



Adapter for Portal Infranet 사용자 안내서

버전 4.2.x



Adapter for Portal Infranet 사용자 안내서

버전 4.2.x

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 115 페이지의 부록 C 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM WebSphere InterChange Server, 버전 4.2, WebSphere Business Integration Adapters, 버전 2.3.0 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견을 보내려면 ibmkspoe@kr.ibm.com으로 전자 우편을 보내십시오. 고객의 의견을 기대합니다.

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

© Copyright International Business Machines Corporation 1999, 2003. All rights reserved.

목차

이 책의 정보	v
이 책의 사용자	v
관련 서적	v
인쇄 규칙	vi
이 릴리스의 새로운 기능	vii
버전 4.2.x.	vii
이전 릴리스	vii
제 1 장 커넥터 개요	1
커넥터 구성요소	1
커넥터 작동 방법	3
제 2 장 커넥터 설치 및 구성	11
호환성	12
전제조건 소프트웨어	12
Infranet 응용프로그램 구성	13
Portal Infranet 어댑터 및 기타 파일 설치	14
AIX/DB2 환경에서 어댑터 구성	14
AIX/DB2 환경에서 Portal Infranet 응용프로그램 구성	17
Oracle 환경에서 어댑터 구성	18
Windows 환경에서 이벤트 메커니즘 설치 및 구성	20
커넥터 구성	22
새 Business Object에 대한 이벤트 메커니즘 사용자 정의	26
Infranet 사용자 정의 속성의 선택적 구성 선언	30
다중 커넥터 인스턴스 작성	31
커넥터 시작	32
커넥터 중지	34
제 3 장 Business Object 이해	35
Portal Infranet 응용프로그램 배경	35
메타 데이터 구동 커넥터	39
Portal Infranet 응용프로그램 특정 Business Object 구조	39
Business Object 속성 등록 정보	42
Business Object 정의 지침	43
Business Object 응용프로그램 특정 정보	43
제 4 장 PortalODA를 사용한 Business Object 정의 생성	63
설치 및 사용법	63
Business Object Designer에서 PortalODA 사용	66
생성된 정의 내용	75
Business Object 정의에 정보 추가	77
부록 A. 커넥터의 표준 구성 등록 정보	79

신규 및 삭제된 등록 정보	79
표준 커넥터 등록 정보 구성	79
표준 등록 정보 요약	81
표준 구성 등록 정보	84
부록 B. Connector Configurator	97
Connector Configurator 개요	97
Connector Configurator 시작	98
System Manager에서 Configurator 실행	99
커넥터 특정 등록 정보 템플릿 작성	99
새 구성 파일 작성	102
기존 파일 사용	103
구성 파일 완료	104
구성 파일 등록 정보 설정	105
구성 파일 저장	112
구성 파일 변경	112
구성 완료	113
국제화된 환경에서 Connector Configurator 사용	113
부록 C. 주의사항	115
프로그래밍 인터페이스 정보	116
상표 및 서비스표	117

이 책의 정보

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

이 문서에서는 Portal Infranet용 IBM WebSphere Business Integration Adapter의 설치, 구성 및 Business Object 개발 및 문제점 해결에 대해 설명합니다.

이 책의 사용자

이 책은 WebSphere Business Integration System의 일부로 커넥터를 구현하고 있는 WebSphere 컨설턴트와 고객을 위한 것입니다. 이 책의 정보를 사용하려면 다음 영역에 대한 지식이 있어야 합니다.

- 커넥터 개발
- Business Object 개발
- Portal Infranet 응용프로그램 구조

관련 서적

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

다음 사이트에서 관련 문서를 설치할 수 있습니다.

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

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

InterChange Server와 함께 어댑터를 사용하는 경우에는 다음 웹 사이트를 참조하십시오.

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

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

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

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

WebSphere Application Server에 대한 자세한 정보는 다음 웹 사이트를 참조하십시오.

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

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

인쇄 규칙

이 책에서는 다음 규칙을 사용합니다.

courier 글꼴	명령어, 파일 이름, 입력하는 정보 또는 시스템이 화면에 인쇄하는 정보와 같은 리터럴 값을 표시합니다.
굵은체	처음 나타나는 새 용어를 표시합니다.
가울임	변수 이름이나 상호 참조를 표시합니다.
파란색 윤곽선	온라인으로 매뉴얼을 보는 경우에만 볼 수 있는 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 바로 가려면 윤곽선 내부를 누르십시오.
{ }	구문 행에서, 중괄호는 하나만을 선택해야 하는 옵션 세트를 묶습니다.
[]	구문 행에서, 대괄호는 선택적 매개변수를 묶습니다.
...	구문 행에서, 밑줄임표(...)는 이전 매개변수의 반복을 표시합니다. 예를 들어, option[,...]은 쉼표로 구분되는 여러 옵션을 입력할 수 있음을 의미합니다.
< >	이름 지정 규칙에서 꺾쇠 괄호는 <server_name><connector_name> tmp.log에서와 같이 이름의 개별 요소를 묶어 이들을 서로 구별합니다.
/, \	이 책에서는 백슬래시(\)를 디렉토리 경로 규칙으로 사용합니다. UNIX 설치의 경우 백슬래시를 슬래시(/)로 대체하십시오. 모든 WebSphere Business Integration System 제품 경로 이름은 제품이 시스템에 설치된 디렉토리에 상대적입니다.
%text% 및 \$text	퍼센트(%) 부호 내의 텍스트는 Windows text 시스템 변수 또는 사용자 변수의 값을 표시합니다. UNIX 환경에서 이에 해당하는 표기는 \$text로, 이는 text UNIX 환경 변수의 값을 표시합니다.
ProductDir	제품이 설치된 디렉토리를 표시합니다.

이 릴리스의 새로운 기능

버전 4.2.x

Portal Infranet의 어댑터는 이제 HP-UX 11i에서 지원됩니다.

4.2x 버전으로 시작할 때 Portal Infranet용 어댑터는 더 이상 Microsoft Windows NT에 지원되지 않습니다.

이 안내서에서 어댑터 정보가 이동되었습니다. 해당 정보의 새 위치는 2장 14 페이지의 『Portal Infranet 어댑터 및 기타 파일 설치』를 참조하십시오.

이전 릴리스

새로운 기능은 다음과 같습니다.

버전 4.1.x

58 페이지의 『완전한 샘플 Portal Infranet Business Object 정의』는 사용자 정의 Business Object 개발에 대한 모델을 제공하기 위해 추가되었습니다.

이제 어댑터는 WebSphere Application Server를 통합 브로커로 사용할 수 있습니다. 자세한 정보는 12 페이지의 『호환성』을 참조하십시오. 어댑터는 이제 다음 플랫폼에서 실행됩니다.

- Solaris 7, 8
- AIX 5.x

버전 4.0.x

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

커넥터는 PORTALODA로 CWSAPGEN 유틸리티를 바꾸었습니다. 자세한 정보는 63 페이지의 제 4 장 『PortalODA를 사용한 Business Object 정의 생성』을 참조하십시오.

버전 3.1.x

국제화된 커넥터가 IBM WebSphere Business Integration Adapter for Portal Infranet와 함께 제공됩니다.

버전 3.0.x

이 릴리스의 커넥터에는 다음 새 기능이 들어 있습니다. 커넥터가 국제화되었습니다. 자세한 정보는 9 페이지의 『로케일 중속 데이터 처리』 및 79 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오. 커넥터는 AIX 4.3.3에서 다음 소프트웨어를 지원합니다.

- Oracle 8.1.7 및 Portal Infranet 6.2 SP1
- DB2 7.1.0 및 Portal Infranet 6.2 SP1

버전 2.5.x

IBM WebSphere Business Integration Adapter for Portal Infranet에는 Connector for Portal Infranet이 포함되어 있습니다. 이 어댑터는 통합 브로커인 InterChange Server(ICS) 및 WebSphere MQ Integrator Broker 모두와 함께 작동합니다. 이기종의 응용프로그램 세트 통합을 수행하는 응용프로그램인 통합 브로커는 데이터 라우팅을 포함하는 서비스를 제공합니다. 어댑터에는 다음 사항이 포함되어 있습니다:

- Portal Infranet에 고유한 응용프로그램 구성요소
- 샘플 Business Object
- 다음을 포함하는 IBM WebSphere Adapter Framework
 - 개발 도구(Business Object Designer 및 Connector Configurator 포함)
 - API(ODK, JCDK 및 CDK 포함)

이 매뉴얼은 InterChange Server(ICS) 및 WebSphere MQ Integrator Broker라는 두 통합 브로커와 함께 이 어댑터를 사용하는 방법에 대한 정보를 제공합니다.

주: 커넥터는 이제 Portal Infranet 6.2.0을 지원합니다.

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

버전 2.4.x

이 커넥터의 버전 2.4.x에서 수행된 변경사항은 이 책의 내용에 영향을 주지 않습니다.

버전 2.3.x

결함을 수정하고 IBM CrossWorlds infrastructure 버전 4.0.0과의 호환성을 제공하기 위해 약간의 변경이 수행되었습니다.

버전 2.2.x

커넥터 버전이 이제 Portal Infranet 6.1.0을 지원합니다.

버전 2.1.x

- 커넥터 버전이 이제 Portal Infranet 6.0.1을 지원합니다.
- 커넥터를 UNIX 시스템에 설치하고 실행할 수 있습니다.
- 이 책은 전반적으로 재구성되고 다시 쓰여졌습니다.

제 1 장 커넥터 개요

이 장에서는 IBM WebSphere Business Integration Adapter for Portal Infranet의 커넥터 구성요소 개요에 대해 설명하며, 다음 섹션이 포함됩니다.

- 『커넥터 구성요소』
- 3 페이지의 『커넥터 작동 방법』
- 3 페이지의 『메타 데이터 구동 커넥터 작동』
- 3 페이지의 『Business Object 처리』
- 7 페이지의 『이벤트 공고』
- 8 페이지의 『이벤트 검색』
- 9 페이지의 『Infranet 응용프로그램에 연결』
- 9 페이지의 『로케일 종속 데이터 처리』

커넥터는 커넥터 프레임워크와 응용프로그램 특정 구성요소 두 부분으로 이루어집니다. 코드가 모든 커넥터에 공통인 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소의 매개체로 작동합니다. 응용프로그램 특정 구성요소에는 특정 응용프로그램에 맞게 조정된 코드가 있습니다. 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이에 다음 서비스를 제공합니다.

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

이 책에는 커넥터 프레임워크와 커넥터로서 참조되는 응용프로그램 특정 구성요소에 관한 정보가 있습니다.

커넥터를 사용하면 WebSphere MQ Integrator Broker 또는 IBM WebSphere ICS(InterChange Server)가 Business Object의 교환을 통해 Portal Infranet과 통신할 수 있습니다. Infranet 응용프로그램은 고객 계정 관리를 위해 Portal Infranet 소프트웨어로 개발한 소프트웨어 프로그램 세트입니다. 계정 번호와 청구 정보 같은 고객 정보가 Infranet 데이터베이스에 저장됩니다.

커넥터와 Portal Infranet 응용프로그램은 Infranet 소켓 기반 API를 사용하여 통신합니다. 커넥터는 Infranet API가 제공하는 기능을 사용하여 트랜잭션을 처리하고, Infranet 응용프로그램은 변경사항이 발생하면 이벤트 모듈을 통해 통합 브로커(WebSphere MQ Integrator Broker 또는 ICS)에 알립니다.

커넥터 구성요소

Portal Infranet용 커넥터에는 다음 구성요소가 포함되어 있습니다.

- 커넥터: Business Object Verb 지원 및 이벤트 폴링 메커니즘을 구현하는 Java .jar 파일.
- WebSphere Business Integration Adapters 이벤트 기능 모듈: Infranet 응용프로그램에서 이벤트 공고 메커니즘을 구현하는 Windows의 C++ DLL 및 UNIX의 SO 파일. 이 모듈은 통합 브로커와 관련된 Infranet 이벤트를 선택하고 Infranet 데이터베이스의 테이블에 저장합니다. 커넥터는 이 테이블을 정기적으로 폴링합니다.

커넥터는 통합 브로커로 전송하는 Business Object를 생성합니다. 커넥터는 또한 통합 브로커의 Business Object 요청에 응답합니다. 파일 또는 커넥터 콘솔에 기록하거나 통합 브로커로 전송하는 로깅 및 추적 메시지를 생성합니다.

그림 1은 커넥터 구조와 Infranet 응용프로그램에서 커넥터의 이벤트 메커니즘을 보여줍니다.

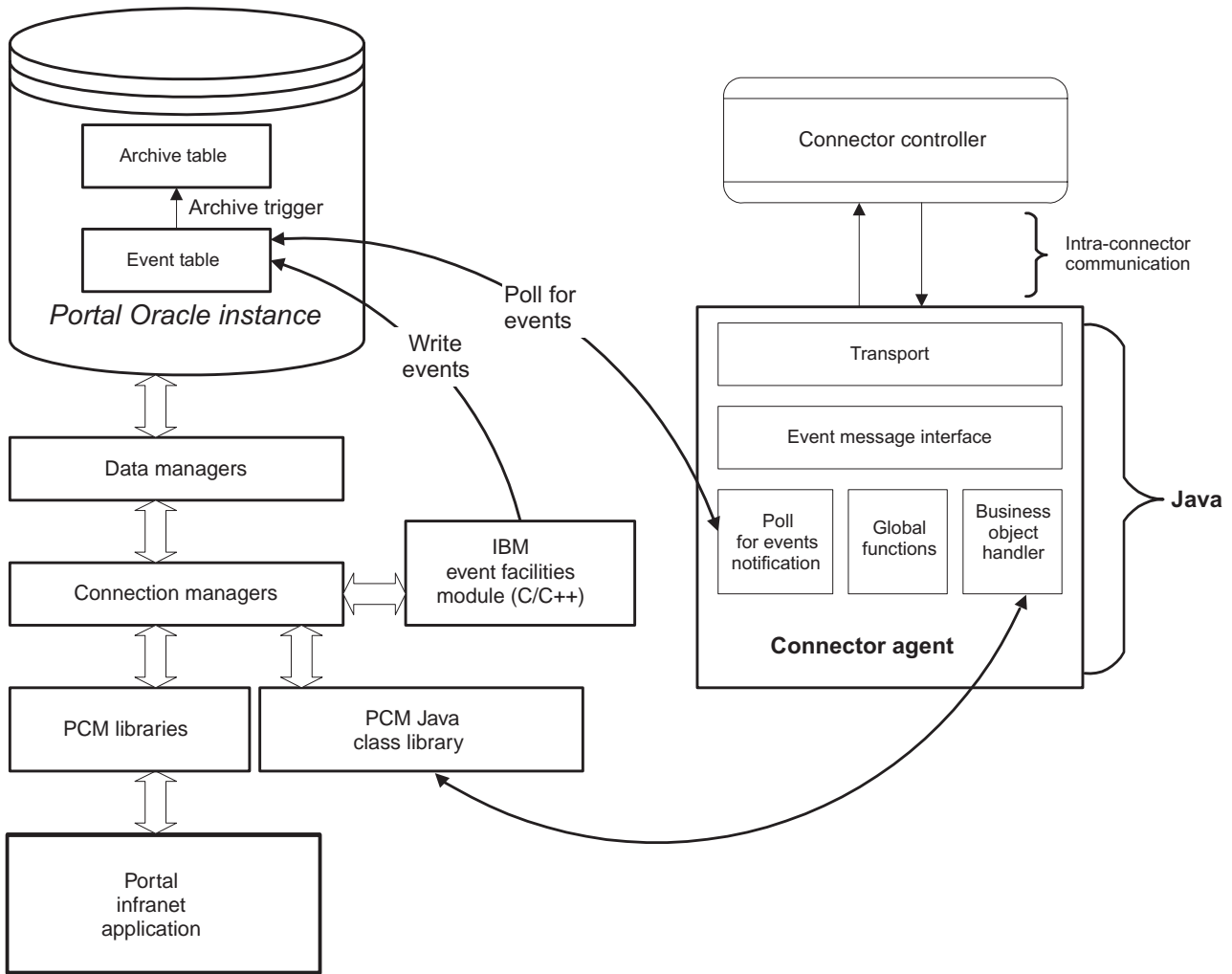


그림 1. 커넥터 구조

커넥터는 PCM(Portal Communications Module) Java 클래스 라이브러리를 API로서 사용하여 Portal Infranet 연결 관리자와 상호작용합니다. 이 구조의 장점은 Portal Infranet이 Windows 및 UNIX 경우 모두 JVM(Java Virtual Machine)에서 PCM Java 클래스 라이브러리를 지원한다는 점입니다. 이를 통해 커넥터는 두 플랫폼에서 실행될 수 있습니다. 이 API는 커넥터의 Business Object 핸들러에 의해 사용되어 통합 브로커와 Portal 간에 정보를 교환할 수 있도록 합니다. WebSphere Business Integration Adapters 이벤트 기능 모듈은 C++ PCM 라이브러리를 사용합니다.

커넥터 작동 방법

다음 절에서는 커넥터가 Business Object 요청을 처리하는 방법 및 커넥터가 이벤트 공고를 처리하는 방법에 대해 설명합니다.

메타 데이터 구동 커넥터 작동

커넥터는 메타 데이터에 의해 구동됩니다. 커넥터는 메타 데이터가 전달하는 Business Object의 유형이나 변수와 상관없이 모든 Business Object의 검색과 제출을 처리하도록 설계됩니다. 커넥터가 메타 데이터에 의해 구동되기 위해서는 Portal Infranet에 대한 Business Object에 다음 정보가 포함되어 있어야 합니다.

- Infranet의 데이터 사전으로 식별되는 각 속성에 대한 필드 이름. 여기에는 API 문서에 설명되어 있는 각각의 pin 필드 이름에 대한 pin 필드 번호가 있습니다. 필드 이름은 속성 레벨에서 응용프로그램 특정 정보로서 지정되며 이름은 데이터 사전을 사용하여 커넥터에 의해 번호로 변환됩니다.
- Business Object가 지원하는 opcode(조작 코드). Opcode는 Business Object의 Verb 레벨에서 지정됩니다. Infranet opcode는 고객 관련 정보 관리, 온라인 계정 작성, 고객 정보 수집 및 추적하고 제3자 시스템을 Infranet과 통합하기 위해 클라이언트 응용프로그램과 스크립트에서 사용되는 조작입니다.

Portal에 대한 Business Object 메타 데이터에 대한 자세한 정보는 35 페이지의 제 3 장 『Business Object 이해』를 참조하십시오.

Business Object 처리

커넥터가 WebSphere Business Integration System으로부터 Business Object 요청을 수신하면 커넥터 Business Object 핸들러가 Business Object를 처리합니다. Business Object 핸들러는 응용프로그램 특정 오브젝트와 Portal Infranet API 사이의 매개체입니다. 이 핸들러는 Portal Infranet 조작을 API에 제출하고 Infranet 이벤트의 결과로서 WebSphere Business Integration System으로 전송되는 응용프로그램 특정 Business Object를 작성하는 작업을 담당합니다. Business Object 핸들러는 Business Object와 모든 메타 데이터의 데이터를 사용하여 Portal에 저장 가능한 오브젝트를 제출하기 위해 Infranet Java API를 호출합니다. 이 조작이 완료될 때 상태가 통합 브로커로 리턴됩니다.

그림 2의 순서도는 상위 레벨에서 Business Object 핸들러가 Business Object 요청을 처리하는 방법을 보여줍니다. Business Object 핸들러는 Business Object에서 Verb와 키 속성을 추출합니다. Verb를 사용하여 Business Object를 처리하기 위해 수행되는 기능 호출을 판별합니다. 이 예에서 Update Verb인 경우 UpdateObject 기능이 호출됩니다.

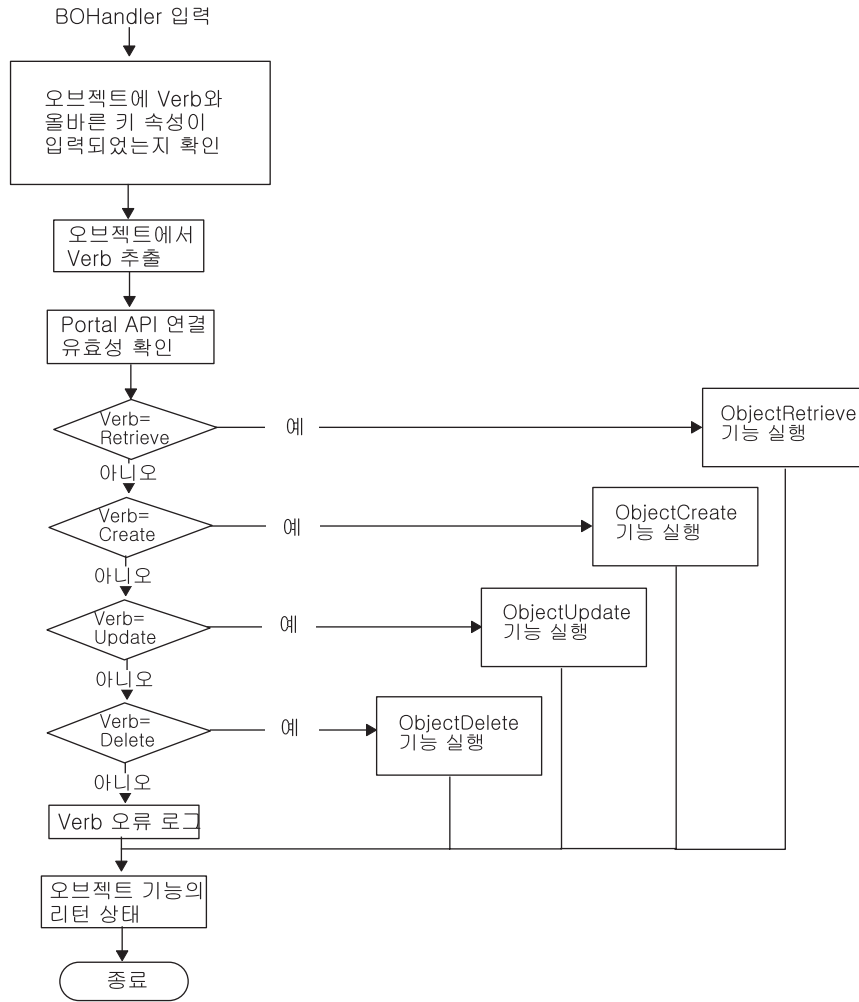


그림 2. Business Object 처리의 상위 레벨 보기

Retrieve Verb 처리

Business Object 핸들러의 Retrieve 메소드는 Portal Infranet에서 오브젝트를 검색하고 WebSphere Business Integration Adapters Business Object를 응용프로그램 정보로 채웁니다. 커넥터의 Retrieve 메소드는 다음을 수행합니다.

1. Portal Infranet 연결이 유효한지 확인합니다. 유효하지 않은 경우 연결이 다시 인스턴스화되어야 합니다. 연결이 처리 중에 끊어지는 경우, 커넥터는 Infranet 연결 문제점을 표시하는 BON_FAIL 상태를 리턴합니다.

2. 오브젝트에 대한 응용프로그램 특정 정보를 검색합니다. Verb에 대해 지정된 응용 프로그램 특정 정보는 Portal Infranet 오브젝트의 검색을 위해 호출되어야 하는 조작 코드 또는 opcode를 제공합니다.
3. Business Object 및 해당 속성의 응용프로그램 특정 정보를 근거로 하여 opcode의 flist를 준비합니다.

주: flist는 필드 및 값 쌍의 변수 길이 목록입니다. Flists는 Infranet opcode 및 기능에 입력과 출력 매개변수를 제공합니다.

4. flist가 입력으로서 지정된 opcode 및 빈 출력 flist를 호출합니다.
5. 이전 단계가 성공하면 리턴 flist 구조에는 WebSphere Business Integration Adapter Business Object로 정의되는 저장 가능한 오브젝트에 대응하는 완전히 채워진 flist 오브젝트가 들어 있게 됩니다. flist가 WebSphere Business Integration Adapter Business Object에 대해 일대일로 대응하므로, flist는 flist 필드 이름과 유형을 식별하는 속성 레벨 응용프로그램 특정 정보를 바탕으로 WebSphere Business Integration Adapter Business Object의 모든 속성을 검색하기 위해 좌우로 이동합니다.
6. Business Object를 채우고 통합 브로커에 전송합니다.

그림 3은 Retrieve 메소드가 작동하는 방법을 보여줍니다. 이 메소드는 속성의 유형에 따라서 속성에서 수행될 조치를 판별합니다. 기본 속성 유형(예: 문자열)인 경우 Business Object 핸들러가 동적으로 필드를 채웁니다. 메소드가 하위 Business Object를 발견하는 경우, 해당 하위 Business Object의 기본 속성에 도달할 때까지 오브젝트를 내려갑니다. 그런 다음 상위 오브젝트의 기본 속성에서 계속하기 전에 하위 오브젝트의 모든 기본 속성을 순환합니다.

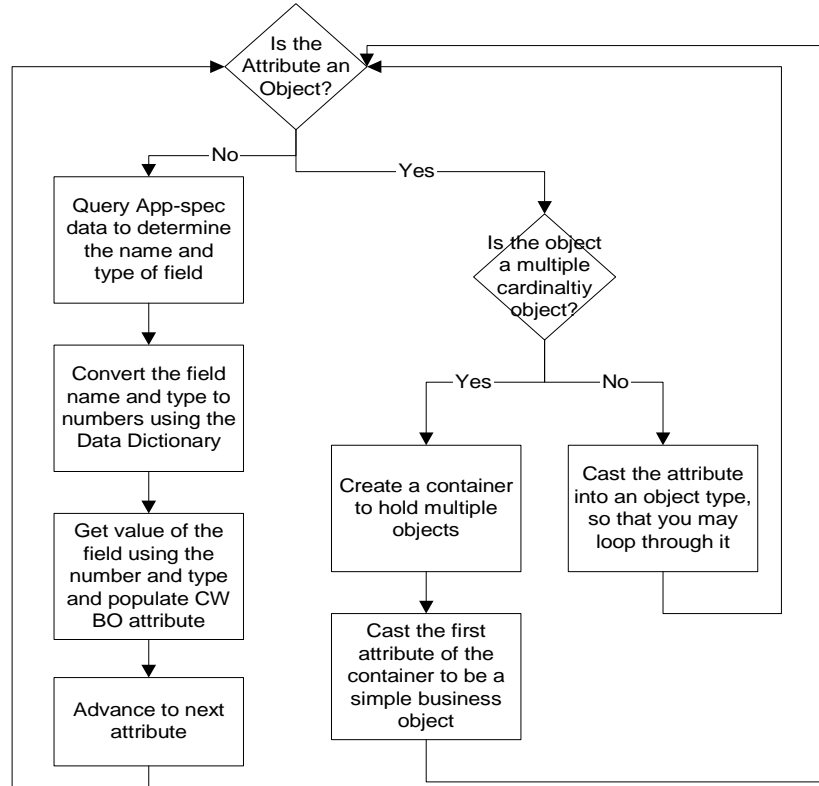


그림 3. Retrieve Verb 처리를 위한 순서도

Create 및 Update Verb 처리

Create 또는 Update Verb를 처리하기 위해 Business Object 핸들러는 Infranet API 오브젝트(Flist)를 구성하고 이것을 Infranet API에 전달합니다. Business Object 핸들러는 다음 단계를 수행합니다.

1. WebSphere Business Integration Adapters Business Object에서 응용프로그램 특정 Business Object를 검색합니다.
2. Portal Infranet API 오브젝트를 채웁니다. Flist가 인스턴스화될 때 커넥터는 속성별로 오브젝트를 반복하고 Flist를 채울 값을 찾습니다. 이 프로세스는 오브젝트의 여러 레이어일 수 있는 속성을 찾기 위해 오브젝트를 검색해야 합니다.
3. Portal flist 연결이 여전히 유효한지 확인합니다. 연결이 처리 중에 끊어지는 경우 커넥터는 Infranet 연결 문제점을 표시하는 BON_FAIL 상태를 리턴하며, 연결이 다시 인스턴스화되어야 합니다.
4. Business Object Verb에 대한 응용프로그램 특정 정보를 사용하여 Create 또는 Update 조작에 대한 적당한 opcode를 동적으로 호출합니다. 매개변수가 수집되고 배열에 배치되며 기능 문자열이 어셈블되면, 적당한 opcode를 갖는 문맥 함수가 호출됩니다.

5. 고유한 opcode를 갖는 하위 Business Object를 처리하기 위해 Business Object 핸들러는 각 하위에 대해 별도의 Flist를 채운 다음 적당한 opcode를 호출합니다.
6. 조작이 완료될 때 상태를 리턴합니다.

이벤트 공고

Infranet은 응용프로그램에서 발생하는 조치를 추적할 수 있게 하는 이벤트 메커니즘을 갖고 있습니다. 사용자가 Infranet에서 조치를 수행할 때 응용프로그램이 연관된 이벤트를 생성합니다.

WebSphere Business Integration Adapters 이벤트 모듈은 이벤트를 시험하고 커넥터가 관심을 갖는 이벤트인지 여부를 판별합니다. 그런 경우 이벤트 모듈은 이벤트에 대한 WebSphere Business Integration Adapters 이벤트 테이블에 항목을 생성합니다.

Infranet에서 이벤트 발견

Infranet에서의 이벤트 감지는 이벤트 공고 메커니즘에 의해 호출되는 사용자 정의 Infranet 기능 모듈을 사용하여 구현됩니다. 이 기능 모듈은 통합 브로커에 의해 제공되며, Infranet 이벤트를 식별하고 WebSphere Business Integration Adapters 이벤트 테이블에 이벤트를 기록하기 위해 두 개의 구성 파일과 함께 작동합니다.

Infranet 오브젝트에 대한 변경이 발생할 때 지속적 이벤트가 발생합니다. Infranet은 지정된 이벤트가 발생할 때 특정 opcode를 호출하도록 구성될 수 있으므로, 통합 브로커가 Portal에 대한 Business Object와 연관된 이벤트에 대해 WebSphere Business Integration Adapters 이벤트 기능 모듈을 호출하도록 Infranet을 구성하는 구성 텍스트 파일을 제공합니다. 이 구성 파일을 “pin_notify_cw”라고 하며, 이 파일은 Infranet에서 제공하는 load_pin_notify 유틸리티를 사용하여 Infranet에 로드됩니다.

이벤트 모듈은 이벤트를 수신할 때 이벤트 오브젝트에서 정보를 추출하고 WebSphere Business Integration Adapters 이벤트 테이블에 새 항목을 작성합니다. Infranet의 이벤트는 실제로는 Infranet 저장 가능 클래스의 인스턴스이며, 각각 작성, 수정 또는 삭제 이벤트가 특정 Infranet 저장 가능 클래스와 연관됩니다. 예를 들어, 사용자가 고객과 관련된 특정 문의처를 수정하는 경우 Infranet은 저장 가능 클래스 /event/customer/nameinfo의 인스턴스를 생성합니다.

이벤트 모듈은 고유한 이벤트 모듈 구성 파일을 사용하여 발생한 이벤트를 판별하고, 저장 가능 클래스(및 관련 Business Object)의 어떤 부분이 수정되었는지를 식별하며, 발생한 조치의 유형을 판별합니다. 구성 파일 event_code.txt를 사용하여 이벤트 모듈은 Infranet 이벤트를 시험하고 해당 이벤트를 반영하는 레코드로 WebSphere Business Integration Adapters 이벤트 테이블을 채웁니다.

커넥터에 대한 이벤트 공고 메커니즘은 Infranet에서 사용되는 Oracle 데이터베이스 인스턴스 안에 작성되는 세 개의 테이블을 사용합니다.

- XWORLDS_Events - 모든 보류 이벤트를 저장합니다.

- XWORLDS_Archive_Events - 커넥터가 처리한 이벤트를 아카이브합니다.
- XWORLDS_Current_Event_ID - 마지막 이벤트 ID 번호를 저장합니다.

첫 번째 테이블의 스키마는 커넥터가 관심을 갖고 있는 Infranet에 의해 전송된 각 이벤트에 대해 기록되는 정보를 지정합니다. 이 테이블 레이아웃이 아카이브 테이블에도 사용됩니다.

이벤트 발견 및 연관된 처리는 Infranet 트랜잭션 내에서 수행됩니다. Infranet은 트랜잭션 안에서 사용자 정의 프로세스를 호출하고 처리 결과를 기다립니다. 사용자 정의 프로세스가 오류를 리턴하는 경우, 트랜잭션은 중단됩니다. 이것은 커넥터가 어떤 이벤트도 잃어버리지 않도록 보장해 줍니다.

주:

알려진 문제 - 이벤트 공고 모듈은 pin_notify_cw 파일에 정의되어 있는 전송된 Portal 이벤트의 사용자 ID를 확인합니다. 모듈에 전송된 이벤트와 연관된 PIN_FLD_USERID가 없으면, 오류가 발생하여 오브젝트를 온라인으로 저장하는데 문제가 생깁니다. FList 또는 저장 가능한 클래스를 사용하여 그러한 ID를 포함하도록 이러한 유형의 이벤트를 조정할 필요가 있습니다. crossworlds.cnf 구성 파일에 정의된 로그 파일에서 이러한 오류를 확인하십시오.

이벤트 모듈은 사용자 ID를 확인하여 커넥터에서 응용프로그램에 전송한 이벤트가 이벤트 대기열에 추가되지 않도록 합니다. 이를 핑퐁 방식이라 합니다.

"/event/customer/billinfo"는 그러한 문제가 있는 이벤트 유형입니다.

이벤트 검색

커넥터는 Infranet 데이터베이스 인스턴스에서 설정된 XWORLDS_Events 테이블을 폴링하여 이벤트를 확인합니다. 커넥터는 XWORLDS_Events 테이블에서 항목을 추출하는 SQL SELECT문을 사용하여 폴링합니다. 선택된 이벤트 수는 커넥터의 PollQuantity 등록 정보에 의해 지정됩니다.

폴링은 커넥터의 pollForEvents() 메소드에 의해 수행됩니다. 커넥터는 WebSphere Business Integration Adapters 커넥터 등록 정보에 설정되어 있는 PollFrequency에서 이벤트 테이블을 폴링합니다. 테이블에서 새 행이 발견되는 경우, 해당 이벤트 데이터가 검색되고 커넥터가 이벤트를 다음과 같이 처리합니다.

1. 폴 기능이 빈 Business Object를 작성하며, Verb를 Retrieve로 설정하고, 이벤트 레코드를 사용하여 키를 설정합니다. Business Object가 커넥터의 Business Object 핸들러로 전송됩니다.
2. Business Object 핸들러는 이벤트 데이터를 사용하여 Portal flist 저장 가능 오브젝트를 검색하기 위한 Infranet Java API를 호출합니다.

3. Business Object 핸들러가 저장 가능 오브젝트를 WebSphere Business Integration Adapters 응용프로그램 특정 Business Object로 변환하고, Verb를 이벤트 레코드의 조치로 설정한 후 Business Object를 통합 브로커에 전송합니다.

Business Object가 WebSphere Business Integration System으로 전송된 후, 이벤트 테이블 항목이 XWORLDS_Archive_Events 테이블에 아카이브되고 이벤트 테이블에서 삭제됩니다.

폴 메소드가 호출되는 시간 간격은 PollFrequency 커넥터 등록 정보를 변경하여 조정할 수 있습니다. 이 등록 정보는 통합 브로커를 사용하여 설정됩니다.

Infranet 응용프로그램에 연결

API를 사용하여 Portal Infranet 연결 관리자에 연결할 때 커넥터는 다음을 수행합니다.

1. Portal Infranet 문맥의 새 인스턴스를 작성합니다.
2. opcode가 실행될 때, 커넥터는 폴에서 문맥을 사용하여 사용 중인 폴에 추가합니다.
3. 타스크 완료 시, 문맥은 사용 가능한 폴로 리턴됩니다.

Connect 명령문은 저장소에서 정의되는 커넥터 응용프로그램 특정 등록 정보의 값을 사용합니다.

폴의 문맥 인스턴스는 커넥터가 종료될 때 닫힙니다.

로케일 종속 데이터 처리

지정된 언어로 2바이트 문자 세트 및 전달 메시지 텍스트를 지원할 수 있도록 커넥터가 국제화되었습니다. 커넥터가 1문자 코드 세트를 사용하는 위치에서 다른 코드 세트를 사용하는 위치로 데이터를 전송하면, 데이터의 의미를 유지하기 위해 문자가 변환됩니다. JVM 내의 Java 런타임 환경에서는 Unicode 문자 코드 세트로 데이터를 나타냅니다. Unicode에는 잘 알려진 문자 코드 세트의 문자 인코딩(단일 바이트 및 다중 바이트 모두)이 들어 있습니다. IBM CrossWorlds 시스템에 있는 대부분의 구성요소들은 Java로 작성되었습니다. 그러므로 대부분의 IBM CrossWorlds 구성요소 사이에서 데이터가 전송될 때 문자 변환이 필요 없습니다. 해당 국가나 지역에 대해 해당 언어로 오류 및 정보 메시지를 기록하려면, 사용자 환경의 로케일 표준 구성 등록 정보를 구성하십시오. 이들 등록 정보에 대한 자세한 정보는 79 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

제 2 장 커넥터 설치 및 구성

이 장에서는 커넥터를 설치 및 구성하는 방법에 대해 설명하며, 다음 주제가 들어 있습니다.

- 12 페이지의 『호환성』
- 12 페이지의 『전제조건 소프트웨어』
- 13 페이지의 『Infranet 응용프로그램 구성』
- 14 페이지의 『Portal Infranet 어댑터 및 기타 파일 설치』
- 14 페이지의 『AIX/DB2 환경에서 어댑터 구성』
- 17 페이지의 『AIX/DB2 환경에서 Portal Infranet 응용프로그램 구성』
- 18 페이지의 『Oracle 환경에서 어댑터 구성』
- 20 페이지의 『Windows 환경에서 이벤트 메커니즘 설치 및 구성』
- 22 페이지의 『커넥터 구성』
- 26 페이지의 『새 Business Object에 대한 이벤트 메커니즘 사용자 정의』
- 30 페이지의 『Infranet 사용자 정의 속성의 선택적 구성 선언』
- 31 페이지의 『다중 커넥터 인스턴스 작성』
- 32 페이지의 『커넥터 시작』
- 34 페이지의 『커넥터 중지』

IBM WebSphere Business Integration Adapter for Portal Infranet의 커넥터 구성요소는 설치 및 구성해야 하는 두 개의 구성요소가 있습니다.

- 커넥터. 커넥터는 Java.jar 파일입니다. 커넥터는 커넥터 Verb 지원과 이벤트 폴링 메커니즘을 구현합니다.
- WebSphere Business Integration Adapters 이벤트 기능 모듈. 이벤트 기능 모듈은 이벤트 공고를 구현하는 실행 파일입니다. 이 모듈은 통합 브로커와 관련된 Infranet 이벤트를 선택하고 데이터베이스 테이블에 저장합니다.

이 장에서는 커넥터 구성요소를 설치 및 구성하는 방법과 커넥터와 함께 작동하도록 Portal Infranet 응용프로그램을 구성하는 방법에 대해 설명합니다.

주: 이 책에서 일부 예제 코드 파일의 경우를 제외하고 백슬래시(\)가 디렉토리 경로의 규칙으로 사용됩니다. UNIX 설치의 경우 슬래시(/)를 백슬래시(\)로 대체하십시오. 모든 파일 경로 이름은 별도로 언급되지 않는 한 WebSphere Business Integration Adapters 제품이 시스템에 설치되는 디렉토리에 상대적입니다.

호환성

어댑터에 사용되는 어댑터 프레임워크는 어댑터가 통신하고 있는 통합 브로커의 버전과 호환되어야 합니다. Portal Infranet용 어댑터의 버전 4.2는 다음 버전의 어댑터 프레임워크 및 통합 브로커에서 지원됩니다.

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

통합 브로커:

- WebSphere InterChange Server, 버전 4.2.x
- WebSphere MQ Integrator, 버전 2.1.0
- WebSphere MQ Integrator Broker, 버전 2.1.0
- WebSphere Business Integration Message Broker, 버전 5.0
- WebSphere Studio Application Developer Integration Edition, 버전 5.0.1이 있는 WebSphere Application Server Enterprise, 버전 5.0.2

예외는 릴리스 정보를 참조하십시오.

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

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

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

전제조건 소프트웨어

컨넥터에는 다음 전제조건 소프트웨어가 필요합니다.

- Java PCM(Portal Communications Module) 라이브러리 pcm.jar 및 pcmext.jar.
- WebLogic Oracle JDBC 드라이버와 라이브러리를 포함하여 WebSphere Business Integration Adapters에 필요한 모든 일반적인 타사 소프트웨어.

이벤트 모듈에는 다음 전제조건 소프트웨어 및 설정이 필요합니다.

- Oracle 데이터베이스와 함께 설치된 Infranet 응용프로그램.

- 환경 변수 ORACLE_SID의 기본값이 Portal Infranet 데이터베이스의 별명 이름에 설정되어야 합니다.

Windows 시스템에 설치하려는 경우, 이벤트 모듈에는 또한 다음 전제조건 소프트웨어가 필요합니다.

- PCN(Portal Communications Module) 라이브러리 pcm.dll
- WebSphere Business Integration Adapter for Portal Infranet의 커넥터 구성요소와 함께 제공되는 Microsoft MFC 및 종속 제품: MFC42.dll. 시스템이 경로에 이 파일을 아직 갖지 않은 경우, 이 파일은 PATH 시스템 변수에 정의되어 있는 디렉토리에 있어야 합니다. 파일을 *ProductDir* 디렉토리에 저장하면 됩니다.

Infranet 응용프로그램 구성

커넥터가 사용할 Infranet 응용프로그램을 설정하려면 커넥터에 대한 사용자 계정을 정의하고 Infranet이 사용하는 Oracle 데이터베이스에 이벤트 및 아카이브 테이블을 작성해야 합니다.

Infranet 계정 설정

Infranet 관리자를 사용하여 모든 권한을 갖는 고객 서비스 담당자(CSR) 사용자를 정의하십시오. 이 사용자는 커넥터에서 사용되고 커넥터를 식별합니다. 이 사용자 ID는 이벤트 모듈 구성 파일 *crossworlds.cnf* 및 커넥터 구성 매개변수에 설정됩니다. 커넥터에서 응용프로그램으로 전송한 이벤트가 다시 커넥터로 재분배되는 것을 막으려면 사용자 정의 이벤트 기능 모듈이 이벤트를 삽입하기 전에 이 값을 확인해야 합니다. 이 시나리오를 “핑퐁 방식”이라고도 합니다.

데이터베이스에 이벤트 및 아카이브 테이블 작성

이벤트 및 아카이브 테이블은 커넥터가 선택할 이벤트를 대기열로 만드는 데 사용됩니다. 커넥터에 대한 이벤트 광고 메커니즘에서는 Infranet이 사용하는 Oracle 데이터베이스 인스턴스 안에 세 개의 이벤트 테이블이 작성되어야 합니다. 이들 테이블은 다음과 같습니다.

- XWORLDS_Events - 커넥터가 관심을 갖는 모든 보류 Infranet 이벤트를 저장하는 데 사용되는 이벤트 테이블입니다.
- XWORLDS_Archive_Events - 커넥터가 이벤트를 처리한 후에 이벤트가 기록되는 아카이브 테이블입니다.
- XWORLDS_Current_Event_ID - 마지막 이벤트 ID 번호를 저장하는 데 사용되는 테이블입니다.

주: 이 테이블은 0으로 초기화되어야 합니다.

처음 두 테이블은 커넥터가 관심을 갖는 각 Infranet 이벤트에 대해 기록될 정보를 지정합니다. 아카이브 테이블은 커넥터가 처리한 모든 이벤트를 갖고 있습니다.

이벤트 및 아카이브 테이블을 작성하려면

%ProductDir%\connectors\Portal\dependencies\config_files에 EventTable.sql을 로드하십시오.

이벤트 및 아카이브 테이블 스키마에 대한 설명

이벤트 테이블에는 다음 열이 있습니다. 이 테이블 레이아웃이 아카이브 테이블에도 사용됩니다.

표 1. 이벤트 및 아카이브 테이블 스키마

이름	유형	설명
Event_id	정수	이벤트에 대한 고유 키. 키 값은 XWORLDS_Current_Event_ID 테이블에 생성됩니다.
Object_name	char 80	응용프로그램 특정 Business Object의 이름.
Object_verb	char 80	이벤트와 연관된 Verb.
Object_key	VARCHAR	오브젝트에 대한 1차 키(POID).
Event_time	날짜 시간	이벤트가 발생한 시간.
Archive_time	날짜 시간	아카이브 테이블 전용. Portal Infranet에 의해 이벤트를 수신한 시간.
Event_status	정수	이벤트의 상태: READY_FOR_POLL 0 SENT_TO_INTERCHANGE 1 UNSUBSCRIBED_EVENT 2 IN_PROGRESS 3 ERROR_PROCESSING_EVENT -1 ERROR_SENDING_EVENT_TO_INTERCHANGE -2
Event_comment	char 255	이벤트에 대한 추가 정보를 제공하는 데 사용되는 문자열. 사용자가 이벤트 모듈 구성 파일에서 이 주석을 정의할 수 있습니다.
Event_priority	정수	이벤트와 연관된 우선순위. 숫자가 낮을수록 우선순위가 더 높습니다. 사용자가 이벤트 모듈 구성 파일에서 이 우선순위를 정의할 수 있습니다.

Portal Infranet 어댑터 및 기타 파일 설치

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

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

AIX/DB2 환경에서 어댑터 구성

AIX 시스템에 어댑터를 구성하려면 다음 지시사항을 순서대로 따르십시오.

DB2 사용자로서

1. Portal Infranet의 인스턴스에 사용된 DB2 사용자 ID로 AIX 시스템에 로그인하십시오(예: db2inst1).

2. 다음 명령을 입력하십시오.

```
db2 catalog tcpip node local_DB2_hostname remote remote_DB2_hostname  
server DB2_instance_portnumber
```

예:

```
db2 catalog tcpip node db2host remote db2host server 50000
```

3. 다음 명령을 입력하십시오.

```
db2 catalog database DB_name as DB_name at node DB2_hostname
```

예:

```
db2 catalog database CWPortal as CWPortal at node db2hos
```

4. 로그오프하십시오.

사용자 “pin”으로서

1. AIX 시스템에 pin으로 로그인하십시오(필요하면 pin 사용자 계정을 작성하십시오).
2. db2profile 파일을 사용자 pin의 홈 디렉토리로 복사하고, ../db2profile 행을 .profile 파일에 추가하여 커넥터를 시작하기 전에 db2profile 쉘 스트립트 파일을 실행하십시오.

파일은 DB2용 Portal Infranet을 설치할 때 사용된 계정의 홈 디렉토리에서 sqllib 디렉토리에 있습니다(예: /home/db2inst1/sqllib).

database.bnd 파일을 \$COSSWORKDS/connectors/Portal/dependencies/config_files/DB2에서 사용자 pin의 홈 디렉토리로 복사하십시오.

3. 다음 명령을 입력하십시오.

```
db2 connect to DB_name user pin using pin
```

예:

```
db2 connect to CWPortal user pin using pin
```

4. 다음 명령을 입력하십시오.

```
dbw bind database.bnd
```

5. 다음 명령을 입력하십시오.

```
db2 disconnect all
```

6. 다음 행을 사용자 pin의 .profile 파일에 추가하십시오.

```
export DB2DBDFT=DB_name
```

예:

```
export DB2DBDFT=CWPortal
```

7. pin을 로그오프하십시오.

사용자 “cadmin”으로서

1. 커넥터 시작 시 사용되는 사용자 cadmin으로 AIX 시스템에 로그인하십시오.
2. 커넥터를 시작하기 전에 db2profile 셸 스크립트 파일을 실행하십시오.
파일은 DB2용 Portal Infranet을 설치할 때 사용된 계정의 홈 디렉토리에서 sqllib 디렉토리에 있습니다(예: /home/db2inst1/sqllib).
connector_manager_Portal 파일을 *\$ProductDir/connectors/Portal*에서 *\$ProductDir/connectors/bin*으로 복사하십시오.
3. *\$ProductDir/connectors/Portal*로 디렉토리를 변경하십시오. start_Portal.sh 파일에서 *POTAL_HOME=/opt/portal/6.1* 행을 다음과 같은 적합한 값으로 변경하십시오.
POTAL_HOME=PortalInfranet_home_dir
예: /opt/portal/6.2
4. cadmin을 로그오프하십시오.

사용자 “pin”으로서

1. AIX 시스템에 pin으로 로그인하십시오.
2. Infranet 파일 pcm.jar 및 pcmext.jar를 *\$ProductDir/connectors/Portal/dependencies*에 복사하십시오. 이들 파일은 Infranet 서버의 *\$INFRANET/jars* 디렉토리에 있습니다.
3. custom_opcode.h 파일을 *\$INFRANET/include*에 복사하십시오.
4. custom_opcode_mem_map 파일을 *\$INFRANET/lib*에 복사하십시오.
5. .profile 파일을 pin의 홈 디렉토리(예: /home/pin)에 복사하십시오. 필요한 경우 사용자 시스템의 환경 변수 설정을 반영하도록 .profile 파일을 수정하십시오. 텍스트 편집기(예: vi)를 사용하여 변경하십시오. 환경 변수가 맞으면, 명령 프롬프트에 다음 명령을 입력하여 시스템에 환경 변수를 로드하십시오.
../.profile
6. fm_crossworlds.so 파일을 *\$INFRANET/lib* 디렉토리에 복사하십시오. 이 파일에는 이벤트에 대한 트리거가 있습니다.

주: AIX는 대소문자를 구분하므로, 파일이 없으면 모든 디렉토리 및 파일 이름에 올바른 대소문자가 사용되었는지 확인해야 합니다.

- 시스템이 .so 파일을 인식할 수 있도록 \$LIBRARY_PATH 변수에 \$INFRANET/lib 경로가 들어 있는지 확인하십시오.
7. 시스템이 .so 파일을 인식할 수 있도록 \$LIBRARY_PATH 변수에 \$INFRANET/lib 경로가 들어 있는지 확인하십시오.
 8. 다음 파일을 \$CW_PORTAL_PATH 디렉토리로 복사하십시오(예: /opt/portal/6.2/sys/cm).
 - crossworlds.cnf - 이 파일에는 이벤트 모듈의 구성 정보가 들어 있습니다.
 - event_code.txt - 이 파일에는 이벤트 모듈이 WebSphere Business Integration System 이벤트 테이블에서 항목을 생성하기 위해 사용한 Infranet 이벤트에 대한 설명이 들어 있습니다.
 9. pin_notify_cw 파일을 \$INFRANET/sys/test 디렉토리에 배치하십시오. 이 파일에는 커넥터 이벤트의 이름이 있습니다. 임의의 이벤트가 추가 또는 삭제되어야 하는 경우, 이 파일의 표준 형식을 따르십시오. /event가 /event/customer 및 /event/status와 같은 모든 서브클래스를 캡슐화함에 유의하십시오.

AIX/DB2 환경에서 Portal Infranet 응용프로그램 구성

DB2를 데이터베이스로, AIX를 운영 체제로 사용 중인 경우, 다음 지시사항에 따라 Portal Infranet 응용프로그램을 구성하십시오.

1. Infranet 응용프로그램을 중지한 후 다시 시작하십시오. \$INFRANET/bin이 \$PATH 변수에 있는지 확인하십시오. 다음 단계를 따르십시오.
 - a. 다음 명령으로 Infranet을 중지하십시오.


```
stop_all
```
 - b. 다음 명령을 입력하여 모든 Infranet 프로세스가 중지되었는지 확인하십시오. 실행 중인 모든 Infranet 프로세스의 프로세스 번호(PID)를 기록하십시오.


```
ps -ef | grep protal
```
 - c. 다음 명령으로 실행 중인 모든 Infranet 프로세스를 종료(kill)하십시오.


```
kill -9 PID
```
 - d. 다음 명령으로 Infranet을 다시 시작하십시오.


```
start_all
```
2. \$CM 디렉토리에서 다음 행을 fm_required 섹션에 있는 pin.conf 파일에 추가하십시오.


```
-cm fm_module $INFRANET/lib/fm_crossworlds.so fm_cw_config -pin
```

 반드시 \$INFRANET에 대한 전체 디렉토리 경로를 입력하십시오.
3. Infranet이 다음 명령을 입력하여 실행되고 있는지 확인하십시오.


```
ps -ef | grep portal
```

4. 디렉토리를 `$INFRANET/sys/test`로 변경하고 `pin.conf` 파일을 연 다음, 파일에 다음과 유사한 행이 있는지 확인하십시오.

```
nap_cm_ptr ip Infranet_cm_machine cm_port
```

예:

```
nap_cm_ptr ip roadrunner 11960
```

여기서 `roadrunner`는 `Infranet_cm_machine`이고 `11960`은 `cm_port`입니다.

5. `pin.conf` 파일에 다음이 들어 있는지 확인하십시오.
6. `nap login_type 1 nap login_name root.0.0.0.1 nap login_pw password`
이것은 Infranet 연결에 대한 로그인 정보를 식별합니다. 디렉토리에 `pin.conf` 파일이 없으면 디렉토리로 복사하십시오.
7. 다음 명령을 입력하여 구성 정보를 Infranet 응용프로그램으로 로드하십시오.

```
load_pin_notify pin_notify_cw
```

응답은 성공해야 합니다. 다른 응답이 표시되면 `pin_notify_cw`를 검사하십시오. 이 파일에는 특정 이벤트가 발생할 때 Infranet이 호출하는 opcode가 들어 있습니다. `pin_notify_cw`는 `load_pin_notify` 실행 파일과 동일한 디렉토리에 위치해야 함에 유의하십시오.

8. `$INFRANET_VAR/cm` 디렉토리에서 로그 파일을 검사하고 `$CM`에 코어가 있는지 확인하십시오(예: `cm.pinlog`에 있는 `cm started at "Tue Jul 9 20:49:37 2002"`). Infranet 관리자를 시작하십시오.
9. 커넥터를 테스트하려면 계정을 입력하거나 수정하고 적당한 이벤트 항목에 대한 이벤트 테이블 `xworlds_events`를 확인하십시오. 더미 이벤트를 발생시킬 수 있으므로, 테스트가 완료되면 이벤트 항목을 삭제하십시오.

Oracle 환경에서 어댑터 구성

Oracle을 데이터베이스로 사용 중인 경우, 다음 지시사항에 따라 커넥터를 구성하십시오.

1. UNIX 시스템에 `pin`으로 로그인하십시오. 필요한 경우 이 계정을 작성하십시오.
2. Infranet 파일 `pcm.jar` 및 `pcmext.jar`를 `$ProductDir/Connector/Portal/dependencies`에 복사하십시오. 이 파일은 Infranet 6.1.0 서버의 `infranet/jars` 디렉토리에 있습니다.
3. `.profile` 파일을 `pin` 사용자의 홈 디렉토리(예: `/home/pin`)에 복사하십시오. 필요한 경우 사용자 시스템의 환경 변수 설정을 반영하도록 `.profile` 파일을 수정하십시오. 텍스트 편집기(예: `vi`)를 사용하여 모든 변경을 수행하십시오.

환경 변수가 맞을 때 명령 프롬프트에 다음 명령을 입력하여 시스템에 환경 변수를 로드하십시오.

```
source .profile
```

4. `fm_crossworlds.so` 파일을 `$INFRANET/lib` 디렉토리에 배치하십시오. 이 파일에는 이벤트에 대한 트리거가 있습니다.

UNIX는 대소문자를 구분하므로, 파일이 없으면 모든 디렉토리 및 파일 이름의 대소문자가 올바른지 확인해야 합니다.

5. 시스템이 커넥터 `.so` 파일을 인식할 수 있도록 `$LIBRARY_PATH` 변수에 `$INFRANET/lib` 경로가 들어 있는지 확인하십시오.
6. 다음 파일을 `$CW_PORTAL_PATH` 디렉토리에 복사하십시오.
 - `crossworlds.cnf`. 이 파일에는 이벤트 모듈에 대한 구성 정보가 있습니다.
 - `event_code.txt`. 이 파일에는 이벤트 모듈이 WebSphere Business Integration Adapters 이벤트 테이블에 항목을 생성하는 데 사용할 Infranet 이벤트에 대한 설명이 있습니다.
7. `pin_notify_cw`를 `$INFRANET/sys/test` 디렉토리에 배치하십시오. 이 파일에는 커넥터 이벤트의 이름이 있습니다. 임의의 이벤트가 추가 또는 삭제되어야 하는 경우, 이 파일의 표준 형식을 따르십시오. `/event`가 `/event/customer`, `/event/status`와 같은 모든 서브클래스를 캡슐화함에 유의하십시오.
8. Infranet 응용프로그램을 중지한 후 다시 시작하십시오. `$INFRANET/bin`이 `$PATH` 변수에 있는지 확인하십시오. 다음 단계를 따르십시오.
 - a. 다음 명령으로 Infranet을 중지하십시오.

```
stop_all
```
 - b. 다음 명령을 입력하여 모든 Infranet 프로세스가 중지되었는지 확인하십시오. 실행 중인 모든 Infranet 프로세스의 프로세스 번호(PID)를 기록하십시오.

```
ps -ef|grep portal
```
 - c. 다음 명령으로 실행 중인 모든 Infranet 프로세스를 종료(kill)하십시오.

```
kill -9 <PID>
```
 - d. 다음 명령으로 Infranet을 다시 시작하십시오.

```
start_all
```
9. `$CM` 디렉토리에서 `pin.conf` 파일을 편집하여 `fm_required` 섹션에 다음 행을 추가합니다. 반드시 `$INFRANET`에 대한 전체 디렉토리 경로를 입력하십시오.

```
- cm fm_module $INFRANET/lib/fm_crossworlds.so fm_cw_config -pin
```
10. `ps -ef|grep portal` 명령을 입력하여 Infranet이 실행 중인지 확인하십시오.
11. 디렉토리를 `$INFRANET/sys/test`로 변경하고 `pin.conf` 파일을 연 다음, 파일에 다음과 유사한 행이 있는지 확인하십시오.

```
- nap cm ptr ip Infranet_cm_machine cm_port
```

예:

```
- nap cm ptr ip roadrunner 11960
```

여기서 roadrunner는 *Infranet_cm_machine*이고, *cm_port*는 11960입니다.

위의 명령문 외에, *pin.conf* 파일은 다음을 포함해야 합니다.

```
- nap login_type 1
- nap login_name root.0.0.0.1
- nap login_pw password
```

이것은 Infranet 연결에 대한 로그인 정보를 식별합니다. 디렉토리에 *pin.conf* 파일이 없으면 디렉토리에 복사하십시오.

12. 구성 정보를 Infranet 응용프로그램에 로드하려면 다음 명령을 입력하십시오.

```
load_pin_notify pin_notify_cw
```

응답이 *successful*이어야 합니다. 다른 응답이 표시되면 *pin_notify_cw*를 검사하십시오. 이 파일에는 특정 이벤트가 발생할 때 Infranet이 호출하는 opcode가 들어 있습니다. *pin_notify_cw*는 *load_pin_notify* 실행 파일과 동일한 디렉토리에 위치해야 함에 유의하십시오.

13. *\$INFRANET_VAR/cm* 디렉토리에서 로그 파일을 검사하고 *\$CM*에 코어가 있는지 확인한 후 Infranet 관리자를 시작하십시오.

14. 커넥터를 테스트하려면 계정을 입력하거나 수정하고 적당한 이벤트 항목에 대한 이벤트 테이블 *xworlds_events*를 확인하십시오. 결국 더미 이벤트를 발생시킬 수 있으므로, 테스트가 완료되면 이벤트 항목을 삭제해야 합니다.

커넥터를 시작하려면 32 페이지의 『커넥터 시작』을 참조하십시오.

Windows 환경에서 이벤트 메커니즘 설치 및 구성

Infranet은 응용프로그램에서 발생하는 조치를 추적할 수 있게 하는 이벤트 메커니즘을 갖고 있습니다. 커넥터가 이벤트 전달을 처리할 수 있으려면 Infranet에 이벤트 공고 메커니즘을 구성해야 합니다. 다음 단계를 따르십시오.

1. 다음 WebSphere Business Integration Adapters 이벤트 모듈 파일을 복사하십시오.
 - *FmCw.dll*을 *%INFRANET%\lib*에 복사하십시오. 이 DLL은 커넥터 이벤트 기능 모듈입니다.
 - *event_code.txt*를 *%INFRANET%\sys\cm*에 복사하십시오. 이 파일은 이벤트 모듈에 대한 Infranet 이벤트를 설명하며, 이 모듈이 WebSphere Business Integration Adapters 이벤트 테이블에 Infranet 이벤트에 대응하는 항목을 생성합니다.
*%INFRANET%*은 Infranet 홈 디렉토리에 있음에 유의하십시오.
 - *custom_opcode.h*를 *%INFRANET%\include*에 복사하십시오.
 - *custom_opcode_mem_map*을 *%INFRANET%\lib*에 복사하십시오.

- Infranet 파일 pcm.jar 및 pcmext.jar를 \$CROSSWORKDS/Connector/Portal/dependencies에 복사하십시오. 이 파일은 Infranet 서버의 infranet/jars 디렉토리에 있습니다.
2. %INFRANET%\sys\cm에서 연결 관리자가 커넥터 이벤트 기능 DLL인 FmCw.dll을 실행할 수 있도록 연결 관리자 pin.conf 구성 파일을 편집하십시오. 이 파일에 다음 행을 추가하십시오.


```
- cm fm_module ..\..\lib\FmCw.dll fm_cw_pol_config_func - pin --
ops_fields_extension_file ..\..\lib\custom_opcode_mem_map
```
 3. pin_notify_cw 파일을 %INFRANET%\sys\test 디렉토리에 배치하십시오. 이 파일은 특정 이벤트가 발생했을 때 Infranet이 호출할 opcode를 지정합니다. 임의의 opcode를 추가 또는 삭제해야 하는 경우, 이 파일의 표준 형식을 따르십시오. /event가 /event/customer, /event/status와 같은 모든 서브클래스를 캡슐화함에 유의하십시오.
 4. %INFRANET%\sys\test의 pin.conf 파일을 다음과 유사하게 갱신하십시오.


```
- nap cm ptr ip <Infranet_cm_machine> <port>
```

예:

```
- nap cm ptr ip cwengtest 11960
```

여기서 cwengtest는 <Infranet_cm_machine>이고 <port>는 11960입니다.

위의 명령문 외에, pin.conf 파일은 다음을 포함해야 합니다.

```
- nap login_type 1
- nap login_name root.0.0.0.1
- nap login_pw password
```

이것은 Infranet 연결에 대한 로그인 정보를 식별합니다. 디렉토리에 pin.conf 파일이 없으면 디렉토리에 복사하십시오.

5. crossworlds.cnf 환경 구성 파일을 %INFRANET%\sys\cm에 복사하십시오. 이 파일은 이벤트 기능 모듈 FmCw.dll이 환경에 관한 정보를 얻을 수 있게 합니다. 필요한 경우 사용자 시스템에 대해 이 파일을 편집하십시오. 이 파일에 대한 내용 예는 다음과 같습니다.

```
db user = pin
db password = pin
crossworlds id = 0.0.0.1 \service\admin_client 14088
log level = 3
log file = D:\pinlog.log
```

여기서

db user	Portal Infranet 데이터베이스에 연결하는 사용자의 이름입니다.
db password	암호입니다.

crossworlds id	Portal Infranet에서 WebSphere Business Integration Adapters 사용자를 표시하는 POID입니다.
log level	로그 레벨을 표시하는 숫자입니다. 0 : 추적 없음 1 : 오류만 2 : 오류 및 경고 3 : 오류, 경고 및 디버그(모든 추적)
log file	로그 파일의 이름입니다.

이 파일은 해시 테이블로 해석됩니다. 키워드는 등호 왼쪽의 문자열이고 값은 오른쪽의 문자열입니다.

6. crossworlds.cnf 파일에 대한 경로를 지정하는 CW_PORTAL_PATH라는 환경 변수를 작성하십시오.
7. 다음 순서로 Infranet 서비스를 다시 시작하십시오.
 - Infranet 데이터 관리자
 - Infranet Java 서버
 - Infranet 연결 관리자
8. 다음 구문을 사용하여 pin_notify_cw 파일을 Infranet 응용프로그램에 로드하십시오.

```
load_pin_notify .\pin_notify_cw
```

오류가 발생하는 경우, \bin 디렉토리가 PATH에 있는지 확인하여 실행 파일을 찾을 수 있도록 하십시오. 오류 메시지에 대해서는 실행 디렉토리의 pinlog.log를 확인하십시오.

커넥터 구성

어댑터에는 표준 구성 등록 정보와 어댑터 특정 구성 등록 정보, 두 유형의 구성 등록 정보가 있습니다. 이들 등록 정보 중 일부의 값을 설정해야 커넥터를 실행할 수 있습니다.

Connector Configurator(WebSphere MQ Integrator Broker가 통합 브로커일 때) 또는 System Manager에서 액세스되는 Connector Configurator(ICS가 통합 브로커일 때)에서 커넥터 등록 정보를 구성합니다. 자세한 구성 정보는 97 페이지의 부록 B 『Connector Configurator』 또는 *Connector Development Guide for Java*를 참조하십시오.

표준 커넥터 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이들 등록 정보의 문서에 대해서는 79 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

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

주: 이 커넥터가 단일 스레드이므로 AgentConnections 등록 정보를 활용할 수 없습니다.

표 2에서는 부록에 있는 구성 등록 정보 중 이 커넥터에만 고유한 정보를 제공합니다.

표 2. 이 커넥터에 고유한 등록 정보

등록 정보	주
CharacterEncoding	이 커넥터는 이 등록 정보를 사용하지 않습니다.
Locale	커넥터가 국제화되었으므로, 이 등록 정보 값을 변경할 수 있습니다. 현재 지원되는 커넥터를 알아보려면 커넥터의 릴리스 정보를 참조하십시오.

중요: WebSphere MQ Integrator Broker는 다중 로케일을 지원하지 않습니다. 설치의 모든 구성요소(예: 모든 어댑터, 응용프로그램 및 통합 브로커 자체)는 같은 로케일로 설정되도록 하십시오.

주: 이 커넥터가 모든 통합 브로커를 지원하므로 모든 브로커에 대한 구성 등록 정보가 해당됩니다.

커넥터 특정 등록 정보

커넥터 특정 구성 등록 정보는 런타임 시 커넥터에 필요한 정보를 제공합니다. 커넥터 특정 등록 정보는 또한 커넥터를 다시 코드화하고 재빌드할 필요없이 커넥터 안에서 정적 정보 또는 논리를 변경하는 방법을 제공합니다.

표 3에서는 커넥터에 대한 커넥터 특정 구성 등록 정보를 나열합니다. 등록 정보에 대한 설명은 뒤따르는 절을 참조하십시오.

표 3. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수
Application password	사용자 계정의 암호		예
ApplicationUserName	사용자 계정의 이름		예
CommentFail	실패한 이벤트에 대한 이벤트 테이블 주석	Fail	아니오
CommentSucceed	성공한 이벤트에 대한 이벤트 테이블 주석	Succeed	아니오

표3. 커넥터 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
DatabaseInfo	jdbc:oracle:thin@CWENGTST:1521:portaldb	서버 호스트 이름, Portal 데이터베이스 포트 번호 및 데이터베이스의 이름.	예
dbDriverClass(DB2 데이터베이스 전용)	데이터베이스의 드라이버 클래스 이름		예
DbName	Oracle 리스너 이름		예
DbPassword	데이터베이스 암호		예
dbURL(DB2 데이터베이스 전용)	데이터베이스의 URL		예
DbUser	데이터베이스 사용자 이름	pin	예
InfDatabase	문자열	0.0.0.1	예
InfHost	호스트 시스템 이름 및 포트	//CWENGTST1:11960	예
InfLogFile	로그 파일의 이름	InfConnection.txt	예
InfranetConnections	연결 풀링을 위해 커넥터가 연 Infranet 연결 횟수	5	예
InfService	문자열	service\admin_client 1	예
InfType	0 또는 1	1	예
InfVersion	Infranet 응용프로그램의 버전		예
PollQuantity	선택할 이벤트 수	1	아니오
UseDefaults	true 또는 false	false	아니오

Application password

WebSphere Business Integration System 사용자 계정의 암호입니다.

ApplicationUserName

커넥터에 대한 사용자 계정의 이름입니다.

CommentFail

실패의 경우에 이벤트 테이블 주석입니다. 기본값은 Fail입니다.

CommentSucceed

성공의 경우에 이벤트 테이블 주석입니다. 기본값은 Succeed입니다.

DatabaseInfo

이 등록 정보는 데이터베이스에 연결하기 위해 JDBC 드라이버에서 사용되는 URL을 정의합니다.

예:

`jdbc:oracle:thin:@CWENGTST1:1521:portaldb`

URL의 특정 형식에 대해서는 JDBC 문서를 참조하십시오.

dbDriverClass

데이터베이스의 드라이버 클래스 이름(DB2만)입니다. dbDriverClass 및 dbURL에 연관된 값이 있으면, 커넥터는 이들 값을 사용하고 dbName 및 DatabaseInfo는 무시합니다. 이 등록 정보 중 하나가 null로 설정되면, dbName 및 DatabaseInfo가 읽혀지고, 커넥터는 데이터베이스가 Oracle인 것으로 가정합니다. DB2의 경우 COM.ibm.db2.jdbc.app.DB2Driver를 설정하십시오.

DbName

Infranet 데이터베이스에 대한 Oracle 리스너 이름입니다.

DbPassword

데이터베이스 암호입니다.

dbURL

데이터베이스의 URL(DB2만)입니다. dbDriverClass 및 dbURL에 연관된 값이 있으면, 커넥터는 이들 값을 사용하고 dbName 및 DatabaseInfo는 무시합니다. 이 등록 정보 중 하나가 null로 설정되면, dbName 및 DatabaseInfo가 읽혀지고, 커넥터는 데이터베이스가 Oracle인 것으로 가정합니다. jdbc:db2:DB_name을 설정하십시오(예: jdbc:db2:CWPortal).

DbUser

데이터베이스 사용자 이름으로, 대개 pin입니다.

InfDatabase

Infranet 양식의 문자열입니다. 기본값은 0.0.0.1입니다.

InfHost

호스트 시스템 이름 및 포트입니다(예: //engtest2:11960).

기본값은 //CWENGTEST1:11960입니다.

InfLogFile

기본 로그로 사용할 파일의 이름입니다. 기본값은 InfConnection.txt입니다.

InfranetConnections

이 등록 정보는 커넥터가 연결 풀링을 위해 Infranet 응용프로그램에서 여는 연결 횟수를 정의하는 데 사용됩니다. 커넥터는 지정된 수의 연결을 유지합니다. Business Object 프로세스에 연결이 필요하면, 풀에서 하나를 할당합니다. 그러면 이 연결이 사용 가능한 풀에서 제거되어 사용 중인 풀로 추가됩니다. Business Object 프로세스가 완료되면, 사용 중인 풀에서 연결이 제거되어 다시 사용 가능한 풀로 추가됩니다. 이러한 방식으로 연결 풀을 사용하면 연결이 각 비즈니스 프로세스를 위해 계속 열리거나 닫히지 않아도 되므로 성능이 현저히 향상됩니다.

주: 등록 정보가 설정되지 않으면, 커넥터는 커넥터가 시작될 때
NumberFormatException 예외를 발생시킵니다.

InfService

대개 service\admin_client 1인 문자열입니다. 이것이 기본값입니다.

InfType

연결 유형입니다. 유일하게 가능한 값은 0 또는 1입니다. 기본값은 1입니다.

InfVersion

Infranet 응용프로그램의 버전입니다.

PollQuantity

단일 폴에서 선택되는 이벤트 수입니다. 기본값은 1입니다.

UseDefaults

UseDefaults가 true로 설정되거나 설정되지 않은 경우, 커넥터는 각 Business Object 속성에 대해 유효한 값이나 기본값이 제공되는지 여부를 검사합니다. 값이 제공되는 경우 작성은 성공하고, 그렇지 않으면 실패합니다.

매개변수가 false로 설정되는 경우, 커넥터는 유효한 값에 대해서만 검사합니다. 유효한 값이 제공되지 않는 경우, Create 조작은 실패합니다.

기본값은 false입니다.

새 Business Object에 대한 이벤트 메커니즘 사용자 정의

새 Business Object를 작성하는 경우, Business Object에 대한 각 조치에 대해 생성되는 이벤트를 판별해야 합니다. 이를 판별한 후 이벤트 모듈 구성 파일을 사용자 정의 하여 이벤트 모듈 DLL이 해당 유형의 이벤트를 찾을 수 있도록 해야 합니다. 이벤트 모듈 구성 파일의 이름은 event_code.txt이고, \$INFRANET\$\sys\cm에 있습니다.

Infranet은 이벤트 모듈이 저장 가능 클래스의 어떤 부분이(즉, 어떤 Business Object가) 호출되었는지를 식별할 수 있도록 이벤트에 대한 충분한 데이터를 생성합니다. 이벤트 모듈이 이벤트를 확보할 때 계정, 사용자 및 호출 프로그램 같은 정보가 들어 있는 저장 가능 클래스의 인스턴스를 확보합니다.

갱신을 삭제 또는 작성과 구별하기 위해 이벤트 모듈은 원래 값과 갱신된 값을 비교합니다. 그러나 작성 또는 삭제가 발생했는지 알려면 이벤트 모듈은 조치가 루트 레벨에서 수행될 때 저장 가능 클래스를 검색해야 하며, 그렇지 않으면 하위 오브젝트에 대한 요소 ID를 찾을 수 있습니다. 하위가 추가될 때 요소 ID는 양수이며 배열에 이 요소 위치가 있습니다. 음수인 경우 삭제됩니다.

이벤트 모듈 구성 파일의 구문

이벤트 모듈 구성 파일을 수정할 때 파일에 대한 변경사항은 다음 구문 규칙을 엄격하게 준수해야 합니다.

1. 주석 행은 두 개의 대시로 시작합니다.
2. 한 이벤트는 한 행에 기술됩니다. 매개변수는 파이프 문자(|)로 분리됩니다.
3. 행은 다음 구문을 따릅니다.

```
<event>|<Inf.action>|<array>|<key poid>|<constraints>|<BO.verb>|<priority>|<comment>
```

여기서

event	이벤트의 저장 가능 클래스 이름
Infranet action	C(작성), U(갱신) 또는 D(삭제)입니다. 이것은 Portal Infranet에서 수행되는 조치를 표시합니다.
array	조치가 수행되는 배열을 표시하는 Portal Infranet 코드입니다. 배열은 이벤트 정보에서 검색되어야 하는 Infranet 요소입니다.
poid	Portal Infranet에서 각 필드는 연관된 번호를 갖습니다. 예를 들어, PIN_FLD_NAMEINFO는 Infranet 코드 156과 연관됩니다. 필드 및 연관된 번호의 목록은 \$INFRANET\$\Include\pin_flds.h 파일에서 찾을 수 있습니다.
constraints	<p>작성, 갱신 또는 삭제된 저장 가능 클래스의 키인 필드를 표시하는 Portal Infranet 코드입니다.</p> <p>무엇이 발생했는지를 정확하게 판별하기 위해 필요한 제한조건의 목록입니다. 제한조건은 다음 키워드를 지원합니다.</p> <ul style="list-style-type: none"> • exists 또는 not_exists: 오브젝트가 존재하는지 여부를 지정하는 true 또는 false. • =, >, >=, <=, < : 숫자 유형에 대한 비교 연산자 • equal, nequal, contains: 문자열 유형에 대한 비교 연산자. • &는 제한조건을 분리하는 데 사용됩니다. 제한조건은 &로 분리되는 모든 조건이 true일 때만 true입니다. <p>or를 지정하려는 경우 여러 행을 사용하십시오. 제한조건이 true인 첫 번째 행이 실행됩니다.</p>
BO.verb	Portal Infranet의 조치에 대응하는 Business Object 및 Verb의 이름입니다.
priority	이벤트의 우선순위
comment	이벤트 테이블에 삽입할 주석

이벤트 모듈 구성 파일 예

아래 텍스트는 event_code.txt 파일의 예를 보여줍니다. 이 예는 account 저장 가능 클래스에 대한 이벤트를 지정하는 행을 포함합니다. 대시로 시작하는 행은 주석입니다.

```
-- Account creation: PIN_FLD_STATUSES[0].PIN_FLD_STATUS[0] = 0 &
   PIN_FLD_SYS_DESCR = "Set Status (acct)": OK
/event/customer/status |U |144 |40 |40 exists&55;5 equal "Set Status
(acct)"&144:0-145;3 = 0 |Portal_Account.Create |1 |Account Creation

-- Account Updated (status updated) : PIN_FLD_STATUSES[1].PIN_FLD_STATUS[0] =
```

```

10100 or 10103 or 10102 & & PIN_FLD_SYS_DESCR = "Set Status (acct)" : OK
/event/customer/status |U |144 |40 |40 exists&55;5 equal "Set Status (acct)
"&144:1-145;3 > 0 |Portal_Account.Update |1 |Status Updated

-- Account Updated (new contact added):OK
/event/customer/nameinfo |C |156 |40|40 exists|Portal_Account.Update|1|new contact

-- Account Updated (contact updated): OK
/event/customer/nameinfo |U |156 |40 |40 exists&17;5 equal "Customer Mngmt. Event
Log" |Portal_Account.Update |2 |contact_update

-- Account Updated (contact deleted):OK
/event/customer/nameinfo |D |156 |40 |40 exists&67;5 nequal "Automatic Account
Creation"|Portal_Account.Update |2 |contact_delete

-- Account Updated (billinfo updated) : two PIN_FLD_BILLINFO and
PIN_FLD_BILLINFO[0].PIN_FLD_BILL_TYPE[0] <> 0 : OK
/event/customer/billinfo |U |126 |40 |40 exists & 126:0-127;3 > 0
Portal_Account.Update |2 |billinfo_update

```

이벤트 구성 파일 항목 정의

이벤트가 발생할 때 Infranet은 이벤트를 표시하는 Flist를 생성합니다. 이벤트 기능 모
 들이 Flist를 시험하여 Verb와 발생한 조치를 식별합니다.

예를 들어, 이벤트를 표시하는 Flist가 다음과 같다고 가정하십시오.

```

0 PIN_FLD_POID          POID [0] 0.0.0.1 /event/customer/nameinfo -1 0
0 PIN_FLD_NAME         STR [0] "Customer Mngmt. Event Log"
0 PIN_FLD_USERID      POID [0] 0.0.0.1 /service/admin_client 2 1
0 PIN_FLD_SESSION_OBJ POID [0] 0.0.0.1 /event/session 10366 0
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 9406 32
0 PIN_FLD_PROGRAM_NAME STR [0] "Admin Manager"
0 PIN_FLD_START_T      TSTAMP [0] (942104217) 11/08/99 15:36:57
0 PIN_FLD_END_T        TSTAMP [0] (942104217) 11/08/99 15:36:57
0 PIN_FLD_SYS_DESCR   STR [0] "Set Name Info"
0 PIN_FLD_NAMEINFO     ARRAY [0] allocated 20, used 17
1   PIN_FLD_SALUTATION  STR [0] ""
1   PIN_FLD_LAST_NAME   STR [0] "Event Test1"
1   PIN_FLD_LAST_CANON  STR [0] "event test1"
1   PIN_FLD_FIRST_NAME  STR [0] "Event Test1"
1   PIN_FLD_FIRST_CANON STR [0] "event test1"
1   PIN_FLD_MIDDLE_NAME STR [0] ""
1   PIN_FLD_MIDDLE_CANON STR [0] ""
1   PIN_FLD_TITLE       STR [0] "Event Test1 "
1   PIN_FLD_COMPANY     STR [0] "Event Test1"
1   PIN_FLD_ADDRESS     STR [0] "Event Test1"
1   PIN_FLD_CITY        STR [0] "Event Test1"
1   PIN_FLD_STATE       STR [0] "CA"
1   PIN_FLD_ZIP         STR [0] "00000"
1   PIN_FLD_COUNTRY     STR [0] "US"
1   PIN_FLD_EMAIL_ADDR  STR [0] ""
1   PIN_FLD_CONTACT_TYPE STR [0] "Billing"
1   PIN_FLD_ELEMENT_ID  UINT [0] 1
0   PIN_FLD_NAMEINFO   ARRAY [1] allocated 20, used 19
1   PIN_FLD_SALUTATION  STR [0] ""
1   PIN_FLD_LAST_NAME   STR [0] "Event Test1"
1   PIN_FLD_LAST_CANON  STR [0] "event test1"
1   PIN_FLD_FIRST_NAME  STR [0] "Event Test1"
1   PIN_FLD_FIRST_CANON STR [0] "event test1"

```



```

1 PIN_FLD_MIDDLE_NAME STR [0] ""
1 PIN_FLD_MIDDLE_CANON STR [0] NULL str ptr
1 PIN_FLD_TITLE STR [0] "Event Test1"
1 PIN_FLD_COMPANY STR [0] "Event Test1"
1 PIN_FLD_ADDRESS STR [0] "Event Test1"
1 PIN_FLD_CITY STR [0] "Event Test1"
1 PIN_FLD_STATE STR [0] "CA"
1 PIN_FLD_ZIP STR [0] "00000"
1 PIN_FLD_COUNTRY STR [0] "US"
1 PIN_FLD_EMAIL_ADDR STR [0] ""
1 PIN_FLD_CONTACT_TYPE STR [0] "Billing"
1 PIN_FLD_ELEMENT_ID UINT [0] 1
1 PIN_FLD_CANON_COUNTRY STR [0] "US"
1 PIN_FLD_CANON_COMPANY STR [0] "event test1"
0 PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1 /item 11454 0
0 PIN_FLD_CURRENCY UINT [0] 840

```

이벤트 모듈 파일에 행을 추가할 때 필드 이름 대신 필드와 연관된 번호를 지정해야 합니다. 예를 들어, 앞의 Flist에서 굵은체로 표시되는 PIN_FLD_NAMEINFO 필드는 코드 156으로 표시됩니다.

Flist의 첫 번째 레벨보다 더 깊은 필드의 경우 제한조건을 사용하여 필드를 식별해야 합니다. 제한조건의 예외로 각 필드가 간단한 번호로 식별되고, 이 번호가 필드와 연관된 번호입니다.

지정된 이벤트가 다양한 이유로 생성될 수 있으므로, 때로는 이벤트가 특정 Business Object의 결과인지 여부를 판별하기 위한 제한조건이 필요합니다. 그러므로 exists 같은 제한조건을 추가할 필요가 있을 수 있습니다.

다음 구문이 필드를 식별하기 위해 제한조건에서 사용되어야 합니다.

```
[<array code:array element>-field;type]
```

예를 들어, Infranet 코드 156이 PIN_FLD_NAMEINFO를 표시하고 코드 161이 PIN_FLD_LAST_NAME을 표시하는 경우, 요소 ID 1의 PIN_FLD_NAMEINFO 배열에 있는 PIN_FLD_LAST_NAME 필드의 표시가 156:1-161;5 = PINFLDNAMEINFO[1].PIN_FLD_LASTNAME과 함께 이 필드와 관련된 제한조건에 표시됩니다.

제한조건에서 세미콜론이 뒤따르는 마지막 번호는 오브젝트의 유형을 표시합니다. 유형은 비교가 수행되어야 하는 데이터의 유형을 표시합니다. 예를 들어, 번호 5는 문자열을 표시합니다. 자세한 정보는 \$INFRANET\$\Include\pin_type.h에 있는 연관된 유형의 목록을 참조하십시오.

pin_notify_cw 파일에 이벤트 추가

새 Business Object를 추가할 때 해당 이벤트를 pin_notify_cw 파일에도 추가해야 합니다. 이 파일에는 커넥터 이벤트의 이름이 있습니다. 이벤트를 추가하려면 28 페이지의 『이벤트 구성 파일 항목 정의』에 나와 있는 파일의 표준 형식을 따르십시오.

CrossWorlds 이벤트 공고 모듈(FmCw.dll)은 전송된 Portal 이벤트의 사용자 ID를 확인합니다(pin_notify_cw 파일에 정의됨). 모듈에 전송된 이벤트와 연관된 PIN_FLD_USERID가 없으면, 오류가 발생하여 오브젝트를 온라인으로 저장하는 데 문제가 생깁니다. 그러한 ID를 포함하도록 이 유형의 이벤트를 조정할 필요가 있습니다 (f-list 또는 저장 가능한 클래스). crossworlds.cnf 구성 파일에 정의된 로그 파일에서 이러한 오류를 확인하십시오.

이벤트 모듈은 사용자 ID를 확인하여 커넥터에서 응용프로그램에 전송한 이벤트가 이벤트 대기열에 추가되지 않도록 합니다. 이를 핑퐁 방식이라 합니다. "/event/customer/billinfo"는 그러한 문제가 있는 이벤트 유형의 한 예입니다.

주: Infranet 조작에 대해 생성되는 모든 이벤트를 식별하려면 모든 Infranet 이벤트에 대한 이벤트 모듈을 호출하십시오(load_pin_notify 구성 파일에서 /event 사용). 그러면 이벤트 모듈이 발생하는 모든 이벤트에 대해 트리거됩니다. 이벤트 모듈 구성 파일에서 로그 레벨을 3으로 설정한 후 pinlog.log 파일에서 이 조작에 대해 Infranet이 전송하는 이벤트 Flist가 있어야 합니다.

Infranet 사용자 정의 속성의 선택적 구성 선언

Infranet 응용프로그램이 사용자 정의된 경우에 대응하는 Java 클래스가 생성되어야 합니다. 이와 같은 클래스를 생성하기 위해 Infranet이 도구를 제공합니다. 커넥터는 런타임 시 이 클래스에 대해 알아야 하므로, 해당 클래스가 CLASSPATH에 선언되어야 합니다.

1. 디렉토리(예: c:\CustomAttributes)를 작성하십시오.
2. 사용자 정의 필드 및 해당 값을 갖는 #include 파일을 작성하십시오.
3. com.portal.pcm.fields를 패키지 이름으로 표시하여 이 파일에 대해 custom_fields.pl 스크립트를 실행하십시오.
4. Java 파일을 컴파일하여 클래스 파일을 작성하십시오.

```
javac -classpath c:\program files\infranet\java\pcmext.jar *.java
```
5. 현재 디렉토리 com\portal\pcm\fields 아래에 디렉토리 계층 구조를 작성하십시오.
6. 컴파일러가 생성하는 .class를 이 디렉토리에 복사하십시오.
7. 현재 디렉토리(예: c:\CustomAttributes)를 커넥터의 CLASSPATH에 추가하십시오.
8. 필요한 경우 \bin\start_Portal.bat 파일을 수정하십시오.

다중 커넥터 인스턴스 작성

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

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

새 디렉토리 작성

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

```
ProductDir\connectors\connectorInstance
```

여기서 connectorInstance는 고유하게 커넥터 인스턴스를 식별합니다.

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

```
ProductDir\repository\connectorInstance
```

Business Object 정의 작성

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

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

```
ProductDir\repository\initialConnectorInstance
```

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

커넥터 정의 작성

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

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

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

시작 스크립트 작성

시작 스크립트를 작성하려면 다음을 수행하십시오.

1. 초기 커넥터의 시작 스크립트를 복사한 다음 시작 스크립트 이름에 커넥터 디렉토리의 이름이 포함되도록 이름을 지정하십시오.
dirname
2. 시작 스크립트를 31 페이지의 『새 디렉토리 작성』에서 작성한 커넥터 디렉토리에 배치하십시오.
3. 시작 스크립트 단축 아이콘을 작성하십시오(Windows 전용).
4. 초기 커넥터의 단축 아이콘 텍스트를 복사한 후 새 커넥터 인스턴스의 이름과 일치하도록 초기 커넥터의 이름을 변경하십시오(명령행에서).

이제 통합 서버에서 두 커넥터 인스턴스를 동시에 실행할 수 있습니다.

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

커넥터 시작

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

ProductDir\connectors\connName

여기서 *connName*은 커넥터를 식별합니다. 시작 스크립트의 이름은 표 4에 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.

표 4. 커넥터의 시작 스크립트

운영 체제	시작 스크립트
UNIX 기반 시스템	connector_manager_connName
Windows	start_connName.bat

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

- Windows 시스템의 경우, 시작 메뉴에서 다음을 수행하십시오.
프로그램 > **IBM WebSphere Business Integration Adapters** > 어댑터 > 커넥터를 선택하십시오. 기본적으로 프로그램 이름은 “IBM WebSphere Business Integration Adapters”입니다. 그러나 이 이름을 사용자 정의할 수 있습니다. 대안으로, 커넥터에 대한 데스크탑 단축 아이콘을 작성할 수 있습니다.
- 명령행에서 커넥터를 시작할 경우에는 다음을 수행하십시오.

- Windows 시스템의 경우:

```
start_connName connName brokerName [-cconfigFile ]
```

- UNIX 기반 시스템의 경우:

```
connector_manager_connName -start
```

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

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

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

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

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

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

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

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

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

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

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

커넥터 중지

커넥터를 중지하는 방법은 다음과 같이 커넥터를 시작한 방법에 따라 다릅니다.

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

```
connector_manager_connName -stop
```

여기서 *connName*은 커넥터의 이름입니다.

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

제 3 장 Business Object 이해

이 장에서는 커넥터가 Business Object를 처리하는 방법에 대해 설명하고 Portal에 대한 Business Object 구현 지침을 제공합니다. 다음 주제에 대해 다룹니다.

- 『Portal Infranet 응용프로그램 배경』
- 39 페이지의 『Portal Infranet 응용프로그램 특정 Business Object 구조』
- 43 페이지의 『Business Object 정의 지침』
- 43 페이지의 『Business Object 응용프로그램 특정 정보』

주: Infranet 응용프로그램에 익숙하지 않은 경우, Portal Infranet에 대한 Business Object 설계 및 구현이 어려울 수 있습니다. 이 경우 응용프로그램 전문가와 함께 작업하는 것이 바람직합니다. Infranet 개념 또는 프로그래밍 요소에 대한 정보는 Infranet 문서를 참조하십시오.

Portal Infranet 응용프로그램 배경

이 절에서는 Portal Infranet 응용프로그램 특정 Business Object의 설계 및 구현에 영향을 주는 Infranet 응용프로그램의 몇 가지 기본 요소 개요에 대해 간략하게 설명합니다. Infranet은 시스템에서 기능을 정의, 확장 또는 액세스하는 데 사용되는 네 가지 기본 프로그래밍 요소를 정의합니다. Portal에 대한 Business Object를 설계하려면 이들 요소에 익숙해야 합니다. 요소들은 뒤따르는 절에서 간략하게 설명됩니다.

- 저장 가능 클래스
- 저장 가능 오브젝트
- 필드 및 필드 목록(Flist)
- Opcode

저장 가능 클래스 및 오브젝트

Infranet의 저장 가능 클래스에는 클래스에 관한 정보를 저장하는 필드가 있습니다. 표준 저장 가능 클래스는 계정, 서비스, 청구, 송장 및 Infranet에 의해 사전정의되는 기타 클래스를 포함합니다. Infranet 기능을 확장하기 위해 새 저장 가능 클래스를 작성하거나 기존 클래스의 서브세트를 작성할 수 있습니다.

저장 가능 클래스는 실제 데이터를 포함하지 않습니다. 즉, WebSphere Business Integration Adapters Business Object 정의가 Business Object 구조를 정의하지만 데이터는 포함하지 않는 것처럼 저장 가능 클래스는 오브젝트 스펙입니다. 저장 가능 클래스는 많은 필드를 포함하고 있는데, 필드는 단순한 필드(예: 정수 또는 문자열 필드), 배열 또는 하위구조일 수 있습니다.

저장 가능 클래스가 인스턴스화되고 실제 데이터 값을 포함할 때 저장 가능 클래스는 저장 가능 오브젝트가 됩니다. 각 저장 가능 오브젝트는 고유한 Portal 오브젝트 ID 또는 POID로 식별됩니다. POID에는 데이터베이스 번호, 저장 가능 클래스의 이름, 저장 가능 오브젝트의 인스턴스 번호 및 오브젝트 개정 번호가 있습니다.

그림 4에서 저장 가능 클래스와 저장 가능 오브젝트 간의 차이점을 볼 수 있습니다.

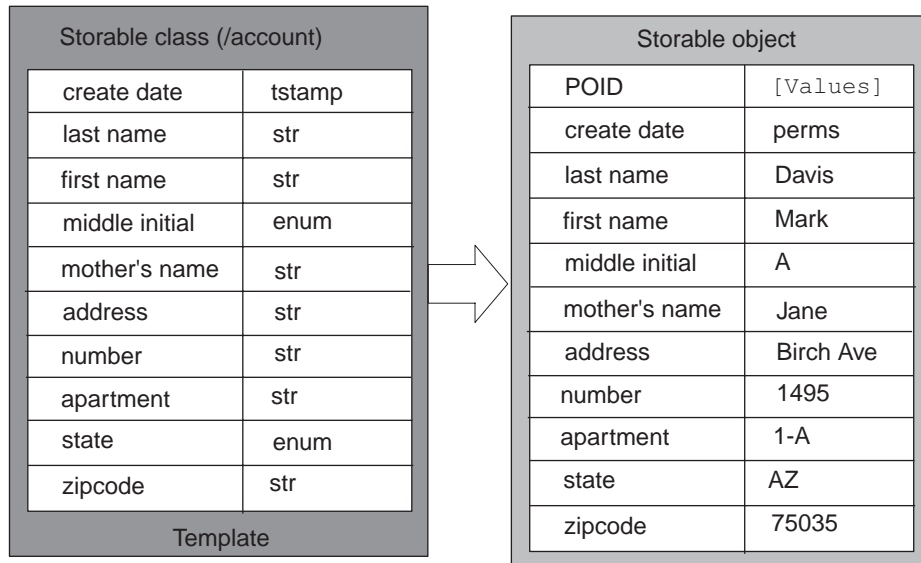


그림 4. Infranet 저장 가능 클래스 및 저장 가능 오브젝트

저장 가능 클래스는 클래스에 대해 상속되고 확장되는 기능을 정의할 수 있습니다. 예를 들어, /account/email 저장 가능 클래스는 account 클래스의 모든 정보와 함께 특히 email 확장 클래스에 적용되는 추가 정보를 포함합니다. 그러므로 /account/email 저장 가능 클래스는 그림 5에 표시된 것처럼 /account의 서브클래스가 됩니다.

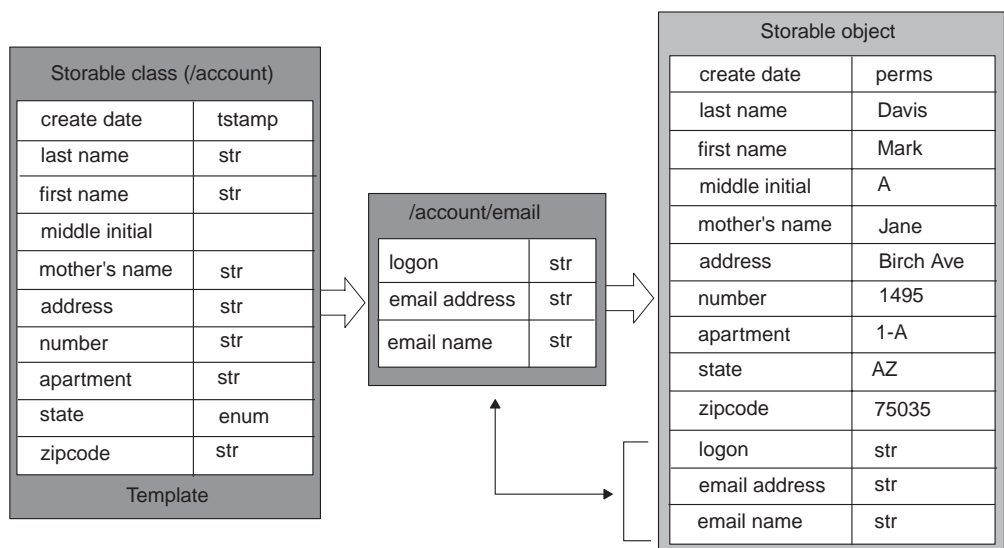


그림 5. /account 저장 가능 클래스 확장

저장 가능 오브젝트는 Infranet 응용프로그램, 스크립트 및 도구 또는 모든 사용자 정의 프로그램 및 프로세스를 사용하여 조작됩니다. 유형과 상관없이 모든 클라이언트 프로그램은 PCM API 및 프로그래밍 라이브러리를 사용하여 저장 가능 오브젝트를 조작합니다. 저장 가능 오브젝트는 opcode에 의해 조작되는데, opcode는 저장 가능 오브젝트를 조작하는 필드의 목록이 있는 루틴입니다.

필드 및 Flist

필드는 Infranet에서 가장 간단한 데이터 값입니다. 시스템의 각 필드 이름은 고유 ID, 이름, 유형 및 정의를 갖습니다. 필드 이름은 여러 가지 많은 클래스 및 opcode 정의에서 공유되고 사용됩니다.

시스템에 새 필드를 작성하는 데 사용할 수 있는 필드 유형의 기본 세트가 있습니다. 표 5에서는 필드 유형을 나열합니다. 처음 6개 유형은 C 같은 프로그래밍 언어에 있는 데이터 유형에 대응합니다. 나머지는 더 복잡한 데이터를 보유하며 C 구조를 값으로 가리킬 수 있습니다. 배열과 하위구조는 다른 필드 목록에 대한 포인터를 보유합니다.

표 5. Infranet 필드 및 데이터 유형

필드 유형	데이터 유형
PIN_FLDT_INT	부호화 정수
PIN_FLDT_UINT	비부호화 정수
PIN_FLDT_ENUM	열거된 정수
PIN_FLDT_NUM	부동 소수점 숫자
PIN_FLDT_TSTAMP	시간 소인
PIN_FLDT_STR	문자열
PIN_FLDT_BINSTR	2진 문자열
PIN_FLDT_BUF	임의 크기의 데이터 버퍼
PIN_FLDT_POID	Portal 오브젝트 ID
PIN_FLDT_ARRAY	배열
PIN_FLDT_SUBSTRUCT	하위구조

필드 목록(Flist)은 Infranet 프로그래밍 API에서 사용되는 기본 데이터 구조입니다. Flist는 데이터 필드와 값의 쌍 및 일부 경우에는 다른 Flist를 보유하는 컨테이너입니다. Flist는 메모리에 맞지 않는 부동 소수점 계산, 버퍼 또는 큰 데이터 부분을 표시할 수 있습니다. Flist는 저장 가능 오브젝트와 해당 저장 가능 오브젝트를 조작하는 루틴 또는 프로그램 사이에서 정보를 전달합니다.

저장 가능 오브젝트(예를 들어, /account 저장 가능 클래스에 있는)는 저장 가능 클래스 스펙을 사용하는 Flist(또는 Flist의 파트)를 구성합니다. Flist는 각각 고유한 속성, 권한 및 데이터 값을 갖는 필드의 목록입니다. 이들 필드는 그림 6에 표시된 것처럼 함께 저장 가능 오브젝트의 기능을 정의합니다.

Flist는 복수 저장 가능 오브젝트를 포함할 수 있습니다. Flist 구조는 응용프로그램에서 올바른 저장 가능 오브젝트로 정보가 전달되도록 합니다.

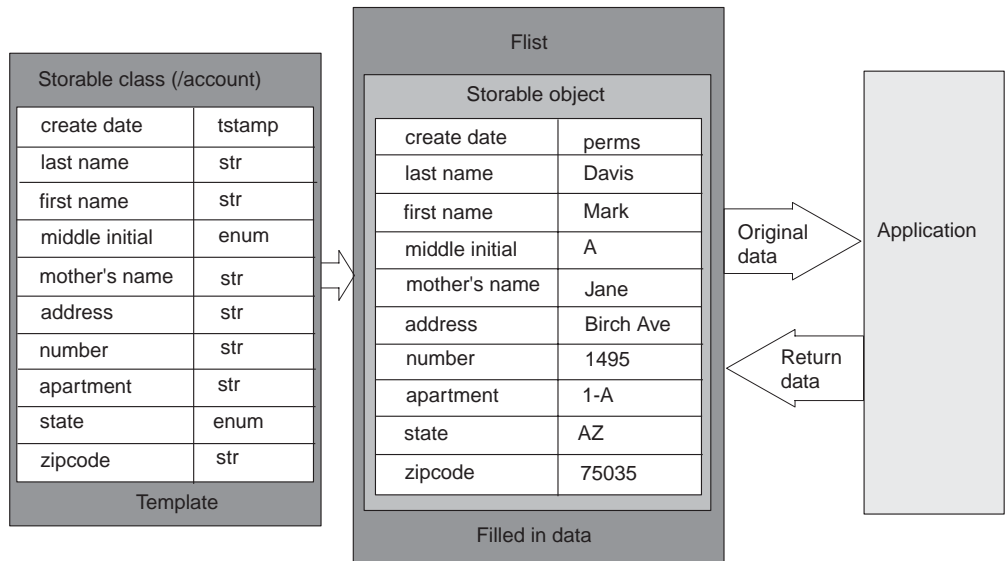


그림 6. 저장 가능 오브젝트 및 Flist

Opcode

응용프로그램은 Infranet 시스템 opcode를 사용하여 저장 가능 오브젝트 및 그 안에 있는 필드에 대한 조작을 수행합니다. 여러 세트의 opcode가 있는데, 이들은 Base, Customer 기능 모듈(FM), Activity FM, Billing FM, Terminal FM 및 Email FM의 기능 카테고리 그룹화됩니다.

오브젝트에 대한 기본 조작은 작성, 삭제, 쓰기, 읽기 및 검색을 포함합니다. 다른 모든 opcode는 로깅 활동, 제품 구매에 대한 계정 청구, 신용 카드 정보 확인, 이름 및 주소 변경, 암호 확인 또는 회계 정보 기록 같은 비즈니스 레벨(상위 레벨) 시멘틱을 구현합니다. 이들 상위 레벨 opcode는 기능 모듈에서 구현되는데, 여기서 기본 opcode 코드가 SM(Storage Manager)에 의해 직접 구현됩니다. 상위 레벨 opcode는 Communication Manager의 기능 모듈 루틴에 의해 다양한 기본 opcode 번호로 변환된 후 Storage Manager로 전달됩니다.

모든 시스템 opcode는 연관된 입력 및 출력 Flist를 갖습니다. 클라이언트 응용프로그램이 관심있는 이벤트를 판별하고, 적당한 opcode와 대응하는 Flist와 함께 Infranet 시스템을 호출하며, 리턴 Flist와 오류 버퍼를 처리합니다.

메타 데이터 구동 커넥터

커넥터는 메타 데이터 구동 커넥터입니다. 이것은 Business Object의 메타 데이터가 커넥터의 작동을 구동함을 의미합니다. 메타 데이터는 Business Object에 저장되며 커넥터가 응용프로그램과 상호작용하는 것을 돕는 응용프로그램에 관한 데이터입니다. 메타 데이터 구동 커넥터는 커넥터에 하드 코딩되는 지시사항이 아니라 Business Object 정의에 인코딩되는 메타 데이터를 바탕으로 커넥터가 지원하는 각 Business Object를 처리합니다.

커넥터는 속성 레벨에 있는 Infranet PIN_FIELDNAME 및 Verb 레벨에 있는 opcode 값을 사용하여 메타 데이터에 의해 구동됩니다. 커넥터가 메타 데이터에 의해 구동되기 때문에 커넥터 코드에 대한 수정없이 새로운 또는 수정된 Business Object를 처리할 수 있습니다. 그러나 커넥터는 Business Object의 다음 측면에 대해 가정합니다.

- Business Object의 구조
- 상위 및 하위 Business Object 사이의 관계
- 응용프로그램 특정 정보의 형식
- Business Object의 데이터베이스 표시

그러므로 Portal Infranet을 위한 Business Object를 작성 또는 수정할 때 사용자 수정은 커넥터가 따르도록 설계된 규칙을 준수해야 하며, 그렇지 않으면 커넥터는 새로운 또는 수정된 Business Object를 올바르게 처리할 수 없습니다. 다음 절에서는 Portal에 대한 Business Object 구현 정보를 제공합니다.

Portal Infranet 응용프로그램 특정 Business Object 구조

WebSphere Business Integration Adapters Business Object는 계층 구조입니다. 즉, 상위 Business Object가 하위 Business Object를 포함할 수 있으며, 하위 Business Object가 다시 하위 Business Object를 포함할 수 있는 방식입니다. 상위 Business Object의 속성이 하나의 하위 오브젝트를 참조할 때 카디널리티 1 컨테이너가 발생합니다. 상위 Business Object의 한 속성이 하위 Business Object의 배열을 참조할 때 카디널리티 n 컨테이너 오브젝트가 발생합니다.

커넥터는 Business Object 사이의 카디널리티 1 및 카디널리티 n 관계를 모두 지원합니다.

WebSphere Business Integration Adapters Business Object에 해당하는 Portal Infranet 오브젝트

Infranet은 다음 컨테이너 유형을 갖습니다.

- 저장 가능 클래스
- 배열

- 하위구조

모든 필수 속성 및 관계를 갖는 대응하는 저장 가능 오브젝트에 대한 Infranet Flist에 맵핑하도록 WebSphere Business Integration Adapters Portal Infranet 응용프로그램 특정 Business Object를 정의해야 합니다. Flist와 Business Object 사이의 관계는 일대일 관계입니다.

처리하는 동안 커넥터는 Business Object를 Infranet 오브젝트에 대해 대응하는 Flist와 비교하고, 구조가 일치하지 않는 경우 예외를 발생시킵니다. Flist 구조의 서브셋인 WebSphere Business Integration Adapters Business Object를 정의하는 것은 가능하지만 반대는 지원되지 않습니다.

각 Infranet 컨테이너 유형에 대해 응용프로그램 특정 Business Object가 필요한 대로 작성됩니다. 일반적으로 저장 가능 클래스가 최상위 레벨 Business Object가 됩니다. 유형 substruct의 컨테이너는 카디널리티 1 하위 Business Object가 될 수 있고, 유형 array의 컨테이너는 카디널리티 n 하위 Business Object가 될 수 있습니다. 그러나 서브컨테이너가 중요하지 않고 상위 opcode가 하위를 조작하기에 충분한 경우 하위 Business Object는 필요 없습니다.

그림 7은 WebSphere Business Integration Adapters Business Object의 구조와 Infranet Flist가 대응할 수 있는 방법을 보여줍니다. Infranet Flist에 대한 정보는 Portal Infranet 문서를 참조하십시오.

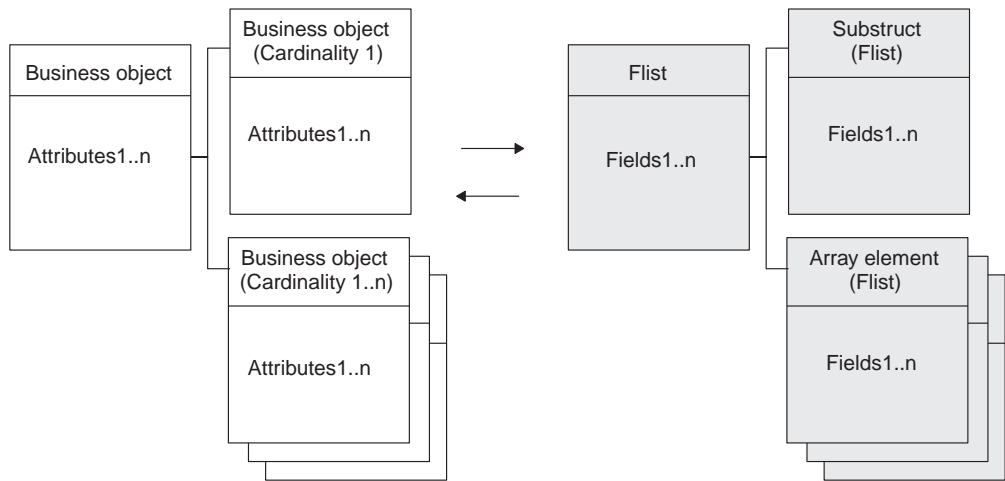


그림 7. Business Object 및 Flist의 구조

그림 8 항목은 Infranet /account 저장 가능 클래스와 Portal_Account 계층 구조 Business Object 사이의 가능한 조정의 예를 보여 줍니다. 저장 가능 클래스의 NameInfo

배열이 최상위 레벨 Business Object에서 카디널리티 n 하위 Business Object가 되고, Balances 하위구조가 카디널리티 1 하위 Business Object가 됩니다.

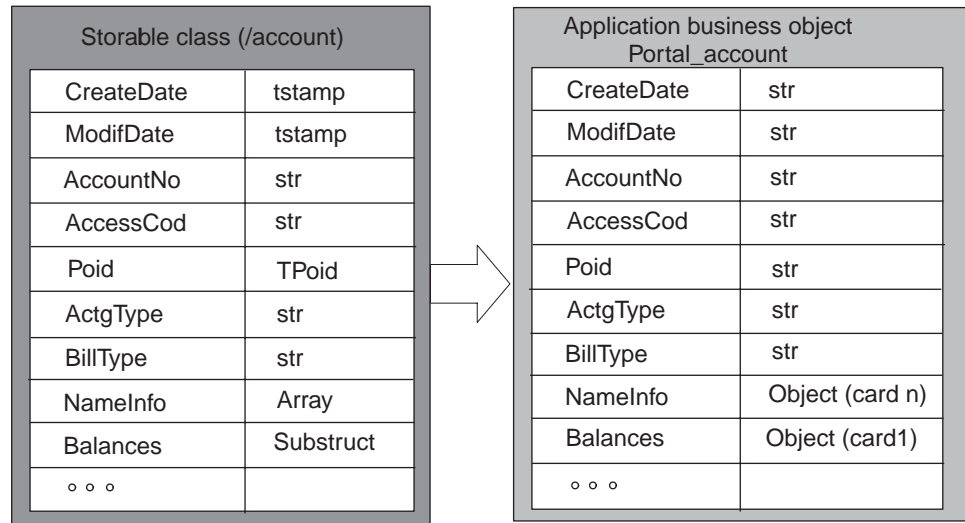


그림 8. WebSphere Business Integration Adapters Business Object에 대한 저장 가능 클래스의 조정

그림 9는 NameInfo 배열과 Portal_Contact 하위 Business Object 사이의 가능한 대응 관계를 보여줍니다. NameInfo 배열은 Phones라는 배열을 포함하는데, 이 배열이 상위 Portal_Contact Business Object인 하위 Business Object가 됩니다.

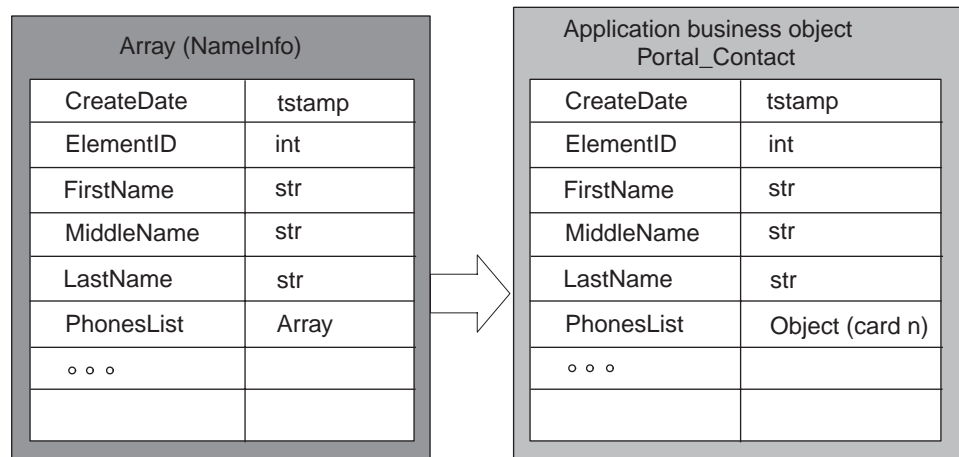


그림 9. 하위 Business Object에 대한 Flist 배열의 대응

특정 속성이 일부 Flist와 opcode의 경우에는 필요하고 다른 경우에는 필요하지 않음에 유의하십시오. 이 경우에 추가 유틸리티 응용프로그램 특정 Business Object가 Verb 매개변수로서 사용될 수 있습니다. 이 오브젝트는 어떤 지속 데이터에도 대응하지 않습

니다. 즉, Flist에 대한 일부 필수 필드만을 기술합니다. 유틸리티 Business Object에 대한 자세한 정보는 50 페이지의 『커넥터 유틸리티 Business Object』를 참조하십시오.

Business Object 속성 등록 정보

Business Object 구조는 속성에 적용되는 다양한 등록 정보를 정의합니다. 이 절에서는 커넥터가 이들 등록 정보를 해석하는 방법을 설명하고 Business Object를 수정할 때 등록 정보를 설정하는 방법을 설명합니다.

Key 등록 정보

Portal Infranet에 대한 모든 Business Object는 최소한 하나의 키 속성을 가져야 합니다. 키인 각 속성에 대해 Key 등록 정보를 True로 설정하십시오.

주: 커넥터는 하위 Business Object 또는 하위 Business Object의 배열을 표시하는 속성을 키 속성으로 지정하는 것을 지원하지 않습니다.

최상위 레벨 Business Object에 대한 키는 저장 가능 오브젝트 Portal 오브젝트 ID(POID)입니다. POID는 각 Infranet 데이터베이스 오브젝트가 작성될 때 지정되는 고유한 64비트 ID입니다. POID는 Infranet 오브젝트에 대한 고유 키입니다.

POID는 동일한 저장 가능 클래스의 복수 인스턴스화를 가능케 합니다. 예를 들어, 모든 계정 오브젝트는 고유한 POID를 갖습니다. POID는 데이터베이스 번호, 오브젝트 유형, 고유 ID 및 개정 번호의 네 가지 구성요소를 갖습니다.

하위 Business Object의 키 값

Infranet 배열은 요소 ID로 식별되고, 요소 ID가 배열에 대한 고유 키입니다. 배열이 일반적으로 하위 Business Object에 대응하기 때문에, Portal Infranet 계층 구조 Business Object에서 하위 Business Object 키는 배열 요소 ID와 상위 POID를 지정합니다.

일반적으로 2차 레벨 하위 Business Object에는 단순히 요소 ID에 대한 속성이 필요합니다. Create 또는 Update 조작 중에 모든 ElementId 및 POID 값이 Business Object에서 채워집니다.

Foreign key 등록 정보

이 등록 정보는 상위 Business Object와 관련시키기 위해 하위 Business Object에서 사용됩니다. Verb 처리 중, 하위가 개별적으로 실행되는 경우 외부 키 필드가 상위 Business Object에서 채워집니다. 상위 Business Object의 속성이 사용되는 이름은 외부 키 속성의 응용프로그램 특정 정보에서 지정됩니다.

Required 등록 정보

커넥터는 Required 등록 정보를 사용하지 않습니다.

Max Length 등록 정보

Max Length 등록 정보를 255로 설정하십시오.

기본값 등록 정보

지정되는 경우 커넥터는 속성의 기본값을 사용합니다.

Business Object 정의 지침

Portal Infranet 응용프로그램 특정 Business Object를 정의할 때 다음 지침을 준수하십시오.

- 오브젝트의 POID 속성은 오브젝트 정의의 첫 번째 속성이어야 합니다.
- 오브젝트의 다른 모든 ID 속성이 오브젝트 정의에서 뒤따라야 합니다.
- 모든 키 또는 외부 키 속성이 다음에 뒤따릅니다.
- 참조 및 포함된 오브젝트의 경우를 제외한 나머지 속성은 논리적으로 정렬되어 키 속성을 뒤따라야 합니다.
- 모든 참조된 오브젝트(카디널리티 1을 갖는 것) 또는 포함된 오브젝트(카디널리티 n을 갖는 것)가 다음에 뒤따라야 합니다.
- 마지막 속성은 ObjectEventId여야 합니다.

Business Object 응용프로그램 특정 정보

Business Object 정의에 있는 응용프로그램 특정 정보에서는 Business Object 처리 방법에 대한 지시사항을 응용프로그램에 따라 다르게 커넥터에 제공합니다. 이 메타 데이터는 Business Object의 속성 등록 정보 및 구조와 함께 사용됩니다. Portal Infranet 응용프로그램 특정 Business Object를 작성할 때 Business Object 정의의 응용프로그램 특정 정보가 커넥터가 예상하는 구문과 일치하도록 해야 합니다.

이 섹션은 Portal용 Business Object에 대해 오브젝트, 속성 및 Verb 응용프로그램 특정 정보 형식에 대한 정보를 제공합니다.

Business Object 응용프로그램 특정 정보

Business Object 레벨의 응용프로그램 특정 정보에서는 다음과 같이 Infranet 엔터티에 대해 설명합니다.

- 저장 가능한 클래스의 경우, Business Object 응용프로그램 특정 정보는 Create 또는 Update Verb에 대한 입력 Flist가 Business Object의 구조와 유사한 경우 저장 가능한 클래스 이름을 지정합니다.
- Business Object가 Flist 배열 또는 하위구조에 해당하는 하위 오브젝트인 경우, 오브젝트 응용프로그램 특정 정보에는 Infranet 필드 이름이 들어 있습니다.

예를 들어, Portal_Account Business Object는 오브젝트 레벨 응용프로그램 특정 정보에 대해 CN=/account 저장 가능 클래스를 지정하고, Portal_BillInfo 하위 Business Object는 오브젝트 레벨 응용프로그램 특정 정보에 대해 FN=PIN_FLD_BILLINFO 필드를 지정합니다.

속성 레벨 응용프로그램 특정 정보

속성 레벨에서 응용프로그램 특정 정보는 특정 opcode 실행에 사용되는 Flist 구조를 준비시키기 위해 사용됩니다. 응용프로그램 특정 정보는 이름-값 쌍 목록입니다. 이 구조는 opcode의 Flist 구조를 정의하기 위해 더 이상 필요하지 않은 유틸리티 Business Object에 의해 이전에 표시된 제한조건을 제거하게 합니다. 속성에 대한 응용프로그램 특정 정보의 형식은 다음과 같습니다.

```
FN=FIN_FLD_POID;Create=true;Update=false;Delete=true;
Retrieve=true;CreateT=;UpdateT=PIN_FL_NAMEINFO;
DeleteT=;RetrieveT=;0=false;Create0=;Update0=;
Delete0=;Retrieve0=;ParentAtt=;Alone=false
```

이전 Business Object를 새 Business Object로 변환

속성 레벨 응용프로그램 특정 정보의 위의 형식은 커넥터 버전 4.0.x에서 역방향 호환성을 위해 지원됩니다. 그러나 향후 릴리스에서, 역방향 호환성은 지원되지 않으며, PortalODA를 사용하여 이전 Business Object 정의를 새 Business Object 정의로 변환해야 합니다.

PortalODA를 사용하여 이전 Business Object 정의를 새 Business Object 정의로 변환하려면 다음을 수행하십시오.

1. 하위 Business Object를 상위 Business Object로 링크하는 하위 Business Object에서 외부 키 필드를 표시하십시오. 응용프로그램 특정 정보는 ParentAtt라는 태그를 포함합니다. 외부 키에 사용되어야 하는 상위 Business Object에서 속성 이름으로 이 태그의 값을 설정하십시오.
2. 필요한 경우, 특정 Verb 태그로 유형 "object"의 속성을 표시하십시오. CreateT, UpdateT, DeleteT 및 RetrieveT에 대해서는 표 3을 참조하십시오.

23 페이지의 표 3에서는 속성의 응용프로그램 특정 정보의 형식을 설명합니다.

표 6. 속성에 대한 응용프로그램 특정 정보

이름	설명	가능한 값	기본값
FN	Infranet에서 필드 이름을 나타내는 필드 이름		
Create	속성이 Create Verb의 Flist 파트인지 여부를 식별합니다.	true 또는 false	false
Update	속성이 Update Verb의 Flist 파트인지 여부를 식별합니다.	true 또는 false	false

표 6. 속성에 대한 응용프로그램 특정 정보 (계속)

이름	설명	가능한 값	기본값
Delete	속성이 Delete Verb의 Flist 파트인지 여부를 식별합니다.	true 또는 false	false
Retrieve	속성이 Retrieve Verb의 Flist 파트인지 여부를 식별합니다.	true 또는 false	false
CreateT	Create Verb의 Flist 준비에서 속성의 컨테이너로서 역할하는 Infranet 필드 이름		null
UpdateT	Update Verb의 Flist 준비에서 속성의 컨테이너로서 역할하는 Infranet 필드 이름		null
DeleteT	Delete Verb의 Flist 준비에서 속성의 컨테이너로서 역할하는 Infranet 필드 이름		null
RetrieveT	Retrieve Verb의 Flist 준비에서 속성의 컨테이너로서 역할하는 Infranet 필드 이름		null
0	속성이 응답 Business Object에서 갱신되어야 하는지 여부를 식별합니다.	true 또는 false	false
Create0	Create Verb 처리를 위해 응답 Business Object를 갱신하는 데 사용되는 속성의 값을 나타내는 필드를 가진 Infranet 필드 이름을 식별합니다. 예를 들어, FN=PIN_FLD_POID 및 Create0=PIN_FLD_NAMEINFO인 경우, 커넥터는 PIN_FLD_POID 필드를 찾으며, 이 필드의 값은 이 속성의 응답 Business Object에서 채워집니다. Main flist로부터 필드를 갱신해야 할 경우 값을 Main으로 설정하십시오.		Main
Update0	Update Verb 처리를 위해 응답 Business Object를 갱신하는 데 사용되는 속성의 값을 나타내는 필드를 가진 Infranet 필드 이름을 식별합니다. Main flist로부터 필드를 갱신해야 할 경우 값을 Main으로 설정하십시오.		Main
Delete0	Delete Verb 처리를 위해 응답 Business Object를 삭제하는 데 사용되는 속성의 값을 나타내는 필드를 가진 Infranet 필드 이름을 식별합니다. Main flist로부터 필드를 갱신해야 할 경우 값을 Main으로 설정하십시오.		Main
Retrieve0	Retrieve Verb 처리를 위해 응답 Business Object를 검색하는 데 사용되는 속성의 값을 나타내는 필드를 가진 Infranet 필드 이름을 식별합니다. Main flist로부터 필드를 갱신해야 할 경우 값을 Main으로 설정하십시오.		Main

표 6. 속성에 대한 응용프로그램 특정 정보 (계속)

이름	설명	가능한 값	기본값
ParentAtt	이 필드는 키 필드를 채우는 데 사용되는 상위 Business Object의 속성을 정의하기 위해 하위 Business Object에서 사용됩니다. 이 필드는 하위 Business Object에서 외부 키 필드로 표시되어야 합니다.		NULL
Alone	이 필드는 하위 Business Object가 상위 Business Object의 파트가 아니라 개별적으로 실행되어야 함을 나타내기 위해 하위 Business Object에서 사용됩니다.	true 또는 false	false

그림 10은 Portal_Account Business Object를 보여주고 Business Object 및 속성에 대한 응용프로그램 특정 정보에 대해 설명합니다.

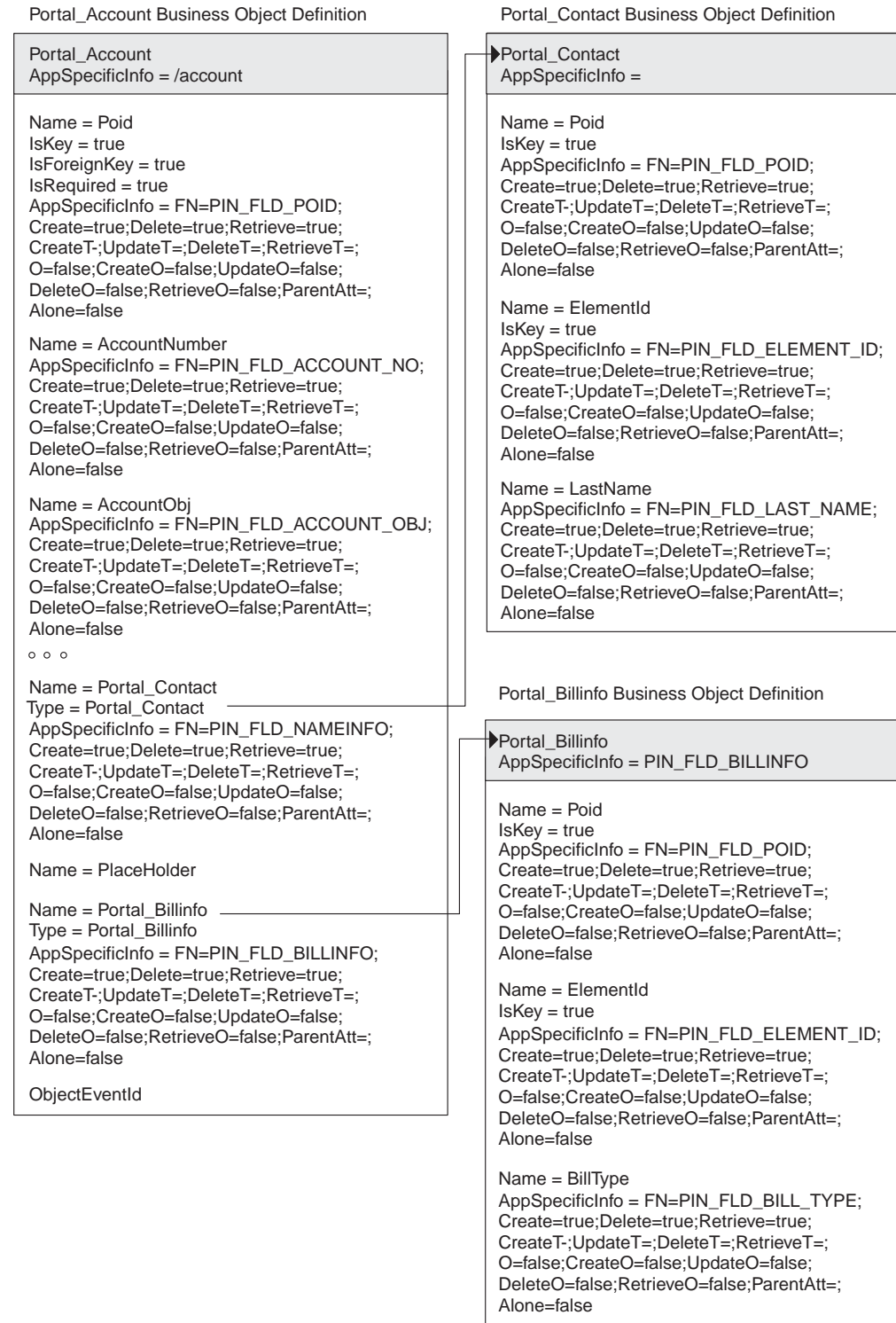


그림 10. Business Object 및 속성에 대한 응용프로그램 특정 정보

Verb 응용프로그램 특정 정보 형식

Portal Infranet에 대한 Business Object의 Verb 레벨 응용프로그램 특정 정보는 Business Object 및 Verb가 수행할 조치에 대한 고유한 Infranet opcode를 지정해야 합니다.

Opcode는 Infranet 응용프로그램과 해당 데이터베이스 사이에서 데이터를 전달하고 저장 가능 오브젝트에 대한 조사를 수행합니다. Infranet 오브젝트에 대한 각 조치는 특정 opcode를 갖습니다.

Opcode는 Flist의 양식으로 Infranet 저장 가능 오브젝트를 전달하는데, Flist는 필드 이름과 값 쌍의 목록입니다. 각 opcode는 특정 입력 및 출력 Flist를 갖습니다. 커넥터는 Business Object 요청을 opcode에 필요한 입력 Flist로 변환해야 합니다. Business Object 요청을 수행할 때 커넥터는 다음 기본 단계를 따릅니다.

1. Business Object 인스턴스의 값과 Business Object 정의의 정보를 사용하여 입력 Flist를 빌드합니다.
2. 입력 Flist를 인수로 갖는 Portal Infranet opcode를 실행합니다.
3. Opcode가 출력 Flist를 리턴하고, 커넥터가 출력 Flist의 정보로 Business Object를 갱신합니다.

Verb 응용프로그램 특정 정보와, 일부의 경우 추가 유틸리티 Business Object를 사용하면 커넥터가 각 opcode에 대해 적당한 Flist를 생성할 수 있습니다. 유틸리티 Business Object는 커넥터가 올바른 입력 Flist를 빌드하거나 출력 Flist로 Business Object를 적절하게 갱신할 수 있게 합니다. 유틸리티 Business Object는 최상위 레벨 Business Object에 대한 Business Object 정의의 파트로서 제공됩니다. 속성 레벨 응용프로그램 특정 정보의 새 정의는 유틸리티 Business Object를 중복시킵니다. 유틸리티 Business Object의 기능은 역방향 호환성을 위해 커넥터 버전 4.0.x에서 지원 중이지만 향후 릴리스에서 제거됩니다.

Verb 응용프로그램 특정 정보의 구문

Verb 응용프로그램 특정 정보의 필수 구문은 다음과 같습니다.

```
<opcode>[' # '<flag>] [' ; 'transaction  
enabled] [' ; '<input_flist_model>] [' ; '<output_flist_model>
```

여기서

opcode	<p>이름의 시작에 PCM_OP_가 없는 모든 Infranet opcode이거나, opcode가 키워드 ERROR일 수 있습니다. 키워드 ERROR는 Infranet 제한조건을 표시합니다. 해당 조치가 발생하는 경우 커넥터는 오류를 발생시킵니다. 이것은 조치가 하위 레벨에서 발생할 수 없음을 표시할 수 있습니다. 예를 들어, Portal_BillInfo Business Object는 단독으로 삭제될 수 없으며, 해당 상위 오브젝트와 함께 삭제되어야 합니다.</p>
flag	<p>Opcod에 대한 선택적 인수입니다. Opcode 플래그에 대한 정보는 Portal Infranet 문서를 참조하십시오.</p>
transaction enabled	<p>Portal Infranet에 있는 opcode의 일부는 opcode 기능의 중요한 본질로 인해 자체 트랜잭션을 유지보수합니다. 이 플래그는 Portal Infranet 커넥터에 해당 정보를 제공하는 데 사용됩니다. 값 “true”는 Opcode가 자체 트랜잭션을 유지보수함을 표시합니다. Portal Infranet 커넥터가 이 등록 정보에 대해 사용하는 기본값은 “false”입니다.</p>
input_flist_model	<p>입력 Flist에 대한 유틸리티 Business Object의 이름 및 선택적으로 일부 매개변수이거나 키워드입니다. 구문은 <input_flist_model>[#<parameter>]입니다. 가능한 키워드는 NotNull, NORMAL 또는 OnlyPoid입니다. 이들 키워드는 다음과 같이 정의됩니다.</p> <ul style="list-style-type: none">• NORMAL은 현재 Business Object가 모델임을 표시합니다. 즉, 커넥터가 현재 Business Object에서 입력 Flist로 직접 변환할 수 있습니다.• OnlyPoid는 현재 opcode에는 Business Object에서 POID만이 필요함을 표시합니다. 그런 다음 커넥터는 해당 처리를 단순화할 수 있습니다.• NotNull은 배열을 Null로 설정하여 이 오브젝트가 삭제되지 않아야 함을 표시합니다. <p>유틸리티 Business Object에 대한 정보는 50 페이지의 『커넥터 유틸리티 Business Object』를 참조하십시오.</p>
output_flist_model	<p>출력 Flist에 대한 유틸리티 Business Object의 이름 및 선택적으로 일부 매개변수이거나 키워드입니다. 구문은 <output_flist_model>[#<parameter>]입니다. 가능한 키워드는 NoRewrite 및 Flat입니다. 이들 키워드는 다음과 같이 정의됩니다.</p> <ul style="list-style-type: none">• NoRewrite는 opcode가 리턴하는 출력 Flist가 Business Object를 겹쳐 쓰는 데 사용되지 않아야 함을 표시합니다.• Flat은 커넥터가 최상위 레벨 Business Object에서 하위 Business Object에 대한 속성을 확보해야 함을 표시합니다. <p>유틸리티 Business Object에 대한 정보는 50 페이지의 『커넥터 유틸리티 Business Object』를 참조하십시오.</p>

opcode 응용프로그램의 규칙

하나 이상의 opcode가 Create, Update 및 Delete 조작에 대한 저장 가능 클래스 레벨에 존재할 수 있습니다. 그러나 커넥터는 각 Verb에 대해 단 하나의 opcode를 지원합니다. 그러므로 각 Verb에 대해 Business Object 및 Verb 조작에 가장 잘 맞는 opcode를 선택해야 합니다.

상위 및 하위 레벨에서 서로 다른 opcode가 필요할 수 있습니다. 하위 오브젝트에 대해 opcode가 존재할 때 Portal Infranet은 상위 opcode를 사용하는 대신 해당 opcode를 사용할 것을 권장합니다. Infranet에서 하위 구성요소를 갱신하는 데 특정 opcode가 필요한 경우, 이 하위 구성요소에 대해 새 응용프로그램 특정 Business Object를 작성하고, Verb 응용프로그램 특정 정보에서 opcode를 지정해야 합니다.

커넥터가 계층 구조 Business Object에 대한 입력 Flist를 빌드할 때, 하위 Business Object Verb가 상위와 동일한 opcode를 갖는 경우 커넥터는 하위를 상위와 동일한 Flist에 넣습니다. 그렇지 않으면 커넥터는 하위에 대해 별도의 Flist를 빌드합니다. Infranet은 Flist의 레벨을 사용하여 배열과 하위구조를 지정합니다. 커넥터가 입력 Flist 빌드를 시작할 때 레벨을 0으로 설정합니다. Business Object가 하위를 갖는 경우, 하위 오브젝트를 처리할 때 레벨이 하나씩 올라갑니다. Business Object가 계층 구조의 파트로 실행될 때보다 단독으로 실행될 때 다른 opcode를 필요로 하는 경우, 다른 Business Object는 유사한 구조로 사용되어야 합니다.

Create 조작의 경우 상위 opcode가 하위 opcode 전에 실행됩니다. Delete 조작의 경우 하위 opcode가 상위 opcode 전에 실행됩니다. Update 및 Retrieve 조작의 경우에는 필수 실행 순서가 없습니다.

Opcode READ_OBJ와 루트 오브젝트의 POID를 사용하여 모든 저장 가능 클래스를 검색할 수 있습니다.

커넥터 유틸리티 Business Object

각 Infranet opcode에는 특정 입력 Flist가 필요하며 특정 출력 Flist를 리턴합니다. 커넥터가 메타 데이터에 의해 구동되기 위해서 Business Object는 커넥터에 Business Object 인스턴스와 Verb를 적당한 Flist로 변환하기 위해 필요한 필드를 제공해야 합니다. Opcode Flist가 다르기 때문에 커넥터가 모든 입력 및 출력 Flist에 필요한 모든 정보를 제공하는 Portal Infranet에 대한 단일 Business Object를 빌드하지 못할 수도 있습니다. 대신, 응용프로그램 특정 Business Object 정의를 보충하는 특수한 유틸리티 Business Object 정의를 정의해야 할 수 있습니다.

주: 커넥터 버전 4.0.x의 속성 레벨 응용프로그램 특정 정보의 정의에는 유틸리티 Business Object가 필요하지 않습니다. 커넥터 버전 4.0.x는 속성 레벨 응용프로그램 특정 정보의 역방향 호환성을 지원하지만, 향후 릴리스의 경우, PortalODA를 사용하여 이전 Business Object 정의를 새 Business Object 정의로 변환하는 것이 필요합니다. 지시사항은 44 페이지의 『이전 Business Object를 새 Business Object로 변환』을 참조하십시오.

유틸리티 Business Object 정의는 통합 브로커를 통해 전송되는 Business Object 인스턴스가 되지 않습니다. 커넥터는 단순히 이들을 사용하여 특정 opcode에 대한 필수 입력 및 출력 Flist를 구성합니다. 응용프로그램 특정 Business Object의 설계 중에 유틸리티 Business Object 정의를 설계하고 빌드해야 하며, 모든 응용프로그램 특정

Business Object뿐 아니라 모든 유틸리티 Business Object를 지원하도록 커넥터를 정의해야 합니다. 유틸리티 Business Object의 예는 『유틸리티 Business Object 예: Create Verb』를 참조하십시오.

유틸리티 Business Object가 필요한지 여부를 판별하려면 Infranet 저장 가능 클래스 및 Verb 조작을 수행하는 데 사용되는 opcode의 입력 및 출력 Flist를 점검하십시오.

Portal Infranet 유틸리티 Business Object는 Portal에 대한 Business Object와는 형식면에서 다른 응용프로그램 특정 정보를 사용합니다. 이 형식은 다음 절에서 설명됩니다.

유틸리티 Business Object에 대한 응용프로그램 특정 정보

유틸리티 Business Object의 속성 응용프로그램 특정 정보는 Flist에 추가될 필드를 지정하고 Flist 필드에 사용할 값을 포함합니다. 유틸리티 Business Object에서 다음과 같이 단순 속성 및 컨테이너 속성에 대한 속성 응용프로그램 특정 정보를 정의해야 합니다.

- 단순 속성의 경우, Flist에 대한 필드 이름을 제공하고 필드에 대한 값을 정의하십시오. 필드는 Business Object 인스턴스의 해당 속성에서 추출되거나 응용프로그램 특정 정보에 제공하는 기본값일 수 있습니다. 이 설명에 대한 구문은 다음과 같습니다.

```
<Flist_fieldname>[:<bus_object_attributename>]:<default_value>]
```

- 배열 또는 구조 속성의 경우, 응용프로그램 특정 정보는 Flist에서 제외되어야 하는 응용프로그램 특정 Business Object의 속성 목록을 포함합니다. 속성은 콜론 분리문자로 분리됩니다.

```
[< bus_object_attributename>]([:< bus_object_attributename>])*
```

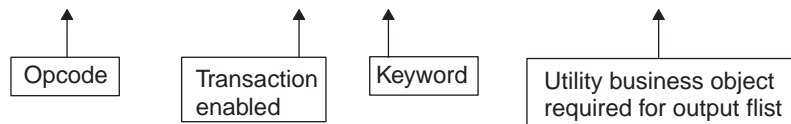
Verb 설명은 유틸리티 오브젝트에 사용되지 않습니다.

다음 섹션에서는 Create, Update, Retrieve 및 Delete Verb에 대한 응용프로그램 특정 정보의 예를 제공합니다. 그 예는 Portal_Account 계층 구조 Business Object를 사용하여 Verb 응용프로그램 특정 정보의 문법을 설명합니다.

유틸리티 Business Object 예: Create Verb

한 예로, Portal_Account 최상위 레벨 Business Object에 있는 Create Verb의 Verb 응용프로그램 특정 정보를 살펴봅시다.

```
CUST_COMMIT_CUSTOMER;true;NORMAL;Portal_CAOutput_Model
```



CUST_COMMIT_CUSTOMER는 커넥터가 새 고객 계정(/account 저장 가능 오브젝트)을 작성하는 데 사용하는 opcode입니다. Opcode는 자체 트랜잭션을 가지므로, “트랜잭션

사용 가능”이 “true”로 설정되었습니다. input_Flist_model 필드의 키워드 NORMAL 은 커넥터가 Business Object에서 입력 Flist로 직접 대응시킬 것임을 표시합니다. 즉, Business Object가 입력 Flist에 필요한 모든 필드를 제공하며, 커넥터에는 입력 Flist 를 작성하기 위해 보충 정보가 필요 없습니다.

계정 Create 조작에 대한 CUST_COMMIT_CUSTOMER opcode는 PIN_FLD_ACCOUNT_OBJ 필드에 새 고객에 대한 ID가 있는 출력 Flist를 리턴합니다. 이 ID의 값이 WebSphere Business Integration Adapters 시스템에 리턴되어야 합니다. 커넥터가 새 ID를 얻을 수 있도록 하기 위해 Business Object Designer가

Portal_C[reate]A[ccount]Output_Model 유틸리티 Business Object 정의를 작성했습니다. 응용프로그램 특정 정보에 있는 output_flist_model 필드에서는 커넥터가 opcode 가 리턴한 출력 Flist를 읽을 때 사용할 유틸리티 Business Object로서 Portal_CAOutput_Model을 지정합니다.

Portal_CAOutput_Model 유틸리티 오브젝트에는 하나의 속성 Poid가 있으며, 이것의 응용프로그램 특정 정보에서는 커넥터에 리턴 Flist에서 PIN_FLD_ACCOUNT_OBJ 값을 추출하여 새 고객의 Portal Infranet 오브젝트 ID를 확보하도록 알려줍니다. 커넥터는 이 값을 커넥터가 통합 브로커에 리턴하는 Business Object에 삽입합니다. 유틸리티 Business Object가 그림 11에 표시되어 있습니다.

Portal_Array_Model
Name = Poid IsKey = true AppSpecificInfo = PIN_FLD_ACCOUNT_OBJ:Poid Name = ObjectEventId

그림 11. Portal_CAOutput_Model 유틸리티 Business Object 정의

Create 조작에 대한 Flist 빌드: Portal_Account Business Object는 그림 12와 유사한 계층 구조 Business Object입니다.

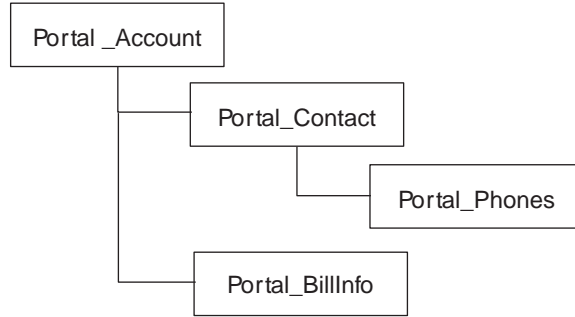


그림 12. Portal_Account 계층 구조 Business Object의 다이어그램

Create 조작 시, 커넥터는 opcode가 상위 Business Object에서 사용된 것과 같은지를 판별하기 위해 하위 Business Object의 Verb 응용프로그램 특정 정보를 점검합니다. Portal_Account Business Object의 경우, opcode는 상위 및 하위 Business Object에 대해 동일하고 커넥터는 전체 Business Object Create 조작에 대해 단일 Flist를 빌드할 수 있습니다. 그림 13은 커넥터가 Infranet에 대해 작성 호출을 수행하는 데 사용하는 단일 opcode 및 Flist를 보여줍니다.

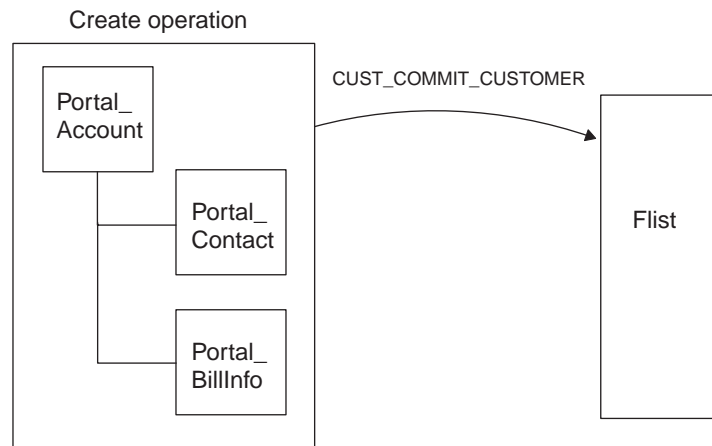


그림 13. Portal_Account 계층 구조 Business Object의 Create 조작을 위한 Flist

그러나 커넥터가 하위 Business Object에 대한 배열을 갖는 Flist를 작성해야 함에 유의하십시오. 그러므로 Verb 응용프로그램 특정 정보에는 배열이 발생해야 하는 Flist에 레벨을 표시하는 필드를 포함시켜야 합니다. 예를 들어, Portal_Contact 하위 Business Object에 대한 Create Verb 응용프로그램 특정 정보는 다음과 같습니다.

CUST_COMMIT_CUSTOMER

Portal_Phone 하위 Business Object에 대한 Create Verb 응용프로그램 특정 정보는 다음과 같습니다.

CUST_COMMIT_CUSTOMER

유틸리티 Business Object 예: Update Verb

이 예는 Portal_Account 최상위 레벨 Business Object에 있는 Update Verb에 대한 Verb 응용프로그램 특정 정보를 보여줍니다.

```
CUST_SET_STATUS;false;Portal_Array_Model#PIN_FLD_STATUSES;NoRewrite
```

CUST_SET_STATUS는 계정 오브젝트를 갱신하기 위해 필요한 opcode입니다. 이 opcode는 자체 트랜잭션을 갖지 않으므로 “트랜잭션 사용 기능”이 “false”로 설정되었습니다. 이 Verb 조작의 경우, Portal_Account Business Object가 입력 Flist에 필요한 모든 정보를 제공하지 않기 때문에 커넥터가 Business Object 인스턴스에서 Flist로 직접 대응시킬 수 없습니다. 커넥터에는 Flist를 작성하기 위한 추가 정보가 필요하기 때문에 input_flist_model 필드에서는 커넥터가 입력 Flist를 구성하는 데 사용하는 유틸리티 Business Object 정의를 지정합니다. 이 유틸리티 오브젝트가 Portal_Array_Model로 이름 지정됩니다.

output_flist_model 필드는 키워드 NoRewrite를 포함하는데, 이것은 opcode가 리턴하는 출력 Flist가 Business Object를 겹쳐쓰는 데 사용되지 않아야 함을 표시합니다.

Portal_Array_Model 유틸리티 Business Object: Portal_Array_Model은 그림 14에 표시된 계층 구조 Business Object 정의입니다. 이것은 커넥터가 Update 조작 opcode에 대한 입력 Flist를 빌드하기 위해 필요한 정보를 포함하고 있습니다. 특히 입력 Flist에는 Portal_Account Business Object 정의가 포함하지 않는 배열이 필요합니다. Portal_Array_Model 유틸리티 오브젝트는 커넥터가 배열을 작성할 수 있게 합니다.

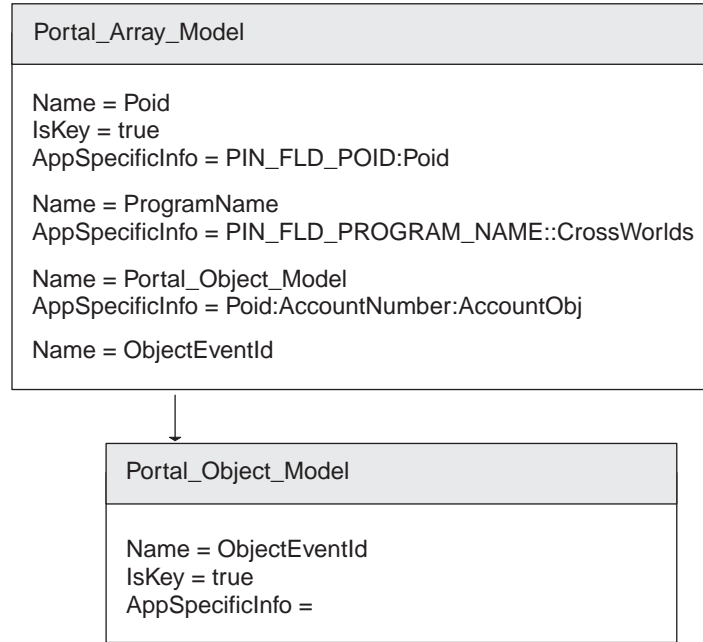


그림 14. Portal_Array_Model 유틸리티 Business Object 정의

input_Flist_model 필드에 대해 Portal_Account Update Verb 응용프로그램 특정 정보에 이 텍스트가 포함됨을 기억하십시오.

Portal_Array_Model#PIN_FLD_STATUSES

이 텍스트는 Flist를 작성하는 데 사용된 유틸리티 Business Object를 식별하고 커넥터가 입력 Flist에 넣어야 하는 배열의 이름을 지정합니다. 런타임 시 커넥터는 유틸리티 Business Object 정의와 Portal_Account 응용프로그램 특정 Business Object 정의를 둘 다 사용하여 입력 Flist를 빌드합니다.

커넥터는 다음과 같이 입력 Flist를 빌드합니다.

1. 필드 PIN_FLD_POID를 사용하여 Flist 구성을 시작하고 Business Object 인스턴스에서 POID의 값을 확보합니다.
2. PIN_FLD_PROGRAM_NAME에 대한 필드를 Flist에 추가합니다. Portal_Account Business Object 정의가 이 속성을 포함하지 않기 때문에 Business Object 인스턴스에는 이것에 대한 값이 없습니다. 그러므로 값은 응용프로그램 특정 정보에서 문자열 CrossWorlds로서 정의됩니다.
3. PIN_FLD_STATUSES로 이름 지정된 배열을 Flist에 추가합니다. Opcode PCM_OP_CUST_SET_STATUS에 대한 Flist에는 PIN_FLD_STATUSES에 대한 배열이 필요하기 때문에 Portal_Array_Model이 컨테이너 속성을 포함하여 커넥터가 Flist에 배열을 작성해야 합니다. 커넥터는 Update Verb에 대한 응용프로그램 특정 정보에 표시되는 것처럼 배열의 이름을 지정합니다.

커넥터는 현재 Business Object인 Portal_Account를 배열에 대한 모델로 사용합니다. 즉, 커넥터는 Flist 배열에 현재 Business Object에 지정되는 필드를 삽입합

니다. 필드 중 일부가 필요하지 않을 수 있으므로, 유틸리티 Business Object의 컨테이너 속성에 대한 응용프로그램 특정 정보에서 무시할 속성을 지정합니다. 컨테이너 속성 Portal_Object_Model이 다음 속성을 지정합니다.

Poid:AccountNumber:AccountObj

그러므로 배열을 구성하기 위해 커넥터는 Portal_Account에 대한 Business Object 정의를 시험하고, POID, AccountNumber 및 AccountObj 속성을 무시하고, Business Object 정의 PIN_FLD_STATUS 및 PIN_FLD_STATUS_FLAGS의 나머지 속성만을 사용하여 배열을 빌드합니다.

결국 Update Verb에 대한 입력 Flist는 다음과 유사합니다.

PIN_FLD_POID	POID	<value from bus obj instance>
PIN_FLD_PROGRAM_NAME	STR	"CrossWorlds"
PIN_FLD_STATUSES	ARRAY	
PIN_FLD_STATUS	ENUM	<value from bus obj instance>
PIN_FLD_STATUS_FLAGS	INT	<value from bus obj instance>

이 Flist는 PCM_OP_CUST_SET_STATUS opcode에 대한 입력 Flist의 필수 필드를 포함하고 있습니다.

하위 Business Object 처리: Update 조작의 경우, 계정 저장 가능 오브젝트를 갱신하고, 계정 저장 가능 오브젝트의 고객 연락처 정보를 갱신하고, 계정 저장 가능 오브젝트의 청구 정보를 갱신하기 위한 필수 Infranet opcode는 서로 다릅니다.

그러므로 고객 연락처 정보와 고객 청구 정보가 동일한 저장 가능 클래스의 일부인 경우에도 커넥터는 Portal_Account 최상위 레벨 Business Object와 Portal_Contact 및 Portal_BillInfo 하위 Business Object를 갱신하기 위해 상이한 opcode를 사용해야 합니다. 또한 커넥터는 각 opcode에 대해 별도의 입력 Flist를 생성해야 합니다. 그림 15는 Portal_Account 계층 구조 Business Object를 갱신하기 위해 필요한 Flist 세트를 보여줍니다.

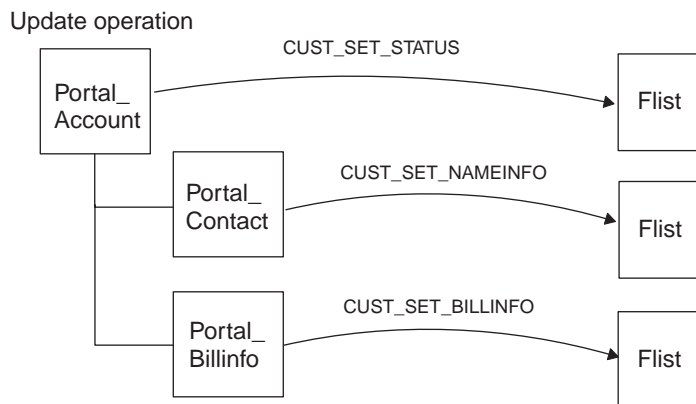


그림 15. Portal_Account 계층 구조 Business Object의 Update 조작을 위한 Flist

유틸리티 Business Object 예: Retrieve Verb

이 예는 Portal_Account 최상위 레벨 Business Object에 있는 Retrieve Verb 응용프로그램 특정 정보입니다.

```
READ_OBJ;false;Only_Poid;Flat#Portal_BillInfo
```

커넥터는 READ_OBJ opcode를 사용하여 데이터베이스에서 저장 가능 오브젝트를 읽습니다. 이 opcode는 해당 트랜잭션을 갖지 않으므로, “트랜잭션 사용 가능”이 “false”로 설정되었습니다. 일반적으로 이 opcode가 모든 Retrieve 조작에 사용될 수 있습니다.

input_Flist_model 필드는 현재 opcode에 Business Object의 POID만이 필요하도록 지정합니다. 다른 필드는 입력 Flist에 필요하지 않습니다.

Opcode는 오브젝트의 POID를 리턴합니다. 모든 배열 요소를 포함하여 오브젝트의 다른 모든 필드는 POID 후의 리턴 Flist에 추가됩니다. output_Flist_model 필드는 매개변수 Portal_BillInfo를 갖는 키워드 Flat을 포함합니다. 이것은 커넥터가 Portal_BillInfo 하위 Business Object를 빌드하기 위해 필요한 정보가 배열이 아닌 일반 Flist에 들어 있음을 표시합니다.

유틸리티 Business Object 예: Delete Verb

마지막 예는 Portal_Account 최상위 레벨 Business Object에 있는 Delete Verb 응용프로그램 특정 정보입니다.

```
CUST_DELETE_ACCT;false;Only_Poid;NoRewrite
```

커넥터는 CUST_DELETE_ACCT opcode를 사용하여 데이터베이스에서 저장 가능 오브젝트를 삭제합니다. 이 opcode는 자체 트랜잭션을 갖지 않으므로 “트랜잭션 사용 가능”이 “false”로 설정되었습니다. input_Flist_model 필드는 현재 opcode에 Business Object의 POID만이 필요하도록 지정합니다. 다른 필드는 입력 Flist에 필요하지 않습니다. output_Flist_model 필드는 키워드 NoRewrite를 포함하는데, 이것은 opcode가 리턴하는 출력 Flist가 Business Object를 겹쳐쓰는 데 사용되지 않아야 함을 표시합니다.

Infranet은 논리 삭제 응용프로그램임에 유의하십시오. 일부 오브젝트의 경우 Delete 조작이 상태의 변경입니다. 예를 들어, Portal_Service Business Object의 경우 Delete Verb 응용프로그램 특정 정보는 다음과 같습니다.

```
CUST_SET_STATUS;Portal_DSInput_Model#PIN_FLD_STATUSES#NotNull
```

이 텍스트는 논리 Delete 조작에 대한 opcode를 CUST_SET_STATUS로 지정하고, input_Flist_model이 Null로 설정되지 않은 PIN_FLD_STATUSES 배열을 갖는 Flist를 정의하는 유틸리티 오브젝트 Portal_D[ele]S[ervice]Input_Model임을 지정합니다.

문맥 구동 Verb 작동

커넥터가 계층 구조 Business Object와 단일 하위 Business Object(예를 들어, Portal_Contact Business Object는 상위없이 전송될 수 있음)를 모두 처리할 수 있어야 하기 때문에 특정 메타 데이터 구동 결정이 Business Object 및 Verb의 문맥에 따라 달라집니다. Verb에 따라서 전체 Business Object 계층 구조에 대해 하나의 고유한 opcode를 지정하거나 각 Business Object에 대해 하나의 opcode를 지정해야 합니다. 이런 이유로 각 Business Object의 각 레벨에서 사용되는 opcode를 지정해야 합니다.

Verb의 경우, 하위 레벨 및 상위 레벨에 대한 opcode 모두가 유사한 경우 글로벌 opcode(상위 opcode)가 적용됩니다. 그렇지 않으면 상위의 상위 opcode가 먼저 적용되고 그 뒤에 각 하위에 대한 하위 opcode가 적용됩니다.

예를 들어, 접속할 때 Portal_Contact Business Object가 개별 Business Object로서 전송되는 경우, Create Verb 응용프로그램 특정 정보에 대해 CUST_SET_NAMEINFO를 사용하십시오. 그러나 계정 Business Object로 접속할 경우 상위 opcode CUST_COMMIT_CUSTOMER를 사용하십시오. 이 opcode는 응용프로그램 특정 정보에 지정되어야 합니다. 위의 기능을 지원하려면 두 개의 Portal_Contact Business Object 사본이 반드시 작성되어야 하며 두 개의 다른 opcode로 사용되어야 합니다. 한 Business Object에는 CUST_SET_NAMEINFO로 설정된 verb ASI가 있어야 하고, 다른 Business Object에는 CUST_COMMIT_CUSTOMER로 설정된 verb ASI가 있어야 합니다.

완전한 샘플 Portal Infranet Business Object 정의

다음 구조는 Portal Infranet 어댑터의 샘플 계정 Business Object에 대한 응용프로그램 특정 정보 및 등록 정보에 대해 설명합니다.

```
[BusinessObjectDefinition]
Name = Portal_Account
Version = 1.0.0
AppSpecificInfo = CN=/account
[Attribute]
Name = Poid
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = true
AppSpecificInfo = eu
IsRequiredServerBound = false
[End]
[Attribute]
Name = AccountNumber
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
```

```

AppSpecificInfo =
    Update0=;FN=PIN_FLD_ACCOUNT_NO;Create=true;0=true;DeleteT=;
    Update=false;RetrieveT=;Alone=false;CreateT=;Retrieve=false;
    Delete0=;ParentAtt=;UpdateT=;Retrieve0=Main;Delete=false;Create0=
IsRequiredServerBound = false
[End]

[Attribute]
Name = AccountObj
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
    Update0=;FN=PIN_FLD_ACCOUNT_OBJ;Create=true;0=true;DeleteT=;Update=false;
    RetrieveT=;Alone=false;CreateT=;Retrieve=false;Delete0=;ParentAtt=;
    UpdateT=;Retrieve0=Main;Delete=false;Create0=
IsRequiredServerBound = false
[End]

[Attribute]
Name = Status
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
    Update0=;FN=PIN_FLD_STATUS;Create=true;0=true;DeleteT=;Update=true;
    RetrieveT=;Alone=false;CreateT=;Delete0=;Retrieve=false;ParentAtt=;
    Retrieve0=Main;UpdateT=PIN_FLD_STATUSES;Create0=;Delete=false
IsRequiredServerBound = false
[End]

[Attribute]
Name = StatusReason
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
    Update0=;FN=PIN_FLD_STATUS_FLAGS;Create=true;0=true;DeleteT=;
    Update=true;RetrieveT=;Alone=false;CreateT=;Delete0=;Retrieve=false;
    ParentAtt=;Retrieve0=Main;UpdateT=PIN_FLD_STATUSES;Create0=;Delete=false
IsRequiredServerBound = false
[End]

[Attribute]
Name = Portal_Contact
Type = Portal_Contact
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = n
MaxLength = 0
IsKey = false

```

```
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
    Update0=Main;FN=PIN_FLD_NAMEINFO;Create=true;O=true;Update=true;
    Alone=false;Retrieve=false;Delete0=Main;ParentAtt=;Retrieve0=Main;
    Create0=Main;Delete=false
IsRequiredServerBound = false
[End]
```

```
[Attribute]
Name = Placeholder
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
```

```
[Attribute]
Name = Portal_BillInfo
Type = Portal_BillInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = n
MaxLength = 0
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
    Update0=Main;FN=PIN_FLD_BILLINFO;Create=true;O=true;Update=true;
    Alone=false;Retrieve=false;Delete0=Main;ParentAtt=;Retrieve0=Main;
    Create0=Main;Delete=false
IsRequiredServerBound = false
[End]
```

```
[Attribute]
Name = ProgramName
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
    Update0=Main;FN=PIN_FLD_PROGRAM_NAME;Create=false;O=true;DeleteT=;
    Update=true;RetrieveT=;Alone=false;CreateT=;Delete0=Main;Retrieve=false;
    ParentAtt=;PAttName=;Retrieve0=Main;UpdateT=;Create0=;Delete=true
DefaultValue = CrossWorlds
IsRequiredServerBound = false
[End]
```

```
[Attribute]
Name = ObjectEventId
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
```



```

IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
AppSpecificInfo =
    IFM=;OpCode=CUST_COMMIT_CUSTOMER;OFP=;TFlag=true;
    IFP=;IF=NORMAL;Flag=0;OF=NORMAL
[End]

[Verb]
Name = Delete
AppSpecificInfo =
    IFM=;OpCode=CUST_DELETE_ACCT;OFP=;TFlag=false;IFP=;
    IF=NORMAL;Flag=0;OF=NORMAL
[End]

[Verb]
Name = Retrieve
AppSpecificInfo =
    IFM=;OpCode=READ_OBJ;OFP=Portal_BillInfo;TFlag=false;IFP=;
    IF=NORMAL;Flag=0;OF=NORMAL
[End]

[Verb]
Name = Update
AppSpecificInfo =
    IFM=;OpCode=CUST_SET_STATUS;OFP=;TFlag=false;
    IFP=;IF=NORMAL;Flag=0;OF=NORMAL
[End]

[End]

```

제 4 장 PortalODA를 사용한 Business Object 정의 생성

이 장에서는 커넥터의 Business Object 정의를 생성하는 ODA(Object Discovery Agent)인 PortalODA에 대해 설명합니다. PortalODA는 Portal Infranet API를 사용하여 Portal Infranet 저장 가능 클래스에 대한 정보를 얻습니다. 그런 다음 새 Business Object 정의를 빌드하기 위해 이 정보를 사용합니다. 또한 PortalODA는 커넥터에서 지원하는 Business Object 정의로 기존 Business Object 정의를 변환할 수 있게 합니다.

다음 주제에 대해 다룹니다.

- 『설치 및 사용법』
- 『PortalODA 설치』
- 65 페이지의 『다중 시스템에서 PortalODA 실행』
- 66 페이지의 『Business Object Designer에서 PortalODA 사용』
- 75 페이지의 『생성된 정의 내용』
- 77 페이지의 『Business Object 정의에 정보 추가』

설치 및 사용법

이 절에서는 다음 주제에 대해 설명합니다.

- 63 페이지의 『PortalODA 설치』
- 64 페이지의 『PortalODA를 사용하기 전에』
- 64 페이지의 『PortalODA 시작』
- 65 페이지의 『다중 시스템에서 PortalODA 실행』
- 65 페이지의 『오류 및 메시지 파일 이름 변경』

PortalODA 설치

PortalODA를 설치하려면 WBIA(WebSphere Business Integration Adapters) 설치 프로그램을 사용하십시오. UNIX용 시스템 설치 안내서 또는 Windows용 시스템 설치 안내서에 있는 지시사항을 따르십시오. 설치가 완료되면 제품을 설치한 시스템의 디렉토리에 다음 파일이 설치됩니다.

- ODA\Portal\PortalODA.jar
- ODA\messages\PortalODAAgent.txt
- ODA\Portal\start_PortalODA.bat(Windows 전용)
- ODA/Portal/start_PortalODA.sh(UNIX 전용)

- bin\CWODAEnv.bat(Windows 전용)
- bin/CWODAEnv.sh(UNIX 전용)

주: 달리 표시되지 않는 한, 이 문서에서는 디렉토리 경로 규칙으로 백슬래시(\)를 사용합니다. UNIX 설치의 경우 백슬래시를 슬래시(/)로 대체합니다. 모든 WBIA 제품 경로 이름은 제품이 시스템에 설치된 디렉토리와 관련이 있습니다.

PortalODA를 사용하기 전에

PortalODA를 실행하기 전에, 필수 Portal Infranet 응용프로그램의 .jar 파일을 %ProductDir%/connectors/Portal/dependencies 디렉토리에 복사해야 합니다. 다음 파일은 이 디렉토리로 복사되어야 합니다.

```
pcm.jar
pcmext.jar
```

위의 파일은 %INFRANET%\jars 폴더에 표시됩니다.

PortalODA를 설치한 후에, Business Object를 생성하거나 변환하려면 다음을 수행해야 합니다.

1. ODA를 시작하십시오.
2. Business Object Designer를 시작하십시오.
3. Business Object Designer의 6단계 프로세스에 따라 ODA를 구성 및 실행하십시오.

다음 절에서는 이 단계를 자세히 설명합니다.

PortalODA 시작

다음 중 한 가지 방법으로 PortalODA를 시작할 수 있습니다.

UNIX:

```
start_PortalODA.sh
```

Windows:

```
start_PortalODA.bat
```

Business Object Designer를 사용하여 i2ADWODA를 구성 및 실행할 수 있습니다. Business Object Designer는 각 스크립트 또는 일괄처리 파일의 AGENTNAME 변수에 지정되어 있는 이름으로 각 ODA를 찾습니다. 이 커넥터의 기본값 ODA 이름은 PortalODA입니다.

다중 시스템에서 PortalODA 실행

네트워크의 로컬 호스트 또는 원격 호스트 중 하나에서 ODA의 다중 인스턴스를 실행할 수 있습니다. 다중 인스턴스가 같은 시스템에서 실행되면, 각 인스턴스는 고유한 포트에서 실행되어야 합니다.

67 페이지의 그림 16은 실행할 ODA를 선택하는 Business Object Designer 창을 보여줍니다.

오류 및 메시지 파일 이름 변경

오류 및 추적 메시지 파일(PortalODAAgent.txt)은 제품 디렉토리 아래의 \ODA\messages\에 있습니다. 이 파일은 다음과 같은 이름 지정 규칙을 사용합니다.

AgentNameAgent.txt

스크립트 또는 일괄처리 파일의 AGENTNAME 변수에서 ODA의 이름을 변경하는 경우, 연관된 오류 및 추적 메시지 파일의 이름을 변경하려면 이 변환을 사용하십시오.

스크립트나 일괄처리 파일의 복수 인스턴스를 작성하고 표시된 각 ODA의 고유한 이름을 제공하는 경우, 이들 각각에 대한 오류 및 추적 메시지 파일의 사본을 작성하십시오. 이 변환에 따라 각 파일을 이름을 지정하십시오. 예를 들어, AGENTNAME 변수가 PortalODA1을 지정하는 경우, 연관된 메시지 파일을 PortalODA1Agent.txt로 이름 지정하십시오.

구성 프로세스에서 다음을 지정하십시오.

- PortalODA가 오류 및 추적 정보를 기록하는 파일 이름
- 추적 레벨(0에서 5까지).

표 7에서는 추적 레벨 값을 설명합니다.

표 7. 추적 레벨

추적 레벨	설명
0	모든 오류 기록
1	메소드의 모든 입력 및 기존 메시지 추적
2	ODA 등록 정보 및 해당 값 추적
3	모든 Business Object 이름 추적
4	모든 하위 스프레드의 세부사항 추적
5	• 관련된 모든 등록 정보의 ODA 초기화 값 표시 • PortalODA가 야기한 각 스프레드의 세부 상태 추적 • Business Object 정의 덤프 추적

이 값 구성 장소에 대한 정보는 67 페이지의 『초기화 등록 정보 구성』을 참조하십시오.

Business Object Designer에서 PortalODA 사용

이 절에서는 기존의 비즈니스 정의를 새 비즈니스 정의로 변환하고 새 Business Object 정의를 생성하기 위해 Business Object Designer에서 PortalODA를 사용하는 방법을 설명합니다. 이것은 Portal Infranet에서 직접 정보를 확보하여 수행됩니다. Business Object Designer 시작에 대한 정보는 *Business Object Development Guide*를 참조하십시오.

ODA를 시작한 후 Business Object Designer를 시작하려면 Business Object Designer를 구성 및 실행해야 합니다. ODA를 사용하여 Business Object Designer로 Business Object 정의를 생성하거나 변환하는 작업은 6단계로 구성됩니다. Business Object Designer에서는 각 단계를 안내하는 마법사를 제공합니다.

ODA를 시작한 후 다음을 수행하여 마법사를 시작하십시오.

1. Business Object Designer를 여십시오.
2. 파일 메뉴에서 ODA를 사용하여 새로 작성... 서브메뉴를 선택하십시오.

Business Object Designer는 마법사에서 에이전트 선택이라는 첫 번째 창을 표시합니다. 67 페이지의 그림 16은 이 창을 보여줍니다.

ODA를 선택, 구성 및 실행하려면 다음 단계를 수행하십시오.

1. 『ODA 선택』
2. 67 페이지의 『초기화 등록 정보 구성』
3. 71 페이지의 『정의 생성』 및 선택적으로 72 페이지의 『추가 정보 제공』
4. 74 페이지의 『정의 저장』

ODA 선택

그림 16은 Business Object Designer 6단계 마법사의 첫 번째 대화 상자를 보여줍니다

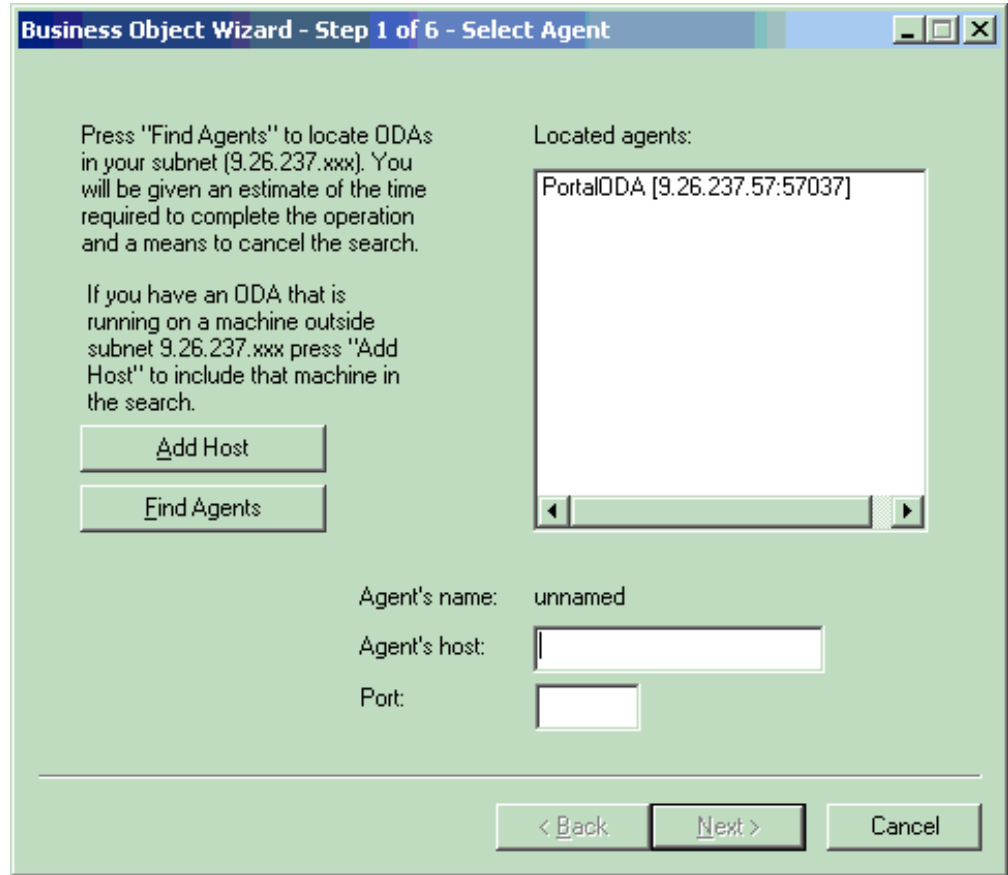


그림 16. ODA 선택

ODA를 선택하려면 다음을 수행하십시오.

1. 에이전트 찾기 단추를 눌러 찾은 에이전트 필드에 등록된 모든 ODA 또는 현재 실행 중인 ODA를 표시하십시오.

또한 호스트 이름 및 포트 번호를 사용하여 에이전트를 찾을 수 있습니다.

주: Business Object Designer가 원하는 ODA를 찾지 못하는 경우 ODA 설정을 확인하십시오.

2. 표시된 목록에서 원하는 ODA를 선택하십시오.

Business Object Designer가 에이전트 이름 필드에 선택사항을 표시합니다.

초기화 등록 정보 구성

처음으로 Business Object Designer가 PortalODA와 통신하게 되면 그림 17에 표시되어 있는 대로 초기화 등록 정보 세트를 입력하도록 프롬프트합니다. 이 등록 정보를 이름을 지정한 프로파일에 저장하여 PortalODA를 사용할 때마다 다시 입력하지 않아도 되게 합니다. ODA 프로파일 지정에 대한 정보는 *Business Object Development Guide*를 참조하십시오.

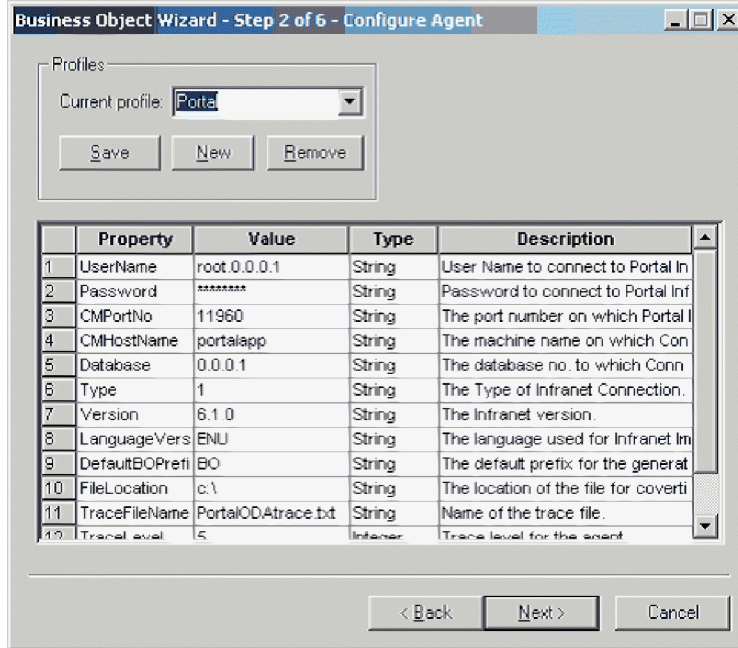


그림 17. 에이전트 초기화 등록 정보 구성

표 8에 설명되어 있는 대로 PortalODA 등록 정보를 구성하십시오.

중요: 표 8의 모든 PortalODA 등록 정보를 입력할 필요가 있습니다.

표 8. PortalODA 등록 정보

행 번호	등록 정보 이름	등록 정보 유형	설명
1	UserName	문자열	Portal Intranet 응용프로그램 로그인 이름
2	암호	문자열	Portal Intranet 응용프로그램 암호
3	CMPortNo	문자열	연결 관리자가 실행 중인 포트 번호
4	CMHostName	문자열	연결 관리자가 실행 중인 시스템의 IP 주소 또는 이름
5	데이터베이스	문자열	연결 관리자가 연결된 데이터베이스 번호
6	유형	문자열	Portal Intranet 연결 유형: 1은 UserName 및 암호의 유효성 검증 을 위한 것이며 0은 유효성 검증을 하지 않는 것입니다.
7	버전	문자열	Portal Intranet의 버전
8	LanguageVersion	문자열	예: 영어의 경우 ENU
9	DefaultBOPrefix	문자열	예: Portal_BO
10	FileLocation	문자열	Business Object 정의의 이전 버전이 있는 파일을 포함하는 절대 경 로. 예를 들어, Windows에서 경로가 C:\PortalBos인 경우, 값 C:\PortalBos\를 입력해야 하며, UNIX에서 경로가 /home/PortalBos인 경우, 값 /home/PortalBos/를 입력해야 합 니다.
11	TraceFileName	문자열	추적 파일의 이름
12	TraceLevel	정수	Business Object 이름을 고유하게 만들기 위해 추가되는 텍스트. 이 등록 정보는 필요한 경우 Business Object Designer가 사용자에게 Business Object 등록 정보를 프롬프트할 때 이후에 변경할 수 있 습니다. 자세한 정보는 72 페이지의 『추가 정보 제공』을 참조하십 시오.
13	MessageFile	문자열	메시지 파일에 대한 경로

노드 펼치기와 저장소 파일 및 저장 가능한 클래스 선택

PortalODA의 모든 초기화 등록 정보를 구성한 후에, Business Object Designer에 의해 다음 화면이 표시됩니다.

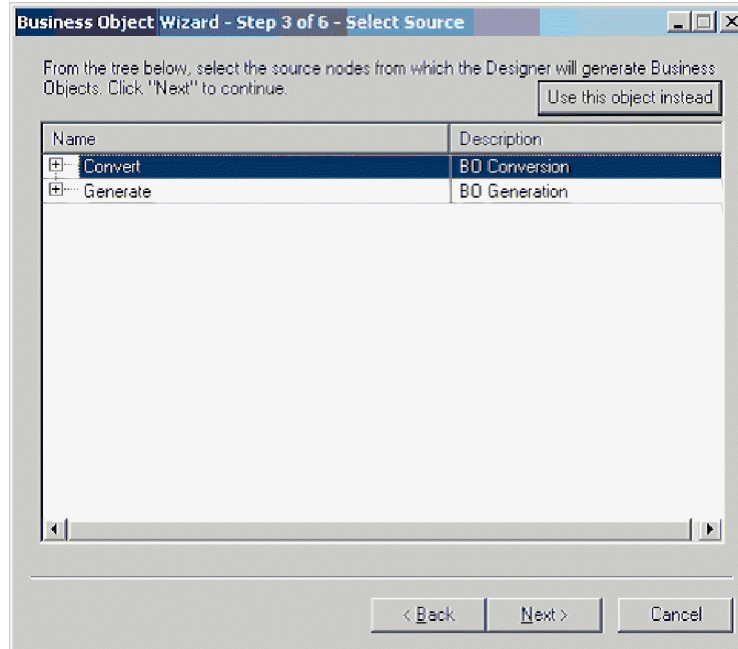


그림 18. BO 변환 및 BO 생성을 위한 두 개의 옵션을 제공하는 트리

이 화면에는 펼칠 수 있는 두 개의 옵션, 변환 및 생성이 있습니다. 이전 Business Object 정의를 새 Business Object 정의로 변환해야 하는 경우, 변환을 펼치십시오. 이것은 변환될 필요가 있는 Business Object 정의가 있는 저장소 파일을 표시합니다.

이전 Business Object 정의 변환

이전 Business Object 정의는 쉼표로 구분된 값으로 응용프로그램 특정 정보를 갖지만 새 Business Object 정의는 쉼표로 구분된 이름-값 쌍으로 응용프로그램 특정 정보를 갖습니다. 또한 이전 Business Object 정의는 메타 Business Object를 사용하여 특정 opcode의 Business Object 구조를 변환하지만 새 Business Object 정의에서는 이 기능이 Business Object의 속성 레벨에서 응용프로그램 특정 정보의 이름-값 쌍으로 바뀝니다.

변환되는 파일을 선택한 후, 다음을 누르십시오.

주: 파일을 선택하면, 해당 파일의 모든 Business Object 정의가 변환됩니다. 변환할 Business Object 정의의 서브세트를 선택할 메소드가 없습니다. 그러나 Business Object 정의의 서브세트만을 변환하려는 경우, Business Object 정의의 서브세트로 새 파일을 작성한 후, 새 파일을 변환할 수 있습니다.

새 Business Object 생성

Portal Infranet에서 정보를 얻어 새 Business Object 정의를 생성할 필요가 있는 경우, 생성을 펼치십시오. 이것은 Portal Infranet에서 모든 저장 가능한 클래스 이름을 얻어 트리를 표시합니다.

트리에서 노드로 표시되는 저장 가능한 클래스 이름을 펼칠 수 있습니다(그림 19 참조). 생성된 Business Object는 커넥터가 Business Object를 사용하기 전에 개별적으로 설정되어야 하는 일부 등록 정보를 갖습니다. 임의의 Business Object의 키 필드는 WebSphere Business Integration System Business Object의 키 필드로 표시되어야 합니다. 다른 Verb용으로 사용 중인 opcode에 따라, 속성 레벨 응용프로그램 특정 정보가 설정되어야 합니다. 예를 들어, 속성이 Create Verb opcode의 파트인 경우, 등록 정보 "Create"의 값이 상위 필드의 이름으로 설정되어야 합니다. 속성의 응용프로그램 특정 정보에서 다양한 등록 정보에 대한 자세한 정보는 44 페이지의 『속성 레벨 응용 프로그램 특정 정보』를 참조하십시오.

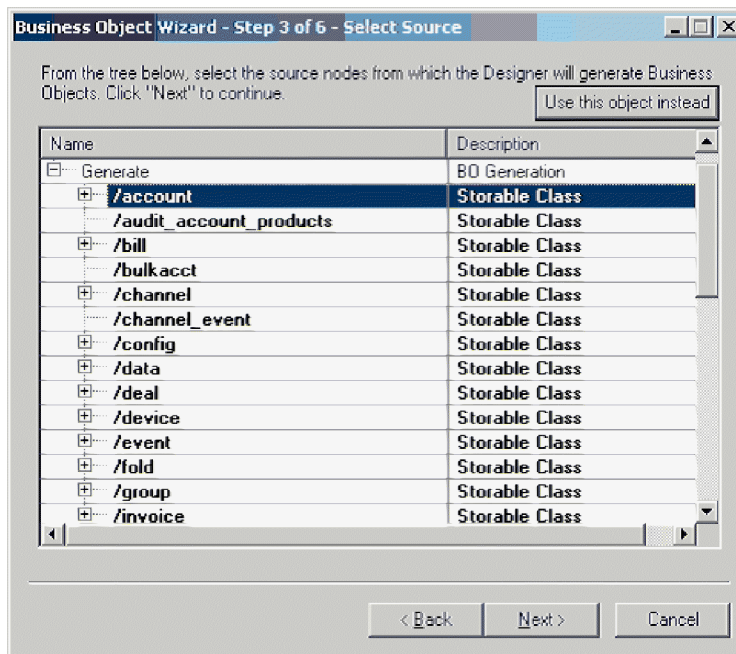


그림 19. 저장 가능한 클래스를 표시하는 화면

이 화면은 생성할 목록에서 저장 가능한 클래스를 선택하게 합니다. 클래스 이름 앞의 "+" 부호는 클래스가 하위 오브젝트를 가지고 있음을 의미합니다. 생성할 복수 클래스를 선택할 수 있습니다.

주: 하위 오브젝트가 있는 클래스를 생성하도록 선택하면, 기본적으로 하위 오브젝트가 선택되지 않습니다. 마찬가지로 하위 오브젝트를 생성하려는 경우 명시적으로 하위 오브젝트를 선택해야 합니다. 하위 오브젝트를 선택하는 중 Shift 키를 보유하여 이를 수행할 수 있습니다.

저장소 파일 및 저장 가능한 클래스의 선택사항 확인

생성된 Business Object 정의와 연관되도록 모든 저장소 파일이나 저장 가능한 클래스를 식별한 후에, Business Object Designer는 다음 확인 화면을 표시합니다(그림 20 참조).

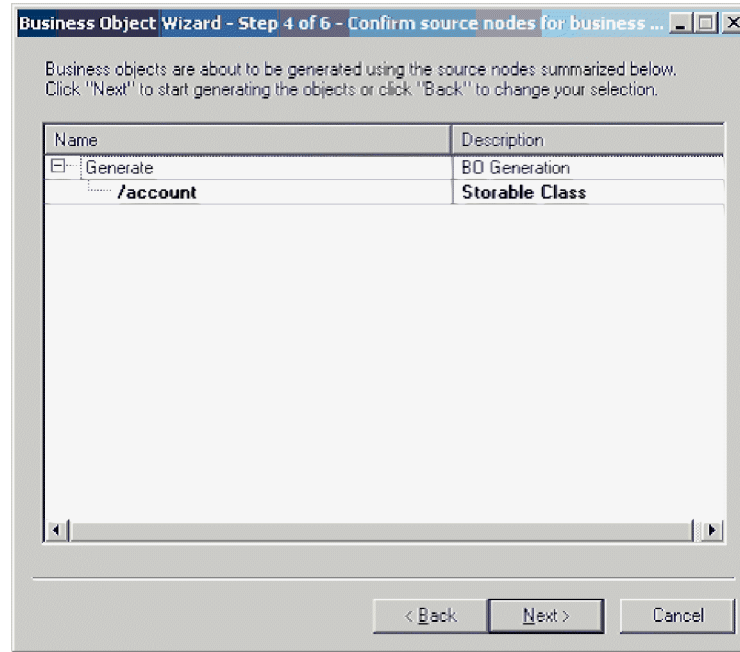


그림 20. 선택사항 확인

이 창에서는 다음 옵션을 제공합니다.

- 선택사항을 확인하려면 다음을 누르십시오.
- 선택사항이 올바르지 않은 경우 이전을 눌러 이전 창으로 돌아간 후 필요한 변경사항을 수행하십시오. 선택사항이 올바른 경우 다음을 누르십시오.

정의 생성

데이터베이스 오브젝트를 확인하고 나면 다음 대화 상자에서 Business Object Designer가 정의를 생성 중임을 알려줍니다.

그림 21은 이 대화 상자를 보여줍니다.

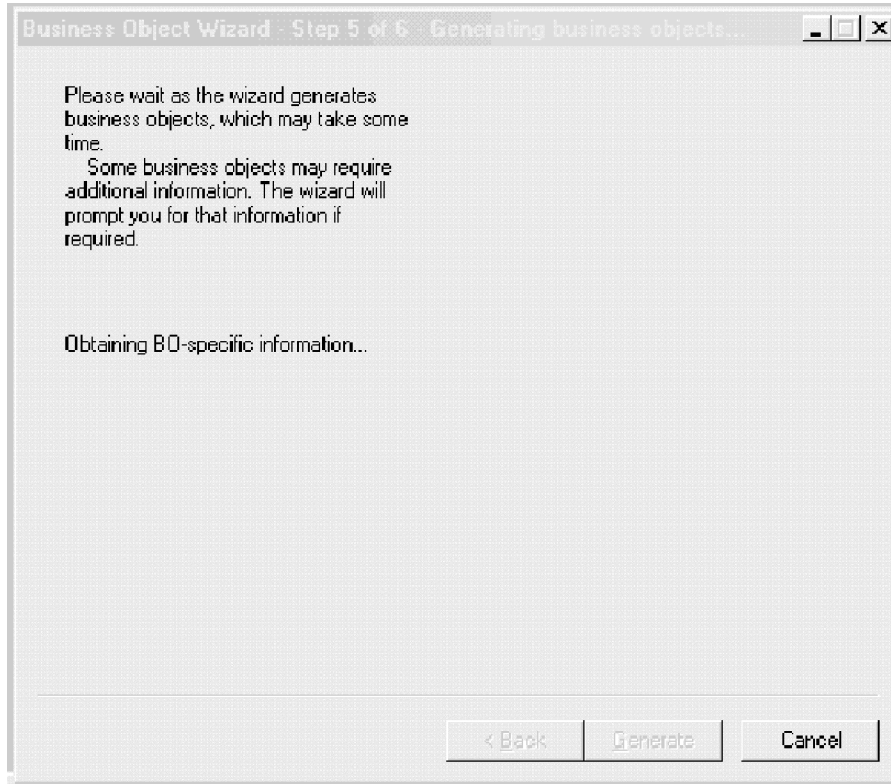


그림 21. 정의 생성

추가 정보 제공

PortalODA에 추가 정보가 필요한 경우, Business Object Designer에는 사용자에게 정보를 프롬프트하는 BO 등록 정보 창이 표시됩니다. 이것은 Business Object 생성의 경우에서만 수행됩니다. 그림 22는 이 창을 보여줍니다.

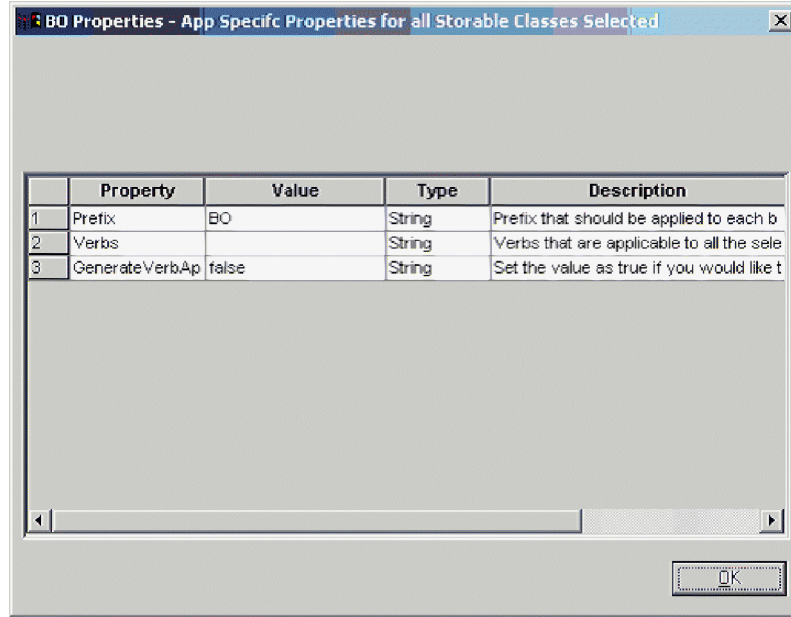


그림 22. 저장 가능한 클래스의 응용프로그램 특정 등록 정보

BO 등록 정보 창에서 다음 정보를 입력 또는 변경하십시오.

- 접두부 - Business Object 이름을 고유하게 만들기 위해 추가되는 텍스트. 에이전트 구성 창(그림 17)의 *DefaultBOPrefix* 등록 정보에 대해 입력한 값에 만족하는 경우, 여기의 값을 변경할 필요가 없습니다.
- Verb - 값 필드를 누른 후 팝업 메뉴에서 하나 이상의 Verb를 선택하십시오. 이 Verb는 Business Object에서 지원하는 Verb입니다.

주: BO 등록 정보 대화 상자의 필드에 여러 값이 있는 경우, 대화 상자가 처음 표시될 때 필드는 비어 있는 것으로 나타납니다. 필드를 눌러 해당 값의 드롭다운 목록을 표시하십시오.

- *GenerateVerbApp* - Verb 레벨에서 응용프로그램 특정 정보를 편집하게 하는 플래그.

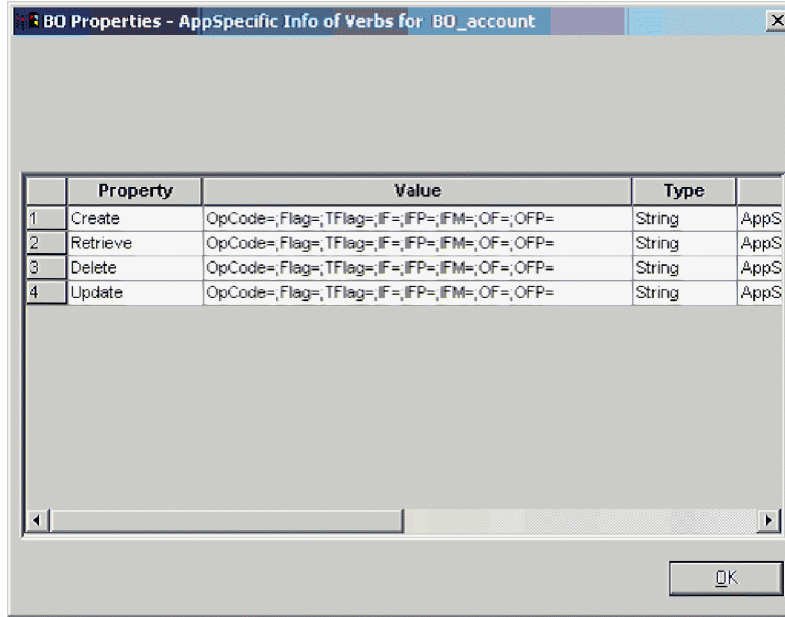


그림 23. Verb의 응용프로그램 특정 정보

Verb 레벨 응용프로그램 특정 정보 형식은 다음과 같습니다.

OpCode=;Flag=;TFlag=;IF=;IFP=;IFM=;OF=;OFP= 기술

표 9에서는 Verb 레벨 응용프로그램 특정 정보의 각 이름을 설명합니다.

표 9. Verb의 응용프로그램 특정 정보

이름	설명
Opcode	이 Verb에서 실행되어야 하는 opcode의 이름
Flag	Opcode에서 사용되어야 하는 플래그 값
TFlag	TFlag는 Opcode가 자체 트랜잭션을 유지보수하는지 여부에 따라 true 또는 false로 설정됩니다.
IF	IF(Input Flist)는 opcode의 입력 Flist를 준비하기 위해 사용되는 Business Object의 이름입니다.
IFP	IFP(Input Flist Parameter)는 입력 Flist를 준비하기 위해 사용될 수 있는 선택적 매개변수의 이름입니다.
IFM	IFM(Input Flist Mode)은 완료된 Flist 변환의 종류를 정의하는 값입니다.
OF	OF(Output Flist)는 opcode 실행의 리턴 Flist가 Business Object로 변환되는 방식을 제어하는 매개변수입니다.
OFP	OFP(Output Flist Mode)은 opcode의 출력 Flist에서 완료된 Business Object 갱신 종류를 정의하는 값입니다.

정의 저장

BO 등록 정보 대화 상자에 모든 필수 정보를 입력하고 확인을 누르고 나면 Business Object Designer가 마법사에 마지막 대화 상자를 표시합니다. 여기서 정의를 서버 또

는 파일에 저장하거나 Business Object Designer에서 편집하기 위해 정의를 열 수 있습니다. 자세한 정보 또는 추가 수정사항은 *Business Object Development Guide*를 참조하십시오.

그림 24는 이 대화 상자를 보여줍니다.

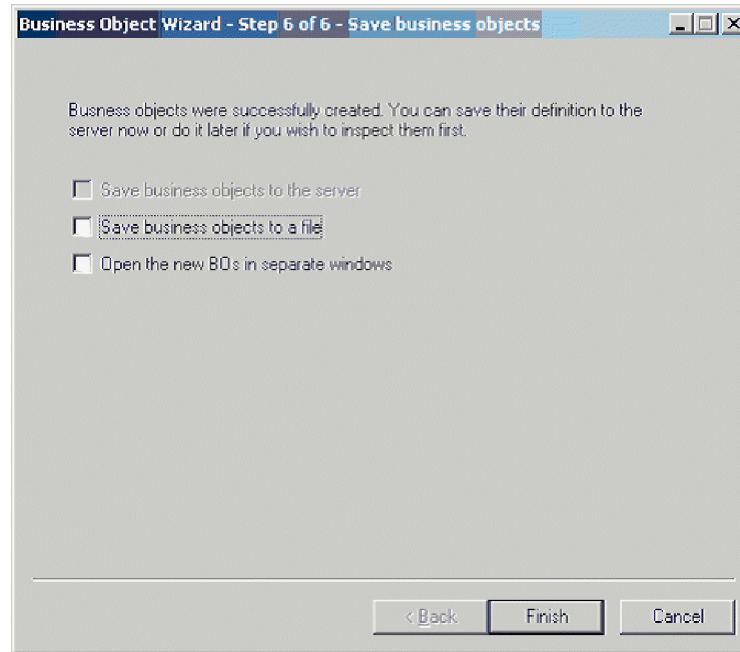


그림 24. Business Object 정의 저장

생성된 정의 내용

PortalODA가 생성하는 Business Object 정의 내용은 다음과 같습니다.

- 지정된 데이터베이스 테이블 및 보기에 있는 각 열의 속성
- BO 등록 정보 창에 지정된 Verb(그림 23)
- 응용프로그램 특정 정보:
 - Business Object 레벨
 - 각 속성
 - 각 Verb

Portal Intranet에서 정보를 가져와 Business Object를 생성할 때, 생성된 Business Object 특정 정보는 단순한 속성 전용입니다. 이 규칙에 대한 예외는 컨테이너 속성이 복수 값 링크인 경우입니다. 기타 모든 경우에, 사용자는 35 페이지의 제 3 장 『Business Object 이해』에서 설명한 대로 응용프로그램 특정 정보를 입력해야 합니다.

이 절에서는 다음 주제에 대해 설명합니다.

- 76 페이지의 『Business Object 레벨 등록 정보』
- 76 페이지의 『속성 등록 정보』
- 77 페이지의 『Verb』

Business Object 레벨 등록 정보

PortalODA에서는 Business Object 레벨에서 다음 정보를 생성합니다.

- Business Object 이름
- 버전 - 기본값은 1.0.0
- 응용프로그램 특정 정보

Business Object 레벨의 응용프로그램 특정 정보는 해당 Portal Infranet 비즈니스 구성요소의 이름을 포함합니다.

속성 등록 정보

이 절에서는 각 속성에 대해 PortalODA가 생성하는 등록 정보에 대해 설명합니다.

중요: 다음 절에서 설명하는 임의의 사용자 편집은 Business Object 생성만을 참조하며 Business Object 변환은 참조하지 않습니다.

Name 등록 정보

PortalODA는 Portal Infranet 비즈니스 구성요소의 해당 속성에서 속성 이름 값을 획득합니다.

Data type 등록 정보

속성 유형 설정 시, PortalODA는 Portal Infranet 비즈니스 구성요소에서 속성의 데이터 유형을 변환하며 표 10에 표시된 대로 해당 데이터 유형으로 변환합니다. Business Object 변환은 기존의 Business Object용이므로, 이것이 유일한 Business Object 생성의 경우입니다.

표 10. 데이터 유형의 대응

응용프로그램	WebSphere Business Integration System	길이
PIN_FLDT_INT	정수	
PIN_FLDT_ENUM	정수	
PIN_FLDT_STR	문자열	Portal Infranet에서 해당 속성의 길이
PIN_FLDT_BUF	문자열	Portal Infranet에서 해당 속성의 길이
PIN_FLDT_POID	문자열	Portal Infranet에서 해당 속성의 길이
PIN_FLDT_TSTAMP	날짜	
PIN_FLDT_ARRAY	오브젝트	
PIN_FLDT_SUBSTRUCT	오브젝트	

표 10. 데이터 유형의 대응 (계속)

PIN_FLDT_BINSTR	문자열	Portal Infranet에서 해당 속성의 길이
PIN_FLDT_DECIMAL	부동	

주: 속성의 데이터 유형이 표 10에 나와 있는 유형이 아닌 경우, PortalODA는 열을 건너뛰며, 열을 처리할 수 없다는 내용의 메시지를 표시합니다.

Cardinality 등록 정보

PortalODA는 모든 단순 속성의 카디널리티를 1로 설정하고 컨테이너 속성을 n으로 설정합니다. 사용자는 필요할 때마다 컨테이너 속성의 카디널리티를 변경해야 합니다.

MaxLength 등록 정보

PortalODA는 Portal Infranet에서 속성의 길이를 획득합니다.

IsKey 등록 정보

PortalODA는 임의 속성을 키 필드로 표시하지 않습니다. Business Object가 생성된 후에 키 필드를 수동으로 표시해야 합니다.

IsRequired 등록 정보

필드가 테이블 또는 보기에서 널이 아닌 것으로 지정되면 PortalODA는 해당 필드를 필수 속성으로 표시합니다. 그러나 레코드를 작성하는 동안 Portal Infranet 응용프로그램은 자체 ID 값을 생성하므로 IPortalODA는 키 필드를 필수로 표시하지 않습니다.

AppSpecificInfo 등록 정보

컨테이너 속성이 생성되지 않은 경우 사용자는 이 등록 정보를 편집하고, 컨테이너 속성이 생성된 경우 정확함을 보장해야 합니다.

Verb

PortalODA는 BO 등록 정보 창에 지정된 Verb를 생성합니다(74 페이지의 그림 23에서 설명한 대로). PortalODA는 각 Verb의 AppSpecificInfo 등록 정보를 작성하지만 채우지는 않습니다.

Business Object 정의에 정보 추가

Portal Infranet 저장 가능한 클래스는 Business Object에 필요한 모든 정보를 갖고 있지 않을 수 있으므로, 특히 새 Business Object 생성 시 PortalODA가 작성하는 Business Object 정의에 정보를 추가할 필요가 있을 수 있습니다.

Business Object 정의를 조사하거나 개정된 정의를 저장소로 재로드하려면, Business Object Designer를 사용하십시오.

주: 대신, ICS가 통합 브로커인 경우, repos_copy 명령을 사용하여 정의를 저장소로 로드할 수 있습니다. WebSphere MQ Integrator Broker가 통합 브로커인 경우, 시

스텝 명령을 사용하여 파일을 저장소 디렉토리로 복사할 수 있습니다.

부록 A. 커넥터의 표준 구성 등록 정보

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

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

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

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

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

신규 및 삭제된 등록 정보

다음 표준 등록 정보가 이 릴리스에 추가되었습니다.

새 등록 정보

- XMLNameSpaceFormat

삭제된 등록 정보

- RestartCount
- RHF2MessageDomain

표준 커넥터 등록 정보 구성

어댑터 커넥터에는 두 가지 유형의 구성 등록 정보가 있습니다.

- 표준 구성 등록 정보
- 커넥터 특정 구성 등록 정보

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

Connector Configurator 사용

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

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

등록 정보 값 설정 및 갱신

등록 정보 필드의 기본 길이는 255자입니다.

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

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

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

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

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

표준 등록 정보 요약

표 11은 표준 커넥터 구성 등록 정보에 대한 빠른 참조를 제공합니다. 모든 커넥터가 다음 등록 정보를 모두 사용하는 것은 아니며 표준 등록 정보 종속성은 RepositoryDirectory를 기본으로 하기 때문에 등록 정보 설정은 통합 브로커에 따라 달라질 수 있습니다.

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

표 11. 표준 구성 등록 정보 요약

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

표 11. 표준 구성 등록 정보 요약 (계속)

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

표 11. 표준 구성 등록 정보 요약 (계속)

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

표 11. 표준 구성 등록 정보 요약 (계속)

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

표준 구성 등록 정보

이 절에서는 표준 커넥터 구성 등록 정보를 나열하고 각각을 정의합니다.

AdminInQueue

관리 메시지를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMININQUEUE입니다.

AdminOutQueue

관리 메시지를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMINOUTQUEUE입니다.

AgentConnections

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

AgentTraceLevel

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

ApplicationName

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

BrokerType

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

CharacterEncoding

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

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

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

ConcurrentEventTriggeredFlows

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

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

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

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

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

ContainerManagedEvents

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

기본값은 값 없음입니다.

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

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

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

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

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

ControllerStoreAndForwardMode

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

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

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

기본값은 true입니다.

ControllerTraceLevel

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

커넥터 제어기의 추적 메시지 레벨입니다. 기본값은 0입니다.

DeliveryQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

Business Object를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/DELIVERYQUEUE입니다.

DeliveryTransport

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

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

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

WebSphere MQ 및 IDL

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

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

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

JMS

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

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

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

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

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

- CWSHaredEnv.sh 스크립트의 LDR_CNTRL 환경 변수.

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

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

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

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

DuplicateEventElimination

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

또한 이 등록 정보는 false로 설정될 수 있습니다.

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

FaultQueue

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

기본값은 CONNECTORNAME/FAULTQUEUE입니다.

JvmMaxHeapSize

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

기본값은 128m입니다.

JvmMaxNativeStackSize

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

기본값은 128k입니다.

JvmMinHeapSize

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

기본값은 1m입니다.

jms.FactoryClassName

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

기본값은 CxCommon.Messaging.jms.IBMMQSeriesFactory입니다.

jms.MessageBrokerName

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

기본값은 crossworlds.queue.manager입니다.

jms.NumConcurrentRequests

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

기본값은 10입니다.

jms.Password

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

기본값은 없습니다.

jms.UserName

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

기본값은 없습니다.

ListenerConcurrency

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

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

Locale

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

로케일 이름의 형식은 다음과 같습니다.

ll_TT.codeset

여기서

ll

2문자 언어 코드(보통 소문자)

TT

2문자 국가 또는 지역 코드(보통 대문자)

codeset

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

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

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

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

LogAtInterchangeEnd

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

MaxEventCapacity

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

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

MessageFileName

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

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

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

MonitorQueue

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

기본값은 CONNECTORNAME/MONITORQUEUE입니다.

OADAutoRestartAgent

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

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

기본값은 false입니다.

OADMaxNumRetry

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 1000입니다.

OADRetryTimeInterval

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 10입니다.

PollEndTime

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

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

PollFrequency

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

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

기본값은 10000입니다.

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

PollQuantity

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

PollStartTime

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

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

RequestQueue

Business Object를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTOR/REQUESTQUEUE입니다.

RepositoryDirectory

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

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

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

ResponseQueue

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

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

RestartRetryCount

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

기본값은 3입니다.

RestartRetryInterval

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

기본값은 1입니다.

SourceQueue

DeliveryTransport가 JMS이고 ContainerManagedEvents가 지정된 경우에만 적용 가능합니다.

JMS 이벤트 저장소를 사용하는 JMS 사용 커넥터에 대해 보증된 이벤트 전달을 지원하는 커넥터 프레임워크의 JMS 소스 대기열을 지정합니다. 자세한 정보는 86 페이지의 『ContainerManagedEvents』를 참조하십시오.

기본값은 CONNECTOR/SOURCEQUEUE입니다.

SynchronousRequestQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

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

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

기본값은 CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE입니다.

SynchronousResponseQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

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

기본값은 CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE입니다.

SynchronousRequestTimeout

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

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

기본값은 0입니다.

WireFormat

전송 시 메시지 형식입니다.

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

WsifSynchronousRequest 제한시간

WAS 통합 브로커에만 해당됩니다.

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

기본값은 0입니다.

XMLNamespaceFormat

WebSphere Message Broker 및 WAS 통합 브로커에만 해당됩니다.

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

기본값은 short입니다.

부록 B. Connector Configurator

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

Connector Configurator를 사용하여 다음을 수행할 수 있습니다.

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

주:

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

이 부록에서 다루는 주제는 다음과 같습니다.

- 97 페이지의 『Connector Configurator 개요』
- 98 페이지의 『Connector Configurator 시작』
- 99 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 102 페이지의 『새 구성 파일 작성』
- 105 페이지의 『구성 파일 등록 정보 설정』
- 113 페이지의 『국제화된 환경에서 Connector Configurator 사용』

Connector Configurator 개요

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

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

Connector Configurator를 사용하여 다음을 수행할 수 있습니다.

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

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

Connector Configurator를 실행하는 모드와, 사용하는 구성 파일 유형은 실행 중인 통합 브로커에 따라 다를 수 있습니다. 예를 들어, WMQ가 브로커인 경우, System Manager 내에서가 아니라 직접 Connector Configurator를 실행합니다(99 페이지의 『독립형 모드로 Configurator 실행』 참조).

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

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

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

그러나 커넥터 특정 등록 정보의 경우 먼저 등록 정보를 정의한 다음 해당 값을 설정해야 합니다. 특정 어댑터의 커넥터 특정 등록 정보 템플릿을 작성하여 이를 수행합니다. 시스템에 템플릿이 이미 설정되어 있으면 이 템플릿을 사용하면 됩니다. 그렇지 않은 경우, 100 페이지의 『새 템플릿 작성』의 단계에 따라 새 템플릿을 설정하십시오.

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

Connector Configurator 시작

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

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

독립형 모드로 Configurator 실행

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

이를 수행하려면 다음 단계를 따르십시오.

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

Connector Configurator를 별도로 실행하도록 선택하여 파일을 생성한 다음 System Manager에 연결하여 파일을 System Manager 프로젝트에 저장할 수 있습니다(104 페이지의 『구성 파일 완료』 참조).

System Manager에서 Configurator 실행

System Manager에서 Connector Configurator를 실행할 수 있습니다.

Connector Configurator를 실행하려면 다음을 수행하십시오.

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

기존 구성 파일을 편집하려면 다음을 수행하십시오.

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

커넥터 특정 등록 정보 템플릿 작성

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

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

- 새 템플리트를 작성하려면 100 페이지의 『새 템플리트 작성』을 참조하십시오.
- 기존 파일을 사용하려면 기존 템플리트를 수정하여 새 이름으로 저장하기만 하면 됩니다.

새 템플리트 작성

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

템플리트를 작성하려면 다음을 수행하십시오.

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

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

일반 특성 지정

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

- 일반
 - 등록 정보 유형
 - 갱신 메소드
 - 설명

- 플래그
 - 표준 플래그
- 사용자 정의 플래그
 - 플래그

등록 정보의 일반 특성을 선택한 후, **값** 탭을 누르십시오.

값 지정

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

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

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

기본값 열을 사용하면 임의의 값을 기본값으로 지정할 수 있습니다.

값 범위는 등록 정보 값 대화 상자에서 입력한 범위를 표시합니다.

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

종속성 설정

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

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

등록 정보가 종속되도록 지정하고 종속 조건을 설정하려면, 다음을 수행하십시오.

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

새 구성 파일 작성

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

브로커를 선택하려면 다음을 수행하십시오.

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

다음 작업도 수행할 수 있습니다.

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

커넥터 특정 템플릿에서 구성 파일 작성

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

1. 파일 > 새로 작성 > 커넥터 구성을 누르십시오.

2. 다음 필드가 있는 새 커넥터 대화 상자가 나타납니다.

- 이름

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

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

- 시스템 연결

ICS, WebSphere Message Brokers 또는 WAS를 누르십시오.

- 커넥터 특정 등록 정보 템플릿

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

사용하려는 템플릿을 선택하고 확인을 누르십시오.

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

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

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

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

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

기존 파일 사용

다음 중 하나 이상의 형식으로 기존 파일을 사용할 수 있습니다.

- 커넥터 정의 파일.

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

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

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

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

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

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

System Manager 프로젝트에서 커넥터 구성을 열려면 다음 단계를 따르십시오.

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

구성 파일 완료

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

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

1. 표준 등록 정보 탭에서 브로커 유형 등록 정보에 대한 값 필드를 선택하십시오. 드롭 다운 메뉴에서 ICS, WMQI 또는 WAS를 선택하십시오.
2. 표준 등록 정보 탭에 선택한 브로커와 연관된 등록 정보가 표시됩니다. 지금 파일을 저장하거나 108 페이지의 『지원되는 Business Object 정의 지정』에 설명된 대로 나머지 구성 필드를 완성할 수 있습니다.
3. 구성을 완료했을 때, 파일 > 저장 > 프로젝트에 또는 파일 > 저장 > 파일에를 누르십시오.

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

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

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

구성 파일 등록 정보 설정

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

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

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

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

ICS에서 실행되는 커넥터의 경우, 다음 등록 정보의 값도 필요합니다.

- 연관된 맵
- 자원
- 메시징(적용 가능할 경우)

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

표준 등록 정보는 다음과 같이 커넥터 특정 등록 정보와 다릅니다.

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

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

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

표준 커넥터 등록 정보 설정

표준 등록 정보의 값을 변경하려면 다음을 수행하십시오.

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

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

응용프로그램 특정 구성 등록 정보 설정

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

1. 격자의 왼쪽 맨 윗부분을 마우스 오른쪽 단추로 누르십시오. 팝업 메뉴 표시줄이 표시됩니다. 등록 정보를 추가하려면 추가를 누르십시오. 하위 등록 정보를 추가하려면 상위 행 번호를 마우스 오른쪽 단추로 누른 다음 하위 추가를 누르십시오.
2. 등록 정보 또는 하위 등록 정보의 값을 입력하십시오.
3. 등록 정보를 암호화하려면 암호화 상자를 선택하십시오.
4. 106 페이지의 『표준 커넥터 등록 정보 설정』에 설명한 대로 변경사항을 저장하거나 취소하십시오.

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

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

커넥터 등록 정보 암호화

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

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

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

갱신 메소드

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

지원되는 Business Object 정의 지정

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

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

ICS가 브로커인 경우

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

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

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

지원되는 목록에서 Business Object를 삭제하려면 다음을 수행하십시오.

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

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

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

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

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

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

대부분의 커넥터의 경우 최상의 노력만이 가능한 선택사항입니다.

트랜잭션 레벨 변경 사항을 적용하려면 서버를 다시 시작해야 합니다.

WebSphere Message Broker가 브로커인 경우

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

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

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

WAS가 브로커인 경우

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

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

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

연관된 맵(ICS에만 해당)

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

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

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

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

연관된 맵 탭은 다음 필드를 표시합니다.

- **Business Object** 이름

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

- **연관된 맵**

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

- **명시**

어떤 경우에는 연관된 맵을 명시적으로 바인드해야 할 수도 있습니다.

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

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

맵을 명시적으로 바인드하려면 다음을 수행하십시오.

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

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

자원(ICS)

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

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

메시징(ICS)

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

추적/로그 파일 값 설정

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

로깅 및 추적 값을 변경하려면 다음을 수행하십시오.

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

- 콘솔에(STDOUT):

로깅 또는 추적 메시지를 STDOUT 화면에 기록합니다.

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

- 파일에:

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

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

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

Data Handler

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

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

구성 파일 저장

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

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

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

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

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

구성 파일 변경

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

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

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

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

구성 완료

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

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

국제화된 환경에서 Connector Configurator 사용

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

Connector Configurator는 다음 위치에서 영어 이외의 문자를 지원합니다.

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

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

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

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

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

부록 C. 주의사항

IBM은 다른 국가에서는 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급하는 것이 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 “현상태대로” 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및(또는) 프로그램을 사전 통지없이 언제든지 개선 및(또는) 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

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

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

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

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

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

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

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위해 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

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

프로그래밍 인터페이스 정보

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

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

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

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

상표 및 서비스표

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

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

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

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

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

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



WebSphere Business Integration Adapter Framework V2.4.0

IBM 한글 지원에 관한 설문

IBM WebSphere Business Integration Adapters
Adapter for Portal Infranet
사용자 안내서
버전 4.2.x



FAX : (02) 3787-0123

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

성 명		직위/담당업무	
회 사 명		부 서 명	
주 소			
전화번호		팩스번호	
전자우편 주소			
사용중인 시스템	o 중대형 서버 o UNIX 서버 o PC 및 PC 서버		

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

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

IBM