IBM WebSphere Business Integration
Adapters

Adapter for Trading Partner Interchange 사용자 안내서

버전 3.4.x

IBM WebSphere Business Integration
Adapters

Adapter for Trading Partner Interchange 사용자 안내서

버전 3.4.x

주!

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

2004년 2월 20일

이 개정판은 새 개정판에 별도로 명시하지 않는 한, 커넥터 버전 3.4.x 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견은, ibmkspoe@kr.ibm.com으로 보내주십시오. 고객의 의견을 기다리고 있겠습니다.

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 2001, 2004. All rights reserved.

목차

이 책의 정보 v
이 책의 사용자 v
관련 서적 v
인쇄 규칙 vi

이 릴리스의 새로운 기능 vii
릴리스 3.4x의 새로운 기능 vii
이전 버전 vii

제 1 장 TPI Connector 개요 1
TPI Server 개요 1
TPI Connector 2

제 2 장 커넥터 설치 및 구성 7
호환성 7
전제조건 8
TPI 어댑터 및 관련 파일 설치 8
TPI Server 이벤트 로깅 기능 구성 9
커넥터 구성 9
다중 커넥터 인스턴스 작성 16
커넥터 시작 17
커넥터 시작에 대한 특수 고려사항 19
커넥터 중지 19

제 3 장 커넥터에 대한 Business Object 개발 21
Data Handler 및 문서 형식 21
Business Object 및 속성 이름 지정 규칙 23
Business Object 구조 23
맵핑 고려사항(WebSphere ICS 통합 브로커 전용) 28
Business Object Verb 처리 29
Business Object 속성 등록 정보 29
Business Object 응용프로그램 특정 정보 29

부록 A. 커넥터 표준 구성 등록 정보 31
신규 및 삭제된 등록 정보 31
표준 커넥터 등록 정보 구성 31
표준 등록 정보 요약 33
표준 구성 등록 정보 36

부록 B. Connector Configurator 49
Connector Configurator 개요 49
Connector Configurator 시작 50
System Manager에서 Configurator 실행 51
커넥터 특정 등록 정보 템플리트 작성 52

© Copyright IBM Corp. 2001, 2004

새 구성 파일 작성 54
기존 파일 사용 56
구성 파일 완료 57
구성 파일 등록 정보 설정 58
구성 파일 저장 65
구성 파일 변경 65
구성 완료 65
국제화된 환경에서 Connector Configurator 사용 66

부록 C. TPI을 통한 RNIF용 IBM WebSphere DataHandler 67
개요 67
소프트웨어 전제조건 68
설치 68
TPI RNIF Data Handler 구성 69
Business Object 구조 이해 70
Business Object 정의 작성 75

부록 D. Adapter for Trading Partner Interchange 샘플 시나리오 77
설치 전 주의사항 및 가정 77
Websphere InterChange Server 사용 78
InterChange Server에 대한 샘플 시나리오 설치 79
WebSphere MQ Integrator Broker 사용 82
요청 처리 시나리오 실행 84
폴링 시나리오 실행 85

주의사항 87
프로그래밍 인터페이스 정보 88
상표 및 서비스표 89

이 책의 정보

IBM(R) WebSphere(R) Business Integration Adapter 포트폴리오는 주요 e-business 기술, 엔터프라이즈 응용프로그램, 레거시 및 메인프레임 시스템에 대한 통합 연결성을 제공합니다. 제품 세트에는 비즈니스 프로세스 통합 구성요소의 사용자 정의, 작성 및 관리를 위한 도구와 템플리트가 포함됩니다.

이 문서에서는 IBM WebSphere Business Integration Adapter for Trading Partner Interchange에 대한 설치, 구성, Business Object 개발 및 문제점 해결에 대해 설명합니다.

이 책의 사용자

이 문서는 WebSphere 비즈니스 통합 시스템의 파트로 커넥터를 구현 중인 WebSphere 컨설턴트 및 고객용입니다. 이 문서의 정보를 사용하려면 다음 영역에 대한 지식이 있어야 합니다.

- 커넥터 개발
- Business Object 개발

관련 서적

이 제품에서 사용할 수 있는 문서의 전체 세트에는 모든 WebSphere Business Integration Adapter 설치에 공통되는 기능 및 구성요소에 대한 설명과 특정 구성요소에 관한 참조 자료가 제공됩니다.

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 경우, 다음 웹 사이트를 참조하십시오.

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

InterChange Server와 함께 어댑터를 사용하는 경우, 다음 웹 사이트를 참조하십시오.

http://www.ibm.com/websphere/integration/wicserver/infocenter
http://www.ibm.com/websphere/integration/wbicollaborations/infocenter

© Copyright IBM Corp. 2001, 2004

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

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

WebSphere Application Server에 대한 자세한 정보는 다음을 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 책은 다음의 규칙을 사용합니다.

courier 글꼴	명령어, 파일 이름, 사용자 입력 정보 또는 시스템이 화면에 인쇄하는 정보와 같은 리터럴 값을 표시합니다.
굵은체	처음으로 나타나는 새 용어를 표시합니다.
기울임	변수 이름이나 상호 참조를 표시합니다.
파란색 윤곽선	온라인 매뉴얼을 보는 경우에만 볼 수 있는 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트를 건너뛰려면 윤곽선 안을 누르십시오.
{ }	구문 행에서, 중괄호는 하나만을 선택해야 하는 옵션 세트를 묶습니다.
[ ]	구문 행에서, 대괄호는 선택적 매개변수를 묶습니다.
...	구문 행에서, 말줄임표(...)는 이전 매개변수의 반복을 표시합니다. 예를 들어, option[,...]은 쉼표로 구분되는 복수 옵션을 입력할 수 있음을 의미합니다.
< >	이름 지정 규칙에서 꺽쇠 괄호는 <server_name><*connector_name*> tmp.log에서와 같이 이름의 개별 요소를 묶어 이들을 서로 구별합니다.
/, \	이 책에서, 백슬래시(\)는 디렉토리 경로에 대한 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체합니다. 모든 WebSphere Business Integration 시스템 제품 경로 이름은 제품이 시스템에 설치된 디렉토리에 상대적입니다.
%text% 및 *$text*	퍼센트(%) 부호 내의 텍스트는 Windows text 시스템 변수 또는 사용자 변수를 표시합니다. UNIX 환경에서의 동등한 표기법은 *$text*로서, *text* UNIX 환경 변수의 값을 표시합니다.
ProductDir	제품이 설치된 디렉토리를 표시합니다.

이 릴리스의 새로운 기능

릴리스 3.4x의 새로운 기능

2004년 2월

TPI를 통한 RNIF(RosettaNet Implementation Framework) 특정 Data Handler가 제공됩니다.

샘플 구현 시나리오가 제공됩니다.

2003년 12월

Trading Partner Interchange의 어댑터가 일반 유지보수 수정사항으로 갱신되었습니다.

Trading Partner Interchange의 어댑터는 버전 3.4부터 더 이상 Microsoft Windows NT와 IBM AIX에서 지원되지 않습니다.

어댑터 설치 정보가 이 안내서에서 다른 곳으로 이동되었습니다. 설치 정보의 새 위치는 제 2 장 8 페이지의 『TPI Server 설치』를 참조하십시오.

이전 버전

다음은 이전 버전의 변경사항 및 새로운 기능에 대한 히스토리입니다.

버전 3.3.x

어댑터는 이제 통합 브로커로 WebSphere Application Server를 사용할 수 있습니다. 추가 정보는 7 페이지의 『호환성』을 참조하십시오. 이제 어댑터는 다음 플랫폼에서 실행됩니다.

- Solaris 7, 8
- AIX 5.x

버전 3.2.x

2003년 3월에 갱신되었습니다. 전체 시스템을 설명하거나 구성요소 또는 도구의 이름을 수정하기 위해 더 이상 "CrossWorlds" 이름을 사용하지 않습니다. 다른 점은 이전과 거의 동일합니다. 예를 들어 "CrossWorlds System Manager"의 경우 이제 "System Manager"이고, "CrossWorlds InterChange Server"는 이제 "WebSphere InterChange Server"입니다.

이 릴리스에 있는 커넥터의 변경사항은 다음과 같습니다.

© Copyright IBM Corp. 2001, 2004

- 어댑터가 일반 유지보수 수정사항으로 갱신되었습니다.
- TPI MCD 2.0 형식을 지원하는 새로운 Data Handler인 tpi_rnif_mcd2가 추가되었습니다. 이 Data Handler는 CwDataHandler.jar 파일(파일 버전 2.3.0)에 포함되어 있습니다.

버전 3.1.x

이 릴리스에 있는 커넥터의 변경사항은 다음과 같습니다.

- 커넥터가 WebSphere MQ Integrator Broker와 함께 작업하는 방식을 설명하기 위해 새 샘플이 추가되었습니다.
- 커넥터는 이제 Trading Partner 구성 파일을 위한 동적 갱신을 지원합니다.
- 커넥터는 이제 복수 이벤트에 대한 병렬 처리를 지원합니다. 이 기능을 구현하기 위해, 4개의 새 커넥터 특정 등록 정보가 추가되었습니다.
  - DeliverOnArrival — 한 번에(PollForEvents 모드를 사용하여) 하나씩 또는 동시에 수신 이벤트가 처리되는지 여부를 지정합니다.
  - DeliverOnArrivalThreads — 병렬 이벤트 처리에 사용할 수 있는 스레드 수를 지정합니다.
  - WaitForController — 제어기가 활성 상태로 되기까지 커넥터가 대기하는 시간을 지정합니다.
  - ControllerWaitCount — 커넥터가 비활성 제어기에 접속을 시도하는 횟수를 지정합니다.

버전 3.0.x

커넥터가 국제화되었습니다. 자세한 정보는 6 페이지의 『로케일 종속 데이터 처리』 및 31 페이지의 부록 A 『커넥터 표준 구성 등록 정보』를 참조하십시오.

TIP 문서가 사용하는 문자 인코딩을 지정하기 위해 새 커넥터 특정 등록 정보 `DataEncoding`이 추가되었습니다. 자세한 정보는 13 페이지의 『커넥터 특정 등록 정보』를 참조하십시오.

버전 2.1.x

버전 2.1.x에서 다음 변경사항이 새로 작성되었습니다. TPI용 IBM WebSphere Business Integration Adapter는 TPI용 커넥터를 포함합니다. 이 어댑터는 2개의 통합 브로커인 ICS(InterChange Server)와 WebSphere MQIntegrator를 지원합니다. 통합 브로커는 이기종 세트의 응용프로그램 통합을 수행하는 응용프로그램입니다. 이는 데이터 라우팅과 같은 서비스를 제공합니다. TPI용 IBM WebSphere Business Integration Adapter는 다음을 포함합니다.

- TPI 특정 응용프로그램 구성요소
- 샘플 Business Object(\connectors\Tpi\Samples 디렉토리에 위치)

- 다음과 같이 구성된 IBM WebSphere Adapter Framework
  - 커넥터 프레임워크
  - 개발 도구(Business Object Designer 및 Connector Configurator 포함)
  - API(ODK, JCDK 및 CDK 포함)

이 매뉴얼은 ICS 및 WebSphere MQIntegrator 통합 브로커와 함께 어댑터의 사용에 대한 정보를 제공합니다. TPI Connector의 오류 보고를 향상시키기 위한 개선사항이 수행되었습니다.

주: 커넥터의 국제화가 이루어지지 않았으므로, ISO Latin-1 데이터만 처리된다고 보장할 수 없는 경우, InterChange Server 버전 4.1.1에 대해 커넥터를 실행하지 마십시오.

버전 2.0.x

커넥터의 버전 2.0.x에서 다음 변경사항이 수행되었습니다.

버전 1.2.x

커넥터의 버전 1.2.x에서 결함을 수정하고 IBM CrossWorlds infrastructure 버전 4.0.0과의 호환성을 제공하기 위해 약간의 사항이 변경되었습니다.

버전 1.1.x

현재 IBM CrossWorlds TPI Connector의 버전 1.1.x가 TPI 버전 4.0을 지원합니다.

제 1 장 TPI Connector 개요

이 장에는 다음 섹션이 포함되어 있습니다.

- 『TPI Server 개요』
- 2 페이지의『TPI Connector』

커넥터는 *커넥터 프레임워크* 및 *응용프로그램 특정 구성요소*로 구성됩니다. 코드가 모든 커넥터에 공용인 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이의 조정자 역할을 합니다. 응용프로그램 특정 구성요소는 특정 응용프로그램이나 기술에 맞게 조정된 코드를 포함합니다(이 경우, Trading Partner Interchange). 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이의 다음 서비스를 제공합니다.

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

이 장에서는 IBM WebSphere Business Integration TPI(Trading Partner Interchange)의 커넥터 구성요소에 대해 설명합니다. 이 문서는 커넥터 프레임워크와 응용프로그램 특정 구성요소 모두에 대한 정보를 포함합니다. 이는 이들 모두를 커넥터로 참조합니다. 커넥터와 통합 브로커의 관계에 대한 자세한 정보는 *IBM WebSphere InterChange Server System Administration Guide* 또는 *IBM WebSphere Business Integration Implementation Guide for WebSphere MQ Integrator Broker*를 참조하십시오.

TPI Server 개요

이 섹션에서는 TPI Server 기능의 간단한 개요를 제공합니다. 커넥터 기능은 다음 섹션에서 설명됩니다.

TPI Server는 인터넷에서 Trading Partner 사이의 안전한 문서 교환을 가능하게 합니다. 응용프로그램은 사용자 구성 스케줄에 따라 Trading Partner 사이에 전송되는 보안 봉투에 문서를 패키지합니다. TPI Server는 XML, EDI 및 2진 데이터 형식을 지원합니다.

TPI는 Trading Partner 간에 문서를 전송하는 데 있어 FTP, SMTP 및 WebSphere MQ를 지원합니다. TPI Server는 문서를 전달하고 검색하기 위해 디렉토리의 시스템을 사용합니다. TPI Server는 각 Trading Partner에 6개의 문서 디렉토리를 유지보수합니다(인바운드에 3개, 아웃바운드에 3개). 각 디렉토리는 특정 형식(XML, EDI 또는 2진)의 수신 또는 전송 문서를 보유합니다.

© Copyright IBM Corp. 2001, 2004

문서 전송

아웃바운드 디렉토리 중 하나에 문서가 있을 때, TPI Server는 전송자 및 수신자 ID를 포함하는 요약을 먼저 작성하여 문서를 패키지합니다. 그런 다음, TPI Server는 문서와 요약을 단일 메시지로 암호화합니다. 문서가 일단 패키지되면, TPI는 메시지를 지정된 Trading Partner에 전송합니다. TPI 저장소의 Trading Partner 프로파일은 메시지를 전송하기 위해 사용하는 전송 메소드를 판별합니다.

문서 수신

TPI Server가 Trading Partner로부터 메시지를 수신할 때, 이는 메시지를 암호 해독하고 디지털 서명을 검증하며 적절한 인바운드 디렉토리에 문서를 씁니다. 메시지가 일단 압축 해제되면, TPI Server가 MDN(Message Disposition Notification)을 메시지 전송자에게 리턴합니다. MDN은 메시지가 수신되어 적절하게 압축 해제되는지 여부만을 검증합니다. Trading Partner가 문서를 완전히 처리했는지 여부는 표시하지 않습니다.

TPI Connector

TPI용 커넥터는 통합 브로커가 Business Object를 TPI Server 버전 3.0.3 이상과 교환할 수 있게 합니다.

커넥터는 Business Object 처리, 이벤트 폴링 및 이벤트 공고를 구현합니다. 커넥터의 응용프로그램 특정 구성요소는 통합 브로커에 전송하는 Business Object를 생성하고, 통합 브로커의 Business Object 요청에 응답하기도 합니다. 이는 파일 또는 커넥터 콘솔에 쓰거나 통합 브로커에 전송하는 로깅 및 추적 메시지를 생성합니다.

TPI Connector는 다음 구성요소를 사용합니다.

- WebSphere Business Integration Adapter Data Handler — TPI Server와 WebSphere Business Integration Adapter Business Object 형식이 지원하는 문서 형식 사이의 형식 변환을 수행하기 위해 커넥터가 호출합니다.
- Trading Partner 구성 파일 — 각 Trading Partner에 대한 Data Handler 정보를 저장합니다.

TPI Connector는 동일한 Java Virtual Machine을 사용하여 TPI Server와 동일한 프로세스 영역에서 실행합니다.

주: 이러한 이유로, TPI Connector를 1보다 큰 값으로 설정된 Parallel Process Degree Resource로 실행하는 것은 바람직하지 않습니다. Parallel Process Degree에 대한 자세한 정보는 *System Administration Guide*를 참조하십시오.

커넥터는 DocumentListener 인터페이스 및 InterchangeEventListener 인터페이스를 사용하여 TPI Server와 통신합니다. 이는 인바운드 TPI 문서를 처리하여 통합 브로커에 전달하는 Business Object를 작성합니다. 이는 요청 Business Object를 처리하여 TPI

Server에 전달하는 문서 스트림을 작성합니다. 그림 1은 TPI Connector 구조를 보여줍니다.

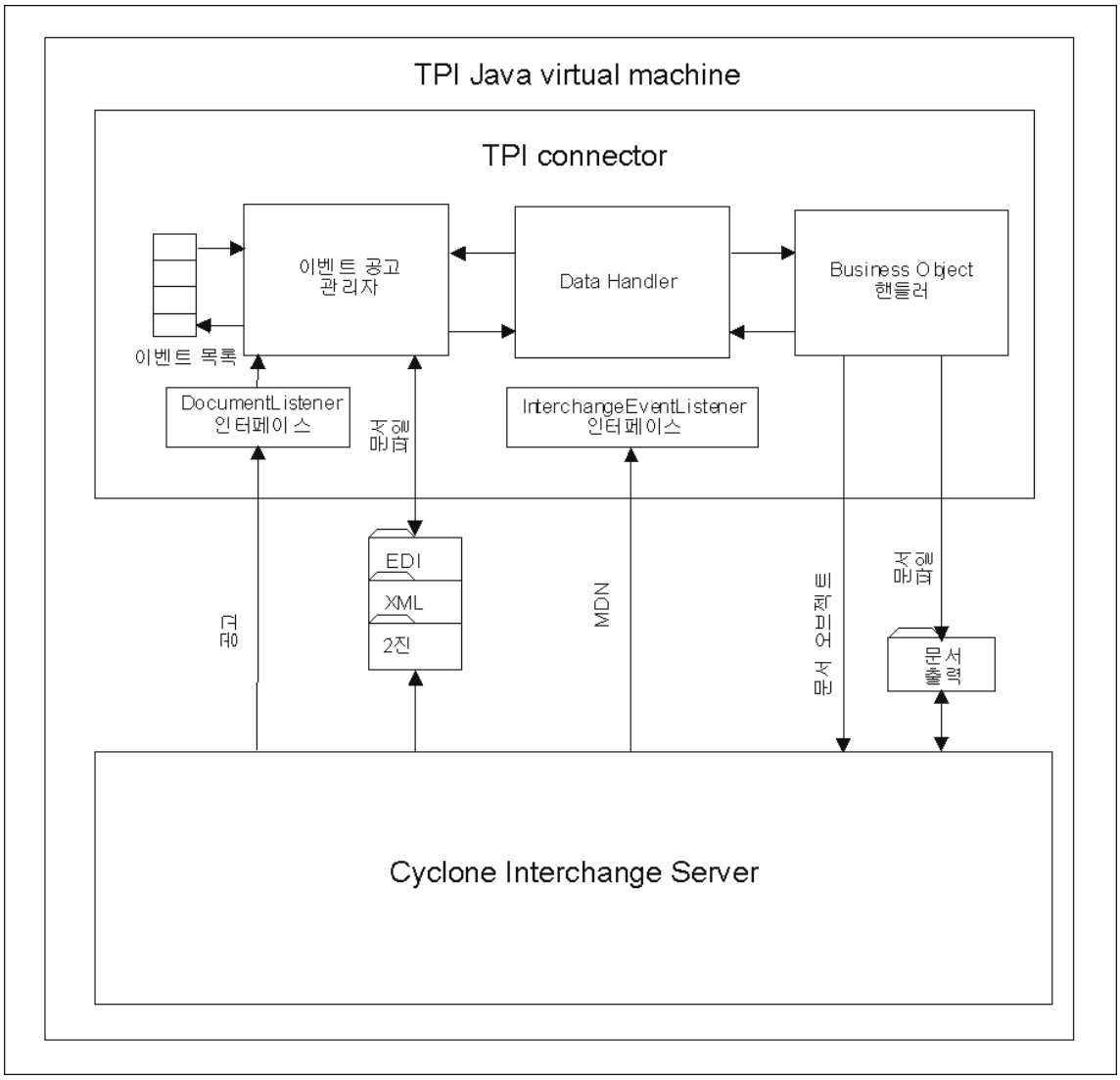


그림 1. TPI Connector 구조

초기화 및 종료

커넥터 구성 등록 정보를 로드하는 것 외에, init() 메소드는 시작 시에 다음 타스크를 수행합니다.

- Trading Partner 구성 파일을 읽고, 각 Trading Partner에 대한 Data Handler 정보와 형식 조합을 메모리에 로드합니다.
- 커넥터는 TPI Server를 시작하고 이벤트 및 요청 처리를 위해 DocmentListener 및 InterchangeEventListener 인터페이스로 등록합니다.

- 커넥터가 장애 후에 시작되고 있는 경우, init() 메소드가 EventRecovery 등록 정보에 지정된 대로 이벤트 복구 타스크를 수행합니다.

terminate() 메소드는 커넥터와 TPI Server를 시스템 종료합니다.

이벤트 공고

이벤트 공고 중 TPI Connector는 파일 시스템을 사용하여 처리되지 않은 이벤트에 대한 메타 데이터를 지속시킵니다. 이는 또한 처리된 이벤트를 해당 상태로 레이블하여 아카이브합니다.

TPI용 커넥터는 이벤트 공고를 위해 콜백 메소드인 documentArriving()을 사용합니다. 이 메소드는 DocumentListener 인터페이스를 통해 제공됩니다. TPI Server가 Trading Partner로부터 문서를 수신할 때, 이는 documentArriving() 메소드를 호출하여 인바운드 문서를 커넥터에 알립니다. 공고는 커넥터가 문서를 검색하고 처리하는 데 필요한 메타 데이터를 포함합니다. 이는 다음 정보를 포함합니다.

- 전송자 ID
- 수신자 ID
- 문서 유형
- 문서 파일 이름 및 경로
- 고유 문서 ID

이는 이벤트 메타 데이터를 구성합니다. 커넥터는 이벤트 메타 데이터를 메모리의 이벤트 목록에 두고, 백업 및 복구를 위해 이벤트의 사본을 .event 확장자가 있는 이벤트 디렉토리의 파일에 씁니다.

커넥터가 이벤트 목록을 폴링할 때마다, 이는 커넥터 도착 시간에 기반하여 가장 먼저 도착한 이벤트를 검색합니다. 커넥터는 In 디렉토리에서 문서를 검색하고, Trading Partner 구성 파일을 사용하여 문서의 MIME 유형을 판별합니다.

커넥터는 MIME 유형을 DataHandler 클래스에 전달하여 적절한 Data Handler의 인스턴스를 작성합니다. 그런 다음 커넥터가 Data Handler를 호출하여 문서를 Business Object로 변환합니다. 각 Data Handler는 다른 기준을 사용하여 Business Object 이름을 판별합니다. Business Object가 생성된 후, 커넥터는 Business Object에 대한 등록을 확인합니다. 등록이 존재하는 경우, 커넥터가 오브젝트를 등록 비즈니스 프로세스에 전달합니다. (통합 브로커 특정 등록 정보는 브로커의 구현 안내서를 참조하십시오.)

병렬 처리

기본적으로, TPI Connector는 동시에 하나의 이벤트만 처리합니다. 그러나 성능을 개선시키기 위해 동시에 복수 수신 이벤트를 처리할 수 있습니다. 이 상황에서 각 이벤트는 자체 스레드에 할당되며, 스레드는 동시에 실행합니다.

커넥터 특정 등록 정보를 사용하여 병렬 처리를 사용 가능하게 하십시오. 자세한 정보는 13 페이지의 『커넥터 특정 등록 정보』를 참조하십시오.

이벤트 복구

커넥터가 이벤트 목록에서 이벤트를 검색할 때, 이는 처리가 완료되어 이벤트 파일이 아카이브 디렉토리에 이동될 때까지 해당 이벤트 파일의 확장자를 .inprogress로 변경합니다. 커넥터 장애 후 다시 시작할 때, 커넥터는 EventRecovery 등록 정보 설정에 따라 모든 `inprogress` 이벤트를 처리합니다. EventRecovery가 `Reprocess`로 설정된 경우, 커넥터는 이벤트 디렉토리를 읽고 모든 `inprogress` 이벤트를 이벤트 목록에 복원합니다. 선택적으로, `inprogress` 이벤트를 무시하거나 실패하도록 커넥터를 구성할 수 있습니다.

주: 커넥터가 `documentArriving` 이벤트 공고를 수신한 후 및 이벤트를 파일에 쓰기 전에, 창이 잠깐 표시되는 동안에 예기치 않게 종료하는 경우, 커넥터는 이 이벤트를 복구할 수 없습니다.

이벤트 아카이브

커넥터는 처리된 이벤트를 저장하는 archive 디렉토리를 유지보수합니다. 이벤트가 처리된 후, 이벤트 파일은 다음 확장자 중 하나로 이름이 지정됩니다.

- archive - 커넥터가 이벤트를 처리했습니다.
- unsubscribed - 이벤트에 대한 등록이 존재하지 않습니다. 통합 브로커 특정 등록 정보는 브로커의 구현 안내서를 참조하십시오.
- fail - 커넥터가 이벤트를 처리해서 통합 브로커에 이를 게시할 수 없었습니다.

Business Object 요청 처리

TPI Connector가 처리한 모든 요청 Business Object에는 구성 하위 오브젝트가 있어야 합니다. 이 오브젝트는 커넥터와 TPI Server가 오브젝트를 처리하는 데 필요한 정보를 포함합니다. 이는 전송자 ID, 수신자 ID 및 문서 유형을 포함합니다.

커넥터가 Business Object를 수신할 때, doVerbFor() 메소드는 Data Handler를 호출하여 Business Object를 적절하게 형식화된 스트림으로 변환합니다. Data Handler는 ReceiverID 및 DocumentType 값에 대한 Trading Partner 구성 파일에 나열된 MIME 유형에 기반하여 호출됩니다. TPI는 SenderID, ReceiverID 및 CycloneID에 대한 EDI 및 XML 문서를 구문 분석합니다. Data Handler는 문서 외부 디렉토리의 파일에 작성되는 문서 스트림을 출력합니다. 커넥터는 TPI Server API에서 sendDocument() 메소드를 호출하여 파일을 참조하는 문서 오브젝트를 TPI Server에 전달합니다.
sendDocument() 메소드는 TPI Server가 생성한 고유 문서 ID를 리턴합니다.

WaitForMDN 커넥터 등록 정보가 `true`로 설정되면, 커넥터는 TPI Server가 Trading Partner로부터 MDN을 수신했음을 표시할 때까지 대기합니다. 선택적으로, 하위 Meta Object의 WaitForMDN 속성을 채워서 Business Object당 하나를 기초로 하여

WaitForMDN 등록 정보를 대체할 수 있습니다. MDN이 일단 수신되면, doVerbFor() 메소드가 상태 코드(성공 또는 오류)를 통합 브로커에 리턴합니다. WaitForMDN 등록 정보가 `false`로 설정되는 경우, 커넥터는 TPI Server가 MDN을 리턴할 때까지 대기하지 않습니다.

로케일 종속 데이터 처리

커넥터가 국제화되어 2바이트 문자 세트를 지원하고, 지정된 언어로 메시지 텍스트를 전달할 수 있습니다. 커넥터가 문자 코드 세트를 사용하는 위치로부터 다른 코드 세트를 사용하는 위치로 데이터를 전송할 때, 문자 변환을 수행하여 데이터의 의미를 보존합니다.

JVM(Java Virtual Machine) 내의 Java 런타임 환경은 유니코드 문자 코드 세트로 데이터를 표현합니다. Unicode에는 널리 알려진 문자 코드 세트(1바이트와 다중 바이트 모두)의 문자에 대한 인코딩이 들어 있습니다. WebSphere Business Integration System에서 대부분의 구성요소는 Java로 작성되었습니다. 따라서, 데이터가 대부분의 WebSphere Business Integration System 구성요소 사이에 전송될 경우, 문자 변환이 필요하지 않습니다.

해당 언어와 해당 국가 또는 지역에 대한 오류 및 정보용 메시지를 로그하려면, 사용자 환경에 대한 로케일 표준 구성 등록 정보를 구성하십시오. 이러한 등록 정보에 관한 자세한 정보는 31 페이지의 부록 A『커넥터 표준 구성 등록 정보』를 참조하십시오.

제 2 장 커넥터 설치 및 구성

이 장에서는 TPI용 IBM WebSphere Business Integration Adapter의 커넥터 구성요소를 설치하고 구성하는 방법에 대해 설명합니다. 다음 섹션이 포함되어 있습니다.

- 『호환성』
- 8 페이지의 『전제조건』
- 8 페이지의 『TPI Server 설치』
- 9 페이지의 『TPI Server 이벤트 로깅 기능 구성』
- 9 페이지의 『커넥터 구성』
- 17 페이지의 『커넥터 시작』

호환성

어댑터가 사용하는 어댑터 프레임워크는 어댑터가 통신하는 통합 브로커(또는 브로커) 버전과 호환 가능해야 합니다. 다음 어댑터 프레임워크 버전과 다음 통합 브로커에서 Adapter for Trading Partner Interchange, 버전 3.4가 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 4.1.1, 4.2.0, 4.2.1, 4.2.2
- WebSphere MQ Integrator, 버전 2.1.0
- WebSphere MQ Integrator Broker, 버전 2.1.0
- WebSphere Business Integration Message Broker, 버전 5.0

예외는 릴리스 정보를 참조하십시오.

주: 통합 브로커 및 전제조건의 설치에 대한 지시사항은 다음 문서를 참조하십시오. ICS(WebSphere 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 *Implementing Adapters with WebSphere Message Brokers* 및 메시지 브로커의 설치 문서를 참조하십시오. 이 문서의 일부는 http://www.ibm.com/software/integration/mqfamily/library/manualsa/ 웹 사이트를 참조하십시오.

© Copyright IBM Corp. 2001, 2004

WebSphere Application Server는 *Implementing Adapters with WebSphere Application Server* 및 http://www.ibm.com/software/webservers/appserv/library.html을 참조하십시오.

전제조건

이 섹션에서는 커넥터를 설치하기 전에 수행할 필수 소프트웨어 구성요소와 타스크에 대해 설명합니다.

TPI Server 설치

TPI(Trading Partner Interchange) Adapter, 버전 3.4.x는 Trading Partner Interchange Server 버전 4.1과 4.2에서 사용할 수 있습니다.

TPI Connector를 설치하기 전에 TPI Server를 설치해야 합니다. 커넥터와 TPI Server는 동일한 시스템에 설치해야 합니다. TPI Server 설치 및 구성에 대한 지시사항은 TPI Server에 포함된 *관리자 안내서*를 참조하십시오.

Solaris에서 계정 및 권한 설정

Solaris 시스템에서 TPI를 설치하기 전에 구성원으로 WebSphere Business Integration Adapter 및 TPI Server 관리자를 모두 포함하는 사용자 그룹을 작성하십시오. 이는 TPI Connector에 TPI Server에 대한 액세스를 부여하는 데 필요합니다.

그룹의 두 구성원 모두가 TPI Server 설치 디렉토리에 대한 읽기/쓰기 사용권한을 갖는지 확인하십시오.

데이터베이스 지원

TPI Server는 표준 저장소로서 런타임 버전의 Sybase SQL Anywhere를 포함합니다. 런타임 버전으로서, 이는 데이터베이스를 관리하고 보기 위한 제한된 액세스만을 제공합니다. 따라서, TPI 저장소로서 Oracle 또는 Microsoft Server SQL Server를 구성하는 것이 바람직합니다. 외부 데이터베이스를 TPI 저장소로 구성하는 방법에 대한 정보는 TPI Server에 포함된 *관리자 안내서*를 참조하십시오.

TPI 어댑터 및 관련 파일 설치

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

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

TPI Server 이벤트 로깅 기능 구성

TPI Connector는 TPI Sever에 의해 로그된 이벤트 메시지를 사용하여 이벤트의 처리를 모니터합니다. 모든 서버 이벤트의 커넥터에 통보할 수 있게 디버그 모드에서 실행하도록 TPI Server를 구성해야 합니다.

주: TPI Server가 디버그 모드에서 실행되지 않는 경우, TPI Server `init()` 메소드가 이벤트 완료 메시지를 전송하는 동안 커넥터는 정지합니다.

디버그 모드에서 실행하도록 TPI Server를 설정하려면, 다음을 수행하십시오.

1. TPI Server 관리자를 시작하십시오.
2. 관리자 메뉴에서 도구 > 환경 설정을 선택하십시오.
3. 일반 탭에서 이벤트 로깅 레벨을 경보, 알림, 트랜잭션, 디버그로 설정하십시오.

커넥터 구성

주: TPI 커넥터와 TPI Server가 동일한 주소 영역에서 실행하기 때문에 TPI 커넥터를 1보다 큰 값으로 설정된 Parallel Process Degree Resource로 실행하는 것은 바람직하지 않습니다. Parallel Process Degree에 대한 자세한 정보는 *System Administration Guide*를 참조하십시오.

커넥터를 시작하기 전에, 다음 구성 타스크를 수행해야 합니다.

- Trading Partner 구성 파일을 작성하십시오.
- 커넥터를 구성하여 TPI Server를 시작하십시오.
- 커넥터 구성 등록 정보를 설정하십시오.

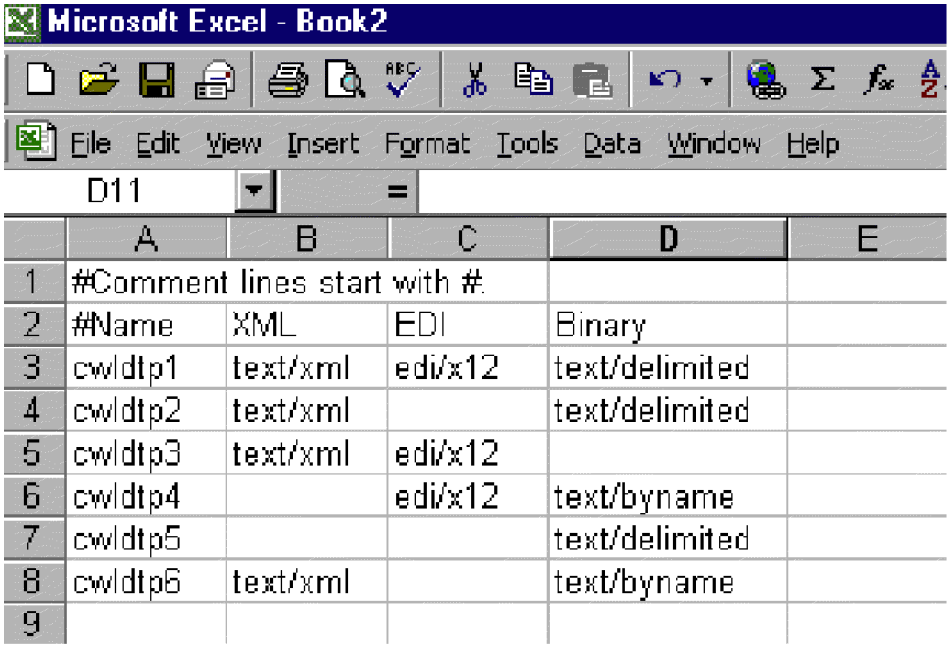
Trading Partner 구성 파일 작성

Trading Partner 구성 파일은 각 Trading Partner가 사용하는 XML, EDI 및 2진 형식에 대한 MIME 유형을 나열합니다. 이는 커넥터가 문서나 Business Object를 처리할 때마다, Trading Partner 구성 파일에서 MIME 유형을 확보합니다. 제공된 문서나 Business Object에 올바른 Data Handler를 호출하기 위해 MIME 유형이 DataHandler 클래스에 전달됩니다.

Trading Partner 구성 파일은 탭으로 구분되는 텍스트 파일입니다. 파일의 각 행은 이 Trading Partner가 사용하는, XML, EDI 및 2진 형식에 대한 MIME 유형이 뒤따르는 TPI Trading Partner ID를 포함합니다. Trading Partner 구성 파일 형식은 다음과 같습니다.

Trading Partner ID <탭> *XML MIME 유형* <탭> *EDI MIME 유형* <탭> *2진 MIME 유형*

적절한 형식화를 확인하려면, Microsoft Excel 스프레드시트로 파일을 작성하십시오. 그런 다음 파일을 텍스트로 저장하십시오(탭으로 구분됨). 그림 2는 Excel로 파일을 작성하는 방법을 보여 줍니다.



	A	B	C	D	E
1	#Comment lines start with #.				
2	#Name	XML	EDI	Binary	
3	cwldtp1	text/xml	edi/x12	text/delimited	
4	cwldtp2	text/xml		text/delimited	
5	cwldtp3	text/xml	edi/x12		
6	cwldtp4		edi/x12	text/byname	
7	cwldtp5			text/delimited	
8	cwldtp6	text/xml		text/byname	
9					

그림 2. Excel로 Trading Partner 구성 파일 작성

TPI가 설치된 시스템의 모든 디렉토리에 파일을 저장할 수 있습니다. TradingPartnerConfigurationFile 커넥터 등록 정보는 파일의 위치를 보유합니다. 자세한 정보는 13 페이지의 『커넥터 특정 등록 정보』를 참조하십시오.

Trading Partner 구성 파일을 갱신하여 새 파트너를 추가하거나 커넥터가 실행되는 동안 문서 정보를 수정할 수 있습니다. 갱신사항을 로드하기 위해 커넥터를 다시 시작하는 것이 필요하지 않으며, 커넥터는 처리 중 구성 파일에 대한 갱신사항을 검색합니다.

TPI Server 시작을 위한 커넥터 구성

TPI Connector는 TPI Server와 함께 제공된 JVM을 사용하여 TPI Server를 실행합니다. 이 JVM을 사용하여 TPI Connector가 java 프로세스로서 시작하고, 이 프로세스 내에서 TPI Server를 시작합니다. 커넥터 설치 후 커넥터 시작 파일인 Windows NT의 `start_TPI.bat` 및 Solaris의 `start_TPI.sh`를 편집하여 이 작동을 구성해야 합니다. TPI Server를 시작하고 시스템 종료하도록 커넥터를 구성하려면 다음을 수행하십시오.

1. `\connectors\TPI directory`에 있는 `start_TPI` 파일을 여십시오.
2. `CYCLONEHOMEDIR` 속성의 값을 TPI 홈 디렉토리 경로로 변경하십시오.
3. 저장하고 파일을 닫으십시오.

TPI Adapter가 TPI Server를 인스턴스로 만들 수 있도록 TPI Server와 함께 제공된 JVM을 구성해야 합니다. TPI에 제공된 JVM을 구성하려면 다음을 수행하십시오.

Windows의 경우

1. \dependencies\win 폴더를 TPI Adapter의 %CROSSWORLDS%\connectors\TPI 디렉토리에 위치시키십시오.
2. \cijre 폴더를 TPI 홈 디렉토리에 위치시키십시오.
3. %CROSSWORLDS%\connectors\TPI\dependencies\win 폴더의 내용을 %CYCLONEHOMEDIR%\cijre 폴더에 복사하십시오.
4. 다음 파일은 이제 %CYCLONEHOMEDIR%\cijre에 존재해야 합니다.
   - \bin\orb.dll
   - \bin\orb.dll
   - \lib\orb.properties
   - \lib\ext\ibmorb.jar
   - \lib\ext\ibmext.jar

Solaris의 경우

1. /dependencies/sol 폴더를 TPI Adapter의 ${CROSSWORLDS}/connectors/TPI 디렉토리에 위치시키십시오.
2. /cijre 폴더를 TPI Server의 홈 디렉토리에 위치시키십시오. 이는 ${CYCLONEHOMEDIR}로서 식별됩니다.
3. ${CROSSWORLDS}/connectors/TPI/dependencies/sol 폴더의 내용을 ${CYCLONEHOMEDIR}/cijre 폴더에 복사하십시오.
4. 다음 파일은 이제 ${CYCLONEHOMEDIR}/cijre에 존재해야 합니다.
   - \lib\orb.properties
   - \lib\ext\ibmorb.jar
   - \lib\ext\ibmext.jar
   - \lib\sparc\liborb.so
   - \lib\sparc\liborb_g.so

시작 파일 편집 외에, TPI Server를 시작하려면 커넥터에 대한 다음 커넥터 등록 정보를 설정해야 합니다.

- MetaEventDir
- DocumentOutDir
- TradingPartnerConfigurationFile
- ArchiveProcessedDocDirSee

자세한 정보는 13 페이지의 『커넥터 특정 등록 정보』를 참조하십시오.

커넥터 및 TPI Server를 Windows 서비스로 실행

TPI 커넥터 및 서버를 Windows 서비스로 실행할 수 있습니다. 커넥터를 서비스로 설치한 후, 다음과 같이 start_TPI.bat 파일을 수정해야 합니다.

- Cyclone JVM에서 커넥터를 실행하려면, *%ProductDir%*\bin\java 값을 %CYCLONEHOMEDIR%\cijre\bin\java로 변경하십시오.
- 다음을 %CYCLONEHOMEDIR%\cijre\bin\java 명령행 끝에 추가하여 JRE 프로세스의 출력 경로를 파일로 재지정하십시오.

```
>"%CONNDIR%"\connectors\TPI\TPITrace.txt
```

커넥터 구성 등록 정보 설정

커넥터의 표준 및 커넥터 특정 구성 등록 정보를 설정해야 커넥터를 실행할 수 있습니다. 다음 도구 중 하나를 사용하여 커넥터의 구성 등록 정보를 설정하십시오.

- Connector Configurator(WebSphere ICS가 통합 브로커인 경우) -- System Manager에서 이 도구에 액세스하십시오.
- Connector Configurator(WebSphere MQ Integrator Broker가 통합 브로커인 경우) -- IBM WebSphere Business Integration Adapter 프로그램 폴더에서 이 도구에 액세스하십시오. Connector Configurator에 대한 자세한 정보는 49 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

표준 커넥터 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이 등록 정보에 대한 자세한 정보는 31 페이지의 부록 A 『커넥터 표준 구성 등록 정보』를 참조하십시오.

중요

커넥터가 ICS 및 WebSphere MQ Integrator Broker 통합 브로커 모두를 지원하기 때문에, 이 두가지 브로커에 대한 구성 등록 정보 모두 커넥터에 해당됩니다.

또한, TPI용 IBM WebSphere Business Integration Adapter에 특정한 구성 정보는 표 1을 참조하십시오. 이 표의 정보는 부록의 정보를 보충합니다.

표 1. 이 커넥터에 특정한 등록 정보

등록 정보	참고
AgentConnections	이 커넥터는 단일 스레드 방식입니다. 따라서, 이는 AgentConnections 등록 정보를 사용할 수 없습니다.
CharacterEncoding	이 커넥터는 Java 기반이므로 CharacterEncoding 등록 정보를 사용하지 않습니다.

표 1. 이 커넥터에 특정한 등록 정보 (계속)

등록 정보	참고
Locale	이 커넥터는 국제화되었기 때문에 로케일 등록 정보의 값을 변경할 수 있습니다. 현재 지원되는 로케일을 알려면 커넥터에 대한 릴리스 정보를 참조하십시오. **주:** WebSphere MQ Integrator Broker를 브로커로 사용할 경우, 어댑터, 브로커 및 응용프로그램에 대해 동일한 로케일을 사용해야 합니다.

커넥터 특정 등록 정보

커넥터 특정 구성 등록 정보는 런타임에 커넥터에 필요한 정보를 제공합니다. 커넥터 특정 등록 정보는 또한 커넥터 내부 구성 또는 논리 변경 방법을 제공하며, 커넥터의 코드 변경 및 재빌드 할 필요가 없습니다.

표 2는 커넥터에 대한 커넥터 특정 등록 정보를 나열합니다. 등록 정보의 설명은 다음 섹션을 참조하십시오.

표 2. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값
ArchiveProcessedDocDir	*<유효한 디렉토리 경로>*	
ArchiveProcessedDocInfo	`true` 또는 `false`	true
BackupRequired	`true` 또는 `false`	true
CycloneServerArgs	커넥터가 사용하는 기타 인수와 충돌하지 않는 유효한 모든 TPI Server 인수	-appagent; -nogui; -console;
DataEncoding	JVM이 지원하는 올바른 인코딩 이름	이 등록 정보에 해당하는 값이 지정되지 않은 경우, 운영 체제에서 설정한 인코딩 유형이 기본값이 됩니다.
DataHandlerDefaultMO		MO_DataHandler_ 기본값
DefaultBinaryMimeType	*<유효한 MIME 유형>*	
DefaultEDIMimeType	*<유효한 MIME 유형>*	
DefaultVerb	커넥터가 지원하는 모든 verb	
DefaultXMLMimeType	*<유효한 DataHandler 클래스 이름>*	
DeliverOnArrival	`true` 또는 `false`	false
DeliverOnArrivalThreads	1 - 10 사이의 모든 숫자	1
DocumentOutDir	*<유효한 디렉토리 경로>*	
EventRecovery	`FailOnStartup`, `Reprocess`, `LogError` 또는 `Ignore`	FailOnStartup
MetaEventDir	*<유효한 디렉토리 경로>*	
PollQuantity		25
TradingPartnerConfigurationFile	*<완전한 파일 이름>*	
WaitForController		500밀리초
WaitForMDN	`true` 또는 `false`	true

ArchiveProcessedDocDir: 메타 이벤트가 아카이브되는 문서를 처리한 디렉토리입니다. ArchiveProcessedDocInfo 등록 정보가 true로 설정된 경우, 이 등록 정보는 필수입니다.

ArchiveProcessedDocInfo: 문서가 처리되어 통합 브로커에 일단 전송된 후 커넥터가 메타 이벤트를 보유하는지 여부를 판별합니다. 이 등록 정보를 false로 설정하면, 메타 이벤트가 처리 후 삭제됩니다. 기본값은 true입니다.

BackupRequired: TPI Server가 각 문서를 전송한 후 백업하는지 여부를 판별합니다. true로 설정된 경우, TPI가 각 문서를 전송한 후 백업합니다. 기본값은 true입니다.

ControllerWaitCount: 제어기가 활성 상태인지 여부를 알아 보기 위해 커넥터를 점검하는 최대 시간을 지정합니다. 이 등록 정보는 WaitForController 등록 정보와의 결합에 사용됩니다. 커넥터가 제어기 활동을 검사할 때마다, WaitForController에서 지정한 시간 동안 대기합니다. 최대 시도 횟수 이후에 제어기가 응답하지 않는 경우, 이벤트 처리에 실패합니다. 기본값은 1입니다.

CycloneServerArgs: 시작 시에 TPI Server로 전달될 세미콜론으로 분리되는 인수의 목록입니다. 이 등록 정보는 필수입니다. 기본값은 -appagent; -nogui; -console; 입니다.

DataEncoding: 커넥터가 사용하는 데이터 인코딩의 유형을 지정합니다. JVM이 지원하는 인코딩 유형이 올바른 값입니다. 비ASCII 문자 인코딩이 사용될 때마다 이 등록 정보를 설정해야 합니다. DataEncoding 등록 정보가 설정되지 않은 경우, 커넥터가 운영 체제 레벨에서 지정한 인코딩 유형을 사용합니다.

DataHandlerDefaultMO: 커넥터가 각 지원 형식에 대한 Data Handler의 구성을 판별하기 위해 사용하는 Meta Object의 기본 이름입니다. 이 등록 정보는 필수입니다. 기본값은 MO_DataHandler_Default입니다.

DefaultBinaryMimeType: Trading Partner 구성 파일에 Trading Partner가 구성되지 않는 경우, Data Handler가 2진 문서에 사용하는 MIME 유형입니다. 2진 MIME 유형이 TPI에 사용되는 경우 이 등록 정보는 필수입니다. 이 등록 정보에는 기본값이 설정되지 않습니다.

DefaultEDIMimeType: Trading Partner 구성 파일에 Trading Partner가 구성되지 않는 경우, Data Handler가 EDI 문서에 사용하는 MIME 유형입니다. EDI MIME 유형이 TPI에 사용되는 경우, 이 등록 정보는 필수입니다. 이 등록 정보에는 기본값이 설정되지 않습니다.

DefaultVerb: 폴링 중 Data Handler가 verb를 설정하지 않은 경우, 수신 Business Object 내에 설정되는 verb를 지정합니다.

DefaultXMLMimeType: Trading Partner 구성 파일에 Trading Partner가 구성되지 않는 경우, Data Handler가 XML 문서에 사용하는 MIME 유형입니다. XML MIME 유형이 TPI에 사용되는 경우, 이 등록 정보는 필수입니다. 이 등록 정보에는 기본값이 설정되지 않습니다.

DeliverOnArrival: 사용할 이벤트 처리의 메소드를 지정합니다. DeliverOnArrival이 false로 설정된 경우, 한 번에 하나의 이벤트가 처리되며 PollForEvent 스레드는 메모리로부터 이벤트를 검색하고 처리한 후 다음 이벤트를 검색합니다. DeliverOnArrival이 true로 설정되는 경우, 복수 이벤트를 동시에 처리할 수 있습니다.
DeliverOnArrivalThreads 등록 정보에서 판별되는 병렬에서 처리할 수 있는 이벤트 수입니다. DeliverOnArrival의 기본값은 false입니다.

DeliverOnArrivalThreads: 동시 이벤트 처리에 할당되는 스레드 수를 지정합니다. DeliverOnArrival 등록 정보가 true로 설정되는 경우, DeliverOnArrivalThreads 등록 정보를 수신 이벤트에 할당하려는 스레드 수로 설정하십시오. 병렬 처리에 대한 최소값은 2이며, 최대값은 10입니다. 기본적으로, DeliverOnArrivalThreads는 1로 설정되고 병렬 처리는 사용 가능하지 않습니다.

DocumentOutDir: TPI가 이를 처리하기 전에 아웃바운드 문서가 임시로 작성되는 디렉토리의 위치입니다. 시스템 장애의 경우, 이 디렉토리에서 문서를 복구할 수 있습니다. 이 등록 정보는 필수입니다.

EventRecovery: 이벤트 복구를 위한 작동을 판별합니다. .inprogress 확장자가 있는 파일에 대한 출력 디렉토리를 확인합니다. 등록 정보가 FailOnStartup으로 설정되는 경우, 커넥터가 시작에 실패합니다. Reprocess로 설정되는 경우, 커넥터가 이들 이벤트를 서버에 다시 제출합니다. LogError로 설정되는 경우, 커넥터가 오류를 로그하지만 시스템 종료하지는 않습니다. Ignore로 설정되는 경우, 커넥터가 이러한 이벤트를 무시합니다. 기본값은 FailOnStartup입니다.

MetaEventDir: 복구 목적으로 TPI 이벤트 정보를 지속시키기 위해 사용되는 디렉토리. 이 등록 정보는 필수입니다.

PollQuantity: 커넥터가 이벤트 목록을 폴링할 때마다 검색하는 이벤트의 수를 지정합니다. 기본값은 25입니다.

TradingPartnerConfigurationFile: Trading Partner 구성 파일의 완전한 이름입니다. 이 파일은 2진, XML 및 EDI 메시지에 대한 각 Trading Partner의 문서에 사용되는 MIME 유형을 포함합니다. MIME 유형은 각 문서에 올바른 Data Handler를 호출하기 위해 사용됩니다. 이 등록 정보는 필수입니다. 등록 정보가 지정되지 않는 경우, 커넥터가 시작에 실패합니다.

WaitForController: 제어기가 활성 상태가 되기를 커넥터에서 대기하는 시간의 양을 밀리초로 지정합니다. 이 등록 정보는 ControllerWaitCount 등록 정보와의 결합에 사용됩니다. 기본값은 500밀리초입니다.

WaitForMDN: 커넥터가 Trading Partner로부터 MDN을 대기하거나, 문서를 TPI Server에 전달한 후 요청 스레드에서 리턴하는지 여부를 판별합니다. MDN은 프로토콜 레벨에서 전송이 초기화되었음을 표시합니다. 하위 Meta Object의 WaitForMDN 속성을 설정하여 Business Object당 하나에 기초하여 이 등록 정보를 대체할 수 있습니다. 기본값은 true입니다.

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, `connectorInstance`는 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 다음을 수행하십시오.

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 16 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 스크립트**를 사용하여 명시적으로 시작해야 합니다. 시작 스크립트는 커넥터의 런타임 디렉토리에 있어야 합니다.

ProductDir\connectors*connName*

여기서, *connName*은 커넥터를 식별합니다. 시작 스크립트의 이름은 표 3에 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.

표 3. 커넥터의 시작 스크립트

운영 체제	시작 스크립트
UNIX 기반 시스템	connector_manager_*connName*

표 3. 커넥터의 시작 스크립트 (계속)

운영 체제	시작 스크립트
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 경우, 다음을 수행하십시오.
  - Windows 시스템의 경우

    ```
    start_connName connName brokerName [-cconfigFile ]
    ```
  - UNIX 기반 시스템의 경우

    ```
    connector_manager_connName -start
    ```

  여기서, *connName*은 커넥터의 이름이며 *brokerName*은 다음과 같이 통합 브로커를 식별합니다.
  - WebSphere InterChange Server의 경우, *brokerName*에 ICS 인스턴스의 이름을 지정하십시오.
  - WebSphere 메시지 브로커(WebSphere MQ Integrator, WebSphere MQ Integrator Broker 또는 WebSphere Business Integration Message Broker) 또는 WebSphere Application Server의 경우, *brokerName*에 브로커를 식별하는 문자열을 지정하십시오.

  주: Windows 시스템의 WebSphere 메시지 브로커 또는 WebSphere Application Server에서는 `-c` 옵션 다음에 커넥터 구성 파일의 이름이 *포함되어야* 합니다. ICS의 경우, `-c`는 선택적입니다.
- System Manager를 시작할 때 실행되는 Adaptor Monitor(WebSphere Business Integration Adapters 제품 전용)에서 다음을 수행할 수 있습니다.

  이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.
- System Monitor(WebSphere InterChange Server 제품 전용)에서 다음을 수행할 수 있습니다.

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

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

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

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

커넥터 시작에 대한 특수 고려사항

Cyclone 4.1 이하를 사용 중인 경우, WebSphere MQ Integrator Broker가 브로커로서 사용되면 TPI의 커넥터를 시작하는 데 어려움이 있을 수 있습니다(WebSphere MQ Integrator Broker 통합 브로커 전용). 로컬 커넥터 구성 파일에 암호화된 등록 정보 값(예를 들어, 암호)이 있는 경우, 시작 중에 java.lang.NullPointerException 예외가 발생하고 커넥터가 종료합니다.

이 문제를 예방하려면, java.security 파일의 제공자 목록을 다음 값으로 바꾸십시오.

```
security.provider.1=com.cyclonecommerce.crossworks.provider.Cyclone
security.provider.2=iaik.security.provider.IAIK
security.provider.3=sun.security.provider.Sun
```

Windows 및 Solaris 시스템에서는 %CYCLONEHOMEDIR%/cijre/lib/security 디렉토리에 파일이 있습니다.

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 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 종료되면 커넥터가 중지됩니다.

제 3 장 커넥터에 대한 Business Object 개발

이 장에서는 TPI(Trading Partner Interchange) Business Object에 필요한 요소와 구조에 대해 설명합니다. 특히 TPI Connector의 Business Object 요구사항에 대해 논의합니다. 다음 섹션이 포함되어 있습니다.

- 『Data Handler 및 문서 형식』
- 23 페이지의 『Business Object 및 속성 이름 지정 규칙』
- 23 페이지의 『Business Object 구조』
- 28 페이지의 『맵핑 고려사항(WebSphere ICS 통합 브로커 전용)』
- 29 페이지의 『Business Object Verb 처리』
- 29 페이지의 『Business Object 속성 등록 정보』
- 29 페이지의 『Business Object 응용프로그램 특정 정보』

Data Handler 및 문서 형식

주: TPI를 통한 RNIF용 Data Handler와 함께 TPI용 어댑터도 전달됩니다(77 페이지의 부록 D 『Adapter for Trading Partner Interchange 샘플 시나리오』 참조).

커넥터는 XML, EDI 및 2진 형식의 문서 교환을 지원합니다. TPI Connector는 Data Handler를 사용하여 TPI 문서를 WebSphere Business Integration Adapter Business Object로 변환하고, WebSphere Business Integration Adapter 오브젝트를 TPI 지원 문서 형식으로 변환합니다.

표 4. WebSphere Business Integration Adapter 전달 Data Handler

문서 형식	**WebSphere Business Integration Adapter 전달 Data Handler**
XML	XML
EDI	EDI
2진	모든 형식

커넥터는 문서 MIME 유형을 DataHandler 클래스에 전달하여 적절한 Data Handler를 호출합니다. 또한, 커넥터가 수신 Business Object에 있게된 경우, BOPrefix 값을 전달할 수 있습니다. 인바운드 문서를 처리할 때, 커넥터는 Trading Partner 구성 파일에서 문서의 MIME 유형을 확보합니다. 아웃바운드 요청 Business Object를 처리할 때, 커넥터는 하위 Meta Object의 DocumentType 및 ReceiverID 속성의 값에 기반하여 Trading Partner 구성 파일에서 MIME 유형을 확보합니다. 자세한 정보는 23 페이지의 『Business Object 구조』를 참조하십시오.

© Copyright IBM Corp. 2001, 2004

Business Object에 대한 Data Handler 요구사항

TPI Connector가 스트림에서 Business Object로 Business Object에서 스트림으로 변환하기 위해 Data Handler를 호출하기 때문에, 각 Business Object는 변환을 수행하기 위해 호출된 Data Handler의 스펙을 따라야 합니다. 특정 Data Handler의 특정 Business Object 요구사항에 대한 정보는 *Data Handler 안내서*를 참조하십시오.

XML 및 EDI 문서에 대한 TPI Server 요구사항

TPI Server에는 EDI 및 XML 문서에 있는 머리글 정보의 형식화와 내용에 대한 특정 요구사항이 있습니다. 모든 XML 및 EDI 문서는 유효한 SenderID, RecieverID 및 고유 CycloneID를 포함해야 합니다. SenderID나 ReceiverID 값이 유효하지 않은 경우 또는 CycloneID 값이 TPI 시스템에서 고유하지 않은 경우, TPI Server가 문서를 처리하지 않습니다. 이들 값은 Data Handler가 문서에 올바르게 넣을 수 있도록 요청 Business Object와 일치해야 합니다.

EDI 요구사항

EDI 문서의 경우, 이들 값의 위치가 EDI 스펙에 의해 지정됩니다. 다음 샘플 EDI 머리글을 살펴 보십시오.

```
ISA*00* *01*XXXXXX *L1*2 *L0*0*961106*2106*U*00302*000087875*
```

SenderID: 이 예제에서 SenderID는 두 개의 열, *L1*2에서 읽혀집니다. 첫 번째 열은 2문자 SenderID 규정자를 포함해야 합니다. 두 번째 열은 SenderID의 모든 기존 문자를 포함해야 합니다.

ReceiverID: SenderID와 마찬가지로, 이 예제에서 ReceiverID도 EDI 문서 머리글의 두 개의 열, *L0*0에서 읽혀집니다. 첫 번째 열은 2문자 ReceiverID 규정자를 포함해야 합니다. 두 번째 열은 ReceiverID의 나머지 모든 문자를 포함해야 합니다.

CycloneID: EDI 문서의 경우, CycloneID는 EDI 제어 ID에 일치합니다. 이 예에서, CycloneID는 단일 열, *000087875*에 표시됩니다. 이 값은 EDI 제어 ID 번호 87875에 일치합니다.

XML 요구사항

XML 문서에서 SenderID, ReceiverID 및 CycloneID의 위치는 TPI Server Administrator를 사용하여 사용자 정의할 수 있습니다. 자세한 정보는 TPI Server에 있는 관리자 안내서를 참조하십시오.

Business Object 및 속성 이름 지정 규칙

요청 처리에 사용되는 Business Object에 대한 특정한 이름 지정 규칙은 없습니다. 이벤트 공고에서 처리되는 Business Object는 수신 문서 형식과 WebSphere Business Integration Adapter Business Object 사이의 변환을 수행하는 Data Handler의 이름 지정 규칙을 따라야 합니다. 자세한 정보는 *Data Handler 안내서*를 참조하십시오.

Business Object 구조

TPI Connector에는 처리하는 Business Object에 대한 2가지 요구사항이 있습니다.

- 각 Business Object는 커넥터가 TPI Server API를 사용하여 내용을 형식화하고 문서를 전송하는 데 필요한 동적 메타 데이터를 보유하는 Meta Object를 포함해야 합니다.
- 각 Business Object는 오브젝트를 데이터 스트림으로 변화하는 데 사용되는 Data Handler에 필요한 Business Object 구조에 합치해야 합니다.

하위 Meta Object 속성

비즈니스 프로세스에서 커넥터에 전송된 각 Business Object는 카디널리티 1 하위로 단일 카디널리티 Meta Object를 포함해야 합니다. 커넥터가 오브젝트를 변환하기 위해 적절한 Data Handler를 호출하고, TIP Server API에서 sendDocument() 메소드를 호출하기 위해 필요한 동적 메타 데이터가 Meta Object에 들어 있습니다. 표 5에는 Meta Object의 속성이 나열되어 있습니다. 나열된 모든 속성은 Business Object 정의에서 정의되어야 합니다.

표 5. TPI Connector에 대한 Meta Object 속성

속성 이름	설명	기본값
DocumentExt	요청 처리 중에 사용되는 문서의 파일 확장자를 지정합니다.	
DocumentType	문서의 형식(XML, EDI 또는 2진)을 지정합니다.	binary
BOPrefix	XML Data Handler의 인스턴스를 작성하기 위해 MIME 유형과 함께 사용됩니다.	
SenderID	문서를 전송 중인 Trading Partner에 대한 고유 TPI ID를 지정합니다.	기본값은 사용자가 설정해야 합니다.
ReceiverID	문서를 수신 중인 Trading Partner에 대한 고유 TPI ID를 지정합니다.	기본값은 사용자가 설정해야 합니다.
UniqueID	TPI Server에 의해 각 문서에 지정된 고유 ID를 지정합니다. 이 속성은 선택적이지만, Business Object 요청 처리에 사용할 수 있습니다.	
OriginalName	문서 외부 디렉토리에 작성되는 문서 오브젝트 파일의 이름을 지정하기 위해 사용되는 접두부를 지정합니다.	사용자가 설정해야 합니다.
WaitForMDN	커넥터가 문서를 전송한 후 TPI Server로부터 MDN을 대기하는지 여부를 판별합니다.	true
BackupRequired	TPI Server가 문서를 Trading Partner에 전송한 후 문서의 백업 사본을 작성하는지 여부를 판별합니다.	true

DocumentExt

DocumentExt 속성을 사용하면 요청 처리 중에 파일 확장자(예: .xml 또는 .edi)를 지정할 수 있습니다. 이는 선택적 속성이며 기본값은 없습니다.

DocumentType

Trading Partner 구성 파일에서 문서 MIME 유형을 확보하기 위해 DocumentType 속성은 ReceiverID 속성과 함께 사용됩니다. 커넥터는 MIME 유형을 사용하여 적절한 Data Handler를 호출합니다. 기본값은 binary입니다.

DocumentType 속성은 특수 값 CW_RNIF를 승인할 수도 있습니다. 이 경우, ReceiverID 속성에 대한 값을 지정할 필요가 없습니다. 커넥터는 이를 사용하지 않습니다.

BOPrefix

커넥터는 적절한 Data Handler 인스턴스를 호출하기 위해 MIME 유형과 함께 BOPrefix를 사용합니다. Data Handler 구현이 한 가지 MIME 유형만을 처리하는 경우, 하위 Meta Object의 BOPrefix 속성은 선택적입니다.

SenderID

SenderID는 문서 전송자의 고유 상대 ID입니다. 커넥터는 이 값을 사용하여 sendDocument() 호출에서 전달되는 DefaultDocument 오브젝트를 빌드합니다.

ReceiverID

ReceiverID는 문서를 수신할 Trading Partner의 고유 ID입니다. Trading Partner 구성 파일에서 문서 MIME 유형을 확보하기 위해 이 값은 DocumentType 속성과 함께 사용됩니다. 이 값을 사용하여 sendDocument() 호출에서 전달되는 DefaultDocument 오브젝트를 빌드합니다. 이 속성은 선택적이지만, Business Object 처리에 사용할 수 있습니다.

UniqueID

TPI Server에 의해 각 문서에 지정된 고유 ID입니다.

Original Name

커넥터가 검색을 위해 DocumentOutDir 디렉토리에 쓰는 출력 파일의 이름을 지정하기 위해서 TPI Server가 사용하는 접두부입니다. 고유성을 보장하기 위해 ObjectEventId가 이 이름에 추가됩니다.

WaitForMDN

이 속성은 TPI로부터 문서에 대한 MDN이 수신되었다는 공고를 커넥터가 대기하는지 여부를 판별합니다. 이 속성은 선택적입니다. 이 속성이 Meta Object에 있는 경우, 이는 커넥터의 WaitForMDN 등록 정보를 대체합니다. 기본값은 true입니다.

BackupRequired

이 속성은 TPI Server가 문서를 전송한 후 이를 백업하도록 하는 플래그를 설정합니다. 이 속성은 선택적입니다. 이 속성이 있는 경우, 이는 BackupRequired 커넥터 등록 정보를 대체합니다. 이 값은 sendDocument() 호출에서 매개변수로 전달됩니다. 기본값은 true입니다.

Business Object 구조에 대한 Data Handler 요구사항

TPI Connector가 사용하는 각 Data Handler에는 Business Object 구조에 대한 자체 요구사항이 있습니다. Business Object는 이들을 변환하기 위해 호출된 Data Handler의 스펙을 따라야 합니다. 이들 요구사항은 *Data Handler 안내서*에 문서화되어 있습니다.

하위 Meta Object가 있는 샘플 Business Object

다음은 카디널리티 1 하위로서의 Meta Object가 있는 TPI Connector Business Object 정의의 예제입니다. 이 Business Object 정의는 한정된 Data Handler를 위해 개발되었습니다.

```
[BusinessObjectDefinition]
Name = TPICustBO
Version = 1.0.0.
AppSpecificInfo = cw_mo_cfg=CustBORouteInfo;

[Attribute]
Name = FirstName
Type = String
IsKey = true
IsRequired = true
AppSpecificInfo =
[end]

[Attribute]
Name = LastName
Type = String
IsKey = true
IsRequired = true
AppSpecificInfo =
[end]

[Attribute]
Name = Company
Type = String
IsKey = true
IsRequired = true
AppSpecificInfo =
[end]

[Attribute]
Name = City
Type = String
IsKey = false
IsRequired = false
```

```
AppSpecificInfo =
[end]

[Attribute]
Name = CustBORouteInfo
Type = TPIRouteInfo
ContainedObjectVersion = 1.0.0.
Relationship = Containment
Cardinality = 1
MaxLength = 0
IsKey = false
IsForeignKey = false
IsRequired = false
DefaultValue =
AppSpecificInfo = type=cw_mo_cfg
[end]

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

[Verb]
Name = Create
[End]

[Verb]
Name = Retrieve
[End]

[End]

[BusinessObjectDefinition]
Name = TPIRouteInfo
Version = 1.0.0.

[Attribute]
Name = SenderId
Type = String
IsKey = true
IsRequired = true
AppSpecificInfo =
[end]

[Attribute]
Name = ReceiverId
Type = String
IsKey = true
IsRequired = true
AppSpecificInfo =
[end]

[Attribute]
Name = DocumentType
```

```
Type = String
IsKey = true
IsRequired = true
AppSpecificInfo =

[end]

[Attribute]
Name = BOPrefix
Type = String
IsKey = false
IsRequired = false
AppSpecificInfo =
[end]

[Attribute]
Name = WaitForMDN
Type = String
IsKey = false
IsRequired = false
AppSpecificInfo =
[end]

[Attribute]
Name = BackupRequired
Type = String
IsKey = false
IsRequired = false
AppSpecificInfo =
[end]

[Attribute]
Name = OriginalName
Type = String
IsKey = false
IsForeignKey = false
IsRequired = false
DefaultValue = TPICustomer
[End]

[Attribute]
Name = UniqueId
Type = String
IsKey = false
IsForeignKey = false
IsRequired = false
[end]

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

[Attribute]
```

```
Name = OriginalName
Type = String
IsKey = false
IsForeignKey = false
IsRequired = false
DefaultValue = TPICustomer
[End]

[Attribute]
Name = UniqueId
Type = String
IsKey = false
IsForeignKey = false
IsRequired = false
[end]

[Verb]
Name = Create
[End]

[Verb]
Name = Retrieve
[End]

[End]
```

맵핑 고려사항(WebSphere ICS 통합 브로커 전용)

일부 TPI 구현에서, 회사에는 Sales_Order와 같이 다른 형식의 동일한 Business Object를 기반으로 문서를 수신하는 둘 이상의 Trading Partner가 있을 수 있습니다. 이 시나리오에는 데이터를 수신 중인 Trading Partner에 따라 Business Object를 다른 형식으로 출력할 수 있는 다형태 맵이 필요합니다.

다형태 맵은 본질적으로 동일한 입력 오보젝트를 사용하지만 다른 유형의 출력 오브젝트를 생성하며 각 출력 유형에 하나씩 있는 둘 이상의 개별 서브맵입니다. 속성 값과 같은 조건을 테스트하는 기본 단일 맵이 이들 서브맵을 호출하여 호출할 서브맵을 판별합니다. 서브맵이 출력 오브젝트를 일단 생성하면, 기본 맵이 커넥터에 오브젝트를 리턴합니다.

TPI를 위해 이를 구현하는 한 가지 방법은 판별 조건으로 ReceiverID 속성 값을 사용하는 것입니다. 기본 맵은 입력 오브젝트의 ReceiverID에 기반하여, 적절한 서브맵을 호출합니다. 다음은 서브맵의 타스크입니다.

- Trading Partner에 의해 지정된 문서 형식에 따라, 적절한 Data Handler에 대한 출력 오브젝트를 생성합니다.
- 적절한 메타 데이터로 하위 Meta Object를 채웁니다.

다형태 맵에 대한 자세한 정보는, *맵 개발 안내서*를 참조하십시오.

Business Object Verb 처리

TPI Connector가 처리한 모든 Business Object에는 Create Verb 설정이 있어야 합니다. Create Verb가 요청에 설정되지 않는 경우, 커넥터가 Business Object 처리에 실패합니다. Data Handler가 아직 설정하지 않은 경우, 커넥터는 모든 이벤트 Business Object에 Create Verb를 설정합니다.

Business Object 속성 등록 정보

TPI Connector에는 Business Object 속성 등록 정보에 대한 특정 요구사항이 없습니다. Business Object 속성 등록 정보는 Business Object를 변환하기 위해 사용되는 Data Handler의 요구사항을 따라야 합니다. 이들 요구사항은 *Data Handler 안내서*에 문서화되어 있습니다.

Business Object 응용프로그램 특정 정보

Business Object 정의의 응용프로그램 특정 정보는 커넥터와 Data Handler에 Business Object를 처리하는 방법에 관한 지시사항을 제공합니다. TPI Business Object에 대한 응용프로그램 특정 정보는 Business Object 처리에 사용되는 커넥터와 Data Handler 모두의 요구를 만족시켜야 합니다.

응용프로그램 특정 정보에 대한 커넥터 요구사항

TPI Connector는 Business Object 레벨의 응용프로그램 특정 정보를 사용하여 하위 Meta Object를 식별합니다. Business Object를 변환할 때, WebSphere Business Integration Adapter 전달 Data Handler는 출력 스트림의 일부로 응용프로그램 특정 정보에서 `type = cw_mo_cfg` 태그가 있는 하위 Meta-Object의 내용을 포함하지 않습니다. 또한 Meta-Object 이름이 상위 오브젝트의 응용프로그램 특정 정보에서 `cw_mo_cfg` 태그와 함께 포함되어야 합니다.

최상위 레벨 Business Object 머리글에서, 응용프로그램 특정 정보는 Meta Object의 이름을 다음 구문으로 지정합니다.

```
cw_mo_cfg=<meta-object attribute name>
```

Meta Object에 대한 하위 Business Object 속성에서, 응용프로그램 특정 정보는 Meta Object의 유형을 지정하고 다음 구문을 따릅니다.

```
type=cw_mo_cfg
```

응용프로그램 특정 정보에 대한 Data Handler 요구사항

응용프로그램 특정 정보에 대한 커넥터의 요구사항 외에, 각 Business Object를 처리하는 데 사용되는 특정 Data Handler의 요구사항도 고려해야 합니다. 각 Data Handler의 응용프로그램 특정 정보 요구사항에 대한 자세한 정보는 *Data Handler 안내서*를

참조하십시오.

부록 A. 커넥터 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 통틀어 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

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.

© Copyright IBM Corp. 2001, 2004

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 메소드**가 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 표의 갱신 메소드 열을 참조하십시오.

표준 등록 정보 요약

표 6은 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 6. 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>
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 서브세트입니다.	ascii7	구성요소 다시 시작	
ConcurrentEventTriggeredFlows	1 - 32,767	1	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>
ContainerManagedEvents	값이 없음 또는 JMS	값이 없음	구성요소 다시 시작	전달 전송이 JMS임
ControllerStoreAndForwardMode	true 또는 false	True	동적	저장소 디렉토리가 <REMOTE>
ControllerTraceLevel	0 - 5	0	동적	저장소 디렉토리가 <REMOTE>
DeliveryQueue		*CONNECTORNAME*/DELIVERYQUEUE	구성요소 다시 시작	JMS 전송
DeliveryTransport	MQ, IDL 또는 JMS	JMS	구성요소 다시 시작	저장소 디렉토리가 로컬이면 값은 JMS 뿐임

표 6. 표준 구성 등록 정보 요약 (계속)

등록 정보 이름	가능한 값	기본값	갱신 메소드	참고
DuplicateEventElimination	True 또는 False	False	구성요소 다시 시작	JMS 전송에만 해당: 컨테이너 관리 이벤트는 <NONE>이어야 합니다.
FaultQueue		*CONNECTORNAME*/FAULTQUEUE	구성요소 다시 시작	JMS임 전송만
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory 또는 CxCommon.Messaging.jms.SonicMQFactory 또는 Java 클래스 이름	CxCommon.Messaging.jms.IBMMQSeriesFactory	구성요소 다시 시작	JMS 전송
jms.MessageBrokerName	actoryClassName이 IBM일 경우, crossworlds.queue.manager를 사용하십시오. FactoryClassName 이 Sonic이면 localhost:2506을 사용하십시오.	crossworlds.queue.manager	구성요소 다시 시작	JMS 전송
jms.NumConcurrentRequests	양의 정수	10	구성요소 다시 시작	JMS 전송
jms.Password	올바른 암호		구성요소 다시 시작	JMS 전송
jms.UserName	올바른 이름		구성요소 다시 시작	JMS 전송
JvmMaxHeapSize	힙 크기(MB)	128M	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>
JvmMaxNativeStackSize	스택 크기(KB)	128K	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>
JvmMinHeapSize	힙 크기(MB)	1M	구성요소 다시 시작	저장소 디렉토리가 <REMOTE>
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 전송에만 해당: DuplicateEvent Elimination이 True이어야 함

표 6. 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	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 커넥터 명령 창에 입력될 때 폴하기 위해)	10000	동적	
PollQuantity	1 - 500	1	에이전트 다시 시작	JMS 전송에만 해당 : 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 메시지 브로커 및 WAS의 경우: C:\crossworlds\ 저장소의 위치
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>
RestartRetryCount	0 - 99	3	동적	
RestartRetryInterval	분별 가능한 값(분): 1 - 2147483547	1	동적	
RHF2MessageDomain	mrm, xml	mrm	구성요소 다시 시작	전달 전송이 JMS이고 WireFormat이 CwXML인 경우에만 해당됨
SourceQueue	올바른 WebSphere MQ 이름	*CONNECTORNAME*/SOURCEQUEUE	에이전트 다시 시작	전달 전송이 JMS 및 컨테이너이고 관리 이벤트가 지정되는 경우에만
SynchronousRequestQueue		*CONNECTORNAME*/ SYNCHRONOUSREQUESTQUEUE	구성요소 다시 시작	전달 전송이 JMS임
SynchronousRequestTimeout	0 - 임의의 수(밀리초)	0	구성요소 다시 시작	전달 전송이 JMS임

표 6. 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>
WsifSynchronousRequest 제한시간	0 - 임의의 숫자 (밀리초)	0	구성요소 다시 시작	WAS만 해당됨
XMLNameSpaceFormat	short, long	short	에이전트 다시 시작	WebSphere MQ 메시지 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 모니터합니다. 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 or 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

`RepositoryDirectory`가 <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 수행하십시오.

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

Parallelism을 사용할 수 있고 복수 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 다음의 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 *크기*를 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(3 세그먼트 * 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 `12`로 설정. 이 등록 정보에 대한 자세한 정보는 *UNIX용 시스템 설치 안내서*를 참조하십시오.

DuplicateEventElimination

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

또한 이 등록 정보는 `false`로 설정될 수 있습니다.

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

FaultQueue

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

기본값은 CONNECTORNAME/FAULTQUEUE입니다.

JvmMaxHeapSize

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

기본값은 128m입니다.

JvmMaxNativeStackSize

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

기본값은 128k입니다.

JvmMinHeapSize

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

기본값은 1m입니다.

jms.FactoryClassName

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

기본값은 CxCommon.Messaging.jms.IBMMQSeriesFactory입니다.

jms.MessageBrokerName

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

기본값은 crossworlds.queue.manager입니다.

jms.NumConcurrentRequests

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

기본값은 10입니다.

jms.Password

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

기본값은 없습니다.

jms.UserName

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

기본값은 없습니다.

ListenerConcurrency

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

이 등록 정보는 MQ 전송을 사용하는 커넥터에만 적용됩니다. `DeliveryTransport` 등록 정보를 `MQ`로 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 올바른 유일한 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/wicserver/infocenter

LogAtInterchangeEnd

RespositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

MaxEventCapacity

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

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

MessageFileName

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

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

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

MonitorQueue

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

기본값은 CONNECTORNAME/MONITORQUEUE입니다.

OADAutoRestartAgent

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

커넥터가 자동 및 원격 다시 시작 기능을 사용할지 여부를 지정합니다. 이 기능은 비정상 종료 이후에 커넥터를 다시 시작하거나 System Monitor로부터 원격 커넥터를 시작할때 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 적용 가능합니다.

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 적용 가능합니다.

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 *HH*는 0 - 23의 시간, *MM*은 0 - 59의 분을 나타냅니다.

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

PollFrequency

각 폴링 조치 사이의 시간 값입니다. 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:MM*이며, 여기서 *HH*는 0 - 23의 시간, *MM*은 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 메타데이터를 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.

RHF2MessageDomain

WebSphere Message Brokers 및 WAS에만 해당됩니다.

이 등록 정보는 JMS 머리글의 필드 도메인 이름 값을 구성할 수 있게 합니다. JMS 전송을 통해 데이터를 WMQI로 전송할 경우, 어댑터 프레임워크는 도메인 이름 및 고정값 mrm을 가진 JMS 머리글을 정보를 씁니다. 구성 가능한 이름을 사용하면 사용자가 WMQI 브로커가 메시지 데이터를 처리하는 방법을 추적할 수 있습니다.

샘플 머리글은 다음과 같습니다.

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

기본값은 mrm이지만 xml로 설정할 수도 있습니다. 이 등록 정보는 DeliveryTransport가 JMS로 설정되고 WireFormat이 CwXML로 설정된 경우에만 나타납니다.

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 38 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 결함 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 결함 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 `short`입니다.

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 템플리트 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.

- 49 페이지의 『Connector Configurator 개요』
- 50 페이지의 『Connector Configurator 시작』
- 52 페이지의 『커넥터 특정 등록 정보 템플리트 작성』
- 54 페이지의 『새 구성 파일 작성』
- 58 페이지의 『구성 파일 등록 정보 설정』
- 66 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 템플리트** 작성
- **커넥터 구성 파일** 작성: 설치한 각 커넥터당 하나의 구성 파일을 작성해야 합니다.
- 구성 파일에 등록 정보 설정
  커넥터 템플리트의 등록 정보에 설정된 기본값을 수정해야 할 경우가 있습니다. 또한

© Copyright IBM Corp. 2001, 2004

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, WMQI가 브로커인 경우, System Manager 내에서가 아니라 직접 Connector Configurator를 실행합니다(51 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 정보**의 경우 먼저 등록 정보를 정의한 다음 해당 값을 설정해야 합니다. 특정 어댑터의 커넥터 특정 등록 정보 템플리트를 작성하여 이를 수행합니다. 시스템에 템플리트가 이미 설정되어 있으면 이 템플리트를 사용하면 됩니다. 그렇지 않은 경우, 52 페이지의 『새 템플리트 작성』의 단계에 따라 새 템플리트를 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.

브로커가 IBM WebSphere InterChange Server일 경우에는 다음을 수행하십시오.

- **시작** > **프로그램**에서 **IBM WebSphere InterChange Server** > **IBM WebSphere Business Integration Toolset** > **개발** > **Connector Configurator**를 누르십시오.
- **파일** > **새로 작성** > **구성 파일**을 선택하십시오.
- **시스템 연결 통합 브로커** 옆의 풀다운 메뉴를 누르면, ICS 연결을 선택할 수 있습니다.

WebSphere Business Integration Adapters 및 다른 브로커가 설치된 경우에는 다음을 수행하십시오.

- **시작** > **프로그램**에서 **IBM WebSphere Business Integration Adapters** > **도구** > **Connector Configurator**를 누르십시오.
- **파일** > **새로 작성** > **커넥터 구성**을 선택하십시오.
- **시스템 연결 통합 브로커** 옆의 풀다운 메뉴를 누르면, 브로커에 따라 WMQI 또는 WAS 연결을 선택할 수 있습니다.

Connector Configurator를 별도로 실행하도록 선택하여 파일을 생성한 다음 System Manager에 연결하여 System Manager 프로젝트에 파일을 저장할 수 있습니다(57 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 **커넥터**를 강조표시하십시오.
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 템플리트 작성

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

커넥터의 커넥터 특정 등록 정보에 맞는 새 템플리트를 작성하거나, 기존 파일을 템플리트로 사용할 수 있습니다.

- 새 템플리트를 작성하려면 52 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 템플리트**를 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 이름**에서 선택사항을 강조표시하고), **다음**을 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 값**에 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 **추가**를 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 **확인**을 누르십시오.
5. **값** 패널이 **최대 길이** 및 **최대 복수 값**에서 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 편집**을 누르십시오.

종속성 설정

일반 및 **값** 탭에서 변경을 수행한 후, **다음**을 누르십시오. **종속성 - 커넥터 특정 등록 정보 템플리트** 대화 상자가 나타납니다.

종속 등록 정보는 다른 등록 정보의 값이 특정 조건에 부합하는 *경우에만* 템플리트에 포함되어 구성 파일에 사용되는 등록 정보입니다. 예를 들어, `PollQuantity`는 JMS가 전송 메커니즘이고 `DuplicateEventElimination`이 `True`로 설정되는 경우에만 템플리트에 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 템플리트에 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**를 누르십시오. Connector Configurator가 Connector Configurator를 설치한 `\bin` 디렉토리의 `\data\app`에 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 구성**을 누르십시오. **새 커넥터** 대화 상자가 나타납니다.
- **통합 브로커** 필드에서 ICS, WebSphere Message Brokers 또는 WAS 연결을 선택하십시오.
- 이 장의 뒷부분에서 설명된 대로 **새 커넥터** 창에서 나머지 필드를 완료하십시오.

다음 작업도 수행할 수 있습니다.

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

커넥터 특정 템플리트에서 구성 파일 작성

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

1. **파일 > 새로 작성 > 커넥터 구성**을 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 템플리트**

     커넥터에 맞게 설계된 템플리트의 이름을 입력하십시오. 사용 가능한 템플리트가 **템플리트 이름** 화면에 표시됩니다. 템플리트 이름 화면에서 이름을 선택할 경우 **등록 정보 템플리트 미리보기** 화면이 해당 템플리트에 정의된 커넥터 특정 등록 정보를 표시합니다.

     사용하려는 템플리트를 선택하고 **확인**을 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 > 프로젝트에**를 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 **저장**을 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 *.cfg입니다.

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

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

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

1. Connector Configurator에서 **파일 > 열기 > 파일에서**를 누르십시오.
2. **파일 커넥터 열기** 대화 상자에서 다음 파일 유형 중 하나를 선택해서 사용 가능한 파일을 보십시오.
   - 구성(*.cfg)
   - ICS 저장소(*.in, *.out)

저장소 파일이 ICS 환경에서 커넥터를 구성하는 데 사용된 경우 이 옵션을 선택하십시오. 저장소 파일은 여러 개의 커넥터 정의를 포함할 수 있으며, 모든 커넥터 정의는 사용자가 파일을 열 때 표시됩니다.

- 모든 파일(*.*)

  `*.txt` 파일이 커넥터용 어댑터 패키지로 전달되거나 확장자가 다른 정의 파일을 사용할 수 있는 경우 이 옵션을 선택하십시오.

3. 디렉토리 화면에서 적당한 커넥터 정의 파일로 이동하고 이를 선택한 후 **열기**를 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 수행하십시오.

1. **표준 등록 정보** 탭에서 브로커 유형 등록 정보에 대한 값 필드를 선택하십시오. 드롭 다운 메뉴에서 `ICS`, `WMQI` 또는 `WAS`를 선택하십시오.
2. 표준 등록 정보 탭에 선택한 브로커와 연관된 등록 정보가 표시됩니다. 지금 파일을 저장하거나 60 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 > 파일에**를 누르십시오.

   파일에 저장할 경우, 확장자로 `*.cfg`를 선택하고 파일의 올바른 위치를 선택한 후 **저장**을 누르십시오.

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

   Connector Configurator는 파일을 저장하기 전에 모든 필수 표준 등록 정보의 값이 설정되었는지 확인합니다. 필수 표준 등록 정보 값이 누락된 경우, Connector

Configurator는 유효성 검증 실패를 알리는 메시지를 표시합니다. 구성 파일을 저장하려면 해당 등록 정보에 값을 제공해야 합니다.

구성 파일 등록 정보 설정

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

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

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

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

ICS에서 실행되는 커넥터의 경우, 다음 등록 정보의 값도 필요합니다.

- 연관된 맵
- 자원
- 메시징(적용 가능할 경우)

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

표준 등록 정보는 다음과 같이 커넥터 특정 등록 정보와 다릅니다.

- 커넥터의 표준 등록 정보는 커넥터의 응용프로그램 특정 구성요소와 해당 브로커 구성 요소에서 모두 공유됩니다. 모든 커넥터에는 동일한 표준 등록 정보 세트가 있습니다. 이러한 등록 정보는 각 어댑터 안내서의 부록 A에 설명되어 있습니다. 일부는 변경할 수 있지만 모든 값을 변경할 수는 없습니다.
- 응용프로그램 특정 등록 정보는 커넥터의 응용프로그램 특정 구성요소, 즉 응용프로그램과 직접 상호작용하는 구성요소에만 적용됩니다. 각 커넥터에는 커넥터의 응용프로그램에 고유한 응용프로그램 특정 등록 정보가 있습니다. 이러한 등록 정보 중 일

부는 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 정보**에 대한 값은 구성 가능한 필드를 표시하기 위해 색상을 사용하여 코딩됩니다.

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

표준 커넥터 등록 정보 설정

표준 등록 정보의 값을 변경하려면 다음을 수행하십시오.

1. 값을 설정하려는 필드를 누르십시오.
2. 값을 입력하거나, 값이 나타나는 경우 드롭 다운 메뉴에서 선택하십시오.
3. 표준 등록 정보의 모든 값을 입력한 후, 다음 중 하나를 수행할 수 있습니다.
   - 변경사항을 취소하고 원래 값을 보존한 상태에서 Connector Configurator를 종료하려면, **파일 > 종료**를 누르고(또는 창을 닫고) 변경사항을 저장할 것인지 묻는 메시지가 표시될 때 **아니오**를 누르십시오.
   - Connector Configurator에서 다른 카테고리의 값을 입력하려면 카테고리 탭을 선택하십시오. **표준 등록 정보**(또는 다른 카테고리)에 입력하는 값은 사용자가 다음 카테고리로 이동할 때 보존됩니다. 사용자가 창을 닫을 때 모든 카테고리에 입력한 값을 전체적으로 저장할 것인지 버릴 것인지 묻는 메시지가 표시됩니다.
   - 변경된 값을 저장하려면, **파일 > 종료**를 누르고(또는 창을 닫고) 변경사항을 저장할지 묻는 메시지가 표시되면 **예**를 누르십시오. 또는 파일 메뉴나 도구 모음에서 **저장 > 파일에**를 누르십시오.

응용프로그램 특정 구성 등록 정보 설정

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

1. 격자의 왼쪽 맨 윗부분을 마우스 오른쪽 단추로 누르십시오. 팝업 메뉴 표시줄이 표시됩니다. 등록 정보를 추가하려면 **추가**를 누르십시오. 하위 등록 정보를 추가하려면 상위 행 번호를 마우스 오른쪽 단추로 누른 다음 **하위 추가**를 누르십시오.

2. 등록 정보 또는 하위 등록 정보의 값을 입력하십시오.
3. 등록 정보를 암호화하려면 **암호화** 상자를 선택하십시오.
4. 59 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 **확인**을 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. **암호화**를 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

32 페이지의 『등록 정보 값 설정 및 갱신』의 부록 *커넥터에 대한 표준 구성 등록 정보*에 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.

브로커가 ICS인 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 저장**을 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 삭제**를 누르십시오. 목록 화면에서 Business Object는 제거됩니다.
3. **파일** 메뉴에서 **프로젝트에 저장**을 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.

브로커가 WebSphere Message Broker인 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 ID**가 선택적 필드이므로 ID를 제공할 경우 고유하지 않아도 됩니다. 그러나 WebSphere MQ Integrator 및 Integrator Broker 2.1의 경우에는 고유한 **ID**를 제공해야 합니다.

브로커가 WAS인 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 저장**을 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 저장**을 누르십시오.
  4. 프로젝트를 ICS에 전개하십시오.
  5. 변경사항이 적용되도록 서버를 다시 시동하십시오.

자원(ICS)

자원 탭에서는 커넥터 에이전트가 커넥터 에이전트 병렬화를 사용하여 동시에 복수 프로세스를 처리할 수 있는지 여부와 그 범위를 판별하는 값을 설정할 수 있습니다.
모든 커넥터가 이 기능을 지원하지는 않습니다. 대개 복수 프로세스보다 복수 스레드를 사용하는 것이 더 효율적이므로 복수 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 **저장**을 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 다음의 세 가지 방식으로 저장할 수 있습니다.

- System Manager에서, 통합 구성요소 라이브러리에 `*.con` 확장자를 갖는 파일로 저장하거나
- 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 템플리트로 사용하여 다른 브로커와 함께 사용할 수 있는 새 구성 파일을 작성할 수 있습니다.

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

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

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

구성 완료

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

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

국제화된 환경에서 Connector Configurator 사용

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

Connector Configurator는 다음 위치에서 영어 이외의 문자를 지원합니다.

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

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

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

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

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

부록 C. TPI을 통한 RNIF용 IBM WebSphere DataHandler

*TPI-RNIF Data Handler*라는 TPI를 통한 RNIF(RosettaNet Implementation Framework)용 IBM WebSphere Business Integration Data Handler는 TPI(Trading Partner Interchange) Server를 사용 시 Trading Partner와 RosettaNet 문서를 교환합니다. Data Handler는 TPI Server 요구사항에 맞게 XML 메시지를 작성 및 구문 분석합니다.

이 부록은 독자가 TPI 제품, RNIF, TPI용 IBM WebSphere Adapter 및 XML용 IBM WebSphere Business Integration Data Handler와 연관된 기능 및 기본 용어에 익숙하다고 가정합니다. 이러한 제품 및 표준에 대한 자세한 정보는 다음을 참조하십시오.

- TPI Server에 있는 *관리자 안내서*
- http://www.rosettanet.org
- 이 안내서의 이전 장, *Trading Partner Interchange용 어댑터 사용자 안내서*
- IBM WebSphere Business Integration Adapters *Data Handler 안내서*

이 부록에 제공된 많은 예는 CDIX(Chemical Industry Data Exchange) 형식을 사용합니다. 이러한 형식은 TPI용 IBM WebSphere Adapter로 폭넓게 구현된 RosettaNet의 전문적인 사용법입니다.

개요

TPI Server와 TPI용 IBM WebSphere Adapter 간에 메시지가 교환될 경우, TPI Server는 MCD(Message Control Document)를 사용하여 관련된 Trading Partners와 RosettaNet 데이터 둘 다 교환되어야 함을 설명합니다. MCD는 TPI Server가 RosettaNet 문서를 프로세스하기 위해 사용하는 TPI 특정 XML 문서입니다. TPI Server는 MCD에 포함되어 있는 정보를 수신되는 RosettaNet 문서로 변환하거나 Trading Partner에 전송합니다.

TPI-RNIF Data Handler는 MCD의 작성 및 구문 분석하는 프로세스를 간략하게 합니다. 이는 다양한 RNIF 머리글, 비즈니스 신호 및 MCD에서 임베드된 PIP(Partner Interface Processes) 메시지(존재할 경우)를 인식하고 조작합니다. 또한, 여러 PIP가 사용될 경우, 사용자가 관리해야 하는 Business Object 정의의 수를 최소화합니다.

TPI-RNIF Data Handler는 MCD가 맵핑되어야 하는 적절한 Business Object 정의를 판별하고 MCD에 임베드되는 RosettaNet 페이로드의 다양한 RNIF 파트, 비즈니스 신호 및 PIP 문서를 식별합니다. 일단 이들 파트를 식별하면, TPI-RNIF Data Handler는 내부적으로 XML용 IBM WebSphere Business Integration Data Handler를 호출하며, 이로 인해 해당 파트가 적절한 Business Object 유형으로 변환됩니다. 따라서,

© Copyright IBM Corp. 2001, 2004

TPI-RNIF Data Handler와 함께 사용되는 Business Object 정의는 이 부록에 설명되어 있는 TPI-RNIF Data Handler의 이름 지정 규칙 및 XML Data Handler의 Business Object 요구사항에 부합해야 합니다.

소프트웨어 전제조건

다음 소프트웨어가 설치되어 있어야 TPI-RNIF Data Handler를 사용할 수 있습니다.

- Trading Partner Interchange용 IBM WebSphere Adapter 버전 3.4.0 이상
- XML용 IBM WebSphere Data Handler 버전 2.5.0 이상
- Trading Partner Interchange 버전 4.1.0 이상(MCD 버전 2.0 지원)

빌드하기 위해, TPI-RNIF Data Handler는 XML 클래스 파일용 WebSphere Business Integration Data Handler 및 WebSphere Business Integration Adapter Framework 라이브러리 모두를 요구합니다.

런타임 시, XML용 WebSphere Data Handler가 나타나야 합니다.

설치

TPI Server가 설치되어 있어야 합니다. RosettaNet이 각 RosettaNet Trading Partner에 대해 인바운드 및/또는 아웃바운드 프로토콜로 나열되어 있는지 확인하십시오. 또한 사용자의 Trading Partner 회사가 지원되는 XML 문서 유형으로 RosettaNet/MCD를 나열하는지 확인하십시오. TPI Server 설치 및 구성에 대한 지시사항은 TPI Server에 포함된 *관리자 안내서*를 참조하십시오.

WebSphere Business Integration Adapter 제품 설치에 대한 자세한 정보는 다음 사이트의 WebSphere Business Integration Adapters InfoCenter에서 *WebSphere Business Integration Adapters 설치 안내서*를 참조하십시오.

http://www.ibm.com/websphere/integration/wbiaadapters/infocenter

TPI-RNIF Data Handler 설치 파일은 어댑터 설치의 DataHandlers 디렉토리에 설치되어 있는 CwDataHandler.jar의 일부로 포함되어 있습니다. 설치된 디렉토리 구조는 다음과 같습니다.

\DataHandlersw\CwDataHandler.jar

어댑터 설치 도중 다음 타스크를 완료했는지 확인하십시오.

1. TPI-RNIF Data Handler에 대해 구성 정보를 가지는 Data Handler 구성 Meta Object를 사용하도록 어댑터 구성

   설치 시 포함되는 샘플 Business Object

   `MO_DataHandler_CIDX`

에 어댑터를 등록하여 이를 수행할 수 있으며 커넥터 특정 등록 정보

`DataHandlerDefaultMO=MO_DataHandler_CIDX`

를 설정할 수 있습니다.

2. Data Handler Meta Object의 TPI-RNIF Data Handler 구성

   예를 들어,

   `MO_DataHandler_CIDX`

   를 사용할 경우, `MO_DataHandler_ChemXMLConfig`에 Business Object 정의와 완전히 일치하는 `TopBOPrefix`, `BOPrefix` 등의 값이 포함되어 있는지를 확인하십시오(9 페이지의 『Trading Partner 구성 파일 작성』 참조).

3. Trading Partner를 전송 및 수신하기 위해 TPI-RNIF Data Handler를 사용하도록 어댑터 구성

   어댑터에 대해 지정된 Trading Partner 구성 파일을 통해 이를 수행할 수 있습니다.

   `MO_DataHandler_CIDX`

   를 사용할 경우, 서로 데이터를 교환하려는 Trading Partner에 대해 이 파일에서 ″text/tpi_rnif″ MIME 유형을 지정하십시오.

4. 어댑터로 전송되는 각 요청에 포함된 TPI 하위 Meta Object에

   `DocumentType=CW_RNIF`

   속성을 설정하십시오.

   이 속성을 설정하지 않은 경우, TPI Server는 문서를 MCD로 인식하지 않고 이를 대상 Trading Partner에 원시 XML로 전송합니다(23 페이지의 『하위 Meta Object 속성』 참조).

TPI RNIF Data Handler 구성

Data Handler 메타 오브젝트를 통해 TPI-RNIF Data Handler를 구성합니다. 아래 나열된 필수 등록 정보는 TPI-RNIF Data Handler에 특정합니다. 그러나 TPI-RNIF Data Handler가 내부적으로 기존 XML Data Handler 기능을 확장하므로, XML Data Handler가 필요로 하는 추가 Meta Object 등록 정보에 대한 정보는 *Data Handler 안내서*를 참조하십시오.

표 7. TPI-RNIF Data Handler는 Meta Object 등록 정보를 필요로 합니다.

등록 정보	설명	필수	기본값

위쪽 BO 접두부	제공된 서비스 내용 문서에 대해 랩퍼 Business Object를 해석합니다. 샘플 값은 다음을 포함합니다.
	`RNETOBJ`
	및
	`CIDXOBJ`
		없음(권장됨)	없음

BO 접두부	TopBOPrefix와 같이 Business Object에 대해 MCD에서 임베드된 RNIF 메시지 파트(Preamble, Service Header 등)로 변환됩니다. 샘플 값은 다음을 포함합니다.
	`RNET`
	및
	`CIDX`
		없음(권장됨)	없음

PIP 특정 신호	랩퍼 Business Object 이름 해석 시 PIP ID를 포함해야 하는 Data Handler를 위해 세미콜론으로 분리된 문서 유형을 나열하십시오.	아니오	없음
클래스 이름		예	com.crossworlds.DataHandlers.text.tpi_rnif_mcd2

Business Object 구조 이해

TPI-RNIF Data Handler는 처리하는 모든 Business Object의 최상위 레벨 구조에 대해 특정 요구사항을 가집니다. 이는 최상위 레벨 Business Object가 랩퍼로 활동하는 것과 MCD에 포함되어 있는 여러 RNIF 파트 및 MCD를 나타내는 여러 하위 오브젝트가 랩퍼 오브젝트에 포함되어 있도록 요구합니다.

랩퍼 오브젝트 구조의 예:

```
Wrapper BO
     |MessageControlDocument BO
     |Preamble BO
     |ServiceHeader BO
     |ServiceContent BO
```

랩퍼 Business Object 이름 지정

Data Handler는 인바운드 MCD를 Business Object 인스턴스로 변환할 때 사용할 적절한 Business Object 정의를 식별하기 위해 특정 이름 지정 규칙을 요구합니다. 랩퍼 오브젝트는 다음 규칙에 따라 이름 지정되어야 합니다.

```
<TopBOPrefix_>Version_ServiceContentDoctype
```

여기서, TopBOPrefix는 선택적입니다. TPI-RNIF Data Handler는 TopBOPrefix 값을 문서 유형으로 prepend하여 적절한 랩퍼를 판별합니다. 오브젝트 이름을 유도하는 데 사용되는 문서 유형은 항상 MCD(예: OrderCreate, ReceiptAcknowledgement 등) 내에 포함되어 있는 RNIF 서비스 내용 파트의 문서 유형입니다.

이름의 Version 부분은 MCD의 패키지화 프로토콜 설명에서 제공된 값에 의해 판별됩니다.

패키지화 프로토콜 설명의 예:

```
<MessageControlDocument>
     <PackagingProtocol>
          <Standard>CIDX</Standard>
          <Version>2.0</Version>
     </PackagingProtocol>
</MessageControlDocument>
```

주: 버전 이름에 마침표가 포함되어 있을 경우, Business Object 이름이 해석될 때에는 밑줄로 바뀝니다.

Business Object 이름 해석의 예:

1. TPI-RNIF Data Handler에 TopBOPrefix=CIDXOBJ를 설정하십시오.
2. RNIF 2.0 구조 OrderCreate 요청을 포함하는 MCD를 Data Handler에 전달하십시오.

Data Handler는 CIDXOBJ_2_0_OrderCreate에 MCD를 맵핑합니다.

하위 Business Object 이름 지정

하위 Business Object는 Data Handler가 제대로 처리할 수 있도록 이름 지정 규칙도 준수해야 합니다. MCD 봉투를 나타내는 하위 오브젝트도 다음 구조와 일치해야 합니다.

```
BOPrefix_Doctype
```

MCD 봉투를 나타내는 하위 오브젝트의 예:

```
CIDX_MessageControlDocument
```

서문, 서비스 머리글 및 서비스 내용을 포함하는 RNIF 메시지 파트를 나타내는 하위 오브젝트도 제대로 이름 지정되어야 하며 다음 구조와 일치해야 합니다.

```
BOPrefix_Version_Doctype
```

여기서, DocType은 RNIF 메시지 파트를 나타냅니다.
RNIF 메시지 파트를 나타내는 하위 오브젝트의 예:

```
CIDX_2_0_Preamble
```

하위 오브젝트 속성 정의

Data Handler는 사용자가 최소한 다음 네 개의 하위 오브젝트 속성을 랩퍼에 정의하도록 요구합니다.

- `MessageControlDocument`
- `Preamble`
- `ServiceHeader`
- `ServiceContent`

다음 예는 RNIF 2.0 스펙을 준수하는 RosettaNet Business Object 정의에 일치하는 필수 속성을 표시합니다. `RNETOBJ_2_0_OrderCreate`는 랩퍼 오브젝트입니다.

```
TopBOPrefix=RNETOBJ

BOPrefix=RNET
```

표 8. 유형과 일치하는 필수 속성

속성 이름	속성 유형
MessageControlDocument	RNET_MessageControlDocument
Preamble	RNET_2_0_Preamble
ServiceHeader	RNET_2_0_ServiceHeader
ServiceContent	RRNET_2_0_OrderCreate

주: 속성 정의 시, 예에 표시되지 않은 필드에 대해 기본값을 변경하지 마십시오. 이들 필드에는 카디널리티, 응용프로그램 특정 정보 등이 포함되어 있습니다.

PIP 특정 신호 사용

Data Handler를 구성하여 최상위 레벨 오브젝트 랩퍼에 PIP ID를 포함시킬 수 있습니다. 이는 선택적이지만, RosettaNet 신호 메시지가 교환 중일 때 유용할 수 있습니다. PIP 특정 신호를 포함시키려면, Data Handler Meta Object 오브젝트 내에서

```
PIPSpecificSignals
```

옵션을 선택한 후 모든 문서 유형을 지정해야 합니다. 문서 유형은 공백 없이 세미콜론으로 분리되어야 합니다. Data Handler가 MCD 문서를 수신할 때, 이는 서비스 내용 문서 유형이 옵션 PIPSpecificSignals에서 지정한 유형인지 확인합니다. 지정된 유형일 경우, Data Handler는 MCD에서 PIP ID를 추출하고 다음 구문을 기본으로 하여 랩퍼 오브젝트 이름을 찾습니다.

```
<TopBOPrefix_>Version_PIPType_ServiceContentDoctype
```

RosettaNet 신호 메시지가 교환될 경우와 포함된 PIP ID를 사용하도록 Data Handler가 구성될 경우, Data Handler는 신호가 생성된 PIP에 따라 별개의 Business Object로 일반 신호 메시지를 변환할 수 있습니다. 이는 프로세스가 선호하는 신호만 승낙할 수 있음을 의미합니다.

PIP 특정 신호 없이, Data Handler는 수신 승인 신호를 다음의 일반 랩퍼 오브젝트로 변환할 수도 있습니다.

```
CIDXOBJ_1_1_ReceiptAcknowledgement
```

PIP 특정 신호를 사용하여, Data Handler는 수신 승인 신호를 다음의 특정 랩퍼 오브젝트에 맵핑할 수 있습니다.

```
CIDXOBJ_1_1_2A1_ReceiptAcknowledgement
```

다음 예는 RNIF 1.1 스펙을 준수하는 RosettaNet Business Object 정의에 일치하는 필수 속성을 표시하며 PIP 특정 ID를 사용합니다. `PIPSpecificSignals`에 `OrderCreate` 문서 유형이 포함되어 있고, `RNETOBJ_1_1_E41_OrderCreate`가 랩퍼 오브젝트임을 가정합니다.

```
TopBOPrefix=RNETOBJ

BOPrefix=RNET
```

표 9. PIP 특정 ID를 사용하는 유형과 일치하는 필수 속성

속성 이름	속성 유형
MessageControlDocument	RNET_MessageControlDocument
Preamble	RNET_1_1_Preamble
ServiceHeader	RNET_1_1_ServiceHeader
ServiceContent	RNET_1_1_E41_OrderCreate

주: RosettaNet 표준은 OrderCreate 문서가 PIP E41 아래에 있도록 지정합니다.

전달 머리글 사용

랩퍼 오브젝트에서 다섯 번째 속성인 전달 머리글을 정의할 수 있습니다. 이 속성은 선택적입니다.

요청이 처리되는 동안, Data Handler가 기존 하위 속성을 찾아 이 이름으로 채울 경우, Data Handler는 하위 오브젝트를 일련화한 다음, 이를 아웃바운드 MCD에 추가합니다.

이벤트 공고 도중 MCD에 전달 머리글 구조가 포함되어 있는 경우, Data Handler는 랩퍼 오브젝트에 전달 머리글 속성이 포함되어 있는지 확인합니다. 포함되어 있을 경우, Data Handler는 전달 머리글 구조를 이 속성에 맵핑합니다.

Data Handler가 랩퍼 오브젝트 내에서 전달 머리글 속성을 찾지 못한 경우, 경고가 생성됩니다. 그러나 MCD에 전달 머리글 구조가 없을 경우, Data Handler는 랩퍼 오브젝트에 전달 머리글 속성이 없더라도 경고를 생성하지 않습니다.

다음 예는 RNIF 2.0 스펙을 준수하는 RosettaNet Business Object 정의에 일치하는 필수 속성을 표시합니다. 이러한 랩퍼 오브젝트는 PIP 특정 신호 및 DeliveryHeader 속성을 사용합니다. RNETOBJ_2_0_E41_OrderCreate는 랩퍼 오브젝트입니다.

```
TopBOPrefix=RNETOBJ

BOPrefix=RNET
```

표 10. 전달 머리글을 사용하는 유형과 일치하는 필수 속성

속성 이름	속성 유형
MessageControlDocument	RNET_MessageControlDocument
Preamble	RNET_2_0_Preamble
ServiceHeader	RNET_2_0_ServiceHeader
ServiceContent	RNET_2_0_E41_OrderCreate
DeliveryHeader	RNET_2_0_ServiceHeader

키 링크 속성 사용

의미있는 키를 최상위 레벨 Business Object에 포함시키는 것이 유용할 수 있습니다. 키 링크를 사용하여 TPI-RNIF Data Handler와 함께 이를 수행할 수 있습니다. 키 링크는 랩퍼에서만 정의 및 지원되며 Data Handler는 하위 Business Object 중 하나에서 랩퍼 오브젝트로 키 정보를 복사하도록 지시합니다.

Data Handler가 MCD를 수신하고 해당 랩퍼 오브젝트를 구성할 때, 이는 하위 오브젝트에 내장된 속성을 가지는 랩퍼에서 발견되는 모든 키 링크 속성을 채웁니다.

키 링크를 정의하려면 다음을 수행하십시오.

1. 랩퍼 Business Object에서 속성을 작성하십시오.

   앞에 설명된 네 개의 필수 속성에 대해 예약된 것을 제외하고 모든 이름을 사용할 수 있습니다.

2. 응용프로그램 특정 텍스트를 다음 구문에 따라 지정하십시오.

   ```
   type=key;ref=someInnerAttribute
   ```

   여기서, someInnerAttribute는 랩퍼 오브젝트와 관련된 속성으로 경로를 정의합니다.

다음 예는 TPI-RNIF Data Handler가 키 링크를 처리하는 방법을 표시합니다.

키 링크 속성 이름:

```
InstanceId
```

응용프로그램 특정 정보:

```
type=key;ref=MessageControlDocument.TLO.ProcessInfo.PIPInstanceID
```

Data Handler는 랩퍼 속성 InstanceId에 하위 속성 PIPInstanceID 값을 복사하여 이를 해결합니다.

Business Object 정의 작성

TPI-RNIF Data Handler와 호환 가능한 Business Object 구조를 작성하려면, 세 가지 유형의 Business Object 정의를 작성해야 합니다.

- MCD 봉투에 대한 Business Object 정의를 작성하십시오.

  하나의 MCD Business Object 정의만 필요합니다. XML ODA(Object Discovery Agent)를 사용하여 이를 작성하고 TPI Server 설치에서 입력한대로 MCD XSD를 사용하십시오. XML ODA에 대해 사용자가 지정한 Business Object 접두부가 TPI-RNIF Data Handler Meta Object에서 지정한 것과 동일한지 확인하십시오. *Data Handler 안내서*의 "XML ODA를 사용하여 Business Object 정의 작성"을 참조하십시오.

  대부분의 MCD Business Object 정의를 사용하여, Business Object Designer 도구로써 하나의 추가 단계를 완료해야 합니다. 이 도구를 사용하여

  `MessageControlDocument.ROOT.ManifestInfo.MessageContentInfo.Body.Body`

  속성의 응용프로그램 특정 정보를 수정함으로써 이것이 "cdata" 유형이 되도록 하십시오. 예를 들어, `mcd:Body;type=pcdata;notag`는 `mcd:Body;type=cdata;notag`로 수정되어야 합니다.

- RNIF 머리글, 비즈니스 신호 및 메시지에 대한 Business Object 정의를 작성하십시오.

  교환하려는 각 유형의 서비스 내용 메시지에 대해 하나의 Business Object 정의가 필요하며, 지원하려는 각 RNIF 버전에 대해 하나의 서문 및 서비스 머리글 Business Object가 필요합니다. BOPrefix의 RNIF 버전 번호(예: `RNET_2_0`)를 포함하십시오. XML ODA를 사용하여 이러한 Business Object 정의도 작성하십시오.

- Business Object Designer 도구를 사용하여 랩퍼 Business Object 정의를 작성하십시오.

  앞에 설명된 이름 지정 규칙에 따라 랩퍼 오브젝트의 이름을 지정한 다음, 필요할 경우 MCD 봉투 Business Object, 적절한 RNIF 머리글 Business Object 및 신호 또는 PIP 특정 신호를 추가하십시오.

  하위 Meta Object도 정의하여 TPI용 어댑터가 필요로 하는 라우팅 정보를 제공해야 합니다(23 페이지의 『하위 Meta Object 속성』 참조).

부록 D. Adapter for Trading Partner Interchange 샘플 시나리오

이 부록은 TPI(Trading Partner InterChange)용 어댑터를 사용하는 거래 파트너 간의 XML 문서 교환에 대한 이해를 돕기 위해 샘플 시나리오를 제공합니다. 시나리오는 TPI용 어댑터를 사용하여 통합 솔루션을 구현하는 두 가지 방법에 대해 다룹니다. 이는 실제 설치 또는 실제 거래 상황을 반영하지는 않으며 기타 가능한 솔루션이 있을 수 있습니다. 시나리오는 다음 구현에 대해 다룹니다.

1. IBM WebSphere InterChange Server와 함께 Adapter for Trading Partner Interchange 사용
2. IBM WebSphere MQ Integrator Broker와 함께 Adapter for Trading Partner Interchange 사용

설치 전 주의사항 및 가정

1. IBM 연결 및 사용할 통합 소프트웨어에 대해 익숙하며 TPI Server에도 익숙합니다.
2. 두 대의 컴퓨터(실제 시스템)가 사용 가능합니다.
3. 첫 번째 컴퓨터에서, IBM 통합 및 사용할 연결 소프트웨어를 설치했습니다.

   이 컴퓨터는 이 부록의 시나리오 전체에서 "IBM"으로 언급됩니다.

   WebSphere InterChange Server를 사용하여 설치하는 경우

   - WebSphere InterChange Server 4.x
   - TPI용 WebSphere Adapter
   - TPI Server
   - WebSphere MQ

   IBM WebSphere MQ Integrator Broker를 사용하여 설치하는 경우

   - WebSphere Business Integration Adapters 4.x
   - TPI용 WebSphere Adapter
   - TPI Server
   - WebSphere MQ Integrator Broker
4. 두 번째 컴퓨터에 TPI Server를 설치했습니다.

   이 컴퓨터는 이 부록의 시나리오 전체에서 "TPI"로 언급됩니다.

© Copyright IBM Corp. 2001, 2004

주: 이 문서에서 %CROSSWORLDS%가 언급될 때마다 이는 다음 중 하나를 의미합니다.

- **WebSphere MQ Integrator Broker를 사용 중일 경우:** 이는 WebSphere Business Integratation Adapter 설치를 포함하는 폴더를 의미합니다.
- **WebSphere InterChange Server를 사용 중일 경우:** 이는 IBM WebSphere InterChange Server 설치를 포함하는 폴더를 의미합니다.

모든 환경 변수 및 파일 분리자는 Windows 형식으로 지정되어 있습니다. AIX 또는 Solaris에서 실행 중일 경우에는 적절하게 변경하십시오(예: %CROSSWORLDS%\connectors를 ${CROSSWORLDS}/connectors로).

Websphere InterChange Server 사용

이러한 가상 시나리오에는 주문 정보를 교환하는 두 개의 Trading Partner 회사가 포함됩니다(**IBM Corporation(IBM)** 및 **IBM Trading Partner(IBMTP)**). **IBM**은 "Port"라는 가상의 백엔드 응용프로그램에 대한 TPI용 IBM WebSphere Adapter 및 통합용 IBM Websphere InterChange Server를 사용 중입니다. 이러한 응용프로그램은 PortConnector를 사용합니다.

데이터 플로우 요약

repos_copy 파일에는 두 개의 협업 오브젝트가 포함되어 있으며, 각각은 통합의 한 방향을 처리하도록 디자인되었습니다.

IBM에서 IBMTP로

첫 번째 협업 오브젝트는 `Port_To_TPI`라는 오브젝트를 통한 단순한 전달입니다. 이는 다음 데이터 플로우에 따라 처리됩니다.

- PortConnector에서 `TPI_Order` 오브젝트를 승인합니다.
- TPI용 어댑터로 `TPI_Order` 오브젝트를 전송합니다.
- TPI용 어댑터가 `TPI_Order`를 수신하여 다음을 수행합니다.
  1. XML DataHandler를 사용하여 이를 XML 문서로 변환합니다.
  2. XML 문서를 HTTP로 번들합니다.
  3. Trading Partner 사이트인 **IBMTP**에서 실행 중인 TPI Server로 번들된 HTTP를 전송합니다.

IBMTP에서 IBM으로

두 번째 협업 오브젝트는 `TPI_To_Port`이며 이는 다음 데이터 플로우에서 사용됩니다.

- **IBMTP**는 주문 XML 문서가 포함되어 있는 번들된 HTTP를 **IBM**에 상주하는 TPI용 어댑터로 전송합니다.
- TPI용 어댑터가 번들된 HTTP를 수신하여 다음을 수행합니다.

1. XML Data Handler를 사용하여 XML 문서를 Business Object로 변환합니다.
2. Business Object를 두 번째 협업 오브젝트인 TPI_To_Port로 전송합니다.

- TPI_To_Port는 PortConnector를 사용하여 Business Object를 포트 응용프로그램으로 전송합니다.

InterChange Server에 대한 샘플 시나리오 설치

1. 다음과 같이 **IBM Corporation** TPI Server를 설정하십시오.
   a. **IBM** 시스템에서 TPI 관리자 도구를 시작하십시오.
   b. 회사 프로파일 탭에서, %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBM\IBMCorporation_company.xml 파일로부터 회사 프로파일을 가져오십시오.
   c. 가져오기 암호를 공백으로 두십시오.
   d. 관리자 도구에서도, 파트너 탭을 선택하고 %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBM\IBMTradingPartner_partner.xml 파일에서 파트너 프로파일을 가져오십시오.

      이러한 TPI Server에 대한 회사 ID는 **IBM**입니다.
   e. 파트너 프로파일에 대해 등록 정보를 열고 아웃바운드 전송 탭 아래에서 번들된 HTTP를 탐색하십시오. URL을 "http://IBMHost:4080/exchange/IBM"에서 시스템 이름에 맞게 변경하십시오.
   f. TPHost를 IBM Trading Partner 시스템의 이름으로 갱신하십시오.
2. 다음과 같이 **IBM Trading Partner** TPI Server를 설정하십시오.
   a. **IBMTP** 시스템에서 TPI 관리자 도구를 시작하십시오.
   b. 회사 프로파일 탭에서, %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBMTP\IBMTradingPartner_company.xml 파일로부터 회사 프로파일을 가져오십시오.
   c. 관리자 도구 내에서도, 파트너 탭을 선택하고 %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBMTP\IBMCorporation_partner.xml 파일에서 파트너 프로파일을 가져오십시오.

      이러한 TPI Server에 대한 회사 ID는 **IBMTP**입니다.
   d. 파트너 프로파일에 대해 등록 정보를 열고 아웃바운드 전송 탭 아래에서 번들된 HTTP를 탐색하십시오. URL을 "http://IBMHost:4080/exchange/IBM"에서 시스템 이름에 맞게 변경하십시오.
   e. IBMHost를 IBM 시스템 이름으로 갱신하십시오.

3. 다음과 같이 Business Object를 저장소로 로드하십시오.
   a. IBM 시스템에서 WebSphere InterChange Server를 시작하십시오.
   b. Business Object Designer를 사용하여

      %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBM

      폴더에서

      Sample_TPI_Order_Objects.in

      저장소 파일을 로드하십시오.
   c. Business Object가 로드되었는지 확인하십시오.
4. 커넥터를 저장소로 로드하십시오.
   a. Connector Configurator를 사용하여

      %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBM

      폴더에서

      Sample_TPI_Order_Connectors.in

      저장소 파일을 로드하십시오.
   b. TPI Connector 및 PortConnector 정의가 로드되었는지 확인하십시오.
5. 다음과 같이 TPI Connector를 구성하십시오.
   a. System Manager를 사용하여 TPI 어댑터 정의를 선택하고 Connector Designer를 실행하십시오. 다음 어댑터 구성 등록 정보를 파일 구조에 맞도록 수정하십시오. 존재하지 않을 경우, 다음 경로 및 파일을 작성하십시오.
      - TradingPartnerConfigurationFile
      - MetaEventDir
      - DocumentOutDir
      - ArchiveProcessedDocDir
6. 다음과 같이 협업 오브젝트 및 템플리트를 저장소로 로드하십시오.
   a. System Manager를 사용하여 Sample_TPI_Order_Collaborations.in으로 레이블된 저장소 파일을 %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBM 폴더로 로드하십시오.
   b. Order_PassThrough 템플리트 정의, Port_To_TPI 및 TPI_To_Port 협업 오브젝트가 로드되었는지 확인하십시오.
7. 협업 템플리트를 컴파일하십시오. System Manager를 사용하여 협업 템플리트 폴더를 선택한 다음, 모두 컴파일을 선택하십시오.
8. WebSphere InterChange Server를 다시 시작하십시오.

서비스 호출 요청 시나리오 실행

다음 단계를 완료하여 서비스 호출 시나리오를 실행하십시오.

1. **IBM 시스템에서:**
   a. WebSphere InterChange Server를 시작한 다음, TPI용 어댑터를 시작하십시오. (그러면 TPI Server가 자동으로 시작됩니다.)
   b. Visual Test Connector 중 하나의 인스턴스를 시작하십시오.
2. **IBMTP 시스템에서:** TPI Server를 시작하십시오.
3. 다음과 같이 Port Connector를 시뮬레이트하십시오.
   a. Test Connector를 사용하여 PortConnector에 대한 프로파일을 정의하십시오.
   b. Test Connector의 파일 메뉴에서 에이전트 연결을 선택하여 에이전트를 시뮬레이트하십시오.
4. 다음과 같이 테스트 데이터를 로드하십시오.
   a. PortConnector를 시뮬레이트하는 Visual Test Connector를 사용하고 편집 메뉴에서 BO 로드를 선택하십시오.
   b. 다음 파일을 선택하여 로드하십시오.

      ```
      %CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBM\sampleOrderData.bo
      ```
5. 다음과 같이 테스트 데이터를 전송하십시오.
   a. PortConnector를 시뮬레이트하는 Visual Test Connector를 사용하고 로드된 테스트 Business Object를 선택하십시오.
   b. 요청 메뉴에서 전송을 선택하십시오.
6. XML 문서에 대해 **IBMTP 시스템**에서 XLMin 폴더를 검사하여 처리가 성공적인지 확인하십시오.

폴링 시나리오 실행

다음 단계를 완료하여 폴링 시나리오를 실행하십시오.

1. **IBM 시스템에서:**
   a. WebSphere InterChange Server를 시작한 다음, TPI용 어댑터를 시작하십시오. (그러면 TPI Server가 자동으로 시작됩니다.)
   b. Visual Test Connector 중 하나의 인스턴스를 시작하십시오.
2. **IBMTP 시스템에서:** TPI Server를 시작하십시오.
3. 다음과 같이 **IBM 시스템에서** Port Connector를 시뮬레이트하십시오.
   a. Test Connector를 사용하여 PortConnector에 대한 프로파일을 정의하십시오.
   b. Test Connector의 파일 메뉴에서 에이전트 연결을 선택하여 에이전트를 시뮬레이트하십시오.
4. 다음과 같이 샘플 데이터를 전송하십시오.

a. **IBMTP 시스템에서:** 다음 파일을 XMLout 디렉토리로 이동시키십시오.

```
%CROSSWORLDS%\connectors\TPI\samples\WebSphereICS\IBMTP\sampleOrderData.xml
```

b. **IBMTP 시스템**에서 실행 중인 TPI Server는 이러한 이벤트를 처리하여 **IBM 시스템**에서 실행 중인 TPI용 어댑터로 전달해야 합니다.

5. PortConnector를 시뮬레이트하는 Visual Test Connector를 사용하여 요청을 승인하십시오. 성공으로 응답하십시오.
6. 승인된 요청에서 데이터를 검사하여 성공적으로 처리되었는지 확인하십시오.

WebSphere MQ Integrator Broker 사용

이러한 가상 시나리오에는 주문 정보를 교환하는 두 개의 Trading Partner(″IBM Corporation″ 및 ″IBM Trading Partner″)가 포함됩니다. **IBM Corporation(IBM)**은 WebSphere MQ Integrator Broker와 함께 TPI 어댑터를 사용하여 **IBM Trading Partner(IBMTP)**라는 Trading Partner와 자체 XML 문서를 교환합니다.

이 샘플 시나리오에는 두 파트가 포함되어 있습니다. 각각은 메시지 교환의 한 방향을 처리하도록 설계되었습니다. 먼저, WebSphere MQ Integrator Broker는 TPI용 어댑터가 읽을 수 있도록 MQ Series 대기열에 이벤트를 둡니다. 어댑터는 XML DataHandler를 사용하여 메시지를 XML 문서로 변환하고 이를 **IBM Trading Partner**로 전송합니다.

두 번째 파트는 **IBM Trading Partner** 회사가 **IBM Corporation** 회사에 XML 문서를 전송하도록 요청합니다. **IBM Corporation**에 있는 TPI 어댑터는 XML 문서를 MQ Series 메시지로 변환하여 WebSphere MQ Integrator Broker가 읽을 수 있도록 MQ Series 대기열에 해당 메시지를 둡니다.

주: 이 샘플에서, 실제 WebSphere MQ Integrator Broker는 사용되지 않습니다. TPI 어댑터는 MQ Series 대기열에서 읽고 쓰기만 합니다. Visual Test Connector는 WebSphere MQ Integrator Broker를 시뮬레이트하는 데 사용됩니다.

샘플 시나리오 설치

1. 다음과 같이 **IBM Corporation** TPI Server를 설정하십시오.
   a. **IBM** 시스템에서 TPI 관리자 도구를 시작하십시오.
   b. 회사 프로파일 탭에서, `%CROSSWORLDS%\connectors\TPI\samples\WebSphereMQIntegratorBroker\IBM\IBMCorporation_company.xml` 파일로부터 회사 프로파일을 가져오십시오.
   c. 가져오기 암호를 공백으로 두십시오.
   d. 관리자 도구에서도, 파트너 탭을 누르고 `%CROSSWORLDS%\connectors\TPI\samples\`

WebSphereMQIntegratorBroker\IBM\IBMTradingPartner_partner.xml 파일에서 파트너 프로파일을 가져오십시오.

e. 이러한 TPI Server에 대한 회사 ID는 **IBM**입니다.

2. 다음과 같이 **IBM Trading Partner** TPI Server를 설정하십시오.
   a. **IBMTP** 시스템에서 TPI 관리자 도구를 시작하십시오.
   b. 회사 프로파일 탭에서, %CROSSWORLDS%\connectors\TPI\samples\WebSphereMQIntegratorBroker\IBMTP\IBMTradingPartner_company.xml 파일로부터 회사 프로파일을 가져오십시오.
   c. 관리자 도구에서도, 파트너 탭을 누르고 %CROSSWORLDS%\connectors\TPI\samples\WebSphereMQIntegratorBroker\IBMTP\IBMCorporation_partner.xml 파일에서 파트너 프로파일을 가져오십시오.
   d. 이러한 TPI Server에 대한 회사 ID는 **IBMTP**입니다.
3. 다음과 같이 MQ Series를 설정하십시오.
   a. 실행 중인 채널 초기자 및 리스너를 사용하여 MQ Series 대기열 관리자를 작성 및 시작하십시오.
   b. 다음 이름의 대기열을 작성하십시오.
      - ADMININQUEUE
      - ADMINOUTQUEUE
      - DELIVERYQUEUE
      - FAULTQUEUE
      - REQUESTQUEUE
4. %CROSSWORLDS%\connectors\TPI 디렉토리에 있는 start_TPI 파일을 여십시오. CYCLONEHOMEDIR 값을 TPI 홈 디렉토리 경로로 변경하십시오. 저장하고 파일을 닫으십시오.
5. 다음과 같이 TPI 어댑터 및 포트 커넥터 CFG 파일을 구성하십시오.
   a. %CROSSWORLDS%\connectors\TPI\samples\WebSphereMQIntegratorBroker\TPIConnector.cfg 파일을 여십시오.
   b. 파일의 다음 등록 정보는 특정 설정 시 수정될 수도 있습니다. 자세한 정보는 TPI용 IBM WebSphere Business Integration Adapter의 사용자 안내서를 참조하십시오. 등록 정보가 경로 또는 파일 이름을 나타내고 아직 존재하지 않을 경우, 이를 작성해야 합니다.
      - QueueManager
      - RepositoryDirectory
      - TradingPartnerConfigurationFile
      - MetaEventDir

- DocumentOutDir
- ArchiveProcessedDocDir

c. %CROSSWORLDS%\connectors\TPI\samples\WebSphereMQIntegratorBroker\PortConnector.cfg 파일을 여십시오. 파일의 다음 등록 정보는 특정 설정 시 수정될 수도 있습니다. 등록 정보가 경로 또는 파일 이름을 나타내고 아직 존재하지 않을 경우, 이를 작성해야 합니다.

- QueueManager
- RepositoryDirectory

6. 다음과 같이 TPI 어댑터 CFG 파일을 지정하십시오.
   - Windows의 경우: TPI 어댑터에 대해 단축 아이콘의 등록 정보를 여십시오. 대상의 마지막 인수로, 다음과 같이 "-c" + <TPIConnector.cfg 파일에 대한 전체 경로 및 파일 이름을 추가하십시오.>
     - 예: -c%CROSSWORLDS%\connectors\TPI\Samples\WebSphereMQIntegratorBroker\IBM\TPIConnector.cfg
   - UNIX의 경우: ${CROSSWORLDS}/bin/connector_manager_TPI 파일을 여십시오. 다음과 같이 "-c" + < TPIConnector.cfg 파일에 대한 전체 경로 및 파일 이름으로 AGENTCONFIG_FILE 등록 정보 값을 설정하십시오.>
     - AGENTCONFIG_FILE=-c${CROSSWORLDS}/connectors/TPI/samples/WebSphereMQIntegratorBroker/IBM/TPIConnector.cfg

요청 처리 시나리오 실행

다음 단계를 완료하여 요청 처리 시나리오를 실행하십시오.

1. **IBM 시스템에서:**
   a. TPI용 어댑터를 시작하십시오. (그러면 TPI Server가 자동으로 시작됩니다.)
   b. Visual Test Connector 중 하나의 인스턴스를 시작하십시오.
2. **IBMTP 시스템에서:** TPI Server를 시작하십시오.
3. 다음과 같이 Port Connector를 시뮬레이트하십시오.
   a. Visual Test Connector를 사용하여 PortConnector에 대한 프로파일을 정의하십시오.
   b. Test Connector 파일 메뉴에서 프로파일 열기를 선택한 다음, 다음과 같이 커넥터 구성 파일로 지정하십시오.

      ```
      %CROSSWORLDS%\connectors\TPI\samples\
      WebSphereMQIntegratorBroker\IBM\PortConnector.cfg
      ```

   c. 에이전트를 연결하십시오.
4. 다음과 같이 테스트 데이터를 로드하십시오.

a. PortConnector를 시뮬레이트하는 Visual Test Connector를 사용하고 편집 메뉴에서 BO 로드를 선택하십시오.

b. 다음 파일을 선택하여 로드하십시오.

```
%CROSSWORLDS%\connectors\TPI\samples\
WebSphereMQIntegratorBroker\IBM\sampleOrderData.bo
```

5. 다음과 같이 테스트 데이터를 전송하십시오.

a. PortConnector를 시뮬레이트하는 Visual Test Connector를 사용하고 로드된 테스트 Business Object를 선택하십시오.

b. 요청 메뉴에서 전송을 선택하십시오.

6. XML 문서에 대해 **IBMTP 시스템**에서 XLMin 폴더를 검사하여 처리가 성공적인지 확인하십시오.

폴링 시나리오 실행

다음 단계를 완료하여 폴링 시나리오를 실행하십시오.

1. **IBM 시스템에서:**

a. TPI용 어댑터를 시작하십시오. (그러면 TPI Server가 자동으로 시작됩니다.)

b. Visual Test Connector 중 하나의 인스턴스를 시작하십시오.

2. **IBMTP 시스템에서:** TPI Server를 시작하십시오.

3. 다음과 같이 Port Connector를 시뮬레이트하십시오.

a. Visual Test Connector를 사용하여 PortConnector에 대한 프로파일을 정의하십시오.

b. Test Connector 파일 메뉴에서 프로파일 열기를 선택한 다음, 다음과 같이 커넥터 구성 파일로 지정하십시오.

```
%CROSSWORLDS%\connectors\TPI\samples\
WebSphereMQIntegratorBroker\IBM\PortConnector.cfg
```

c. 에이전트를 연결하십시오.

4. 다음과 같이 샘플 데이터를 전송하십시오.

a. **IBMTP 시스템에서:** 다음 파일을 XMLout 디렉토리로 이동시키십시오.

```
%CROSSWORLDS%\connectors\TPI\samples\
WebSphereMQIntegratorBroker\IBMTP\sampleOrderData.xml
```

b. **IBMTP 시스템**에서 실행 중인 TPI Server는 이러한 이벤트를 처리하여 **IBM 시스템**에서 실행 중인 TPI용 어댑터로 전달해야 합니다.

5. PortConnector를 시뮬레이트하는 Visual Test Connector를 사용하여 요청을 승인하십시오. 성공으로 응답하십시오.

6. 승인된 요청에서 데이터를 검사하여 성공적으로 처리되었는지 확인하십시오.

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 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상으로 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 라이센스까지 부여하는 것은 아닙니다. 라이센스에 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, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통고없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 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.

© Copyright IBM Corp. 2001, 2004

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

135-270
서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터

이러한 정보는 해당 조항 및 조건에 따라(예를 들면, 사용료 지불 포함) 사용할 수 있습니다.

이 정보에 기술된 라이센스가 있는 프로그램 및 이 프로그램에 대해 사용 가능한 모든 라이센스가 있는 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이센스 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 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 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 진단, 수정 및 성능 조정에 대한 정보를 포함할 수도 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 버전 2.4.0

IBM 한글 지원에 관한 설문

IBM WebSphere Business Integration Adapters
Adapter for Trading Partner Interchange 사용자 안내서
버전 3.4.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.