어댑터가 특정 데이터베이스 응용프로그램에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 재구성할 수 있습니다. 그런 다음 구성된 어댑터 응용프로그램을 시작할 수 있습니다.

## J2C 연결 팩토리 등록 정보

J2C 연결 팩토리 등록 정보는 대상 EIS(Enterprise Information System) 인스턴스를 구성하는 데 사용됩니다. 이러한 등록 정보는 아웃바운드 처리에 영향을 주며 J2EE 컨넥터 구조 스펙의 ManagedConnectionFactory 인터페이스에 해당합니다.

J2C 연결 팩토리는 연결 풀링을 관리합니다. 어댑터를 통해 응용프로그램에서 단일 JDBC 응용프로그램 인스턴스로 아웃바운드 연결에 대한 구성 정보를 제공합니다.

『J2C 연결 팩토리 등록 정보』 테이블에서는 J2C 연결 팩토리에 관련된 JDBC 구성 등록 정보를 정의합니다.

## J2C 연결 팩토리 등록 정보

등록 정보	유형	세계화	설명
AutoCommit	부울	아니오	연결에 설정될 AutoCommit 값
DatabaseURL	문자열	예	데이터베이스에 연결하는 데 사용되는 데이터베이스 URL. 이것은 양방향 언어에 대해 사용 가능합니다.
JDBCDriverClass	문자열	아니오	데이터베이스에 연결하는 데 사용되는 드라이버의 JDBC 드라이버 클래스
암호	문자열	예	해당 사용자 이름에 대한 암호. 이것은 양방향 언어에 대해 사용 가능합니다.
UserName	문자열	예	EIS(Enterprise Information System)에 로그인하는 사용자 이름. 이것은 양방향 언어에 대해 사용 가능합니다.
XADataSourceName	문자열	아니오	데이터베이스에 대해 XA 연결을 설정하는 데이터 원본 이름.

등록 정보	유형	세계화	설명
XADatabaseName	문자열	예	XA 연결에 사용되는 데이터베이스 이름. DB2 데이터베이스를 사용 중이면 이 등록 정보가 설정되어야 합니다.

### 관련 개념

15 페이지의 『트랜잭션 관리』

Adapter for JDBC는 로컬 및 XA 트랜잭션을 모두 지원합니다.

### 관련 태스크

55 페이지의 『서버에 어댑터 구성』

어댑터 프로젝트를 위한 프로젝트 EAR 파일이 응용프로그램 서버에 설치되면 어댑터가 특정 데이터베이스 응용프로그램에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 재구성할 수 있습니다. 그런 다음 구성된 어댑터 응용프로그램을 시작할 수 있습니다.

## J2C 활성화 스펙 등록 정보

J2C 활성화 스펙 등록 정보는 메시지 엔드포인트를 활성화합니다. 이러한 등록 정보는 J2EE 커넥터 구조 스펙의 ActivationSpec 인터페이스에 해당합니다.

『J2C 활성화 스펙 등록 정보』 테이블에서는 메시지 엔드포인트 활성화에 관련된 JDBC 어댑터 특정 구성 등록 정보를 설명합니다.

### J2C 활성화 스펙 등록 정보

등록 정보	유형	세계화	설명
AutoCreateEDT	부울	예	EDT 테이블이 없는 경우 EDT가 자동으로 작성해야 하는지 여부를 표시하는 플래그. 기본값은 true입니다.
DatabaseURL	문자열	예	EIS에 연결을 작성하는 드라이버 특정 URL. 이것은 양방향 언어에 대해 사용 가능합니다.
전달 유형	문자열	예	ORDERED 또는 UNORDERED. 이것은 이벤트를 발행할 순서를 결정합니다. Ordered는 한 번에 하나씩을 의미하며 Unordered는 동시에 모두를 의미합니다. 기본값은 ORDERED입니다.
EDTDatabaseName	문자열	예	이벤트 분배 테이블의 데이터베이스 이름.
EDTDriverName	문자열	예	인바운드 이벤트를 위해 이벤트 분배 테이블에 연결하는 데 사용하는 XA 데이터베이스 드라이버의 이름. 예: com.ibm.db2j.DB2jXADataSource 값이 없으면 이벤트 관리자는 복구를 수행할 수 없습니다.
EDTSchemaName	문자열	예	이벤트 분배 테이블의 스키마 이름.
EDTTableName	문자열	예	이벤트 분배 테이블을 나타내는 테이블의 이름.

등록 정보	유형	세계화	설명
EDTUserName	문자열	예	이벤트 분배 테이블의 데이터베이스에 연결하는 데 필요한 사용자 이름.
EDTUserPassword	문자열	예	이벤트 분배 테이블의 데이터베이스에 연결하는 데 필요한 암호.
EventOrderBy	문자열	아니오	이벤트가 검색 및 처리되는 순서. 기대 값은 선택표로 구분된 이벤트 테이블의 열이 됩니다. 이것은 양방향 언어에 대해 사용 가능합니다.
EventTableName	문자열	예	인바운드 처리를 위해 EIS에서 생성한 이벤트가 들어 있는 EIS의 테이블.
JDBCDriverClass	문자열	아니오	EIS에 연결하는 데 사용되는 JDBC 드라이버 클래스.
암호	문자열	예	EIS에서 이벤트를 검색하도록 사용자에게 권한을 부여하는 암호. 이것은 양방향 언어에 대해 사용 가능합니다.
PollPeriod	0보다 크거나 0과 같은 정수.	예	새 인바운드 이벤트를 위해 EIS 이벤트를 스토어를 폴하는 밀리초 단위 비율. 0이면 어댑터는 주기가 돌아오기를 기다리지 않습니다. 폴 주기는 고정 비율로 설정되어, 폴 주기의 실행이 연기되는 경우(즉, 이전 폴 주기가 완료 예상 시간보다 오래 걸리는 경우) 다음 주기가 즉시 발생하여 따라잡음을 의미합니다. 이것은 필수 등록 정보입니다. 기본값은 500입니다.
PollQuantity	0보다 큰 정수.	예	이 등록 정보는 폴 주기당 각 엔드포인트에서 전달할 이벤트의 수를 결정하는 데 사용됩니다. 이것은 필수 등록 정보입니다.
Username	문자열	예	인바운드 이벤트를 위해 EIS에 로그인하는 사용자 이름. 이것은 양방향 언어에 대해 사용 가능합니다.

### 관련 태스크

55 페이지의 『서버에 어댑터 구성』

어댑터 프로젝트를 위한 프로젝트 EAR 파일이 응용프로그램 서버에 설치되면 어댑터가 특정 데이터베이스 응용프로그램에 대한 통신 채널을 설정하는 데 사용하는 등록 정보를 재구성할 수 있습니다. 그런 다음 구성된 어댑터 응용프로그램을 시작할 수 있습니다.

## 연결 등록 정보

연결 등록 정보는 엔터프라이즈 서비스 찾기 마법사가 대상 EIS(Enterprise Information System) 인스턴스에 연결하는 데 사용됩니다.

### 관련 태스크

70 페이지의 『시나리오 2의 전개 및 구성』

시나리오 2에서는 어댑터 연결 등록 정보를 설정하고 Business Object를 생성합니다. 프로젝트를 EAR 파일로 내보내고 응용프로그램 서버에 프로젝트를 전개하며 구성 등록 정보를 재설정합니다.

**메타데이터 찾기 연결 등록 정보**

엔터프라이즈 서비스 찾기 프로세스에는 찾기 및 서비스 설명 작성을 위해 EIS(Enterprise Information System)에 연결하는 이러한 등록 정보가 필요합니다.

이러한 등록 정보를 사용하여 찾기 서비스는 오브젝트 선택 및 탐색을 위해 표시할 메타데이터 트리를 준비합니다.

『메타데이터 찾기 연결 등록 정보』 테이블은 이러한 등록 정보를 설명합니다.

**메타데이터 찾기 연결 등록 정보**

등록 정보	필수	세계화	설명
데이터베이스 URL	예	예	연결할 데이터베이스 URL. URL은 드라이버마다 다를 수 있습니다. 드라이버 지시사항에 따라 이 등록 정보를 지정하십시오. 이 등록 정보는 양방향 언어에 대해 사용 가능합니다.
JDBCDriverClass	예	아니오	데이터베이스에 연결하는데 사용된 데이터베이스 드라이버의 이름
암호	예	예	해당 사용자 이름에 대한 암호. 이 등록 정보는 양방향 언어에 대해 사용 가능합니다.
접두부	아니오	예	Business Object의 이름에 추가될 접두부. 이 등록 정보는 양방향 언어에 대해 사용 가능합니다.
UserName	예	예	데이터베이스에 로그인하는 사용자 이름. 이 등록 정보는 양방향 언어에 대해 사용 가능합니다.

**관련 태스크**

45 페이지의 『연결 등록 정보 설정』

어댑터 프로젝트를 작성한 후, JDBC를 위해 엔터프라이즈 서비스 찾기 마법사를 초기화하고 데이터베이스 인스턴스를 위해 연결 등록 정보 값을 설정해야 합니다.

## 양방향 연결 등록 정보

이러한 등록 정보를 사용하여 엔터프라이즈 서비스 찾기 마법사는 EIS(Enterprise Information System)로 전달된 데이터에 대해 알맞은 양방향 변환을 적용할 수 있습니다.

다음 테이블은 연결 구성에서 사용되는 양방향 등록 정보를 설명합니다.

### 양방향 연결 등록 정보

등록 정보	값	문자 위치	설명
BiDi OrderingSchema	암시적 시각적	1	논리적(암시적). 기본값은 암시적입니다. 시각적
BiDi Direction	LTR RTL contextual_LTR contextual_RTL	2	Left to Right. 기본값은 LTR(L)입니다. Right to Left Contextual Left to Right Contextual Right to Left
BiDi SymmetricSwapping	예 아니오	3	대칭 스위핑 작동. 기본값은 예(Y)입니다. 대칭 스위핑 작동 안 함
BiDi Shaping	nominal shaped initial middle final isolated	4	Nominal (N). 기본값입니다. 문자는 위치에 따라 형태가 달라집니다. 문자가 초기 형태입니다. 문자가 중간 형태입니다. 문자가 최종 형태입니다. 문자가 고립된 형태입니다.
BiDi NumericShaping	nominal national contextual	5	Nominal (N). 기본값입니다. Hindi(National) Contextual

### 관련 태스크

45 페이지의 『연결 등록 정보 설정』

어댑터 프로젝트를 작성한 후, JDBC를 위해 엔터프라이즈 서비스 찾기 마법사를 초  
기화하고 데이터베이스 인스턴스를 위해 연결 등록 정보 값을 설정해야 합니다.

## 오브젝트 선택 등록 정보

이 등록 정보는 Business Object 선택 프로세스 동안 사용됩니다. 여기에는 필터, 노드 및 선택 등록 정보가 있습니다.

### 필터 및 노드 등록 정보

Business Object를 찾는 동안 트리에 표시된 스키마 목록을 축소하려면 필터 등록 정보를 지정할 수 있습니다. 그렇지 않으면 스키마가 모두 표시됩니다. 표시할 데이터베이스 오브젝트 목록을 축소하려면 노드 등록 정보를 설정할 수 있습니다.



『필터 등록 정보』 테이블은 스키마를 필터링하는 데 필요한 등록 정보를 설명합니다.

### 필터 등록 정보

필터 등록 정보	유형	설명
SchemaNameFilter	문자열	스키마를 필터링하는 데 사용되는 문자열. 지정된 문자열로 시작하는 스키마만 표시됩니다. 이 등록 정보를 설정하지 않으면 모든 스키마가 표시됩니다.
유형	다중 값	테이블, 보기, 저장 프로시저 및 동의어/별명 등 데이터베이스 오브젝트 유형을 나열합니다. 값은 엔터프라이즈 서비스 찾기에서 설정합니다. 다중 선택도 가능합니다.
DefineASI	부울	true이면 생성을 위해 오브젝트를 선택할 때, Business Object에 대한 ASI 값을 설정하라는 메시지가 나타납니다.

『노드 등록 정보』 테이블은 필터 데이터베이스 오브젝트 사용에 필요한 등록 정보를 설명합니다.

### 노드 등록 정보

노드 등록 정보	유형	설명
ObjectNameFilter	문자열	데이터베이스 오브젝트를 필터링하는 데 사용되는 문자열. 지정된 텍스트로 시작하는 데이터베이스 오브젝트만 표시됩니다. ObjectNameFilter가 지정되지 않으면 모든 오브젝트가 표시됩니다.

### 관련 태스크

39 페이지의 『시스템 성능 찾기』

어댑터는 데이터베이스를 분석하여 스키마를 식별하여 데이터베이스에서 Business Object를 찾습니다. 그런 다음 데이터베이스에서 오브젝트를 모두 나열하고 트리 구조로 이를 표시합니다.

### 선택 등록 정보

데이터베이스 오브젝트를 선택한 후 선택 등록 정보의 값을 설정해야 합니다.

엔터프라이즈 서비스 찾기 마법사는 선택 등록 정보를 조회합니다. 『선택 등록 정보』 테이블에서 이러한 등록 정보를 설명합니다.

### 선택 등록 정보

선택 등록 정보	유형	설명
BOLocation	문자열	생성된 .xsd 파일을 저장할 위치의 경로. DataDescription에 RelativePath로서 설정됩니다.

선택 등록 정보	유형	설명
MaxRecords	정수	RetrieveALL 조작에서 검색할 레코드의 최대 수.
NameSpace	문자열	스키마를 필터링하는 데 사용할 문자열을 입력하십시오.
조작	다중 값	기본값은 선택한 서비스 유형에 대해 어댑터에서 지원하는 조작의 목록입니다. 조작에서 선택하십시오. 지정된 조작은 생성 중인 모든 Business Object에 대해 설정됩니다.
서비스 유형	문자열	인바운드 또는 아웃바운드를 선택하십시오. 값은 엔터프라이즈 서비스 찾기에서 설정합니다.

NameSpace 등록 정보는 처음에 모든 Business Object의 기본 Namespace로 설정됩니다. 기본 namespace는 <http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/>입니다.

NameSpace는 Business Object 스키마가 논리적으로 구별되도록 Business Object에 제공됩니다. 예: <http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/Schema1Customer>

#### 관련 태스크

39 페이지의 『오브젝트 선택 및 생성』

Business Object를 생성하려면 데이터베이스 오브젝트 노드를 선택하십시오. 그러면 엔터프라이즈 서비스 찾기 마법사는 선택된 노드의 오브젝트에 대해 Business Object를 생성합니다.

50 페이지의 『선택 등록 정보 설정』

데이터베이스 오브젝트를 선택하였으면 가져오기 및 내보내기 파일을 위해 선택 등록 정보의 값을 지정해야 합니다.

---

## 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 나라에서는 이 자료에 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 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운용에 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

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106-0032, Japan

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.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및  
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩  
한국 아이.비.엠 주식회사  
고객만족센터

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

이 정보에 기술된 라이선스가 있는 프로그램 및 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 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 사용자의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 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.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이 예제에는 가능한 완벽하게 개념을 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.

저작권 라이선스:

이 정보에는 여러 가지 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스에 부합하는 응용프로그램을 개발, 사용, 마케팅 및 배포하기 위한 목적으로 이러한 샘플 프로그램을 추가 비용없이 어떤 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지 않습니다.

이 정보를 소프트카피로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

---

## 프로그래밍 인터페이스 정보

프로그래밍 인터페이스 정보는 본 프로그램을 사용하는 응용프로그램 소프트웨어 작성을 돕기 위해 제공됩니다.

귀하는 범용 프로그래밍 인터페이스를 통해 본 프로그램 툴의 서비스를 제공하는 응용프로그램 소프트웨어를 작성할 수 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버그를 위해 제공된 것입니다.

경고 본 진단, 수정 및 조정 정보는 변경될 수 있으므로 프로그램 인터페이스로서 사용할 수 없습니다.

---

## 상표 및 서비스표

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

i5/OS

IBM

IBM 로고

AIX

AIX 5L

CICS

CrossWorlds

DB2

DB2 Universal DatabaseDomino

HelpNow

IMSInformix

iSeries

Lotus  
Lotus  
NotesMQIntegrator  
MQSeries  
MVSNotes  
OS/400  
Passport Advantage  
pSeries  
Redbooks  
SupportPac  
WebSphere  
z/OS

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

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

Intel, Intel 로고, Intel Inside, Intel Inside 로고, Intel Centrino, Intel Centrino 로고, Celeron, Intel Xeon, Intel SpeedStep, Itanium 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.

UNIX는 미국 또는 기타 국가에서 Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 상표입니다.

기타 회사, 제품 및 서비스 이름은 타사의 상표 또는 서비스표입니다.

본 제품에는 Eclipse Project(<http://www.eclipse.org/>)에서 개발한 소프트웨어가 포함되어 있습니다.



WebSphere Adapter, 버전 6.0









**IBM**