



Adapter for HTTP 사용자 안내서

버전 1.0.x



Adapter for HTTP 사용자 안내서

버전 1.0.x

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 131 페이지의 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM WebSphere Business Integration Adapter for HTTP, 버전 1.0.x 및 모든 후속 릴리스와 수정판에 적용됩니다.

IBM CrossWorlds 문서에 대한 의견을 보내시려면 ibmkspe@kr.ibm.com으로 전자 우편을 보내주십시오. 여러분의 의견을 기다립니다.

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 2003. All rights reserved.

이 책의 정보

IBM^(R) WebSphere^(R) Business Integration Adapter 포트폴리오는 선도적인 e-business 기술 및 엔터프라이즈 응용프로그램의 통합 연결성을 제공합니다. 이 책에서는 Adapter for HTTP의 설치 구성 및 Business Object 개발에 대해 설명합니다.

이 책의 사용자

이 책은 IBM WebSphere 고객, 상담 개발자, WebSphere Business Integration Adapter for HTTP를 구현하려는 모든 사람을 대상으로 합니다.

이 책을 읽기 전에

여러 전제조건이 이 책에서 언급됩니다. 이 중 많은 것은 http 프로토콜에 관한 정보나 자원을 포함하는 웹 사이트에 대한 참조사항으로 구성됩니다. 또한 WebSphere Business Integration 시스템 구현과 친숙해져야 합니다. 시작하기에 좋은 곳은 더 많은 세부 문서에 대한 상호 참조가 있는 *Technical Introduction to IBM WebSphere InterChange Server*입니다.

관련 서적

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

다음 사이트에서 관련 문서를 설치할 수 있습니다.

- 일반 어댑터 정보:
 - <http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- InterChange Server와 함께 어댑터를 사용하는 경우:
 - <http://www.ibm.com/websphere/integration/wicsserver/infocenter>
 - <http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>

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

인쇄 규칙

이 문서는 다음과 같은 규칙을 사용합니다.

<code>courier</code> 글꼴	사용자가 입력한 명령 이름, 파일 이름, 정보와 같은 리터럴 값이나 시스템이 화면에 인쇄하는 정보를 표시합니다.
굵은체	처음 표시하는 새 용어를 표시합니다.
기울임	변수 이름이나 상호 참조를 표시합니다.
파란색 윤곽선	매뉴얼을 온라인으로 볼 때에만 가시적인 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 점프하려면 윤곽선 안을 클릭하십시오.
{ }	구문 행에서, 사용자가 하나 및 오직 하나만을 선택해야 하는 옵션의 설정은 물결 모양의 중괄호로 묶습니다.
[]	구문 행에서, 꺾쇠 괄호([])는 선택적 매개변수를 묶습니다.
...	구문 행에서, 타원은 이전 매개변수의 반복을 표시합니다. 예를 들어, <code>option[,...]</code> 은 쉼표로 구분된 여러 옵션을 입력할 수 있음을 의미합니다.
< >	이름 지정 규칙에서, 꺾쇠 괄호는 <code><server_name><connector_name></code> <code>tmp.log</code> 에서와 같이 서로를 구분하기 위해 이름의 각 요소를 묶습니다.
/, \	이 문서에서 백슬래시(\)는 디렉토리 경로의 규약으로 사용됩니다. UNIX 설치의 경우 백슬래시를 슬래시로 대체합니다. 모든 IBM 제품 경로 이름은 제품을 사용자 시스템에 설치한 디렉토리와 관련됩니다.
<i>ProductDir</i>	제품이 설치된 디렉토리를 표시합니다.
->	다음과 같이 메뉴에서 선택을 표시합니다. 파일 선택 -> 갱신 -> SGML 참조

목차

이 책의 정보	iii	시작 문제점	77
이 책의 사용자	iii	런타임 오류	78
이 책을 읽기 전에	iii		
관련 서적	iii	부록 A. 커넥터의 표준 구성 등록 정보	79
인쇄 규칙	iv	신규 및 삭제된 등록 정보	79
		표준 커넥터 등록 정보 구성	79
이 릴리스에서 새로 작성	vii	표준 등록 정보 요약	81
		표준 구성 등록 정보	84
제 1 장 어댑터의 개요	1	부록 B. Connector Configurator	97
Adapter for HTTP 환경	1	Connector Configurator 개요	97
전문용어	3	Connector Configurator 시작	98
Connector for HTTP의 구성요소	4	System Manager에서 Configurator 실행	99
Connector for HTTP의 구조	8	커넥터 특정 등록 정보 템플릿 작성	99
설치, 구성 및 설계 체크리스트	10	새 구성 파일 작성	102
		기존 파일 사용	103
제 2 장 설치 및 시작	13	구성 파일 완료	104
설치 작업의 개요	13	구성 파일 등록 정보 설정	105
커넥터 및 관련 파일 설치	13	구성 파일 저장	112
설치된 파일 구조	14	구성 파일 변경	112
구성 작업의 개요	15	구성 완료	113
어댑터의 다중 인스턴스 실행	16	국제화된 환경에서 Connector Configurator 사용	113
커넥터 시작	17		
커넥터 중지	19	부록 C. Adapter for HTTP 학습서	115
		학습서 정보	115
제 3 장 Business object 요구사항	21	시작하기 전에	116
Business Object 메타 데이터	21	설치 및 구성	117
커넥터 Business Object 구조	21	비동기 시나리오 실행	121
Business Object 개발	43	동기 시나리오 실행	123
		부록 D. HTTPS/SSL 구성	127
제 4 장 Connector for HTTP	45	Keystore 설정	127
커넥터 처리	45	TrustStore 설정	128
HTTP(S) 서비스	47	공용 키 인증을 위한 CSR(certificcate signing request) 생성	129
이벤트 처리	48		
요청 처리	55	주의사항	131
SSL	61	프로그래밍 인터페이스 정보	132
커넥터 구성	63	상표 및 서비스표	133
시동 시 커넥터	74		
로깅	75		
추적	75		
제 5 장 문제점 해결	77		

이 릴리스에서 새로 작성

버전 1.0.x는 Adapter for HTTP의 첫 번째 릴리스입니다.

제 1 장 어댑터의 개요

- 『Adapter for HTTP 환경』
- 3 페이지의 『전문용어』
- 4 페이지의 『Connector for HTTP의 구성요소』
- 8 페이지의 『Connector for HTTP의 구조』
- 10 페이지의 『설치, 구성 및 설계 체크리스트』

커넥터는 WebSphere Business Integration Adapter for HTTP의 런타임 구성요소입니다. 커넥터는 비즈니스가 해당 조직이나 거래 파트너가 사용할 HTTP(S) 메시지를 모으고, 공개하고 소비하도록 허용합니다. 이 문서에서 설명한 커넥터와 기타 구성요소는 HTTP 및 HTTPS 프로토콜을 통해 전달할 수 있는 메시지의 본문에서 Business Object 정보를 교환하는 데 필요한 기능을 제공합니다.

이 장에서는 WebSphere Business Integration Adapter for HTTP를 구현하는 데 사용되는 범위, 구성요소, 설계 도구 및 구조에 대해 설명합니다. 또한 이 문서에서 설명한 HTTP 구성요소를 설치 및 구성하기 위해 완료해야 하는 작업의 개요를 제공합니다. 구성요소 설치 및 구성에 대한 정보는 10 페이지의 『설치, 구성 및 설계 체크리스트』에서 참조하십시오.

주: HTTP의 어댑터는 표준 Adapter Framework API를 구현합니다. 이러한 이유로 어댑터는 프레임워크가 지원하는 통합 브로커로 조작할 수 있습니다. 그러나 어댑터가 제공하는 기능은 ICS(IBM WebSphere InterChange Server) 통합 브로커를 지정하기 위해 특별히 설계되었습니다.

Adapter for HTTP 환경

어댑터를 설치, 구성 및 사용하기 전에 요구 환경을 이해해야 합니다.

- 『브로커 호환성』
- 2 페이지의 『소프트웨어 전제조건』
- 2 페이지의 『어댑터 플랫폼』
- 2 페이지의 『표준 및 API』
- 3 페이지의 『로케일 종속 데이터』

브로커 호환성

어댑터가 사용하는 어댑터 프레임워크는 어댑터가 통신 중인 통합 브로커(또는 브로커)의 버전과 호환 가능해야 합니다. Adapter for HTTP의 1.0 버전은 다음과 같은 어댑터 프레임워크와 통합 브로커에서 지원됩니다.

- 어댑터 프레임워크: WebSphere Business Integration Adapter Framework, 버전:
 - 2.2.0
 - 2.3.0
 - 2.3.1
 - 2.4.0
- WebSphere InterChange Server 전용, 버전:
 - 4.2
 - 4.2.1
 - 4.2.2

예외사항은 릴리스 정보를 참조하십시오.

소프트웨어 전제조건

Connector for HTTP를 설치하기 전에 다음과 같은 가정과 소프트웨어 요구사항을 검토하십시오.

- HTTPS/SSL을 사용하려는 경우, keystore 및 truststore를 작성하려면 사용자 자신의 타사 소프트웨어가 필요합니다.

어댑터 플랫폼

어댑터는 다음 플랫폼*운영 체제)에서 실행합니다.

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

표준 및 API

다양한 표준 및 기술로 네트워크에서 기능에 대한 액세스를 제공합니다.

어댑터에서 사용하는 표준은 다음과 같습니다.

- HTTP 1.0

어댑터에서 사용하는 API는 다음과 같습니다.

- IBM JSSE 1.0.2

사용자 구성에 따라, 추가 소프트웨어의 설치가 필요할 수도 있습니다. 아래 섹션에서는 이러한 가능성에 대해 설명합니다.

SSL

SSL을 사용하려고 계획하는 경우 keystore, 인증 및 키 생성을 관리하려면 타사 소프트웨어를 사용해야 합니다. keystore, 인증서 또는 키 생성을 설정하기 위한 어떤 도구

도 제공되지 않습니다. 자체 서명 인증서를 작성하고 keystore를 관리하기 위해 keytool(IBM JRE에서 제공) 사용을 선택할 수 있습니다. 자세한 정보는 61 페이지의 『SSL』에서 참조하십시오.

로케일 종속 데이터

커넥터를 글로벌화하여 2바이트 문자 세트를 지원할 수 있습니다. 한 문자 코드를 사용하는 위치에서 다른 코드 세트를 사용하는 위치로 커넥터가 데이터를 전송하면, 데이터의 의미를 보존하기 위해 문자 변환을 수행합니다.

JVM(Java Virtual Machine)의 Java 런타임 환경은 Unicode 문자 코드 세트에 데이터를 표시합니다. Unicode는 가장 잘 알려진 문자 코드 세트(단일 바이트 및 다중바이트)에 문자의 인코딩을 포함합니다. WebSphere Business Integration 시스템의 대부분의 구성요소는 Java로 쓰여집니다. 그러므로 대부분의 통합 구성요소 간에 데이터가 전송되면 문자 변환이 필요하지 않습니다.

주: 커넥터가 국제화되지 않았습니다. 이것은 추적 및 로그 메시지가 변환되지 않았음을 의미합니다.

Connector for HTTP

이 섹션에서는 글로벌화와 커넥터에 대해 설명합니다.

이벤트 공고: 커넥터는 이벤트 공고를 위해 플러그 가능한 프로토콜 리스너를 사용합니다. 프로토콜 리스너는 전송에서 메시지를 추출하며 메시지 메타 데이터에 지정된 Data Handler를 호출합니다. 리스너 처리에 대한 자세한 정보는 49 페이지의 『HTTP 및 HTTPS 프로토콜 리스너 처리』에서 참조하십시오.

요청 처리: 커넥터는 요청 처리를 위해 플러그 가능한 HTTP-HTTPS 프로토콜 핸들러 프레임워크를 사용합니다. 프로토콜 핸들러는 Data Handler를 호출합니다. 자세한 정보는 56 페이지의 『HTTP-HTTPS 프로토콜 핸들러 처리』에서 참조하십시오.

Data Handler

Data Handler를 사용하기 위해 HTTP 어댑터를 구성할 수 있습니다. Data Handler 구성의 개요에 대해서는 11 페이지의 『Data Handler 구성』에서 참조하십시오.

전문용어

다음 용어가 이 안내서에서 사용됩니다.

- **ASI(Application-Specific Information, 응용프로그램 특정 정보)**는 특정 응용프로그램이나 기술에 맞게 조정된 코드입니다. ASI는 Business Object 정의의 속성 레벨 및 Business Object 레벨에 모두 존재합니다.
- **ASBO(Application-Specific Business Object, 응용프로그램 특정 Business Object)** ASI를 가질 수 있는 Business Object.

- **BO(Business Object)** 비즈니스 엔티티(예: 고객) 및 데이터의 조치(예: 작성 또는 갱신 조작)를 나타내는 속성 세트. IBM WebSphere 시스템의 구성요소는 정보 및 트리거 조치를 교환하기 위해 비즈니스 오브젝트를 사용합니다.
- **Content-Type** 유형/부속유형 및 선택적 매개변수를 포함하는 HTTP 프로토콜 머리글. 예를 들어, Content-Type 값 `text/xml; charset=ISO-8859-1`, `text/xml`은 유형/부속 유형이며 `charset=ISO-8859-1`은 선택적 Charset 매개변수입니다.
- **ContentType**은 Content-Type 머리글 값의 유형/부속 유형 부분만을 나타냅니다. 예를 들어, Content-Type 값 `text/xml; charset=ISO-8859-1`, `text/xml`은 이 문서에서 ContentType으로 나타납니다.
- **GBO(Generic Business Object, 일반 Business Object)** ASI가 없으며 모든 응용프로그램에 구속되지 않는 Business Object.
- **MO_DataHandler_Default** 예를 들어 설명할 Data Handler를 결정하기 위해 커넥터 에이전트가 사용하는 Data Handler Meta Object. 이것은 커넥터의 `DataHandlerMetaObjectName` 구성 등록 정보에서 지정됩니다.
- **프로토콜 Config MO** 요청 처리 중 HTTP-HTTPS 프로토콜 핸들러는 프로토콜 Config MO를 사용하여 목적지를 결정합니다. 이벤트 처리 중 협업을 노출하려는 경우 커넥터는 프로토콜 Config MO를 사용하여 HTTP 또는 HTTPS 프로토콜 리스너에서 협업으로 메시지 머리글 정보를 전달합니다.
- **최상위 레벨 Business Object** 최상위 레벨 Business Object에는 요청, 응답(선택적) 및 하나 이상의 결합(선택적) Business Object가 들어 있습니다. TLO는 이벤트 처리 및 요청 처리를 위해 커넥터에 의해 사용됩니다.

Connector for HTTP의 구성요소

그림 1에서는 프로토콜 핸들러 및 리스너 프레임워크를 포함하여 Connector for HTTP를 예증합니다.

주: Adapter for HTTP는 XML Data Handler의 제한된 사용 라이선스와 함께 제공됩니다. 그러나 어댑터는 XML Data Handler가 기능하도록 요구하지 않습니다.

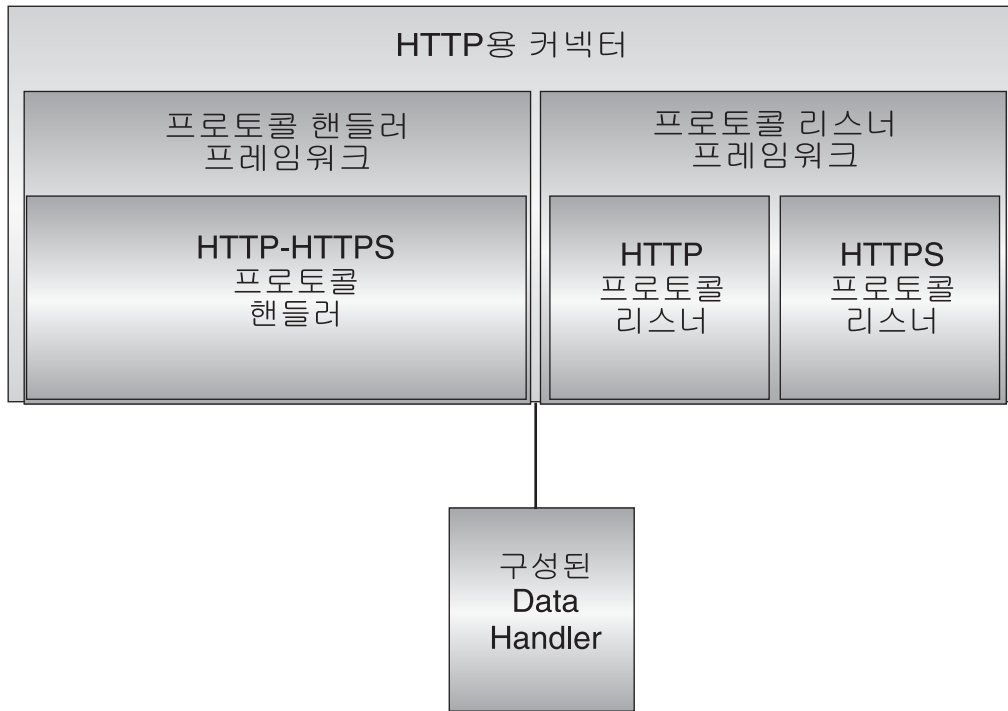


그림 1. Connector for HTTP

다음 구성요소는 인터넷에서 데이터 교환이 가능하도록 상호작용합니다.

- 구성된 Data Handler, 프로토콜 리스너 및 핸들러를 포함하는 Connector for HTTP
- HTTP 가능 협업
- Business Object 및 HTTP(S) 메시지
- WebSphere Business Integration InterChange Server

Connector for HTTP

요청 처리 중 커넥터는 메시지를 요청하고 지정된 목적지로 전달하기 위해 Business Object를 변환하여 협업 서비스 호출에 응답합니다. 선택적으로(동기 요청 처리의 경우) 커넥터는 응답 메시지를 응답 Business Object로 전달하고 협업으로 리턴합니다.

이벤트 처리 중, 커넥터는 클라이언트에서 요청 메시지를 요청 Business Object로 변환하고 처리를 위한 협업으로 전달함으로써 처리합니다. 커넥터는 선택적으로 응답 메시지로 변환되어 클라이언트로 리턴된 협업에서 선택적으로 응답 Business Object를 수신합니다.

자세한 정보는 45 페이지의 제 4 장 『Connector for HTTP』에서 참조하십시오.

주: 이 문서에서 커넥터의 언급은 다르게 지정된 경우를 제외하고 Connector for HTTP에 대한 참조사항입니다.

프로토콜 리스너 및 핸들러

커넥터는 다음 프로토콜 리스너 및 핸들러를 포함합니다.

- HTTP 프로토콜 리스너
- HTTPS 프로토콜 리스너
- HTTP-HTTPS 프로토콜 핸들러

프로토콜 리스너는 HTTP 또는 HTTPS 형식으로 내부 또는 외부 클라이언트에서 이벤트를 발견합니다. 협업으로 처리를 요구하는 이벤트의 커넥터에 알립니다. 프로토콜 리스너는 프로토콜 구성 오브젝트에 임베드된 Business Object 레벨 및 속성 레벨 ASI, 커넥터 등록 정보 및 변환 규칙을 읽어 트랜잭션의 협업, Data Handler, 처리 모드(동기/비동기) 및 전송 특정 측면을 결정합니다. 프로토콜 리스너 처리의 세부 계정은 48 페이지의 『프로토콜 리스너』에서 참조하십시오.

프로토콜 핸들러는 협업 대신 HTTP 또는 HTTPS 형식으로 HTTP 서비스를 호출합니다. HTTP(S) 프로토콜 핸들러는 프로토콜 구성 오브젝트에 임베드된 TLO ASI 및 변환 규칙을 읽어 (동기 또는 비동기로) 요청을 처리하는 방법, 메시지의 Business Object 변환 및 역변환에 사용하는 Data Handler를 결정하며, (요청 Business Object 오브젝트 프로토콜 Config MO의 목적지 속성으로) 목적지를 결정합니다. 동기 트랜잭션의 경우 프로토콜 핸들러는 응답 메시지를 처리하여 응답 Business Object로 변환하며 다시 협업으로 전달합니다.

프로토콜 핸들러에 대한 자세한 정보는 56 페이지의 『프로토콜 핸들』을 참조하십시오.

Data Handler

Data Handler를 사용하기 위해 HTTP 어댑터를 구성할 수 있습니다. 예증을 위해 이 문서는 주로 text/xml MIME 유형 및 XML Data Handler를 참조합니다.

구성된 Data Handler는 Business Object를 메시지로 변환하며 그 역으로도 변환합니다. 자세한 정보는 HTTP 어댑터에서 사용하려는 Data Handler의 문서를 참조하십시오.

Object Discovery Agent

ODA(Object Discovery Agent)가 있는 경우 Data Handler를 사용하려는 경우 해당 ODA를 사용하여 Business Object를 생성할 수 있습니다. 예를 들어, 사용자 요구사항이 XML 인코딩을 포함하며 XML Data Handler로 어댑터를 구성하는 경우, XML ODA를 사용하여 Business Object를 작성하고 수정할 수 있습니다.

커넥터 전개

Connector for HTTP를 전개하는 두 가지 방식이 있습니다.

- HTTP 또는 HTTPS 형식으로 비즈니스 프로세스가 통신하는 엔터프라이즈에서 인트라넷 기반 솔루션으로 방화벽 뒤에서(그림 2 참조).

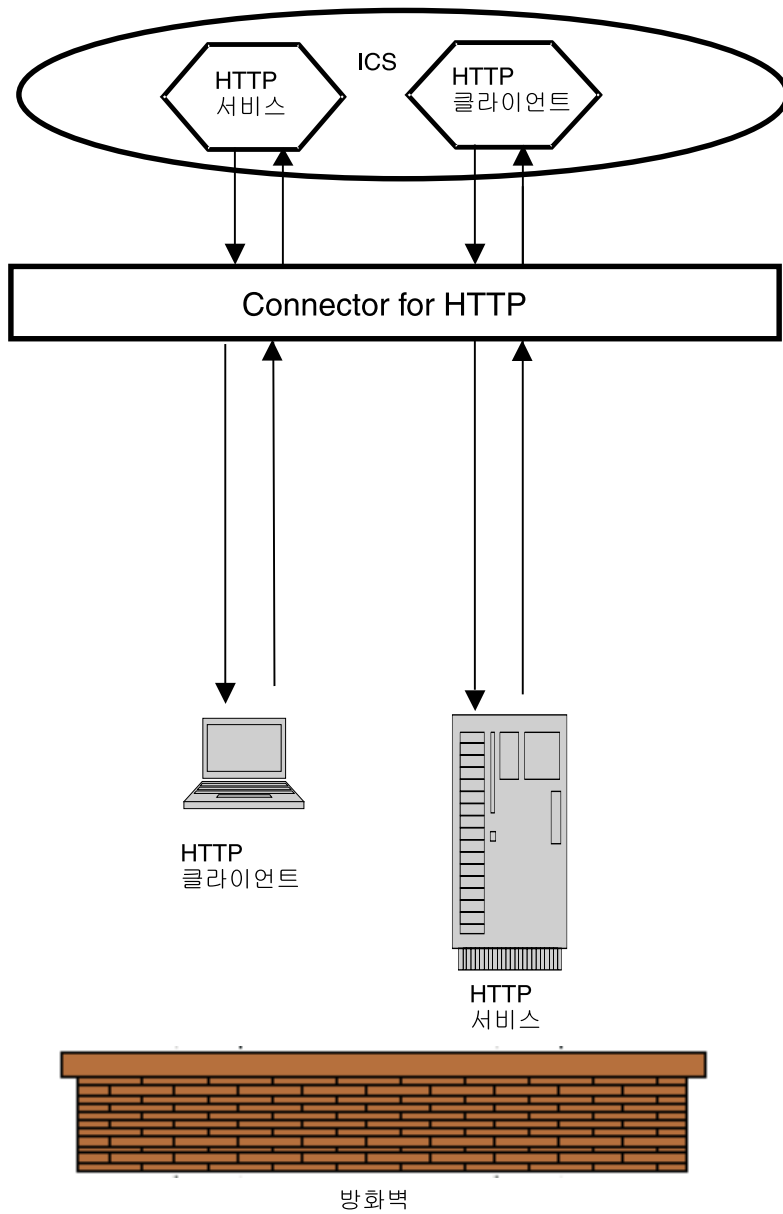


그림 2. 인트라넷 솔루션으로 HTTP 어댑터

- 엔터프라이즈 외부의 통신을 처리, 필터, 그외 관리하기 위해 프론트 엔드 또는 게이트웨이 서버가 있는 방화벽 뒤에서.

주: Connector for HTTP는 외부 클라이언트 사이의 수신 또는 전송 메시지를 관리하기 위한 게이트웨이나 프론트 엔드를 포함하지 않습니다. 반드시 사용자 자신의 계

이트웨이를 구성하고 전개해야 합니다. 커넥터는 엔터프라이즈에서만 전개되어야 하며, DMZ에서나 방화벽의 외부에서는 전개되지 않습니다.

Connector for HTTP의 구조

상위 레벨에서 구성요소의 구조를 예시하기 위해, 이 섹션에서는 두 가지 데이터 플로우에 대해 설명합니다. 그림 3에서는 두 가지 시나리오를 예시합니다. 이 두 가지 시나리오가 아래 설명됩니다.

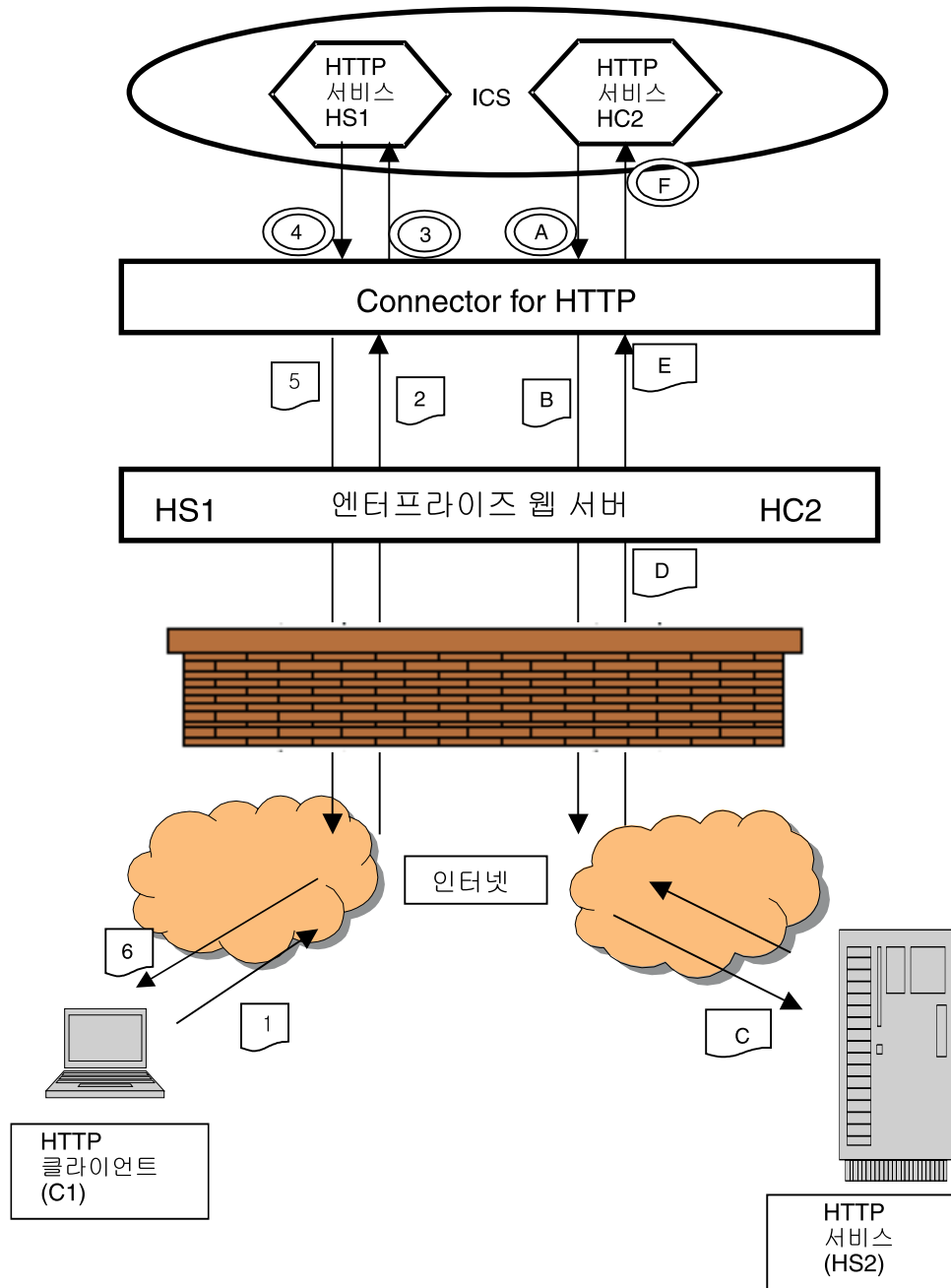


그림 3. HTTP 메시지의 플로우

요청 처리는 협업이 커넥터에 대한 서비스 호출 요청을 작성할 때 발생하는 이벤트 순서를 예시합니다. 이 시나리오에서 협업은 클라이언트의 역할을 하며 요청을 서버로 전송합니다.

- A 협업은 Data Handler를 호출하여 Business Object를 요청 메시지로 변환하는 커넥터로 서비스 호출 요청을 전송합니다.
- B 커넥터는 요청 메시지를 전송하여 엔터프라이즈 웹 서버의 URL을 호출합니다.

- C 엔터프라이즈 웹 서버는 HTTP 서버(HS2)의 URL을 호출합니다.
- D HTTP 서버 HS2는 요청을 처리하여 응답을 리턴합니다. 응답은 동일한 연결 파트로 리턴됩니다.
- E 엔터프라이즈 웹 서버는 응답 메시지를 어댑터로 리턴합니다.
- F 커넥터는 응답(또는 결합) 메시지를 수신하고, Data Handler를 호출하여 메시지를 Business Object로 변환하고 협업으로 리턴합니다.

이벤트 처리에서는 HTTP가 협업을 호출했을 때 발생하는 이벤트의 순서를 예시합니다. 이 시나리오에서 협업은 서버의 역할을 하며 클라이언트, 외부 또는 내부에서 요청을 승인하며 필요 시 응답합니다.

- 1 HTTP 클라이언트(C1)는 요청 메시지를 목적지로 전송합니다. -- 협업.
- 2 HTTP 클라이언트가 외부에 있으면, 게이트웨이는 메시지를 수신하여 커넥터로 라우트합니다.
- 3 커넥터는 Business Object로의 변환을 위해 Data Handler로 메시지를 전송합니다. 커넥터는 협업을 호출합니다.
- 4 협업은 응답(또는 결합) Business Object를 리턴합니다.
- 5 커넥터는 Data Handler를 호출하여 응답(또는 결합) Business Object를 응답 메시지로 변환합니다. 커넥터는 응답을 게이트웨이로 리턴합니다.
- 6 클라이언트가 외부에 있으면, 게이트웨이는 응답 메시지를 HTTP 클라이언트(C1)로 라우트합니다.

설치, 구성 및 설계 체크리스트

이 섹션에서는 HTTP 솔루션을 설치, 구성 및 설계하기 위해 수행해야 하는 작업을 요약합니다. 각 섹션에서는 작업에 대해 간단하게 설명한 후 작업을 수행하는 방법에 대해 설명하거나 배경 정보를 제공하는 섹션에 대한 링크를 이 문서에서 (그리고 관련 문서에 대한 상호 참조를) 제공합니다.

어댑터 설치

설치해야 하는 내용과 위치에 대한 설명은 13 페이지의 제 2 장 『설치 및 시작』에서 참조하십시오.

커넥터 등록 정보 구성

커넥터는 두 가지 유형의 구성 등록 정보를 갖습니다. 표준 등록 정보 및 커넥터 특정 구성 등록 정보입니다. 이 등록 정보 중 일부는 변경할 필요가 없는 기본값을 갖습니다. 커넥터를 실행하기 전에 이 등록 정보의 일부 값을 설정할 필요가 있습니다. 자세한 정보는 45 페이지의 제 4 장 『Connector for HTTP』를 참조하십시오.

프로토콜 핸들러 및 리스너 구성

이 구성요소의 작동을 제어하는 커넥터 구성 등록 정보에 값을 지정할 때 프로토콜 핸들러 및 리스너를 구성합니다. 자세한 정보는 45 페이지의 제 4 장 『Connector for HTTP』를 참조하십시오.

Business Object 작성 또는 수정

Connector for HTTP에서 사용 중인 Data Handler에 따라 ODA를 사용 할 수 있습니다. ODA는 Business Object를 작성하고 수정하는 프로세스를 자동화합니다. 그렇지 않으면 Business Object Designer를 사용하여 수동으로 Business Object를 작성하거나 수정할 수 있습니다. 자세한 정보는 사용 중인 Data Handler의 문서와 *Business Object Development Guide*를 참조하십시오.

Data Handler 구성

제품 파일을 설치한 후에 그러나 시작하기 전에 Data Handler Meta Object를 구성합니다. 커넥터 특정 구성 등록 정보, DataHandlerMetaObjectName을 지정하여 시작합니다. Data Handler가 구성 등록 정보를 검색하기 위해 사용하는 최상위 레벨 Meta Object(MO_DataHandler_Default)의 이름을 지정합니다. 그런 다음 사용 중인 Data Handler에서 필요로 하는 추가 구성 단계를 따르십시오. Mime 유형 및 TLO 속성을 사용하여 선택적으로 Data Handler를 지정할 수 있습니다. 자세한 정보는 24 페이지의 표 5에서 참조하십시오.

Data Handler 구성에 대한 자세한 정보는 64 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

제 2 장 설치 및 시작

- 『설치 작업의 개요』
- 『커넥터 및 관련 파일 설치』
- 15 페이지의 『구성 작업의 개요』
- 16 페이지의 『어댑터의 다중 인스턴스 실행』
- 17 페이지의 『커넥터 시작』
- 19 페이지의 『커넥터 중지』

이 장에서는 Connector for HTTP의 구현을 위해 구성요소를 설치하는 방법에 대해 설명합니다. 일반적으로 ICS 시스템을 설치하는 정보는 사용자 플랫폼에 적절한 시스템 설치 안내서를 참조하십시오.

설치 작업의 개요

브로커 호환성, 어댑터 프레임워크, 소프트웨어 전제조건, 종속성 및 표준 및 API에 대한 정보는 1 페이지의 『Adapter for HTTP 환경』을 참조하십시오.

Connector for HTTP를 설치하려면, 다음 작업을 수행해야 합니다.

ICS 설치

시스템 설치 및 ICS 시작을 포함하는 이 작업은 시스템 설치 안내서에서 설명됩니다. ICS, 버전 4.2를 설치해야 합니다.

파일을 저장소에 로드하려면 *Implementation Guide for WebSphere InterChange Server* 를 참고하십시오.

커넥터 및 관련 파일 설치

이 작업은 시스템의 소프트웨어 패키지에서 커넥터(및 관련 구성요소)의 파일을 설치하는 것을 포함합니다. 『커넥터 및 관련 파일 설치』를 참조하십시오.

커넥터 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>

설치된 파일 구조

이 섹션의 테이블은 설치된 파일 구조를 표시합니다.

Windows 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

유틸리티는 커넥터를 설치하며 커넥터의 단축 아이콘을 시작 메뉴에 추가합니다.

표 1에서는 커넥터가 사용하는 Windows 파일 구조에 대해 설명하며, 설치 프로그램을 통해 커넥터를 설치하도록 선택할 때 자동으로 설치되는 파일을 표시합니다.

표 1. 어댑터의 설치된 Windows 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors\HTTP\BIA_HTTP.jar	WebSphere Business Integration 어댑터 jar 파일
connectors\HTTP\start_HTTP.bat	커넥터의 시작 파일
bin\Data\App\HTTPConnectorTemplate	Connector for HTTP 템플릿
connectors\HTTP\dependencies\ibmjse.jar	IBM의 JSSE(Java Secure Socket Extension) API 라이선스
connectors\HTTP\dependencies\IBMReadme.txt	샘플의 저장소 파일
connectors\HTTP\samples\WebSphereICS\HTTPSample.jar	Test Connector의 샘플 (동기) Business Object
connectors\HTTP\samples\WebSphereICS\CLIENT_SYNC_TLO_OrderStatus.bo	Test Connector의 샘플 (비동기) Business Object
connectors\HTTP\samples\WebSphereICS\CLIENT_ASYNC_TLO_Order.bo	샘플 (동기) Business Object
connectors\HTTP\samples\WebSphereICS\SERVICE_SYNC_TLO_OrderStatus.bo	커넥터 메시지 파일
connectors\messages\HTTPConnector.txt	

주: 모든 제품 경로 이름은 제품을 사용자 시스템에 설치한 디렉토리와 관련됩니다.

UNIX 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

표 2에서는 커넥터가 사용하는 UNIX 파일 구조에 대해 설명하며, 설치 프로그램을 통해 커넥터를 설치하도록 선택할 때 자동으로 설치된 파일을 표시합니다.

표 2. 어댑터의 설치된 UNIX 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors/HTTP/BIA_HTTP.jar	WebSphere Business Integration 어댑터 jar 파일
connectors/HTTP/start_HTTP.sh	커넥터의 시작 파일
bin/Data/App/HTTPConnectorTemplate	Connector for HTTP 템플릿
connectors/HTTP/dependencies/ibmjse.jar	IBM의 JSSE(Java Secure Socket Extension) API 라이선스
connectors/HTTP/dependencies/IBMReadme.txt	샘플의 저장소 파일
connectors/HTTP/samples/WebSphereICS/HTTPSample.jar	Test Connector의 샘플 (동기) Business Object
connectors/HTTP/samples/WebSphereICS/CLIENT_SYNC_TLO_OrderStatus.bo	Test Connector의 샘플 (비동기) Business Object
connectors/HTTP/samples/WebSphereICS/CLIENT_ASYNC_TLO_Order.bo	샘플 (동기) Business Object
connectors/HTTP/samples/WebSphereICS/SERVICE_SYNC_TLO_OrderStatus.bo	커넥터 메시지 파일
connectors/messages/HTTPConnector.txt	

주: 모든 제품 경로 이름은 제품을 사용자 시스템에 설치한 디렉토리와 관련됩니다.

구성 작업의 개요

설치 후 그리고 시작 전에 다음과 같이 구성요소를 구성해야 합니다.

커넥터 구성

이 작업은 커넥터 설정 및 구성을 포함합니다. 63 페이지의 『커넥터 구성』을 참조하십시오.

Business Object 구성

Business Object를 구성하는 단계는 제품 세트를 구현하기 위해 선택하는 방법에 의해 좌우됩니다.

- 요청 처리 다음에 해당하는 Business Object를 작성해야 합니다.
 - 요청 메시지 전송
 - 결함을 포함하는 가능한 각 응답

자세한 정보는 21 페이지의 제 3 장 『Business object 요구사항』을 검토하십시오.

- 이벤트 처리 TLO Business Object를 사용합니다. 자세한 정보는 21 페이지의 제 3 장 『Business object 요구사항』을 검토하십시오.

Business Object Designer를 사용하거나 Business Object Designer를 작성하는 프로세스를 자동화하는 Data Handler, ODA에 따라 수동으로 Business Object를 작성할 수 있습니다. 자세한 정보는 Data Handler 문서를 참조하십시오.

Data Handler 구성

커넥터 특정 구성 등록 정보, DataHandlerMetaObjectName을 지정하여 Data Handler를 구성합니다. Data Handler가 구성 등록 정보를 검색하기 위해 사용하는 최상위 레벨 Meta Object(MO_DataHandler_Default)의 이름을 지정합니다. 그런 다음 사용 중인 Data Handler에서 필요로 하는 추가 구성 단계를 따르십시오.

Mime 유형 및 TLO 속성을 사용하여 선택적으로 Data Handler를 지정할 수 있습니다. 자세한 정보는 24 페이지의 표 5에서 참조하십시오.

Data Handler 구성에 대한 자세한 정보는 64 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

협업 구성

요청 또는 이벤트 처리의 협업을 사용하려면 다음 문서를 참조하십시오.

- *IBM Websphere InterChange Server Collaboration Development Guide*
- *IBM Websphere InterChange Server Map Development Guide*

어댑터의 다중 인스턴스 실행

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

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

새 디렉토리 작성

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

```
ProductDir\connectors\connectorInstance
```

여기서 connectorInstance는 고유하게 커넥터 인스턴스를 식별합니다.

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

```
ProductDir\repository\connectorInstance
```

Business Object 정의 작성

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

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

```
ProductDir\repository\initialConnectorInstance
```

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

커넥터 정의 작성

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

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

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

시작 스크립트 작성

시작 스크립트를 작성하려면 다음을 수행하십시오.

1. 초기 커넥터의 시작 스크립트를 복사한 다음 시작 스크립트 이름에 커넥터 디렉토리의 이름이 포함되도록 이름을 지정하십시오.
dirname
2. 시작 스크립트를 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

중요: 이 장의 앞부분에서 언급한 대로, 적절한 조작용을 보장하기 위해 설치 이후에 커넥터를 시작하기 전에 커넥터, Business Object, Data Handler meta-object 및 협업을 구성해야 합니다. 이 작업의 요약은 15 페이지의 『구성 작업의 개요』에서 참조하십시오. 또한 커넥터 폴링이 사용 가능해야 합니다(커넥터 폴링은 기본값으로 사용 가능합니다).

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

ProductDir\connectors\connName

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

표 3. 커넥터의 시작 스크립트

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

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

- Windows 시스템의 경우, 시작 메뉴에서 다음을 수행하십시오.

프로그램 > **IBM WebSphere Business Integration Adapters** > 어댑터 > 커넥터를 선택하십시오. 기본적으로 프로그램 이름은 “IBM WebSphere Business Integration Adapters”입니다. 그러나 이 이름을 사용자 정의할 수 있습니다. 대안으로, 커넥터에 대한 데스크탑 단축 아이콘을 작성할 수 있습니다.

- 명령행에서 커넥터를 시작할 경우에는 다음을 수행하십시오.

- Windows 시스템의 경우:

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

- UNIX 기반 시스템의 경우:

```
connector_manager_connName -start
```

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

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

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

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

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

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

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

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

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

- WebSphere InterChange Server에 대한 자세한 정보는 *System Administration Guide*를 참조하십시오.

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

커넥터 중지

커넥터를 중지하는 방법은 다음과 같이 커넥터를 시작한 방법에 따라 다릅니다.

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

```
connector_manager_connName -stop
```

여기서 *connName*은 커넥터의 이름입니다.

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

제 3 장 Business object 요구사항

- 『Business Object 메타 데이터』
- 『커넥터 Business Object 구조』
- 22 페이지의 『동기 이벤트 처리 TLO』
- 29 페이지의 『비동기 이벤트 처리 TLO』
- 32 페이지의 『동기 요청 처리 TLO』
- 32 페이지의 『동기 요청 처리 TLO』
- 40 페이지의 『비동기 요청 처리 TLO』
- 43 페이지의 『Business Object 개발』

이 장에서는 커넥터 Business Object의 구조, 요구사항 및 속성에 대해 설명합니다.

Business Object 메타 데이터

HTTP의 커넥터는 메타 데이터 방식 커넥터입니다. Business Object에서 메타 데이터는 Business Object 정의에 저장되어 커넥터가 응용프로그램과 상호작용하도록 돕는 응용프로그램에 대한 데이터입니다. 메타 데이터 방식 커넥터는 커넥터에서 하드 코딩된 명령어보다는 Business Object 정의에서 인코딩된 메타 데이터에 근거하여 지원하는 각 Business Object를 핸들합니다.

Business Object 메타 데이터는 Business Object의 구조, 속성 등록 정보의 설정 및 응용프로그램 특정 정보의 내용을 포함합니다. 커넥터는 메타 데이터 방식이므로, 커넥터 코드에 대한 수정을 요구하지 않고 새로 수정된 Business Object를 핸들할 수 있습니다. 그러나 커넥터 구성 Data Handler는 Business Object의 구조, 오브젝트 카디널리티, 응용프로그램 특정 텍스트 형식 및 Business Object의 데이터베이스 표시에 대한 가정을 작성합니다. 그러므로 http의 Business Object를 작성하거나 수정하면, 사용자 수정은 커넥터가 따르도록 정해진 규칙과 일치하거나 커넥터가 새로 작성하거나 수정한 Business Object를 제대로 처리할 수 없습니다.

커넥터 Business Object 구조

커넥터는 다음 Business Object를 처리할 수 있습니다.

- **TLO** 최상위 레벨 Business Object(TLO)에는 요청 Business Object와 선택적으로 응답 및 결합 Business Object를 포함합니다. 이 하위 오브젝트는 내용 데이터와 선택적으로 프로토콜 Config MO를 포함합니다. 또한 Data Handler 특정 오브젝트이기도 합니다. 예를 들어 XML Data Handler를 사용하려는 경우 요청 하위는 XML Data Handler가 이해할 수 있는 Business Object입니다. 요청 대 이벤트 처

리에 대한 응용프로그램 특정 정보, 속성 및 요구사항은 물론 TLO, 요청, 응답 및 결합 오브젝트가 아래 섹션에서 설명되며 예시됩니다.

주: TLO은 요청 처리 및 이벤트 처리를 위해 사용됩니다.

동기 이벤트 처리 TLO

이벤트 처리의 경우 커넥터는 두 가지 종류의 TLO(동기 및 비동기)를 허용합니다. 이 섹션에서는 동기 이벤트 처리 TLO에 대해 설명합니다.

그림 4에서는 동기 이벤트 처리를 위한 Business Object를 표시합니다. 요청 및 응답 오브젝트가 필요하며, 결합 오브젝트가 선택적입니다.

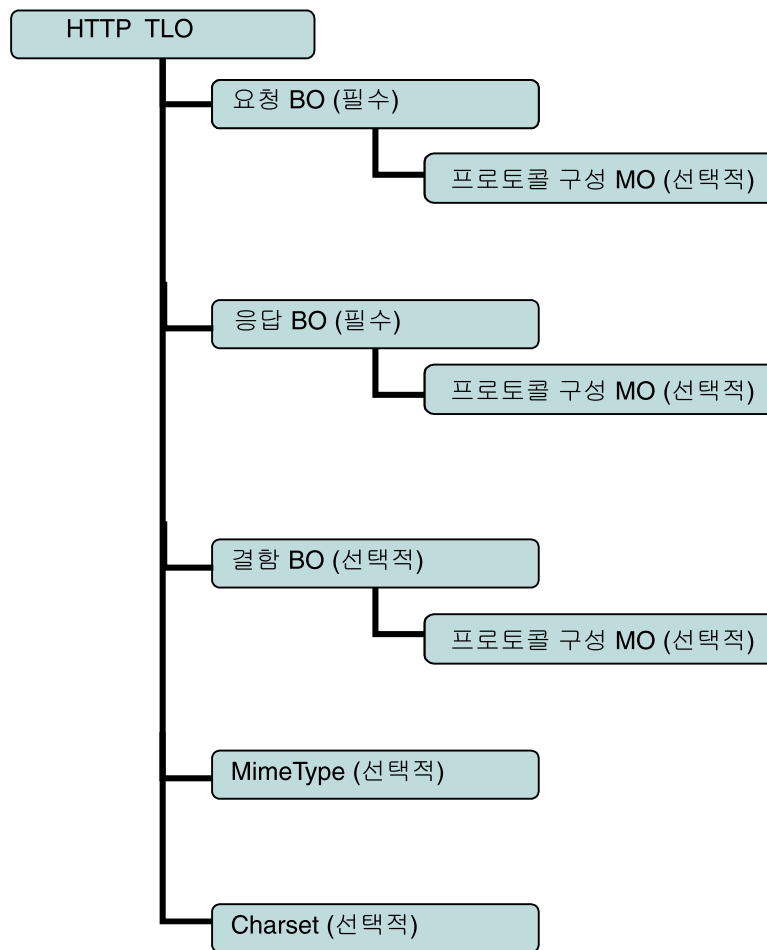


그림 4. 동기 이벤트 처리를 위한 Business Object 계층 구조

TLO에는 속성 레벨 ASI가 있는 속성은 물론 오브젝트 레벨 ASI를 포함합니다. 두 가지 종류의 ASI가 아래 설명됩니다.

동기 이벤트 처리 TLO의 오브젝트 레벨 ASI

오브젝트 레벨 ASI는 TLO의 특성과 포함하는 오브젝트에 대한 기본적인 정보를 제공합니다. 그림 5에서는 SERVICE_SYNCH_OrderStatus의 오브젝트 레벨 ASI, 동기 이벤트 처리의 샘플 TLO를 표시합니다.

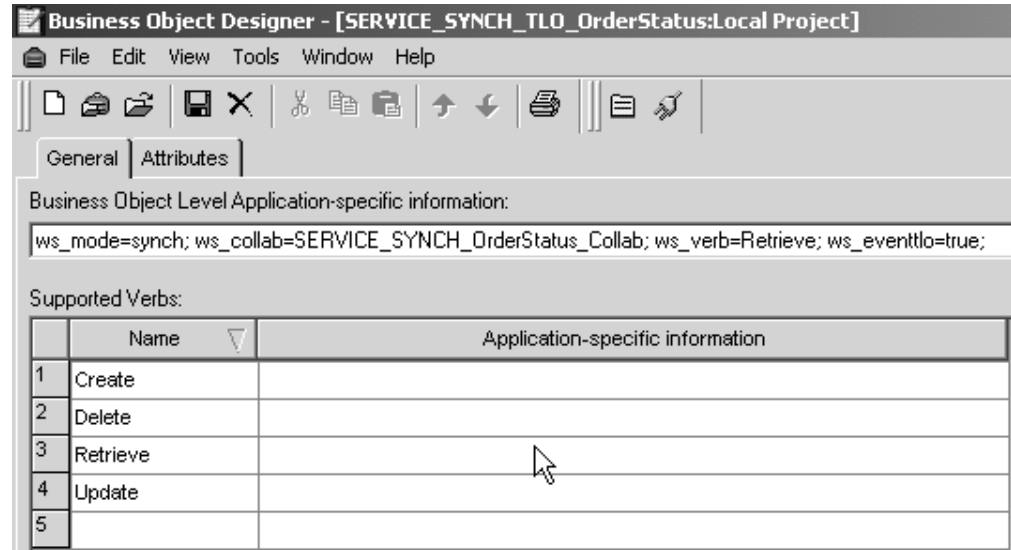


그림 5. 동기 오브젝트 처리를 위한 최상위 레벨 Business Object

아래 표 4에서는 동기 이벤트 처리 TLO의 오브젝트 레벨 ASI에 대해 설명합니다.

표 4. 동기 이벤트 처리 TLO 오브젝트 ASI

오브젝트 레벨 ASI	설명
ws_eventtlo=true	이 ASI 등록 정보가 true로 설정되면, 커넥터는 이벤트 처리를 위해 사용 가능한 TLO로 이 오브젝트를 처리합니다.
ws_collab=collabname	이 ASI에서는 어떤 협업을 호출할 지 커넥터에 알립니다. 해당 값은 협업의 이름입니다. 그림 5에 표시된 샘플에서, 협업 이름은 SERVICE_SYNCH_OrderStatus_Collab입니다.
ws_verb=verb	TLO를 협업으로 전달하기 전에, 커넥터는 이 ASI를 사용하여 TLO에서 verb를 설정합니다. 그림 5에 표시된 샘플에서, verb는 검색입니다.
ws_mode=synch	이벤트 공고 중, 커넥터는 이 ASI 등록 정보를 사용하여 동기적으로(synch) 또는 비동기적으로(asynch) 협업을 호출할 지 여부를 판별하기 위해 이 ASI 등록 정보를 사용합니다. 동기 처리의 경우, 이 ASI는 synch로 설정되어야 합니다. 기본값은 asynch입니다.

동기 이벤트 처리 TLO의 속성 레벨 ASI

각 동기 이벤트 처리 TLO는 속성 및 속성 레벨 ASI를 가집니다. 그림 6에서는 SERVICE_SYNCH_OrderStatus의 속성, 샘플 TLO를 표시합니다. 또한 응용프로그램 특정 정보 열에서 속성 레벨 ASI를 표시합니다.

	Pos	Name	Type	Key	Foreign	Required	Card	App Spec Info
1	1	Request	SERVICE_SYNCH_OrderStatus_Request	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=request
2	2	Response	SERVICE_SYNCH_OrderStatus_Response	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=response
3	3	Fault	SERVICE_SYNCH_OrderStatus_Fault	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=fault
4	4	ObjectEventId	String					
5	5			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

그림 6. 동기 이벤트 처리를 위한 TLO 속성

표 5에서는 동기 이벤트 처리 TLO의 요청, 응답, 결합, MimeType 및 Charset 속성의 속성 레벨 ASI를 요약합니다.

표 5. 동기 이벤트 처리 TLO 속성 ASI

TLO 속성	속성 레벨 ASI	설명
MimeType		선택적 속성이며, 지정된 경우 해당 값은 동기 응답에서 호출할 Data Handler의 MIME 유형으로 사용됩니다.
Charset		유형 문자열의 이 선택적 매개변수는 전송 Business Object를 메시지로 변환할 때 Data Handler에 설정되는 Charset를 지정합니다. 주: 이 속성으로 지정된 Charset 값은 응답 메시지의 Content-Type 프로토콜 머릿글에 전파되지 않습니다.
요청	ws_botype=request	이 속성은 HTTP 서비스 요청에 해당합니다. 커넥터는 이 TLO 속성이 유형 요청 BO인지 여부를 판별하기 위해 ASI를 사용합니다. 속성 이름이 아닌 이 ASI는 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 요청 속성의 ASI를 사용합니다. 동기 이벤트 처리 TLO를 위해 이 속성이 필요합니다.

표 5. 동기 이벤트 처리 TLO 속성 ASI (계속)

TLO 속성	속성 레벨 ASI	설명
응답	ws_botype=response	이 속성은 HTTP 서비스가 리턴한 응답에 해당합니다. 커넥터는 이 TLO 속성이 유형 응답 BO인지 여부를 판별하기 위해 이 ASI를 사용합니다. 속성 이름이 아닌 이 ASI는 속성 유형을 판별합니다. 둘 이상의 응답 속성이 있는 경우, 커넥터는 첫 번째 요청 속성의 ASI를 사용합니다. 동기 이벤트 처리 TLO를 위해 이 속성이 필요합니다.
결함	ws_botype=fault ws_botype=defaultfault	동기 이벤트 처리를 위해 선택적인 이 속성은 정상적으로 응답을 채울 수 없을 때 협업으로 리턴한 결함 메시지에 해당합니다. 커넥터는 속성 이름이 아닌 이 ASI를 사용하여 속성이 유형 결함 BO인지 여부를 판별합니다.

동기 이벤트 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 동기 이벤트 처리에 필요합니다. 요청 Business Object는 오브젝트 레벨 ASI를 갖습니다. 동기 이벤트 처리를 위한 요청 Business Object의 오브젝트 레벨 ASI가 표 6에 설명되어 있습니다. 요청 Business Object의 기본값 verb를 지정할 수 있습니다. 요청 Business Object의 최상위 레벨에서 지원되는 Verb 목록에서 verb의 ASI 필드에서

DefaultVerb=true;

와 같이 지정하여 수행합니다. DefaultVerb ASI가 지정되지 않고 Data Handler가 verb 세트 없이 Business Object를 처리하는 경우, Business Object는 verb 없이 리턴됩니다.

표 6. 동기 이벤트 처리: 요청 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_http=HTTPCfgMO	이 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. ASI는 HTTP 또는 HTTPS 프로토콜 리스너를 지정합니다. ASI 및 프로토콜 Config MO는 둘 다 선택적입니다. 자세한 정보는 27 페이지의 『프로토콜 Config MO』에서 참조하십시오. 주: Business Object 변환을 구성하는 Data Handler는 변환되는 주 데이터의 파트가 아닌 메타 데이터로써 cw_mo로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 오브젝트 cw_mo 메타 데이터를 발견하는 기능을 갖습니다.

표 6. 동기 이벤트 처리: 요청 Business Object의 오브젝트 레벨 ASI (계속)

오브젝트 레벨 ASI	설명
ws_tloname=tloname	이 ASI는 이 오브젝트가 속하는 TLO의 이름을 지정합니다. 이벤트 처리 중, 커넥터는 이 ASI를 사용하여 Data Handler가 전달하는 요청 Business Object가 TLO의 하위인지 여부를 판별합니다. 그런 경우 커넥터는 지정된 TLO를 작성하며, 하위로 요청 Business Object를 설정하며 TLO Business Object ASI를 사용하여 등록 협업으로 전달합니다.

동기 이벤트 처리를 위한 응답 Business Object

응답 Business Object는 TLO의 하위이며 동기 이벤트 처리에 필요합니다. 동기 이벤트 처리를 위한 응답 Business Object의 오브젝트 레벨 ASI가 표 7에 설명되어 있습니다.

표 7. 동기 이벤트 처리: 응답 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_http=HTTPCfgMO	이 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. ASI는 HTTP 또는 HTTPS 프로토콜 리스너를 지정합니다. ASI 및 프로토콜 Config MO는 둘 다 선택적입니다. 자세한 정보는 27 페이지의 『프로토콜 Config MO』에서 참조하십시오. 주: Business Object 변환을 구성하는 Data Handler는 변환되는 주 데이터의 파트가 아닌 메타 데이터로써 cw_mo로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 오브젝트 cw_mo 메타 데이터를 발견하는 기능을 갖습니다.

주: 선택적으로 응답 BO의 프로토콜 Config MO 오브젝트 레벨 ASI를 포함합니다.

동기 이벤트 처리를 위한 결합 Business Object

결합 Business Object는 TLO의 하위이며 동기 이벤트 처리를 위해 선택적입니다. 동기 이벤트 처리를 위한 결합 Business Object의 오브젝트 레벨 ASI가 표 8에 설명되어 있습니다.

표 8. 동기 이벤트 처리: 결합 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_http=HTTPCfgMO	이 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. ASI는 HTTP 또는 HTTPS 프로토콜 리스너를 지정합니다. ASI 및 프로토콜 Config MO는 둘 다 선택적입니다. 자세한 정보는 27 페이지의 『프로토콜 Config MO』에서 참조하십시오. 주: Business Object 변환을 구성하는 Data Handler는 변환되는 주 데이터의 파트가 아닌 메타 데이터로써 cw_mo로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 오브젝트 cw_mo 메타 데이터를 발견하는 기능을 갖습니다.

주: 선택적으로 결합 BO의 프로토콜 Config MO 오브젝트 레벨 ASI를 포함합니다.

프로토콜 Config MO

프로토콜 Config MO는 이벤트 처리를 위한 요청, 응답 또는 결합 Business Object의 하위로 선택적으로 포함됩니다. 일반적으로 사용자는 프로토콜 머리글 및 사용자 정의 등록 정보를 읽거나(요청 메시지에서) 전파(응답 또는 결합 메시지로)하기 위해 필요할 때 지정합니다. 위에서 언급한 대로 요청 Business Object는 선택적으로 Business Object 레벨 ASI로 프로토콜 Config MO의 이름을 선언합니다.

- `cw_mo_http=HTTPProtocolListenerConfigMOAttribute`

이벤트 처리 중, 커넥터는 프로토콜 리스너(HTTP 또는 HTTPS)를 사용하여 전송에서 이벤트를 검색합니다. 이 이벤트는 협업에서 서비스를 요청하는 내부 또는 외부 클라이언트의 메시지입니다. 각 전송은 자체 머리글 요구사항을 갖습니다. 커넥터는 프로토콜 리스너에서 협업으로의 프로토콜 특정 머리글 정보를 전달하기 위해 프로토콜 Config MO를 사용합니다. 프로토콜 Config MO 속성은 인바운드 메시지의 머리글에 대응합니다. 커넥터는 인바운드 메시지 내용을 사용하여 Business Object에서 이 속성 값을 설정합니다.

HTTP(S) 프로토콜의 경우, 프로토콜 Config MO 속성은 다음과 같습니다.

표 9. 이벤트 처리를 위한 HTTP/HTTPS 프로토콜 Config MO 속성

속성	필수	유형	설명
Content-Type	아니오	문자열	이 속성 값은 전송 메시지(메시지 ContentType과 전송 메시지의 경우 0 개 이상의 매개변수(문자세트)를 포함합니다)의 Content-Type 머리글을 정의합니다. 구문은 HTTP 프로토콜의 Content-Type 머리글의 구문과 같습니다. 예: <code>text/xml; charset=ISO-8859-4</code> . 정의된 Content-Type 속성이 없는 경우, 커넥터는 응답/결합 메시지의 ContentType으로 요청의 ContentType을 사용합니다.
UserDefinedProperties	아니오	Business object	이 속성은 사용자가 정의한 프로토콜 등록 정보 Business Object를 보유합니다.
하나 이상의 HTTP 머리글	아니오	문자열	이 속성은 핸들러가 지정된 HTTP 머리글의 값을 전달하거나 검색하게 합니다.
Authorization_UserID	아니오	문자열	이 속성은 HTTP 기본 인증의 사용자 ID에 해당합니다.

표 9. 이벤트 처리를 위한 HTTP/HTTPS 프로토콜 Config MO 속성 (계속)

속성	필수	유형	설명
Authorization_Password	아니오	문자열	이 속성은 HTTP 기본 인증의 암호에 해당합니다.

이 속성은 다음에서 설명됩니다.

- 『이벤트 처리를 위한 사용자 정의 등록 정보』
- 29 페이지의 『이벤트 처리를 위한 HTTP 신임 전파』

프로토콜 리스너에 대한 자세한 정보는 48 페이지의 『프로토콜 리스너』에서 참조하십시오(요청 처리를 위한 프로토콜 Config MO 설명에 대한 정보는 32 페이지의 『동기 요청 처리 TLO』에서 참조하십시오).

이벤트 처리를 위한 사용자 정의 등록 정보: 선택적으로 HTTP(S) 프로토콜 Config MO에서 사용자 정의 등록 정보를 지정할 수 있습니다. UserDefinedProperties 속성을 포함하여 지정할 수 있습니다. 이 속성은 등록 정보 값을 가진 하나 이상의 하위 속성이 있는 Business Object에 해당합니다. 이 Business Object의 모든 속성은 다음과 같이 메시지 머리글의 변수 부분에서 읽는(또는 동기 응답의 경우 작성된) 단일 등록 정보를 정의해야 합니다.

- 속성 유형은 프로토콜 등록 정보 유형과 관계 없이 문자열이어야 합니다. 속성의 응용프로그램 특정 정보는 속성이 맵핑하는 프로토콜 메시지 등록 정보의 이름과 형식을 정의하는 두 개의 이름-값 쌍을 포함할 수 있습니다.

표 10에서는 이 속성의 응용프로그램 특정 정보를 요약합니다.

표 10. 사용자 정의 프로토콜 등록 정보 속성의 응용프로그램 특정 정보: name=value 쌍 내용

이름	값	설명
ws_prop_name (대소문자 구분하지 않음. 지정하지 않으면 속성 이름이 등록 정보 이름으로 사용됩니다)	올바른 등록 정보 이름	이것은 프로토콜 등록 정보의 이름입니다. 일부 벤더는 확장된 기능을 제공하기 위해 특정 등록 정보를 보존합니다.

주어진 사용자 정의 등록 정보 ASI(the ws_prop_name)가 올바르지 않으며 이 머리글을 처리할 논리적 방식이 없는 경우, 커넥터는 경로를 기록하며 이 등록 정보를 무시합니다. ws_prop_name에 대해 필요한 검사가 수행된 이후에 필요한 검사사용자 정의 등록 정보의 값이 설정되거나 검색될 수 없는 경우, 커넥터는 오류를 기록하며 이벤트가 실패합니다.

UserDefinedProperties 속성이 지정된 경우, 커넥터는 UserDefinedProperties Business Object의 인스턴스를 작성합니다. 그런 다음 커넥터는 메시지에서 등록 정보 값을 추출하여 Business Object에 저장합니다. 최소한 하나의 등록 정보 값이 정상적으로 검색되면, 커넥터는 프로토콜 Config MO에서 수정된 UserDefinedProperties 속성을 설정합니다.

동기 이벤트 처리의 경우, UserDefinedProperties 속성이 지정되고 관련 Business Object 실례를 든 경우, 커넥터는 이 하위 Business Object의 각 속성을 처리하며 메시지 등록 정보 값을 이에 따라 설정합니다.

이벤트 처리를 위한 HTTP 신임 전파: 신임 전파의 목적으로, 커넥터는 HTTP 프로토콜 Config MO에서 Authorization_UserID 및 Authorization_Password 속성을 지원합니다. 지원은 HTTP 기본 인증 방식의 파트로 이 신임 전파에 제한됩니다.

HTTP 또는 HTTPS 프로토콜 리스너가 권한 부여 머리글을 포함하는 HTTP 서비스 요청을 처리하는 경우, 리스너는 머리글을 구문 분석하여 HTTP 기본 인증과 일치하는지 여부를 판별합니다. 그런 경우 리스너는 사용자 이름 및 암호를 추출하며 디코드합니다(Base64 사용). 이 디코드된 문자열은 콜론으로 구분된 사용자 이름과 암호로 구성됩니다. 프로토콜 리스너가 Authorization_UserID 및 Authorization_Password 속성을 프로토콜 Config MO에서 찾을 경우, 리스너는 이 값을 이벤트 권한 부여 머리글에서 추출된 값으로 설정합니다.

비동기 이벤트 처리 TLO

그림 7에서는 비동기 이벤트 처리를 위한 Business Object 계층 구조를 표시합니다. 요청 오브젝트만이 필요합니다.

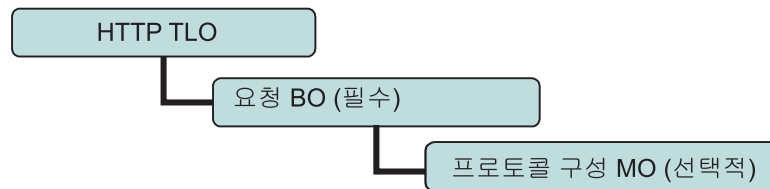


그림 7. 비동기 이벤트 처리를 위한 Business Object 계층 구조

TLO에는 속성 레벨 ASI가 있는 속성은 물론 오브젝트 레벨 ASI를 포함합니다. 두 가지 종류의 ASI가 아래 설명됩니다.

비동기 이벤트 처리 TLO의 오브젝트 레벨 ASI

오브젝트 레벨 ASI는 TLO의 특성과 포함하는 오브젝트에 대한 기본적인 정보를 제공합니다. 그림 8에서는 SERVICE_ASYNC_TLO_Order의 오브젝트 레벨 ASI, 비동기 이벤트 처리를 위한 샘플 TLO를 표시합니다.

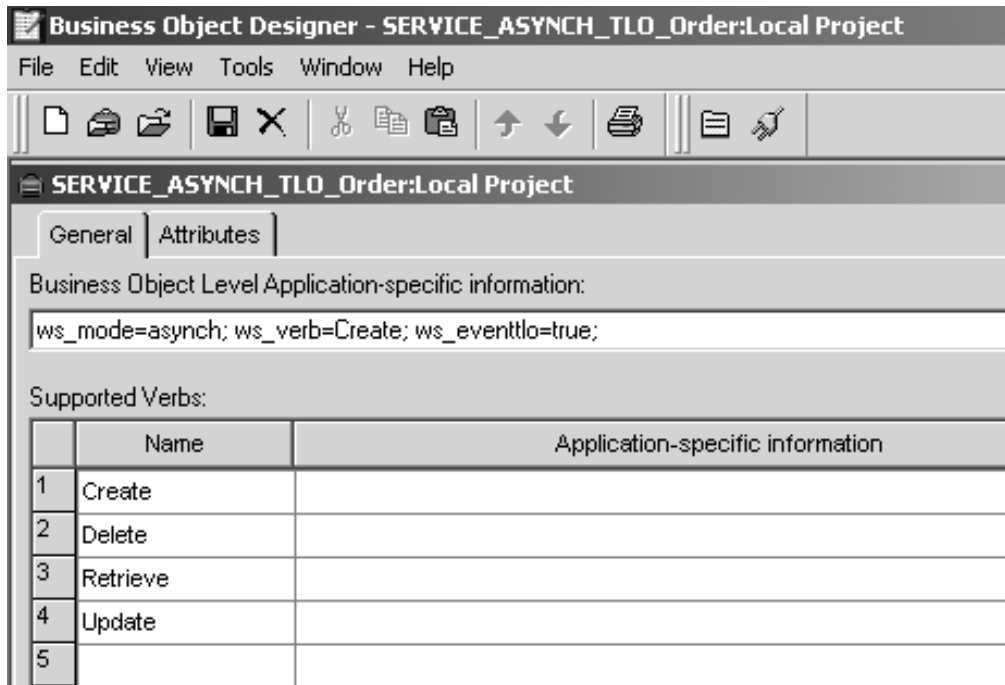


그림 8. 비동기 이벤트 처리를 위한 최상위 레벨 Business Object

아래 표 4에서는 비동기 이벤트 처리 TLO의 오브젝트 레벨 ASI에 대해 설명합니다.

표 11. 비동기 이벤트 처리 TLO 오브젝트 ASI

오브젝트 레벨 ASI	설명
<code>ws_eventtlo=true</code>	이 ASI 등록 정보가 true로 설정되면, 커넥터는 이벤트를 처리하기 위한 TLO로 이 오브젝트를 처리합니다.
<code>ws_verb=verb</code>	TLO를 협업으로 전달하기 전에, 커넥터는 이 ASI를 사용하여 TLO에서 verb를 설정합니다. 그림 8에 표시된 샘플에서, verb는 작성입니다.
<code>ws_mode=asynch</code>	이벤트 공고 중, 커넥터는 이 ASI 등록 정보를 사용하여 동기적으로(synch) 또는 비동기적으로(asynch) 협업을 호출할지 여부를 판별하기 위해 이 ASI 등록 정보를 사용합니다. 비동기 처리의 경우, 이 ASI는 asynch로 설정되어야 합니다. 기본값은 asynch입니다.

주: 동기 이벤트 처리와 달리, 비동기 이벤트를 처리하기 위한 TLO 레벨에서는 협업 이름 ASI가 필요하지 않습니다. 대신 통합 브로커는 그러한 BO-verb 조합에 등록하는 모든 협업에 응용프로그램 이벤트가 도달한다고 확신합니다.

비동기 이벤트 처리 TLO의 속성 레벨 ASI

각 비동기 이벤트 처리 TLO는 요청 Business Object에 해당하는 단일 속성을 갖습니다. 그림 9에서는 SERVICE_ASYNC_TLO_Order의 요청 속성, 샘플 TLO 및 속성의 ASI를 표시합니다.

	Pos	Name	Type	Key	Foreign	Required	Card	Maximum	App Spec Info
1	1	Request	SERVICE_ASYNC_Order	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		ws_botype=request
2	2	ObjectEventId	String						
3	3			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	

그림 9. 비동기 이벤트 처리를 위한 TLO 속성

표 12에서는 비동기 이벤트 처리 TLO의 요청 속성의 속성 레벨 ASI를 요약합니다.

표 12. 비동기 이벤트 처리 TLO 속성 ASI

TLO 속성	속성 레벨 ASI	설명
요청	ws_botype=request	이 속성은 요청에 해당합니다. 커넥터는 이 TLO 속성이 유형 요청 BO인지 여부를 판별하기 위해 ASI를 사용합니다. 속성 이름이 아닌 이 ASI는 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 요청 속성의 ASI를 사용합니다. 동기 이벤트 처리 TLO를 위해 이 속성이 필요합니다.

비동기 이벤트 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 비동기 이벤트 처리에 필요합니다. 요청 Business Object의 기본값 verb를 지정할 수 있습니다. 요청 Business Object의 최상위 레벨에서 지원되는 Verb 목록에서 verb의 ASI 필드에서

DefaultVerb=true;

와 같이 지정하여 수행합니다. DefaultVerb ASI가 지정되지 않고 Data Handler가 verb 세트 없이 Business Object를 처리하는 경우, Business Object는 verb 없이 리턴됩니다. 비동기 이벤트 처리를 위한 요청 Business Object의 오브젝트 레벨 ASI가 표 13에 설명되어 있습니다.

표 13. 비동기 이벤트 처리: 요청 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
<code>cw_mo_http=HTTPCfgMO</code>	<p>이 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. ASI는 HTTP 또는 HTTPS 프로토콜 리스너를 지정합니다. ASI 및 프로토콜 Config MO는 둘 다 선택적입니다. 자세한 정보는 27 페이지의 『프로토콜 Config MO』에서 참조하십시오.</p> <p>주: Business Object 변환을 구성하는 Data Handler는 변환되는 주 데이터의 파트가 아닌 메타 데이터로써 <code>cw_mo</code>로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 오브젝트 <code>cw_mo</code> 메타 데이터를 발견하는 기능을 갖습니다.</p>
<code>ws_tloname=tloname</code>	<p>이 ASI는 이 오브젝트가 속하는 TLO의 이름을 지정합니다. 이벤트 처리 중, 커넥터는 이 ASI를 사용하여 Data Handler가 전달하는 요청 Business Object가 TLO의 하위인지 여부를 판별합니다. 그런 경우 커넥터는 지정된 TLO를 작성하며, 하위로 요청 Business Object를 설정하며 TLO Business Object ASI를 사용하여 등록 협업으로 전달합니다.</p>

동기 요청 처리 TLO

요청 처리의 경우 커넥터는 두 가지 종류의 TLO(동기 및 비동기)를 허용합니다. 이 섹션에서는 동기 요청 처리 TLO에 대해 설명합니다.

그림 10에서는 동기 요청 처리를 위한 TLO Business Object를 표시합니다. 요청, 응답 및 핸들러 오브젝트가 필요하며, 결합 오브젝트가 선택적입니다. 이벤트 처리와 달리, 프로토콜 Config MO는 요청 오브젝트에 필요하며, 응답 및 결합 오브젝트에 선택적입니다.

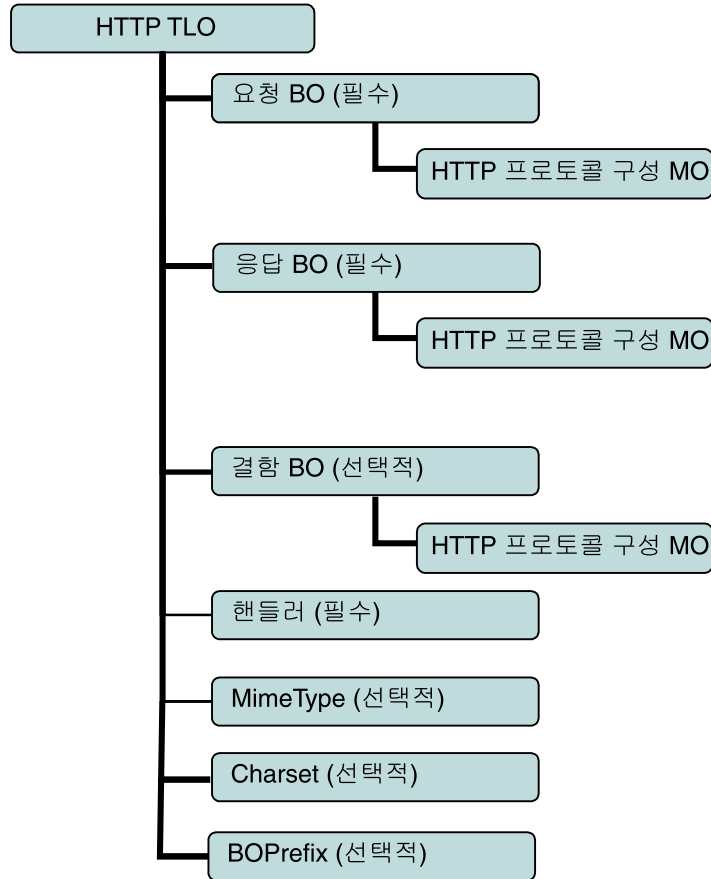


그림 10. 동기 요청 처리를 위한 Business Object 계층 구조

동기 요청 처리 TLO의 오브젝트 레벨 ASI

오브젝트 레벨 ASI는 TLO의 특성과 포함하는 오브젝트에 대한 중요한 정보를 제공합니다. 그림 11은 동기 요청 처리를 위한 TLO인 CLIENT_SYNCH_TLO_OrderStatus를 표시합니다.

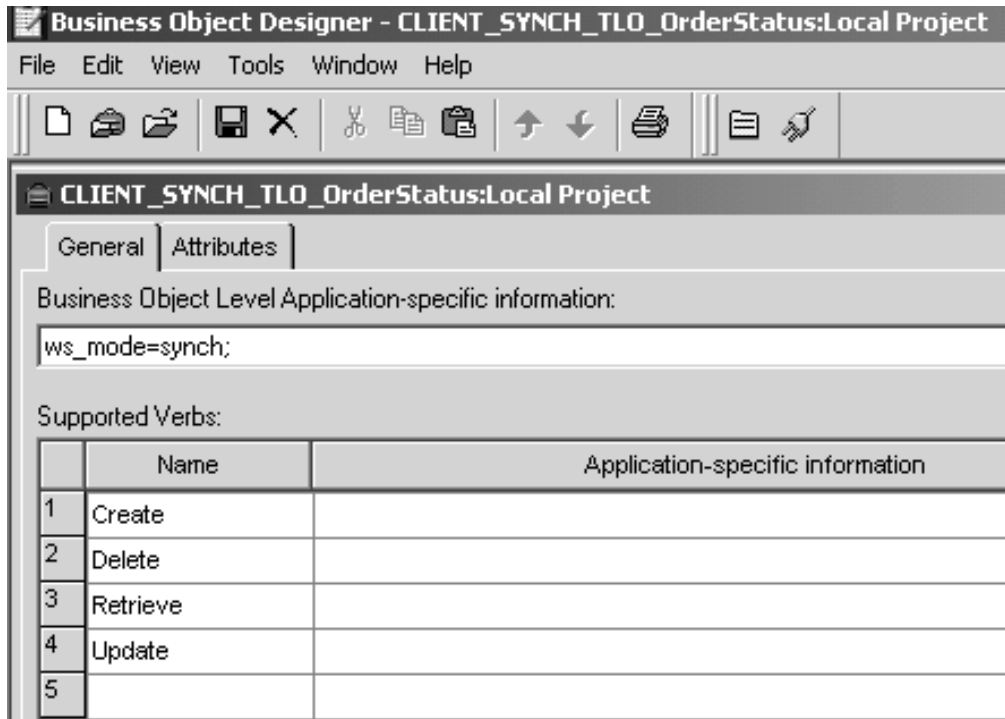


그림 11. 동기 요청 처리를 위한 최상위 레벨 Business Object

표 14에서는 동기 요청 처리 TLO의 오브젝트 레벨 ASI에 대해 설명합니다. 동기 이벤트 처리 TLO의 ASI와 달리, 요청 처리를 위해 이 레벨에서 ws_collab, ws_verb 또는 ws_eventtlo ASI가 필요하지 않습니다.

표 14. 동기 요청 처리 TLO 오브젝트 ASI

오브젝트 레벨 ASI	설명
ws_mode=synch	요청 처리 중, 커넥터는 이 ASI 등록 정보를 사용하여 동기적으로(synch) 또는 비동기적으로(asynch) HTTP 서비스를 호출할 지 여부를 판별하기 위해 이 ASI 등록 정보를 사용합니다. synch가 표시되는 경우, 커넥터는 응답을 예상하며 TLO는 요청 및 응답 Business Object를 포함해야 하며, 선택적으로 하나 이상의 결합 오브젝트를 포함해야 합니다. 기본값은 asynch입니다.

동기 요청 처리 TLO의 속성 레벨 ASI

표 15에서는 동기 요청 처리 TLO의 속성 및 ASI에 대해 설명합니다.

표 15. 요청 처리 TLO 속성

TLO 속성	속성 레벨 ASI	설명
MimeType	없음	이 속성은 커넥터가 요청 비즈니스 오브젝트를 요청 메시지로 변환하기 위해 커넥터가 호출하는 Data Handler의 MIME 유형을 지정합니다. 이 값은 메시지 변환 규칙 구성에 따라 Business Object로 동기 응답/결함 메시지를 변환하는 데 사용될 수 있습니다.
BOPrefix	없음	유형 문자열의 이 속성이 Data Handler로 전달됩니다.
핸들러	없음	이 속성은 요청을 처리하기 위해 사용할 프로토콜 핸들러를 지정하며 요청 처리만을 위한 것입니다. HTTP- HTTPS 프로토콜 핸들러를 지정하는 값 http를 갖습니다. 기본값은 http입니다.
Charset		유형 문자열의 이 선택적 매개변수는 요청 Business Object를 메시지로 변환할 때 Data Handler에 설정되는 Charset를 지정합니다. 주: 이 속성으로 지정된 Charset 값은 요청 메시지의 Content-Type 프로토콜에서 전파되지 않습니다.
요청	ws_botype=request	이 속성은 요청 Business Object에 해당합니다. 커넥터는 이 TLO 속성이 유형 요청 BO인지 여부를 판별하기 위해 이 속성 ASI를 사용합니다. 속성 이름이 아닌 이 ASI는 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 채워진 속성의 ASI를 사용합니다.
응답	ws_botype=response	이 속성은 협업으로 리턴된 응답에 해당하며 동기 요청 처리를 위해 필요합니다. 커넥터는 이 TLO 속성이 유형 응답 BO인지 여부를 판별하기 위해 이 속성 ASI를 사용합니다. 속성 이름이 아닌 이 ASI는 속성 유형을 판별합니다.
결함	ws_botype=fault 또는 ws_botype=defaultfault	동기 요청 처리를 위해 선택적인 이 속성은 정상적으로 응답을 채울 수 없을 때 HTTP 서비스로 리턴한 결함 메시지에 해당합니다. TLO의 속성이 유형 결함 BO인지 여부를 판별하기 위해 이 ASI를 사용합니다. 속성 이름이 아닌 이 ASI는 속성 유형을 판별합니다. 결함 메시지가 세부사항 요소인 경우 defaultfault Business Object가 리턴됩니다. defaultfault는 기본값 Business Object 해상도에서 사용됩니다.

동기 요청 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 동기 요청 처리에 필요합니다. 요청 Business Object는 오브젝트 레벨 ASI를 갖습니다.

표 16에서는 동기 요청 처리를 위한 요청 Business Object의 오브젝트 레벨 ASI에 대해 설명합니다.

표 16. 동기 요청 처리: 요청 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_http= HTTPCfgMO	<p>이 선택적 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. 이 프로토콜 Config MO는 HTTP-HTTPS 프로토콜 핸들러의 목적지를 지정합니다. 이 ASI는 HTTP-HTTPS 프로토콜 핸들러에 의해 사용됩니다. TLO 요청 속성이 요청 처리를 위한 HTTP 프로토콜 Config MO를 가져야 함을 참고하십시오. 자세한 정보는 37 페이지의 『요청 처리를 위한 HTTP 프로토콜 Config MO』에서 참조하십시오.</p> <p>주: Business Object 변환을 위해 구성하는 Data Handler가 변환되는 비즈니스 데이터의 파트가 아닌 메타 데이터로써 cw_mo로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 cw_mo 메타 데이터를 발견하는 기능을 갖습니다.</p>

동기 요청 처리를 위한 응답 Business Object

응답 Business Object는 TLO의 하위이며 동기 요청 처리에 필요합니다. 동기 요청 처리를 위한 응답 Business Object의 오브젝트 레벨 ASI가 표 17에 설명되어 있습니다.

표 17. 동기 요청 처리: 응답 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_http= HTTPCfgMO	<p>이 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. 응답 Business Object의 선택적인 이 프로토콜 Config MO는 HTTP(s) 프로토콜 핸들러의 응답 메시지에서 머릿글을 지정합니다. 자세한 정보는 37 페이지의 『요청 처리를 위한 HTTP 프로토콜 Config MO』에서 참조하십시오.</p> <p>주: Business Object 변환을 구성하는 Data Handler는 변환되는 주 데이터의 파트가 아닌 메타 데이터로써 cw_mo로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 오브젝트 cw_mo 메타 데이터를 발견하는 기능을 갖습니다.</p>

응답 Business Object의 기본값 verb를 지정할 수 있습니다. 응답 Business Object의 최상위 레벨에서 지원되는 Verb 목록에서 verb의 ASI 필드에서

```
DefaultVerb=true;
```

와 같이 지정하여 수행합니다. DefaultVerb ASI가 지정되지 않고 Data Handler가 verb 세트 없이 Business Object를 처리하는 경우, 응답 Business Object는 verb 없이 리턴됩니다.

동기 요청 처리를 위한 결합 Business Object

결합 Business Object는 TLO의 하위이며 동기 요청 처리를 위해 선택적입니다. 동기 요청 처리를 위한 결합 Business Object의 오브젝트 레벨 ASI가 표 8에 설명되어 있습니다.

표 18. 동기 요청 처리: 결합 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_http= HTTPCfgMO	이 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. 결합 Business Object의 선택적인 이 프로토콜 Config MO는 HTTP-HTTPS 프로토콜 핸들러의 응답 메시지에서 머릿글을 지정합니다. 자세한 정보는 27 페이지의 『프로토콜 Config MO』에서 참조하십시오. 주: Business Object 변환을 구성하는 Data Handler는 변환되는 주 데이터의 파트가 아닌 메타 데이터로써 cw_mo로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 cw_mo 메타 데이터를 발견하는 기능을 갖습니다.

요청 처리를 위한 HTTP 프로토콜 Config MO

요청 처리 중, HTTP-HTTPS 프로토콜 핸들러는 HTTP 프로토콜 Config MO를 사용하여 대상 HTTP 서비스의 목적지를 결정합니다. 이 프로토콜 Config MO는 요청 Business Object에 필요합니다. HTTP-HTTPS 프로토콜 핸들러는 HTTP 1.0 POST 요청만을 지원합니다. 표 19에 표시된 대로 단 하나의 필수 속성(목적지)이 대상 HTTP 서비스의 전체 URL입니다. 선택적 권한 부여 속성이 아래 섹션에 설명되어 있습니다.

표 19. 요청 처리를 위한 HTTP 프로토콜 Config MO 속성

속성	필수	유형	설명
목적지	예	String	대상 HTTP 서비스의 목적지 URL. HTTP-HTTPS 프로토콜 핸들러는 이 속성을 사용하여 HTTP 서비스의 목적지를 판별합니다.
Content-Type	요청 Business Object에 필수이며, 그렇지 않으면 선택적입니다.	문자열	이 속성 값은 전송 메시지(메시지 ContentType과 전송 메시지의 경우 선택적으로 Charset를 포함합니다)의 Content-Type 머릿글을 정의합니다. 구문은 HTTP 프로토콜의 Content-Type 머릿글의 구문과 같습니다. 예: text/xml; charset=ISO-8859-4.
Authorization_UserID	아니오	문자열	이 속성은 HTTP 기본 인증의 사용자 ID에 해당합니다. 자세한 정보는 40 페이지의 『요청 처리를 위한 HTTP 신임 전파』에서 참조하십시오.
Authorization_Password	아니오	문자열	이 속성은 HTTP 기본 인증의 암호에 해당합니다. 자세한 정보는 40 페이지의 『요청 처리를 위한 HTTP 신임 전파』에서 참조하십시오.

표 19. 요청 처리를 위한 HTTP 프로토콜 Config MO 속성 (계속)

속성	필수	유형	설명
하나 이상의 HTTP 머리글	아니오	문자열	이 속성은 핸들러가 지정된 HTTP 머리글의 값을 전달하거나 검색하게 합니다.
UserDefinedProperties	아니오	Business object	이 속성은 사용자가 정의한 프로토콜 등록 정보 Business Object를 보유합니다. 자세한 정보는 『요청 처리를 위한 사용자 정의 등록 정보』에서 참조하십시오.
MessageTransformationMap	아니오	단일 카디널리티 Business Object	이것은 0개 이상의 메시지 변환 규칙을 보유하는 Business Object를 가리키는 속성입니다. 규칙은 규칙에 지정된 수신 메시지에 적용할 MIME 유형과 Charset에 대한 정보를 보유합니다. 자세한 정보는 39 페이지의 『메시지 변환 맵』에서 참조하십시오.

HTTP 프로토콜 Config MO 속성이 다음에서 설명됩니다.

- 『요청 처리를 위한 사용자 정의 등록 정보』
- 39 페이지의 『메시지 변환 맵』
- 40 페이지의 『요청 처리를 위한 HTTP 신임 전파』

요청 처리를 위한 사용자 정의 등록 정보: 선택적으로 HTTP 프로토콜 Config MO에서 사용자 정의 등록 정보를 지정할 수 있습니다. UserDefinedProperties 속성을 포함하여 지정할 수 있습니다. 이 속성은 등록 정보 값을 가진 하나 이상의 하위 속성이 있는 Business Object에 해당합니다. 이 Business Object의 모든 속성은 다음과 같이 메시지 머리글의 변수 부분에서 읽는(또는 동기 응답의 경우 작성된) 단일 등록 정보를 정의해야 합니다.

- 속성 유형은 항상 문자열이어야 합니다. 속성의 응용프로그램 특정 정보는 속성이 맵핑하는 프로토콜 메시지 등록 정보의 이름과 형식을 정의하는 이름-값 쌍을 포함할 수 있습니다.

표 20에는 이 속성에 대한 응용프로그램 특정 정보가 요약되어 있습니다.

표 20. 사용자 정의 프로토콜 등록 정보 속성의 응용프로그램 특정 정보: 이름=값 쌍 내용

이름	값	설명
ws_prop_name (대소문자 구분하지 않음. 지정하지 않으면 속성 이름이 등록 정보 이름으로 사용됩니다)	올바른 등록 정보 이름	이것은 프로토콜 등록 정보의 이름입니다. 일부 벤더는 확장된 기능을 제공하기 위해 특정 등록 정보를 보존합니다.

주어진 사용자 정의 등록 정보 ASI(the ws_prop_name)가 올바르지 않으며 이 머리글을 처리할 논리적 방식이 없는 경우, 커넥터는 경고를 기록하며 이 등록 정보를 무시합니다. ws_prop_name에 대해 필요한 검사가 수행된 이후에 사용자 정의 등록 정보의 값이 설정되거나 검색될 수 없는 경우, 커넥터는 오류를 기록하며 이벤트가 실패합니다.

UserDefinedProperties 속성이 지정되고 관련 Business Object 실례를 든 경우, 커넥터는 이 하위 Business Object의 각 속성을 처리하며 메시지 등록 정보 값을 이에 따라 설정합니다.

동기 요청 처리의 경우 응답 메시지 수령 시, UserDefinedProperties 속성이 지정된 경우 커넥터는 UserDefinedProperties Business Object의 인스턴스를 작성하며 메시지에서 등록 정보 값을 추출한 후 새 Business Object에 저장합니다. 최소한 하나의 등록 정보 값이 정상적으로 검색되면, 커넥터는 프로토콜 Config MO에서 수정된 UserDefinedProperties Business Object를 설정합니다.

메시지 변환 맵: MTM(Message Transformation Map) 기능은 요청 처리 HTTP(S) 프로토콜 핸들러만을 위해 지원됩니다. MessageTransformationMap은 Business Object를 지시하는 프로토콜 Config MO의 선택적 속성입니다. Business Object에는 규칙에 지정된 MIME 유형과 Charset로 메시지를 변환하는 규칙이 들어 있습니다. (대소문자 구분) 속성 이름 MessageTransformationMap을 찾고 이 속성이 Business Object 유형이면, 커넥터는 해당 오브젝트의 규칙을 사용하여 메시지를 변환합니다.

MTM 속성은 TransformationRule로 이름 지정된 하나의 카디널리티 N 하위 Business Object 속성을 가져야 합니다. 메시지의 TransformationRule을 찾으려고 시도할 때, HTTP-HTTPS 프로토콜 핸들러는 우선 모든 TransFormationRules에서 지정한 ContentType으로 지정한 그대로 메시지를 일치시키려고 시도합니다. 실패한 경우, 커넥터는 여러 유형의 메시지에 적용하는 규칙을 찾으려고 시도합니다. 프로토콜 핸들러 처리에 대한 자세한 정보는 56 페이지의 『HTTP-HTTPS 프로토콜 핸들러 처리』에서 참조하십시오.

TransformationRule Business Object의 각 인스턴스는 표 21에서 표시한 대로 지정한 속성을 가져야 합니다.

표 21. HTTP 프로토콜 Config MO에서 MessageTransformationMaps의 TransformationRule 속성

속성 이름	필수	유형	기본값	설명
TransformationRule	아니오	Business object, 카디널리티 N		이것은 메시지 변환을 위한 1 규칙을 보유하는 속성입니다. MessageTranformationMap 속성 아래에는 이 속성의 하나 이상의 인스턴스가 있을 수 있습니다.

표 21. HTTP 프로토콜 Config MO에서 MessageTransformationMaps의 TransformationRule 속성 (계속)

속성 이름	필수	유형	기본값	설명
+ContentType	예	문자열	*/*	이 등록 정보의 값은 이 변환 규칙이 적용하는 메시지의 HTTP ContentType을 지정합니다. 이 속성의 기본값 */*는 커넥터가 이 규칙을 임의 ContentType에 적용할 수 있게 합니다. 프로토콜 핸들러 처리에 대한 자세한 정보는 56 페이지의 『HTTP-HTTPS 프로토콜 핸들러 처리』에서 참조하십시오. 프로토콜 핸들러가 다른 규칙과 동일한 ContentType을 갖는 둘 이상의 규칙을 찾는 경우, 프로토콜 핸들러는 경고를 기록하며 중복된 규칙을 모두 무시하지만 고유한 규칙을 사용합니다.
+MimeType	아니오			이 Business Object에 지정된 ContentType의 메시지를 처리하는 동안 Data Handler를 호출할 때 사용할 MIME 유형.
+Charset	아니오			이 Business Object에 지정된 ContentType의 요청을 변환할 때 사용할 Charset.

요청 처리를 위한 HTTP 신임 전파: 신임 전파를 위해 커넥터는 HTTP 프로토콜 Config MO에서 Authorization_UserID 및 Authorization_Password 속성을 지원합니다. 지원은 HTTP 기본 인증 방식의 파트로 이 신임 전파에 제한됩니다.

협업은 프로토콜 Config MO에서 Authorization_UserID 및 Authorization_Password 속성의 값을 설정합니다. 이 속성이 널(null)이 아니거나 비어 있지 않은 경우, 커넥터는 대상 HTTP 서비스로 전송하는 요청에서 권한 부여 머리글을 작성합니다. HTTP/HTTPS 프로토콜 핸들러는 권한 부여 머리글 작성 시 *HTTP Authentication: Basic and Digest Access Authentication (RFC 2617)*을 따릅니다.

주: 요약 인증 방식은 지원되지 않으며, Rfc2617에 정의된 HTTP 인증의 선택적 요청-응답 메커니즘도 아닙니다. HTTP 프로토콜 핸들러가 신임을 요구하는 서버를 호출 중이면, 커넥터는 서버에서 요청 응답을 대기하지 않습니다. 대신 계속하여 신임을 전송합니다.

비동기 요청 처리 TLO

그림 12에서는 비동기 요청 처리를 위한 Business Object 계층 구조를 표시합니다. 요청 및 핸들러 오브젝트가 필수입니다. 요청 오브젝트는 HTTP-HTTPS 프로토콜 핸들러의 프로토콜 Config MO를 포함합니다. 이것은 아래 섹션에서 설명됩니다.

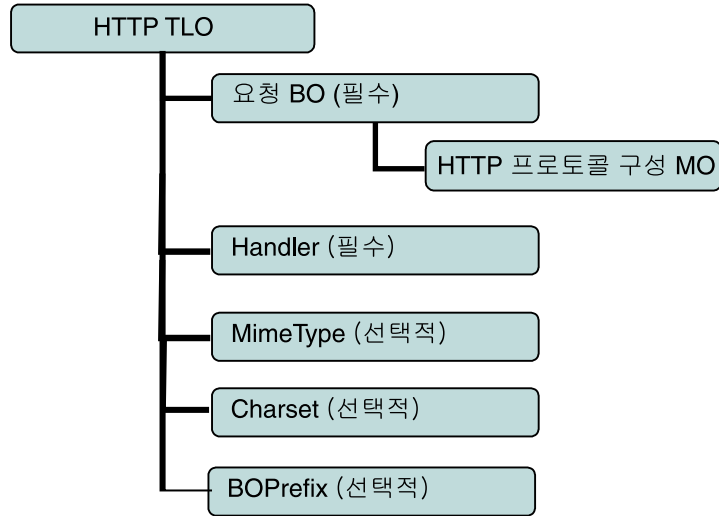


그림 12. 비동기 요청 처리를 위한 Business Object 계층 구조

TLO에는 속성 레벨 ASI가 있는 속성은 물론 오브젝트 레벨 ASI를 포함합니다. 두 가지 종류의 ASI가 아래 설명됩니다.

비동기 이벤트 처리 TLO의 오브젝트 레벨 ASI

그림 13에서는 비동기 요청 처리를 위한 샘플 TLO인 CLIENT_ASYNC_Order_TLO를 표시합니다.

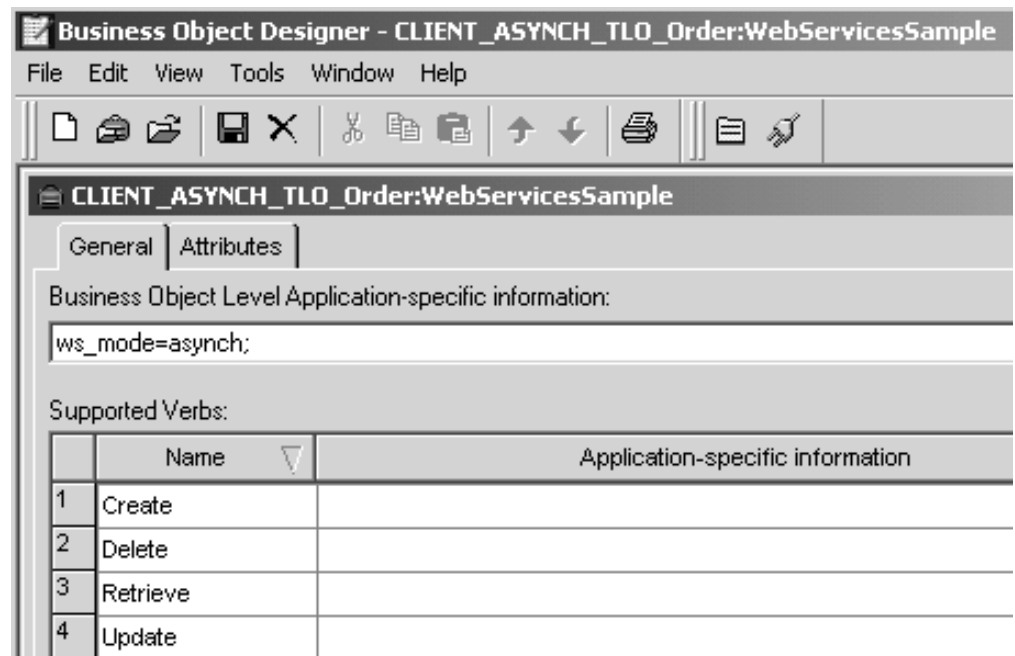


그림 13. 비동기 요청 처리를 위한 최상위 레벨 Business Object

아래 표 22에서는 비동기 요청 처리 TLO의 오브젝트 레벨 ASI에 대해 설명합니다.

표 22. 비동기 요청 처리 TLO 오브젝트 ASI

오브젝트 레벨 ASI	설명
ws_mode=asynch	요청 처리 중, 커넥터는 이 ASI 등록 정보를 사용하여 동기적으로(synch) 또는 비동기적으로(asynch) 협업을 호출할 지 여부를 판별하기 위해 이 ASI 등록 정보를 사용합니다. 비동기 요청 처리의 경우, 이 ASI는 asynch로 설정되어야 합니다. 기본값은 asynch입니다.

비동기 요청 처리 TLO의 속성 레벨 ASI

표 23에서는 비동기 요청 처리 TLO의 요청 속성의 속성 레벨 ASI를 요약합니다.

표 23. 비동기 요청 처리 TLO 속성

TLO 속성	속성 레벨 ASI	설명
MimeType	없음	이 속성은 커넥터가 호출하는 Data Handler의 MIME 유형을 지정합니다. 이 속성은 요청 처리 전용임을 참고하십시오.
BOPrefix	없음	이 속성 값은 Data Handler로 전달됩니다.
핸들러	없음	이 속성은 요청을 처리하기 위해 사용할 프로토콜 핸들러를 지정하며 요청 처리만을 위한 것입니다. 요청을 처리하기 위해 HTTP-HTTPS 프로토콜 핸들러를 지정하는 값 http를 갖습니다. 기본값은 http입니다.
Charset		유형 문자열의 이 선택적 매개변수는 요청 Business Object를 메시지로 변환할 때 Data Handler에 설정되는 Charset를 지정합니다. 주: 이 속성으로 지정된 Charset 값은 요청 메시지의 Content-Type 프로토콜에서 전파되지 않습니다.
요청	ws_botype=request	이 속성은 HTTP 서비스 요청 Business Object에 해당합니다. 커넥터는 이 TLO 속성이 유형 요청 BO인지 여부를 판별하기 위해 이 속성 ASI를 사용합니다. 속성 이름이 아닌 이 ASI는 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 요청 속성의 ASI를 사용합니다.

비동기 요청 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 비동기 요청 처리에 필요합니다. 비동기 요청 처리를 위한 요청 Business Object의 오브젝트 레벨 ASI가 표 24에 설명되어 있습니다.

표 24. 비동기 요청 처리: 요청 Business Object의 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_http= HTTPCfgMO	<p>이 ASI의 값은 프로토콜 Config MO에 대응하는 속성의 이름과 일치해야 합니다. 이 프로토콜 Config MO는 HTTP-HTTPS 프로토콜 핸들러의 목적지를 지정합니다. 이 ASI는 HTTP-HTTPS 프로토콜 핸들러에 의해 사용됩니다. TLO 요청 속성이 요청 처리를 위한 HTTP 프로토콜 Config MO를 가져야 함을 참고하십시오. 자세한 정보는 37 페이지의 『요청 처리를 위한 HTTP 프로토콜 Config MO』에서 참조하십시오.</p> <p>주: Business Object 변환을 구성하는 Data Handler는 변환되는 주 데이터의 파트가 아닌 메타 데이터로써 cw_mo 로 시작하는 ASI를 읽을 수 있어야 합니다. XML Data Handler는 그러한 값이 지시하는 속성을 무시하며 오브젝트 cw_mo 메타 데이터를 발견하는 기능을 갖습니다.</p>

비동기 요청 처리를 위한 프로토콜 Config MO

요청 처리 중, HTTP-HTTPS 프로토콜 핸들러는 HTTP 프로토콜 Config MO를 사용하여 대상 HTTP 서비스의 목적지를 결정합니다. 이 프로토콜 Config MO는 요청 Business Object에 필요합니다. 자세한 정보는 37 페이지의 『요청 처리를 위한 HTTP 프로토콜 Config MO』에서 참조하십시오.

Business Object 개발

Business Object Designer를 사용하여 Business Object를 작성하고 Connector Configurator를 사용하여 Business Object를 지원하는 커넥터를 구성합니다. Business Object Designer 도구에 대한 자세한 정보는 *Business Object Development Guide* 및 97 페이지의 부록 B 『Connector Configurator』에서 참조하십시오.

제 4 장 Connector for HTTP

- 『커넥터 처리』
- 47 페이지의 『HTTP(S) 서비스』
- 48 페이지의 『이벤트 처리』
- 55 페이지의 『요청 처리』
- 61 페이지의 『SSL』
- 63 페이지의 『커넥터 구성』
- 74 페이지의 『시동 시 커넥터』
- 75 페이지의 『로깅』
- 75 페이지의 『추적』

이 장에서는 Connector for HTTP와 커넥터 구성 방법에 대해 설명합니다.

모든 WebSphere Business Integration 커넥터는 통합 브로커로 작동합니다. Connector for HTTP는 *Technical Introduction to IBM WebSphere InterChange Server*에서 설명하는 IBM WebSphere InterChange Server 통합 브로커로 작동합니다.

커넥터는 어댑터의 런타임 구성요소입니다. 커넥터는 응용프로그램 특정 구성요소 및 커넥터 프레임워크로 구성됩니다. 응용프로그램 특정 구성요소에는 특정 응용프로그램에 맞게 조정된 코드가 들어 있습니다. 코드가 모든 커넥터에 공통인 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 간에 매개 역할을 합니다. 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 간에 다음 서비스를 제공합니다.

- Business Object를 수신하고 전송합니다.
- 시작 및 관리 메시지의 교환을 관리합니다.

이 문서에는 응용프로그램 특정 구성요소 및 커넥터 프레임워크에 대한 정보가 들어 있습니다. 이 구성요소 둘다를 커넥터로 나타냅니다.

통합 브로커와 커넥터의 관계에 대한 자세한 정보는 *System Administration Guide*를 참조하십시오.

커넥터 처리

커넥터는 이벤트 처리를 위한 프로토콜 리스너 프레임워크와 요청 처리를 위한 프로토콜 핸들러 프레임워크를 포함합니다. 이 양방향 기능은 커넥터 프레임워크로 하여금 다음을 수행할 수 있게 합니다.

- HTTP 클라이언트에서 호출 처리(이벤트 처리)

- HTTP 서비스를 호출하는 협업에 의한 요청 처리(요청 처리)

이벤트 처리 개요

커넥터 이벤트 처리(또는 이벤트 공고)는 HTTP 클라이언트에서 요청을 핸들하는 데 사용됩니다. 이 이벤트 처리 기능은 이 장에서 나중에 더 자세히 설명하는 다음 구성요소가 포함된 프로토콜 리스너 프레임워크를 포함합니다.

- HTTP 프로토콜 리스너
- HTTPS 프로토콜 리스너

커넥터는 클라이언트에서 협업으로의 호출 전송에서 청취하기 위해 이 구성요소를 사용합니다.

클라이언트에서 요청이 도착하면 리스너는 요청 메시지를 Business Object로 변환하며 협업을 호출합니다. 동기 요청인 경우 커넥터는 요청 Business Object와 동일한 유형의 응답 Business Object를 수신합니다. 리스너는 응답 Business Object를 응답 메시지로 변환합니다. 그런 다음 리스너는 응답 메시지를 클라이언트로 전송합니다. 이벤트 순서는 이 커넥터의 요구사항이 아니며, 커넥터는 임의 순서로 이벤트를 전달할 수 있음을 기억하십시오.

Connector for HTTP는 구성된 Data Handler를 이용하여 수신 요청 메시지를 Business Object로 변환합니다. 수신 요청 메시지 분석을 위한 Business Object 판별에서 Data Handler를 돕기 위해, 커넥터는 지원되는 Business Object에 대한 메타 정보를 Data Handler에 제공합니다. 지원되는 Business Object에서, 커넥터는 먼저 변환의 잠재적인 후보가 되는 모든 Business Object 목록을 작성합니다. 이 목록은 지원되는 TLO만으로 구성됩니다. 지원되는 TLO Business Object는 오브젝트 레벨 `ASIs_ws_eventtlo=true`를 갖는 Business Object입니다.

프로토콜 리스너는 다음과 같이 TLO의 오브젝트 레벨 ASI를 읽습니다.

- `ws_collab=`은 호출할 협업을 판별합니다
- `ws_mode=`은 협업을 동기적으로(synch) 또는 비동기적으로(asynch) 호출하는 방법을 판별합니다.

커넥터는 Data Handler가 리턴한 요청 비즈니스 오브젝트를 검사합니다. 이 Business Object의 `ws_tloname` ASI를 사용하여 상위 TLO의 이름을 추출합니다. 이 TLO는 예를 들어 설명되며 요청 Business Object가 TLO에서 설정됩니다. 마지막으로 구성된 이 TLO는 협업을 호출하기 위해 사용됩니다.

동기 협업 실행의 경우 커넥터는 Data Handler를 이용하여 다시 클라이언트로 송신할 응답 또는 결함 메시지를 작성합니다. 이 경우 커넥터는 단순히 Business Object(TLO의 하위)를 Data Handler로 전달합니다. Data Handler는 전달되는 Business Object에 근거하여 메시지를 리턴합니다.

요청 처리 개요

협업 대신 커넥터는 HTTP(S)에서 HTTP 서비스를 호출할 수 있습니다. 이 요청 처리 기능은 프로토콜 핸들러 프레임워크에서 지원됩니다. 프로토콜 핸들러 프레임워크는 이 장에서 나중에 자세히 설명하는 HTTP-HTTPS 프로토콜 핸들러로 구성된 구성 가능한 런타임 모듈입니다.

항상 TLO에서 설정되는 협업 요청 Business Object의 수령 시 프로토콜 핸들러 프레임워크는 프로토콜 핸들러를 로드합니다. 프로토콜 핸들러는 HTTP 서비스를 호출하고 (선택적으로) 응답을 보안하며, 다음 세 가지 작업을 수행하는 데 필요한 전송 레벨 세부사항을 관리합니다. 협업 요청 비즈니스를 요청 메시지로 변환, 요청 메시지로 HTTP 서비스 호출, 요청/응답(동기) 모드에 있는 경우 응답 메시지를 비즈니스 오브젝트로 변환하고 해당 오브젝트를 협업으로 리턴합니다.

Connector for HTTP는 항상 TLO를 사용하여 협업에서 호출됩니다. 커넥터는 TLO에서 요청 Business Object를 판별하며 비즈니스 오브젝트로 Data Handler를 호출합니다. Data Handler는 HTTP 서비스로 커넥터가 전송한 요청 메시지를 리턴합니다.

동기 실행의 경우 커넥터는 Data Handler를 이용하여 응답 및 결함 메시지를 응답 및 결함 Business Object로 변환합니다. 이 응답/결함을 Business Object로 변환하는 분석을 위한 Business Object 판별에서 Data Handler를 돕기 위해, 커넥터는 특정 메타 정보를 Data Handler에 제공합니다. 특히 커넥터는 호출 TLO의 하위인 모든 응답 및 결함 Business Object의 목록을 작성합니다. 오직 하나의 응답 Business Object와 선택적으로 여러 결함 Business Object가 있어야 합니다. 또한 하나와 오직 하나의 default Business Object가 있을 수 있습니다. default Business Object의 경우, 커넥터는 단순히 default Business Object의 이름을 Data Handler에 알립니다. 다른 결함 Business Object가 이 변환에서 분석되지 않은 경우 최후의 수단으로서 Data Handler가 default Business Object를 분석해야 합니다.

HTTP(S) 서비스

HTTP 서비스는 HTTP 전송 프로토콜을 지원합니다. HTTP는 HTTP 클라이언트가 연결을 열고 요청 메시지를 HTTP 서버로 전송하는 클라이언트 서버 모델을 구체화합니다. 클라이언트 요청 메시지는 HTTP 서비스를 호출하는 것입니다. HTTP 서버는 호출을 포함하는 메시지를 디스패치하며 연결을 닫습니다.

커넥터의 HTTP 및 HTTPS 프로토콜 리스너는 협업에 대한 클라이언트 요청 핸들 시 HTTP 클라이언트-서버 및 요청/응답 모델을 사용합니다. 그러나 HTTP 리스너는 HTTP 서버- 프록시, 매개자 또는 그 외의 것으로 가능하도록 의도되지 않습니다. 호իր HTTP 리스너는 엔터프라이즈 내에서 그리고 방화벽 뒤에서 사용할 엔드포인트로 가능합니다.

이에 따라 별도의 웹 서버나 게이트웨이가 방화벽에서 전개되어 클라이언트 요청을 리스너로 라우트해야 합니다. 자세한 정보는 1 페이지의 제 1 장 『어댑터의 개요』에서 참조하십시오.

동기 HTTP(S) 서비스

커넥터 처리의 perspective에서, 동기 HTTP 서비스는 요청/응답 경로를 따르는 서비스입니다. HTTP 또는 HTTPS 프로토콜 리스너가 HTTP 요청 메시지를 정상적으로 처리하는 경우 본문은 응답과 HTTP 상태 코드 200 OK를 포함합니다. 결함이 리턴되면 본문은 결함 메시지와 상태 코드 500을 포함합니다.

비동기 HTTP(S) 서비스

커넥터 처리의 perspective에서, 비동기 HTTP 서비스는 요청 전용 경로를 따르는 서비스입니다. HTTP 또는 HTTPS 프로토콜 리스너는 정상적으로 요청 전용 조작을 수신하며 처리하며, HTTP 상태 코드 202 Accepted가 생성됩니다. 또한 HTTP 상태 코드 200 OK를 생성하기 위해 커넥터를 구성할 수 있으며, 자세한 정보는 표 32에서 HTTPAsyncResponseCode 등록 정보를 참조하십시오. 결함이 발생하는 경우 HTTP 상태 코드 500이 생성됩니다. 결함 본문이 리턴될 수 있지만 응답은 없습니다.

이벤트 처리

이벤트 처리 중 커넥터는 프로토콜 리스너와 구성된 Data Handler를 사용하여 HTTP 서비스 클라이언트에서 협업으로 조작될 수 있는 Business Object로 요청 메시지를 변환합니다. 프로토콜 리스너는 이벤트 처리에서 결정적인 역할을 합니다.

프로토콜 리스너

HTTP 요청은 HTTP 또는 HTTPS 전송에서 올 수 있습니다. 리스너는 전송 채널에서 그러한 요청의 도착을 모니터링합니다. 두 개의 프로토콜 리스너와 해당 채널이 있습니다.

- HTTP 프로토콜 리스너
- HTTPS 프로토콜 리스너

각각은 전송 시 청취하는 스레드로 구성됩니다. 클라이언트에서 요청 메시지 수신 시 리스너는 프로토콜 리스너 프레임워크로 이벤트를 등록합니다.

프로토콜 리스너 프레임워크는 프로토콜 리스너를 관리하며, 자원이 사용가능하면 요청을 스케줄합니다. 값을 커넥터 특정 등록 정보로 설정할 때 프로토콜 리스너 프레임워크의 양상과 리스너를 구성합니다. 구성할 수 있는 프로토콜 리스너 프레임워크 등록 정보 간에는 다음 항목이 있습니다.

- **WorkerThreadCount** 병렬로 처리할 수 있는 요청 수이며, 프로토콜 리스너 프레임워크에 사용 가능한 총 스레드 수.

- **RequestPoolSize** 프로토콜 리스너 프레임워크로 등록될 수 있는 최대 요청 수. 이 최대 요청을 초과하여 수신하는 경우 더 이상 새 요청을 등록하지 않습니다.

이 두 개의 커넥터 특정 등록 정보는 프로토콜 리스너가 무한한 이벤트로 커넥터를 방해하지 못하게 하는 방식으로 메모리 할당을 제어합니다. 할당 알고리즘은 다음과 같습니다. 언제든지 커넥터는 $WorkerThreadCount + RequestPoolSize$ 와 동일한 총 이벤트 수를 수신할 수 있습니다. 병렬로 $WorkerThreadCount$ 요청 수를 처리할 수 있습니다.

추가 프로토콜 리스너를 프로토콜 리스너 프레임워크에 더할 수 있습니다. 자세한 정보는 73 페이지의 『다중 프로토콜 리스너 작성』 및 64 페이지의 『커넥터 특정 구성 등록 정보』에서 참조하십시오.

HTTP 및 HTTPS 프로토콜 리스너 처리

HTTP(S) 프로토콜 리스너는 연속적으로 클라이언트에서 HTTP(S) 요청을 청취하는 스레드로 구성됩니다. 리스너 스레드는 호스트 및 포트 커넥터 특정 구성(리스너) 등록 정보에서 지정한 호스트 및 포트를 바인드합니다. 다른 구성 등록 정보 ($RequestWaitTimeout$)는 커넥터가 종료했는지 여부를 점검하기 전에 리스너가 요청을 대기하는 동안의 간격을 정의합니다.

그림 14에서는 동기 조작의 HTTP 프로토콜 리스너 처리를 예시합니다.

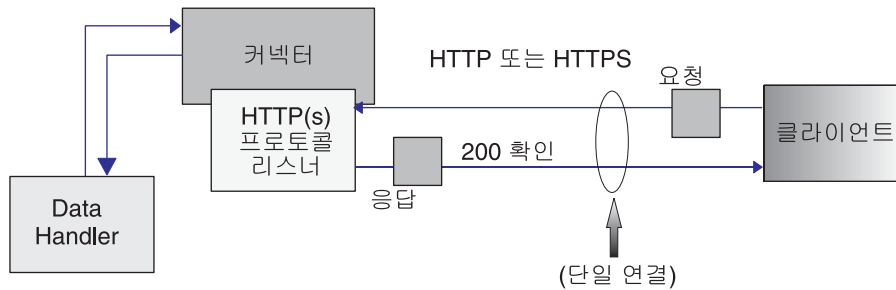


그림 14. HTTP 프로토콜 리스너: 동기 이벤트 처리

그림 15에서는 비동기 조작의 HTTP 프로토콜 리스너 처리를 표시합니다.

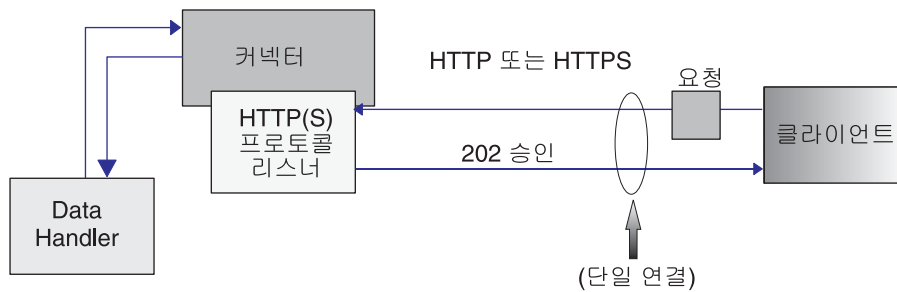


그림 15. HTTP 프로토콜 리스너: 비동기 이벤트 처리

클라이언트가 HTTP 또는 HTTPS 요청을 시작하면, 요청 메시지를 HTTP 또는 HTTPS 리스너에 게시합니다. 클라이언트는 HTTP POST 메시지를 사용하여 프로토콜 리스너 URL을 호출해야 합니다.

HTTP(S) 요청이 도착하면, 리스너는 프로토콜 리스너 프레임워크로 요청을 등록하며, 자원이 사용 가능하게 되면 처리할 이벤트를 스케줄합니다. 그런 다음 리스너는 프로토콜 머리글 및 페이로드를 요청에서 추출합니다.

표 25에서는 인바운드 메시지의 Charset, MimeType, ContentType 및 Content-Type 머리글을 판별하기 위해 리스너가 사용하는 규칙 우선 순위를 요약합니다.

표 25. 인바운드 메시지의 HTTP 프로토콜 리스너 처리

우선 순위	Charset	MimeType	ContentType	Content-Type 머리글
1	수신 HTTP 메시지 Content-Type 머리글 값의 Charset 매개변수 값	이 리스너의 URLsConfiguration 컨텍터 등록 정보 값	Content-Type 머리글 값에서 수신 HTTP 메시징 유형/부속 유형 값	수신 HTTP 메시지 Content-Type 머리글
2	이 리스너의 URLsConfiguration 등록 정보 값			
3	요청 메시지 ContentType의 유형이 부속 유형(예: text/xml, text/plain 등)을 가진 text인 경우, 기본값은 ISO-8859-1입니다. 그렇지 않으면 문자 세트가 사용됩니다.	기본값 ContentType		

표 25에서와 같이

- 프로토콜 리스너는 다음 규칙에 따라 인바운드 메시지의 Charset를 결정합니다.
 1. 리스너는 HTTP 메시지 Content-Type 머리글 값의 Charset 매개변수에서 Charset를 추출하려고 시도합니다.
 2. Content-Type 머리글에서 Charset 값을 얻지 못한 경우, 프로토콜 리스너는 이 리스너의 URLsConfiguration 등록 정보 값을 읽으려고 시도합니다.

3. 이전 단계에서 설명한 메소드를 사용하여 Charset 값을 얻지 않고 메시지 ContentType의 유형이 부속 유형(예: text/xml, text/plain 등)을 가진 text 인 경우, 리스너는 기본 Charset 값 ISO-8859-1을 사용합니다. 그렇지 않으면 Charset 값은 사용되지 않습니다.
- 리스너는 이 규칙에 따라 응답 메시지의 MIMEType을 결정합니다.
 1. 수신 요청 메시지에서 사용하는 URL의 TransformationRules를 구성하고 요청 ContentType이 TransformationRule의 ContentType과 일치하는 경우, 리스너는 TransformationRule을 사용하여 요청 Business Object로의 요청 메시지의 변환을 위해 MIMEType을 추출합니다. 리스너는 요청 URL의 URLConfiguration 등록 정보에서 ContentType 값(예를 들어, text/xml)에 근거하여 완전한 TransformationRule 일치를 찾으려고 시도합니다.
 2. 실패하는 경우 리스너는 요청 URL 아래에서 둘 이상의 ContentType에 적용하는 TransformationRule을 찾으려고 시도합니다.
 3. 이전 모든 단계가 MIMEType을 판별하는 데 실패하면, ContentType 값이 MIMEType으로 사용되어 Data Handler를 호출하고 요청 메시지를 요청 Business Object로 변환합니다.
 - 리스너는 수신 HTTP 메시지 Content-Type 머리글에서 유형/부속 유형을 추출하여 ContentType을 판별합니다.
 - 리스너는 수신 HTTP 메시지 Content-Type 머리글에서 Content-Type 머리글을 판별합니다.

협업이 비동기적으로 호출되면, 리스너는 요청 Business Object를 통합 브로커로 전달 하며 HTTP 상태 코드 202 Accepted로 클라이언트에 응답합니다. 이것은 리스너 처리를 끝냅니다.

동기 호출인 경우, 리스너는 동기적으로 협업을 호출합니다. 협업은 응답 Business Object에 응답합니다.

표 26에서는 응답 메시지의 Charset, MIMEType, ContentType 및 Content-Type 머리글 판별 시 리스너가 사용한 규칙의 우선 순위를 요약합니다.

표 26. 아웃바운드 메시지의 HTTP 프로토콜 리스너 처리

우선 순위	Charset	MIMEType	ContentType	Content-Type 머리글
1	프로토콜 ConfigMO Content-Type 머리글	TLO의 MIMEType 등록 정보	프로토콜 ConfigMO Content-Type 머리글	프로토콜 ConfigMO Content-Type 머리글
2	TLO의 Charset 등록 정보 값	요청 및 응답 ContentType이 일치하는 경우에만 요청 메시지 MIMEType.	요청 메시지 ContentType	ContentType 및 Charset를 사용하여 Content-Type 머리글 구성
3	요청 및 응답 ContentType이 일치하는 경우에만 요청 메시지 Charset.	ContentType 값을 MIMEType으로 사용		

표 26. 아웃바운드 메시지의 HTTP 프로토콜 리스너 처리 (계속)

4	ContentType이 text/*이면, 기본값은 ISO-8859-1입니다. 그렇지 않으면 문자 세트가 사용됩니다.			
---	------------------------------------------------------------------	--	--	--

표 26에서와 같이,

- 리스너는 이 규칙에 따라 응답 메시지의 Charset를 결정합니다.
 1. Charset가 응답 Business Object 프로토콜 Config MO에 지정된 경우, 관련 값이 사용됩니다.
 2. Charset 값이 응답 Business Object 프로토콜 Config MO 머리글에 지정되지 않은 경우, 머리글는 Charset가 TLO에 지정되었는지 여부를 검사합니다.
 3. TLO에 Charset가 지정되지 않은 경우 응답은 요청과 동일한 ContentType을 가지며 요청 Charset가 응답에 사용됩니다.
 4. 이전 단계가 응답 Charset 값 판별에 실패하고 메시지 ContentType의 유형 부분이 부속 유형(예: text/xml, text/plain 등)을 가진 text인 경우, 리스너는 기본 문자 세트 ISO-8859-1 값을 사용합니다. 그렇지 않으면 Charset 값은 사용되지 않습니다.
- 리스너는 이 규칙에 따라 응답 메시지의 MimeType을 결정합니다.
 1. TLO의 MimeType 속성
 2. TLO MimeType 속성이 누락된 경우, 그리고 요청 및 응답 ContentType이 일치하는 경우 리스너는 응답 메시지의 요청 MimeType을 사용합니다.
 3. 그렇지 않으면 리스너는 ContentType 값을 MimeType으로 사용합니다.
- 리스너는 이 규칙에 따라 응답 메시지의 ContentType을 결정합니다.
 1. Content-Type 머리글이 응답 Business Object 프로토콜 Config MO에 지정된 경우 Content-Type 머리글의 유형/부속 유형 부분이 ContentType으로 사용됩니다.
 2. Content-Type 머리글이 응답 Business Object 프로토콜 Config MO에서 지정되지 않은 경우, 리스너는 결정된 ContentType 및 Charset를 사용하여 Content-Type 머리글을 구성합니다(Charset가 응답 메시지로 결정된 경우).

리스너는 HTTP 프로토콜 Config MO를 처리합니다. HTTP 프로토콜 Config MO에 전달된 머리글 값이 요청-응답 이벤트의 컨텍스트에서 올바른지 확인하는 것은 협업의 책임입니다. 리스너는 다음 규칙에 따라 표준 머리글 및 사용자 정의 등록 정보를 채웁니다.

1. 리스너는 특정 속성(예: ObjectEventId)을 무시하기 위해 HTTP 프로토콜 Config MO의 각 항목을 조사합니다.

2. 비어 있지 않은 각 머리글은 전송 메시지로 넣어지며 추가 처리(예: Content-Type 머리글)가 발생합니다.
3. 위의 접근 방식에서 리스너는 메시지에 대한 비표준 머리글을 설정할 수 있지만 메시지가 논리적으로 또는 의미적으로 올바른지 검사하지 않습니다.
4. HTTP 프로토콜 Config MO UserDefinedProperties 속성에 하나 이상의 사용자 정의 등록 정보가 있는 경우, 리스너는 엔티티 머리글 섹션(마지막 머리글 섹션)에 추가합니다. 사용자 정의 등록 정보에 대한 자세한 정보는 28 페이지의 『이벤트 처리를 위한 사용자 정의 등록 정보』에서 참조하십시오.

주: HTTP 프로토콜 Config MO에서 다음 머리글을 지정하는 것은 잘못된 HTTP 메시지 결과를 초래하기 쉽습니다.
 Connection, Trailer, Transfer-Encoding, Content-Encoding, Content-Length, Content-MD5, Content-Range.

그런 다음 리스너는 협업에서 리턴한 응답 Business Object를 응답 메시지로 변환하기 위해 Data Handler를 호출합니다.

리스너는 응답 메시지를 클라이언트로 전달하며 200 OK HTTP 상태 코드를 포함합니다. 협업이 결함 Business Object를 리턴하는 경우 결함 메시지로 변환됩니다. 이 결함 메시지는 500 내부 서버 오류 HTTP 코드로 클라이언트에 전달됩니다.

그런 다음 리스너는 연결을 닫고 이벤트를 처리한 스레드가 사용 가능하게 됩니다.

지원되지 않는 HTTP 프로토콜 리스너 처리 기능

HTTP 프로토콜 리스너는 다음을 지원하지 않습니다.

- 캐싱: 프로토콜 리스너는 HTTP 스펙(RFC2616)에 정의된 대로 캐싱 기능을 수행하지 않습니다.
- 프록시: 프로토콜 리스너는 HTTP 스펙(RFC2616)에 정의된 대로 프록시 기능을 수행하지 않습니다.
- 지속적 연결: 프로토콜 리스너는 HTTP 스펙(RFC2616)에 정의된 대로 지속적 연결을 지원하지 않습니다. 대신 프로토콜 리스너는 각 HTTP 연결의 범위가 단일 클라이언트 요청임을 가정하며 서비스 요청이 완료되면 연결을 닫습니다. 프로토콜 리스너는 서비스 호출에서 연결을 재사용하도록 시도하지 않습니다.
- 방향변경: 프로토콜 리스너는 방향변경을 지원하지 않습니다.
- 큰 파일 전송: 프로토콜 리스너는 큰 파일 전송에는 사용될 수 없습니다. 대신 참조 별 큰 파일을 전달하는 것을 고려할 수 있습니다.
- 상태 관리: 프로토콜 리스너는 RFC2965로 기술한 HTTP 상태 관리 메커니즘을 지원하지 않습니다.
- 쿠키: 프로토콜 리스너는 쿠키를 지원하지 않습니다.

보안 소켓을 사용하여 HTTPS 리스너 처리

HTTPS 프로토콜 리스너 처리는 HTTPS가 보안 소켓을 사용하는 것만 제외하고 HTTP 프로토콜 리스너 처리 섹션에서 설명한 것과 동일합니다. 자세한 정보는 61 페이지의 『SSL』에서 참조하십시오.

이벤트 지속성 및 전달

이벤트 지속성은 프로토콜에 따릅니다.

- **HTTP** 프로토콜 리스너 지속성이 없으므로 전달이 보장되지 않습니다
- **HTTPS** 프로토콜 리스너 지속성이 없으므로 전달이 보장되지 않습니다

이벤트 순서 지정

커넥터는 임의 순서로 이벤트를 전달할 수 있습니다.

이벤트 트리거링

이벤트 트리거링 메커니즘은 프로토콜 리스너가 구성되는 방식에 따릅니다.

- **HTTP** 프로토콜 리스너 HTTP 연결 요청의 ServerSocket에서 청취가 일어납니다.
- **HTTPS** 프로토콜 리스너 청취는 HTTPS 연결 요청의 보안 ServerSocket 계층에서 일어납니다.

주: 커넥터는 작성, 갱신, 검색 또는 삭제 사이에 구별하지 않습니다. 그러한 모든 이벤트는 동일한 접근을 따릅니다.

이벤트 발견

이벤트 발견은 각 프로토콜 리스너에서 수행됩니다. 이벤트 발견 메커니즘은 완전히 전송과 각 리스너의 커넥터 특정 등록 정보를 구성하는 방법에 의존합니다. 이 등록 정보에 대한 자세한 정보는 64 페이지의 『커넥터 특정 구성 등록 정보』에서 참조하십시오.

이벤트 상태

이벤트 상태는 프로토콜 리스너에 의해 관리되며 전송과 리스너 구성 방식에 의존합니다.

- **HTTP** 프로토콜 리스너 HTTP는 본래 비지속적이며 동기적입니다. 이에 따라 이벤트 상태가 유지보수되지 않습니다.
- **HTTPS** 프로토콜 리스너 HTTP는 본래 비지속적이며 동기적입니다. 이에 따라 이벤트 상태가 유지보수되지 않습니다.

이벤트 검색

이벤트 검색은 프로토콜 리스너에 의해 관리되며 전송과 리스너 구성 방식에 의존합니다.

- **HTTP** 프로토콜 리스너 이벤트는 소켓에서 HTTP 요청을 추출하여 검색됩니다.

- **HTTPS** 프로토콜 리스너 이벤트는 소켓에서 HTTP 요청을 추출하여 검색됩니다.

이벤트 아카이브

이벤트 아카이브는 프로토콜 리스너에 의해 관리되며 전송과 리스너 구성 방식에 의존합니다.

- **HTTP** 프로토콜 리스너 전송의 비지속적 및 동기 특성 때문에 아카이브를 수행하지 않습니다.
- **HTTPS** 프로토콜 리스너 전송의 비지속적 및 동기 특성 때문에 아카이브를 수행하지 않습니다.

이벤트 복구

이벤트 복구는 프로토콜 리스너에 의해 관리되며 전송과 리스너 구성 방식에 의존합니다.

- **HTTP** 프로토콜 리스너 전송의 비지속적 특성 때문에 이벤트 복구를 수행하지 않습니다.
- **HTTPS** 프로토콜 리스너 전송의 비지속적 특성 때문에 이벤트 복구를 수행하지 않습니다.

요청 처리

HTTP 서비스의 호출을 위한 협업이 가능하도록 커넥터의 요청 처리 기능을 사용할 수 있습니다. 커넥터 및 관련 요청 처리 구성요소를 구성해야 합니다(프로토콜 핸들러 프레임워크 및 프로토콜 핸들러).

실행 시간에 커넥터는 Business Object 양식의 협업에서 요청을 수신합니다. Business Object(요청, 선택적으로 응답 및 결합 Business Object)는 HTTP 서비스를 사용하도록 구성된 협업에 의해 발생된 TLO에 의해 포함됩니다. TLO 및 하위 Business Object는 처리 모드(동기 또는 비동기)를 지정하는 ASI와 속성, Data Handler MIME 유형, 사용할 프로토콜 핸들러, 대상 주소를 포함합니다. 프로토콜 핸들러는 이 정보를 사용하여 Data Handler의 인스턴스를 호출하며, 요청 Business Object를 요청 메시지로 변환하며 대상 HTTP 서비스를 호출합니다. 모드가 동기이면, 프로토콜 핸들러는 다시 Data Handler를 호출하여 응답 메시지를 응답 Business Object로 변환하고 이를 협업으로 리턴합니다.

요청 메시지에 응답하여, 커넥터는 원격 거래 파트너로부터 다음을 수신할 수 있습니다.

- 데이터를 포함하는 응답 메시지
- 결합 정보를 포함하는 응답 메시지

프로토콜 핸들러는 요청 처리에서 주요 역할을 합니다.

프로토콜 핸들

협업은 HTTP 또는 HTTPS 전송에서 HTTP 서비스를 호출할 수 있습니다. 커넥터는 HTTP 및 HTTPS 서비스를 호출하기 위한 HTTP-HTTPS 프로토콜 핸들러인 하나의 프로토콜 핸들러 및 해당 채널을 갖습니다.

프로토콜 핸들러 프레임워크는 시작 시 로드하는 프로토콜 핸들러를 관리합니다. 커넥터가 요청 Business Object를 수신하면, 요청 스레드(각 협업 요청은 자체 스레드에서 제공됨)는 요청을 처리하기 위해 프로토콜 핸들러 프레임워크를 호출합니다.

프로토콜 핸들러 프레임워크는 사용할 프로토콜 핸들러를 결정하기 위해 TLO 핸들러 속성 ASI를 읽습니다. 일련의 규칙 적용 시(『HTTP-HTTPS 프로토콜 핸들러 처리』 참조), 프로토콜 핸들러는 Data Handler를 호출하여 요청 Business Object를 요청 메시지로 변환합니다. 프로토콜 핸들러는 요청 메시지를 전송(HTTP(S)) 메시지로 패키징합니다.

그런 다음 프로토콜 핸들러는 대상 주소를 결정하기 위해 요청 Business Object 프로토콜 Config MO의 목적지 속성을 읽습니다. 그런 다음 프로토콜은 요청 메시지로 대상 HTTP 서비스를 호출합니다.

ws_mode TLO ASI를 읽음으로서 프로토콜 핸들러는 처리 모드가 동기 또는 비동기인지 여부를 판별합니다. 이 ASI가 asynch로 설정된 경우, 프로토콜 핸들러 처리가 완료됩니다. 그렇지 않으면 프로토콜 핸들러는 응답 메시지를 대기합니다. 응답 메시지가 도착하면 프로토콜 핸들러는 프로토콜 머리글 및 페이로드를 추출합니다. 그리고 Data Handler(MimeType TLO 속성으로 표시됨)를 호출하여 메시지를 응답이나 결합 Business Object로 변환합니다. 프로토콜 Config MO를 다시 사용하여 프로토콜 핸들러는 Business Object에 프로토콜 머리글을 설정합니다. 그런 다음 프로토콜 핸들러는 응답이나 결합 Business Object를 협업으로 리턴합니다.

커넥터 구성에 따라 커넥터에 플러그인되는 하나 이상의 프로토콜 핸들러가 있을 수 있습니다. 커넥터 특정 등록 정보는 프로토콜 핸들러를 구성할 수 있게 합니다.

HTTP-HTTPS 프로토콜 핸들러 처리

HTTP-HTTPS 프로토콜 핸들러는 이 섹션에 있는 예외를 가지고 『프로토콜 핸들』에서 설명한 대로 수행합니다. 그림 16에서는 동기 조작을 위한 HTTP-HTTPS 프로토콜 핸들러를 표시합니다.

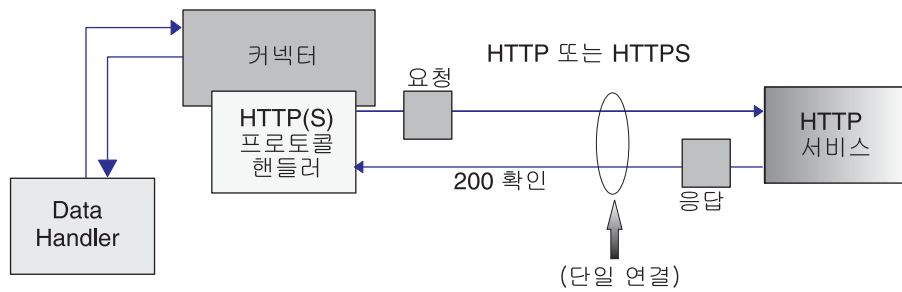


그림 16. HTTP-HTTPS 프로토콜 핸들러: 동기 요청 처리

그림 17에서는 비동기 요청 프로세스를 위한 HTTP-HTTPS 프로토콜 핸들러를 표시합니다.

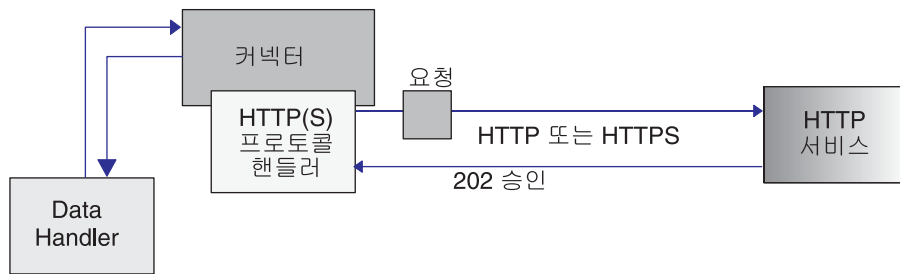


그림 17. HTTP-HTTPS 프로토콜 핸들러: 비동기 요청 처리

주: 이 섹션에서는 HTTP 프로토콜 핸들링에 대해서만 설명합니다.

HTTP-HTTPS 프로토콜 핸들러는 프로토콜 Config MO를 판별하기 위해 요청 Business Object의 오브젝트 레벨 ASI(cw_mo_http)를 사용합니다. HTTP-HTTPS 프로토콜 핸들러는 HTTP 프로토콜 Config MO에서 목적지 속성을 읽어 대상 HTTP 서비스의 URL을 판별합니다. URL이 누락되었거나 불완전한 경우 프로토콜 핸들러는 서비스 호출에 실패합니다. HTTP 프로토콜 Config MO 및 관련 속성에 대한 자세한 정보는 37 페이지의 『요청 처리를 위한 HTTP 프로토콜 Config MO』의 내용을 참조하십시오.

HTTP-HTTPS 프로토콜 핸들러는 Data Handler가 리턴한 요청 메시지를 사용하여 HTTP 서비스를 호출합니다. HTTP 프록시 커넥터 구성 등록 정보가 지정되면, HTTP-HTTPS 프로토콜 핸들러는 이에 따라 작동합니다. 응답이 리턴되면 HTTP-HTTPS 프로토콜 핸들러가 이를 읽습니다.

58 페이지의 표 27에서는 전송 요청 메시지의 Charset, MimeType, ContentType 및 Content-Type 머리를 판별하기 위해 HTTP-HTTPS 프로토콜 핸들러가 사용하는 규칙의 우선 순위를 요약합니다.

표 27. 아웃바운드 메시지의 HTTP-HTTPS 프로토콜 핸들러 처리 규칙

우선 순위	Charset	MimeType	ContentType	Content-Type 머리글
1	프로토콜 Config MO의 Content-Type 머리글	TLO 속성의 MimeType 등록 정보	프로토콜 Config MO의 Content-Type 머리글	프로토콜 Config MO의 Content-Type 머리글
2	TLO 속성의 Charset 등록 정보	기본값 ContentType		
3	ContentType이 text/*이면, 기본값은 ISO-8859-1입니다. 그렇지 않으면 Charset가 사용됩니다.			

표 27에서와 같이,

- HTTP-HTTPS 프로토콜 핸들러는 이 규칙에 따라 응답 메시지의 Charset를 결정합니다.
 1. 요청 Business Object 프로토콜 Config MO 머리글에 지정된 경우, Charset 값이 사용됩니다.
 2. Charset가 이전 단계에서 결정되지 않은 경우, 프로토콜 핸들러는 TLO 속성에서 Charset를 추출하려고 시도합니다.
 3. 이전 단계에서 설명한 조작이 성공적이지 않은 경우, Charset를 결정하기 위해 테이블이 사용됩니다.

표 28. 기본값 요청 처리 Charset

ContentType	기본값 Charset
text/*	ISO-8859-1 자세한 정보는 RFC2616을 참조하십시오.
application/*	기본값 없음
기타 모두	기본값 없음

4. Charset를 이전 단계에서 결정하지 않은 경우 Charset가 Data Handler에 설정됩니다.
 5. 요청을 작성하는 데 필요한 데이터 구조에 따라 Data Handler는 스트림 또는 바이트 배열 API로 호출됩니다.
- HTTP-HTTPS 프로토콜 핸들러는 이 규칙에 따라 요청의 MimeType을 결정합니다.
 1. TLO MimeType 속성.
 2. TLO MimeType 속성이 누락된 경우 프로토콜 핸들러는 ContentType을 사용하여 MimeType을 결정합니다.
 - HTTP-HTTPS 프로토콜 핸들러는 이 규칙에 따라 요청 메시지의 ContentType을 결정합니다.
 - Content-Type 머리글이 요청 Business Object 프로토콜 Config MO에 지정된 경우, 머리글의 유형/부속 유형이 ContentType으로 사용됩니다.

- HTTP-HTTPS 프로토콜 핸들러는 이 규칙에 따라 요청 메시지의 Content-Type을 결정합니다.
 - Content-Type 머리글이 요청 Business Object 프로토콜 Config MO에 지정된 경우, 관련 값이 전송 메시지에 설정됩니다.

표 29에서는 응답 메시지의 Charset, MimeType, ContentType 및 Content-Type 머리글 판별 시 핸들러가 사용한 규칙의 우선 순위를 요약합니다.

표 29. 인바운드 메시지의 HTTP 프로토콜 리스너 처리 규칙

우선 순위	Charset	MimeType	ContentType	Content-Type 머리글
1	수신 HTTP 메시지 Content-Type 머리글 값의 Charset 매개변수 값	요청 Business Object의 프로토콜 Config MO의 메시지 TransformationMap 하위 Business Object	Content-Type 머리글 값에서 수신 HTTP 메시징 유형/부속 유형 값	수신 HTTP 메시지 Content-Type 머리글
2	요청 Business Object의 프로토콜 Config MO의 메시지 TransformationMap 하위 Business Object	요청 및 응답 ContentType이 일치하는 경우에만 요청 메시지 MimeType.		
3	요청 및 응답 ContentType이 일치하는 경우에만 요청 메시지 Charset.	TLO의 MimeType 등록 정보		
4	TLO의 Charset 등록 정보	기본값 ContentType		
5	Content-Type이 text/*이면, 기본값은 ISO-8859-1입니다. 그렇지 않으면 Charset가 사용되지 않습니다.			

표 29에서와 같이,

- 프로토콜 핸들러는 다음 규칙에 따라 동기 응답 메시지의 Charset를 결정합니다.
 1. Charset 매개변수가 수신 응답 메시지의 Content-Type 머리글에 설정된 경우, 프로토콜 핸들러는 Data Handler에 설정된 Charset 값을 사용합니다.
 2. 응답 메시지 머리글에 Charset 값이 없으면, 프로토콜 핸들러는 TLO 요청 프로토콜 Config MO MessageTransformationMap에서 협업 정의 Charset를 읽어려고 시도합니다.
 3. 주어진 요청의 MessageTransformationMap에 Charset가 지정되지 않은 경우, 응답은 요청과 동일한 ContentType을 가지며 요청 Charset가 응답에 사용됩니다.
 4. 이전 단계에 Charset 값을 산출하는 데 실패하면 프로토콜 핸들러는 TLO Charset 속성을 읽어려고 시도합니다.
 5. 이전 단계에서 설명한 메소드를 사용하여 Charset 값을 얻지 않고 메시지 ContentType의 유형이 부속 유형(예: text/xml, text/plain 등)을 가진 text인 경우, 기본값 ISO-8859-1을 사용합니다. 그렇지 않으면 Charset 값은 사용되지 않습니다.

- 프로토콜 핸들러는 다음 규칙에 따라 동기 응답 메시지의 MIMEType을 결정합니다.
 1. 프로토콜 핸들러는 우선 TLO 요청 프로토콜 Config MO의 MessageTransformationMap에서 MIMEType을 추출하려고 시도합니다. 특히 프로토콜 핸들러는 MessageTransformationRule을 추출하기 위해 MTM에서 완전 ContentType 일치점을 찾으려고 시도한 후 MIMEType 등록 정보 값을 사용합니다. 그렇지 않으면 프로토콜 핸들러는 둘 이상의 ContentType(ContentType은 */*임)에 적용하는 MessageTransformationRule을 찾습니다.
 2. MIMEType이 MessageTransformationMap을 사용하여 결정되지 않은 경우 프로토콜 핸들러는 요청 및 응답 ContentType이 일치하는 경우에만 응답의 요청 MIMEType을 사용합니다.
 3. 이전 단계를 사용하여 MIMEType을 추출할 수 없는 경우, 프로토콜 핸들러는 TLO의 MIMEType 속성을 사용합니다.
 4. 모든 이전 단계가 실패하면 프로토콜 핸들러는 ContentType을 사용하여 MIMEType을 설정합니다.
- 핸들러는 수신 HTTP 메시지 Content-Type 머리글에서 유형/부속 유형을 추출하여 ContentType을 판별합니다.

핸들러는 HTTP 프로토콜 Config MO를 처리합니다. HTTP 프로토콜 Config MO에 전달된 머리글 값이 요청-응답 이벤트의 컨텍스트에서 올바른지 확인하는 것은 협업의 책임입니다. 핸들러는 다음 규칙에 따라 표준 머리글 및 사용자 정의 등록 정보를 채웁니다.

1. 핸들러는 특정 속성(예: ObjectEventId)을 무시하기 위해 HTTP 프로토콜 Config MO의 각 항목을 조사합니다.
2. 비어 있지 않은 각 머리글은 전송 메시지로 넣어지며 추가 처리(예: Content-Type 머리글)가 발생합니다.
3. 위의 접근 방식에서 핸들러는 메시지에 대한 비표준 머리글을 설정할 수 있지만 메시지가 논리적으로 또는 의미적으로 올바른지 보증하지 않습니다.
4. HTTP 프로토콜 Config MO UserDefinedProperties 속성에 하나 이상의 사용자 정의 등록 정보가 있는 경우, 핸들러는 엔티티 머리글 섹션(마지막 머리글 섹션)에 추가합니다. 사용자 정의 등록 정보에 대한 자세한 정보는 38 페이지의 『요청 처리를 위한 사용자 정의 등록 정보』에서 참조하십시오.

주: HTTP 프로토콜 Config MO에서 다음 머리글을 지정하는 것은 잘못된 HTTP 메시지 결과를 초래하기 쉽습니다.
 Connection, Trailer, Transfer-Encoding, Content-Encoding, Content-Length, Content-MD5, Content-Range.

SSL

이 섹션에서는 커넥터가 SSL 기능을 구현하는 방식에 대해 설명합니다. 배경 정보는 SSL 문서를 참조하십시오. 이 섹션에서는 SSL 기법과의 친숙함을 가정합니다.

JSSE

커넥터는 JSSE를 사용하여 HTTPS 및 SSL의 지원을 제공합니다. IBM JSSE가 커넥터에 제공됩니다. 이 기능을 사용 가능하게 하려면 커넥터에 설치된 파일 가운데 있는 `java.security` 파일에 다음 항목이 있는지 확인하십시오.

```
security.provider.5=com.ibm.jsse.IBMJSSEProvider
```

`java.security`는 커넥터 설치의 `$ProductDir\lib\security` 디렉토리에 위치합니다. 커넥터는 `JavaProtocolHandlerPackages` 커넥터의 값을 사용하여 시스템 프로토콜 `java.protocol.handler.pkgs`를 설정합니다. 커넥터와 함께 제공되는 IBM JSSE의 경우, 이 등록 정보의 값은 `com.ibm.net.ssl.internal.www.protocol`로 설정되어야 합니다.

`JavaProtocolHandlerPackages` 구성 등록 정보의 기본값은 이 값입니다.

설치 중 커넥터는 JSSE가 지원하는 모든 익명의 암호 세트를 사용 불가능하게 합니다.

KeyStore 및 TrustStore

커넥터로 SSL을 사용하려면 keystore 및 truststore를 설정해야 합니다. keystore, 인증서 및 키 생성을 설정하기 위한 어떤 도구도 제공되지 않습니다. 이 작업을 완료하려면 타사 소프트웨어 도구를 사용해야 합니다.

SSL 등록 정보

다음과 같은 SSL 커넥터 특정 등록 정보를 지정할 수 있습니다.

- SSLVersion
- SSLDebug
- KeyStore
- KeyStoreAlias
- KeyStorePassword
- TrustStore
- TrustStorePassword

이 등록 정보가 커넥터 인스턴스에 적용됨을 참고하십시오. 동일한 SSL 등록 정보 값 설정이 커넥터로 플러그인된 모든 HTTPS 프로토콜 리스너와 각 커넥터 인스턴스의 HTTP-HTTPS 프로토콜 핸들러에 의해 사용됩니다. HTTPS/SSL 설정에 대한 자세한 정보는 127 페이지의 부록 D 『HTTPS/SSL 구성』의 내용을 참조하십시오.

SSL 및 HTTPS 프로토콜 리스너

HTTPS 프로토콜 리스너를 사용하려면 SSL 커넥터 특정 등록 정보를 지정해야 합니다. 사용자가 이 등록 정보에 지정한 값은 SSL 요구사항을 반영해야 합니다.

- **SSLVersion** 사용하려면 SSLVersion이 JSSE에 의해 지원되는지 확인하십시오.
- **KeyStore** HTTPS 프로토콜 리스너는 SSL 통신에서 서버로 역할하므로 keystore를 지정해야 합니다. 리스너는 SSL -> KeyStore 구성 등록 정보에 지정한 keystore를 사용합니다. 이 등록 정보 값은 keystore 파일에 대한 전체 경로이어야 합니다. keystore가 커넥터의 키 쌍(개인용 키 및 공용 키)을 가지는지 확인하십시오. 개인용 키의 별명이 SSL -> KeyStoreAlias 등록 정보로 지정되어야 합니다. SSL -> KeyStorePassword 등록 정보로 keystore를 액세스하려면 필요한 암호를 지정해야 합니다. 또한 keystore를 액세스하는 데 필요한 암호와 개인용 키(keystore에서)가 동일한지 확인하십시오. 마지막으로 커넥터를 인증할 수 있도록 커넥터의 디지털 인증을 사용자 클라이언트에 분배해야 합니다.
- **TrustStore** HTTPS 프로토콜 리스너가 클라이언트를 인증하게 하려는 경우, 클라이언트 인증을 활성화해야 합니다. SSL -> UseClientAuth 등록 정보를 true로 설정하여 이를 수행합니다. 또한 다음을 지정해야 합니다.

- SSL -> TrustStore 구성 등록 정보의 값으로 truststore의 위치
- SSL -> TrustStorePassword 구성 등록 정보의 값으로 truststore에 액세스하는 데 필요한 암호

truststore가 클라이언트의 디지털 인증을 포함하는지 확인하십시오. 클라이언트가 사용하는 디지털 인증은 CA가 발행하거나 자체 서명할 수 있습니다. truststore가 CA의 루트 인증을 신뢰하는 경우 JSSE는 해당 CA가 발행한 모든 디지털 인증서를 인증합니다.

HTTPS/SSL 설정에 대한 자세한 정보는 127 페이지의 부록 D 『HTTPS/SSL 구성』의 내용을 참조하십시오.

SSL 및 HTTP-HTTPS 프로토콜 핸들러

HTTP-HTTPS 프로토콜 핸들러로 SSL을 사용하려면 경우 SSL 커넥터 특정 등록 정보를 지정해야 합니다. 이 등록 정보에 지정하는 값은 HTTP 프로바이더의 요구사항을 반영합니다.

- **SSLVersion** 사용하려면 SSLVersion이 프로바이더 및 JSSE에 의해 지원되는지 확인하십시오.
- **TrustStore** HTTP-HTTPS 프로토콜 핸들러는 SSL 통신에서 클라이언트로 역할하므로 truststore를 설정해야 합니다. 핸들러는 SSL -> Truststore 지정 등록 정보에서 지정한 truststore를 사용합니다. 이 등록 정보 값은 truststore 파일에 대한 전체 경로이어야 합니다. SSL -> TrustStorePassword 등록 정보에서 truststore를 액세스하는 데 필요한 암호를 지정해야 합니다. truststore가 프로바이더의 디지털 인증을 포함하는지 확인하십시오. 프로바이더가 사용하는 디지털 인증은 CA가 발행하거나 자

체 서명할 수 있습니다. truststore가 CA의 루트 인증을 신뢰하는 경우 JSSE는 해당 CA가 발행한 모든 디지털 인증서를 인증합니다.

- **KeyStore** HTTP 서비스 프로바이더가 클라이언트 인증을 필요로 하는 경우, keystore를 설정해야 합니다. HTTP-HTTPS 프로토콜 핸들러는 SSL -> KeyStore 구성 등록 정보에 지정한 keystore를 사용합니다. 이 값은 keystore 파일에 대한 전체 경로이어야 합니다. keystore가 커넥터용으로 구성된 키 쌍(개인용 키 및 공용 키)을 가지는지 확인하십시오. 개인용 키의 별명이 SSL -> KeyStoreAlias 등록 정보에 지정되어야 합니다. keystore를 액세스하는 데 필요한 암호는 SSL -> KeyStorePassword 등록 정보에 지정되어야 합니다. 마지막으로 keystore를 액세스하는 데 필요한 암호와 개인용 키(keystore에서)가 동일한지 확인하십시오. 인증을 위한 HTTP 서비스 프로바이더로 커넥터의 디지털 인증을 분배해야 합니다.

HTTPS/SSL 설정에 대한 자세한 정보는 127 페이지의 부록 D 『HTTPS/SSL 구성』의 내용을 참조하십시오.

커넥터 구성

커넥터 파일을 시스템에 설치하기 위해 설치 프로그램을 사용한 후에 표준 및 응용프로그램 특정 커넥터 구성 등록 정보를 설정해야 합니다.

구성 등록 정보 설정

커넥터는 두 가지 유형의 구성 등록 정보를 갖습니다. 표준 등록 정보 및 커넥터 특정 구성 등록 정보입니다. 커넥터를 실행하기 전에 System Manager(SM)을 사용하여 이 등록 정보의 값을 설정해야 합니다.

표준 구성 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이 등록 정보의 문서는 79 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』에서 참조하십시오. 아래 테이블은 부록에서 구성 등록 정보에 대해 이 커넥터에 특정한 정보를 제공합니다.

등록 정보	설명
CharacterEncoding	이 커넥터는 이 등록 정보를 사용하지 않습니다.
로케일	이 커넥터는 국제화되지 않았으므로 이 등록 정보의 값을 변경할 수 없습니다. 현재 지원되는 로케일을 판별하려면 커넥터의 릴리스 정보를 참조하십시오.

이 커넥터는 통합 브로커로 InterChange Server(ICS)만을 지원하므로, 이와 관련된 유일한 구성 등록 정보는 ICS에 대한 것입니다.

최소한 다음과 같은 표준 커넥터 구성 등록 정보를 설정해야 합니다.

- AgentTraceLevel
- ApplicationName

- ControllerTraceLevel
- DeliveryTransport

커넥터 특정 구성 등록 정보

커넥터 특정 구성 등록 정보는 런타임에 커넥터 에이전트가 필요로 하는 정보를 제공합니다. 또한 커넥터 특정 등록 정보는 에이전트를 재코딩 및 재빌드하지 않고 커넥터 에이전트에서 정적 정보나 논리를 변경하는 방식을 제공합니다.

표 30에서는 커넥터 특정 구성 등록 정보를 나열합니다. 등록 정보의 설명은 다음에 오는 섹션을 참조하십시오. 일부 등록 정보는 다른 등록 정보에 포함된 것을 참고하십시오. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 30. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수
DataHandlerMetaObjectName	<i>Data Handler Meta Object</i> 이름	MO_DataHandler_Default	예
JavaProtocolHandlerPackages	올바른 <i>Java</i> 프로토콜 핸들러 패키지	com.ibm.net.ssl.internal. www.protocol	아니오
ProtocolHandlerFramework	이것은 계층 구조 등록 정보이며 값이 없 습니다	없음	아니오
+ProtocolHandlers	이것은 계층 구조 등록 정보이며 값이 없 습니다		아니오
++Handler1	이것은 계층 구조 등록 정보입니다. 부속 등록 정보에 대한 정보는 65 페이지의 『Handler1』의 내용을 참조하십시오		예
ProtocolListenerFramework	이것은 계층 구조 등록 정보이며 값이 없 습니다		아니오
+WorkerThreadCount	사용 가능한 리스너 스레드의 수를 제공 하는 1보다 큰 정수	10	아니오
+RequestPoolSize	자원 풀 크기를 제공하는 <i>WorkerThreadCount</i> 보다 큰 정수	20	아니오
+ProtocolListeners	이것은 계층 구조 등록 정보이며 값이 없 습니다		
++Listener1	고유하게 이름 지정된 프로토콜 리스너		예
+++프로토콜	http 또는 https		예
+++ListenerSpecific	리스너가 필요로 하거나 고유한 등록 정 보 66 페이지의 『ListenerSpecific』에서 참조하십시오		
ProxyServer	이것은 계층 구조 등록 정보이며 값이 없 습니다		아니오
+HttpProxyHost	<i>HTTP</i> 프록시 서버의 호스트 이름		아니오
+HttpProxyPort	<i>HTTP</i> 프록시 서버의 포트 번호	80	아니오
+HttpNonProxyHosts	직접 연결을 필요로 하는 <i>HTTP</i> 호스트		아니오
+HttpsProxyHost	<i>HTTPS</i> 프록시 서버의 호스트 이름		아니오
+HttpsProxyPort	<i>HTTPS</i> 프록시 서버의 포트 번호	443	아니오
+HttpsNonProxyHosts	직접 연결을 필요로 하는 <i>HTTPS</i> 호스트		아니오
+SocksProxyHost	<i>Socks</i> 프록시 서버 이름		아니오
+SocksProxyPort	<i>Socks</i> 프록시 서버 포트		아니오
+HttpProxyUsername	<i>Http</i> 프록시 서버 사용자 이름		아니오
+HttpProxyPassword	<i>Http</i> 프록시 서버 암호		아니오

표 30. 커넥터 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
+HttpsProxyUsername	Https 프록시 서버 사용자 이름		아니오
+HttpsProxyPassword	Https 프록시 서버 암호		아니오
SSL	이것은 계층 구조 등록 정보이며 값이 없습니다		아니오
+SSLVersion	SSL, SSLv2, SSLv3, TLS, TLSv1	SSL	아니오
+SSLDebug	true, false	false	아니오
+KeyStoreType	임의의 올바른 keystore 유형	JKS	아니오
+KeyStore	KeyStore 파일에 대한 경로.		아니오
+KeyStorePassword	KeyStore에서 개인용 키의 암호		아니오
+KeyStoreAlias	KeyStore에서 키 쌍의 별명		아니오
+TrustStore	TrustStore 파일에 대한 경로		아니오
+TrustStorePassword	TrustStore의 암호		아니오
+UseClientAuth	true false	false	아니오

DataHandlerMetaObjectName: 이것은 Data Handler가 구성 등록 정보를 설정하기 위해 사용하는 Meta Object의 이름입니다.

Default = M0_DataHandler_Default.

JavaProtocolHandlerPackages: 이 등록 정보의 값은 Java 프로토콜 핸들러 패키지를 제공합니다. 커넥터는 이 등록 정보의 값을 사용하여 시스템 프로토콜 `java.protocol.handler.pkgs`를 설정합니다.

Default = `com.ibm.net.ssl.internal.www.protocol`.

ProtocolHandlerFramework: 프로토콜 핸들러 프레임워크는 이 등록 정보를 사용하여 프로토콜 핸들러를 로드하고 구성합니다. 이것은 계층 구조 등록 정보이며 값이 없습니다.

기본값 = 없음

ProtocolHandlers: 이 계층 구조 등록 정보에는 값이 없습니다. 첫 번째 하위 레벨은 분리 프로토콜 핸들러를 표시합니다.

기본값 = 없음

Handler1: HTTP-HTTPS 프로토콜 핸들러의 이름. 이것은 계층 구조 등록 정보를 참고하십시오. 리스너와 달리 프로토콜 핸들러는 중복될 수 없으며 각 프로토콜에는 오직 하나의 핸들러가 있을 수 있습니다. 아래 표 31에서는 HTTP-HTTPS 프로토콜 핸들러의 부속 등록 정보를 표시합니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 31. HTTP-HTTPS 프로토콜 핸들러 구성 등록 정보

이름	가능한 값	기본값	필수
++HTTPHandler	이것은 계층 구조 등록 정보이며 값이 없습니다.		예
+++Protocol	프로토콜 핸들러 종류를 구현 중입니다. HTTP 및 HTTPS http의 경우, 값은 http입니다. 주: 이 등록 정보의 값을 지정하지 않은 경우 커넥터는 이 프로토콜 핸들러를 초기화하지 않습니다.		예
+++HTTPReadTimeout	원격 호스트에서 읽는 동안 제한시간 간격(밀리초)을 지정 하는 HTTP 특정 등록 정보 이 등록 정보가 지정되지 않거나 0으로 설정된 경우, HTTP-HTTPS 프로토콜 핸들러는 원격 호스트에서 읽는 동안 무한히 차단합니다.	0	아니오

ProtocolListenerFramework: 프로토콜 리스너 프레임워크는 프로토콜 리스너를 로드하기 위해 이 등록 정보를 사용합니다. 이것은 계층 구조 등록 정보이며 값이 없습니다.

WorkerThreadCount: 1보다 큰 정수이어야 하는 이 등록 정보는 프로토콜 리스너 프레임워크에 사용 가능한 프로토콜 워커 스레드의 수를 설정합니다. 자세한 정보는 48 페이지의 『프로토콜 리스너』에서 참조하십시오.

Default = 10.

RequestPoolSize: WorkerThreadCount 보다 큰 정수이어야 하는 이 등록 정보는 프로토콜 리스너 프레임워크의 자원 풀 크기를 설정합니다. 프레임워크는 동시에 최대 WorkerThreadCount + RequestPoolSize 요청 수를 처리할 수 있습니다.

Default = 20.

ProtocolListeners: 이것은 계층 구조 등록 정보이며 값이 없습니다. 이 등록 정보의 각 첫 번째 하위 레벨은 분리 프로토콜 리스너를 표시합니다.

Listener1: 프로토콜 리스너의 이름. 다중 프로토콜 리스너가 있을 수 있습니다. 이것은 계층 구조 등록 정보임을 참고하십시오. 이 등록 정보의 다중 인스턴스를 작성하고 추가적으로 고유하게 이름이 지정된 리스너를 작성할 수 있습니다. 그렇게 수행할 때 리스너 특정 등록 정보는 변경할 수 있지만 프로토콜 등록 정보는 변경할 수 없습니다. 다중 리스너의 이름은 고유해야 합니다.

가능한 이름(값이 아님): HTTPListener1, HTTPSListener1.

프로토콜: 이 등록 정보는 이 리스너가 구현 중인 프로토콜을 지정합니다.

가능한 값: http, https.

주: 이 등록 정보의 값을 지정하지 않은 경우 커넥터는 이 프로토콜 리스너를 초기화하지 않습니다.

ListenerSpecific: 리스너 특정 등록 정보는 지정된 프로토콜 리스너에 고유하거나 필수입니다. 예를 들어 HTTP 리스너는 리스너 모니터가 요청하는 포트 번호를 표시하

는 리스너 특정 등록 정보 포트를 갖습니다. 표 32에서는 HTTP-HTTPS 리스너 특정 등록 정보를 요약합니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 32. HTTP 및 HTTPS 프로토콜 리스너 특정 구성 등록 정보

이름	가능한 값	기본값	필수
+++HTTPListener1	HTTP 프로토콜 리스너의 고유한 이름. 이것은 ProtocolListenerFramework -> ProtocolListeners 계층 구조 등록 정보입니다. 여러 리스너가 있을 수 있습니다. 이 등록 정보와 계층 구조의 다른 인스턴스를 작성하여 추가 HTTP 리스너를 플러그인할 수 있습니다.		예
++++Protocol	http HTTP 프로토콜 리스너인 경우 https HTTPS 프로토콜 리스너인 경우 주: 이 등록 정보의 값을 지정하지 않은 경우 커넥터는 이 프로토콜 리스너를 초기화하지 않습니다.		예
++++BOPrefix	이 등록 정보 값이 Data Handler에 전달됩니다.		아니오
++++Host	리스너는 이 등록 정보의 값으로 지정한 IP 주소에서 청취합니다. 호스트가 지정되지 않은 경우, 기본값은 localhost입니다. 리스너가 실행 중인 시스템의 IP 주소나 호스트 이름(DNS 이름)을 지정할 수 있습니다. 시스템은 여러 IP 주소나 여러 이름을 가질 수 있습니다.	localhost	아니오
++++Port	리스너가 요청에 대해 청취하는 포트 지정되지 않은 경우 HTTP 리스너의 HTTP의 경우 포트 기본값은 80이며 HTTPS의 경우 포트 기본값은 443입니다. 커넥터에서 리스너를 복제하는 경우 호스트 및 포트 등록 정보의 조합은 고유하거나 리스너가 요청을 승인하기 위해 포트에 바인딩할 수 없습니다.	HTTP 리스너의 경우 80 HTTPS 리스너의 경우 443	아니오
++++SocketQueueLength	수신 연결 요청의 경우 대기열(소켓 대기열)의 길이. 호스트가 연결을 거부하기 전에 한 번에 수신 연결이 저장되는 수를 지정합니다. 최대 대기열 길이는 운영 체제에 따릅니다.	호스 5	아니오
++++RequestWaitTimeout	요청이 도착하기를 대기하는 동안 리스너 스레드가 클라이언트에서 차단하는 시간 간격(밀리초). 이 간격 전에 요청을 수신하는 경우 리스너가 처리합니다. 그렇지 않은 경우 리스너 스레드는 커넥터 종료 플래그가 설정되었는지 여부를 확인합니다. 설정된 경우 커넥터가 종료합니다. 그렇지 않은 경우 RequestWaitTimeout 간격 동안 계속 차단합니다. 이 등록 정보가 0으로 설정된 경우 영원히 차단합니다. 지정된 경우 기본값은 60000ms입니다.	60000 (ms)	아니오
++++HTTPReadTimeout	클라이언트에서 요청을 읽는 동안 리스너가 차단되는 시간 간격(밀리초). 이 매개변수가 0으로 설정되면 전체 요청 메시지를 수신할 때까지 리스너는 무한히 차단합니다.	0	아니오
++++HttpAsyncResponseCode	리스너에 대한 비동기 요청의 HTTP 응답 코드: 200 (OK) 202 (ACCEPTED)	202 (ACCEPTED)	아니오

표 32. HTTP 및 HTTPS 프로토콜 리스너 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
++++URLsConfiguration	이것은 계층 구조 등록 정보이며 값이 없습니다. 이 리스너가 지원하는 URL에 대한 하나 이상의 구성과, 선택적으로 유형과 Charset 값이 들어 있습니다. 이것이 ProtocolListenerFramework -> ProtocolListeners -> HTTPListener1 계층 구조 등록 정보의 하위 등록 정보를 참조하십시오. 이 등록 정보가 지정되지 않은 경우 리스너는 기본값을 가정합니다.	ContextPath: / 사용 가능: true Data Handler Mime 유형: 요청의 ContentType과 같음 Charset: 없음. 자세한 정보는 49 페이지의 『HTTP 및 HTTPS 프로토콜 리스너 처리』에서 참조하십시오.	아니오
+++++URL1	이것은 계층 구조 등록 정보이며 값이 없습니다. 하위는 이 리스너가 지원하는 URL의 이름을 제공합니다. 지원되는 여러 URL이 있을 수 있습니다. 이 등록 정보와 계층 구조를 복제하여 추가 URL을 플러그인할 수 있음을 참조하십시오.		아니오
++++++ContextPath	리스너가 수신한 HTTP 요청의 URI. 이 값은 URLsConfiguration 등록 정보 아래에서 ContextPath 값 간에 고유해야 합니다. 그렇지 않으면 오류를 기록하며 시작에 실패합니다. ContextPath는 대소문자 구분됩니다. 그러나 대소문자가 구분되지 않는 프로토콜, 호스트 이름 및 포트를 포함할 수 있습니다. 프로토콜이 ContextPath에서 지정된 경우 http이어야 합니다. 호스트가 지정된 경우 호스트 리스너 등록 정보의 값과 같아야 합니다. 포트가 지정된 경우, 포트 리스너 등록 정보의 값과 같아야 합니다.		아니오
++++++Enabled	이 등록 정보의 값은 상위 URL 계층 구조 등록 정보가 커넥터에 대해 사용 가능한지 여부를 판별합니다.	True	아니오
++++++TransformationRules	이것은 계층 구조 등록 정보이며 값이 없습니다. 하나 이상의 변환 규칙을 보유합니다.		
++++++TransformationRule1	이것은 계층 구조 등록 정보이며 값이 없습니다. 변환 규칙을 보유합니다.		아니오
++++++ContentType	이 등록 정보의 값은 특정 핸들링(Data Handler MIME 유형 또는 Charset)이 적용되어야 하는 수신 요청의 ContentType을 지정합니다. ContentType이 TransformationRuleN 계층 구조 등록 정보로 지정되지 않은 경우, 커넥터는 경고 메시지를 기록하며 esTransformationRuleN 등록 정보를 무시합니다. 이 등록 정보의 특정 값 */*을 지정하면 프로토콜 리스너가 이 규칙을 ContentType에 적용하게 합니다. 리스너가 ContentType을 공유하는 동일한 문맥에 대해 둘 이상의 규칙을 찾은 경우, 리스너는 오류를 기록하여 초기화에 실패함을 참조하십시오.		아니오
++++++MimeType	Data Handler가 지정된 ContentType의 요청을 처리하기 위해 호출할 때 사용할 MIME 유형.		아니오
++++++Charset	지정된 ContentType의 요청을 Business Object로 변환할 때 사용할 Charset.		아니오

그림 18에서는 Connector Configurator에 표시된 대로 등록 정보를 표시합니다.

Standard Properties		Connector-Specific Properties	Supported Business Objects	Associated Maps	Resources
	Property	Value	Encrypt	Update Method	Description
1	[-] ProtocolHandlerFramework		<input type="checkbox"/>	agent restart	
2	DataHandlerMetaObjectName	MO_DataHandler_Default	<input type="checkbox"/>	agent restart	
3	[-] ProtocolListenerFramework		<input type="checkbox"/>	agent restart	
4	WorkerThreadCount	10	<input type="checkbox"/>	agent restart	
5	RequestPoolSize	20	<input type="checkbox"/>	agent restart	
6	[-] ProtocolListeners		<input type="checkbox"/>	agent restart	
7	[-] HTTPListener1		<input type="checkbox"/>	agent restart	
8	[-] HTTPSListener1		<input type="checkbox"/>	agent restart	
9	Protocol	https	<input type="checkbox"/>	agent restart	
10	Host	localhost	<input type="checkbox"/>	agent restart	
11	Port	8443	<input type="checkbox"/>	agent restart	
12	SocketQueueLength	5	<input type="checkbox"/>	agent restart	
13	HTTPReadTimeout	0	<input type="checkbox"/>	agent restart	
14	RequestWaitTimeout	60000	<input type="checkbox"/>	agent restart	
15	BOPrefix		<input type="checkbox"/>	agent restart	
16	[-] URLsConfiguration		<input type="checkbox"/>	agent restart	
17	[-] ProxyServer		<input type="checkbox"/>	agent restart	
18	[-] SSL		<input type="checkbox"/>	agent restart	

그림 18. HTTP(S) 프로토콜 리스너 등록 정보

ProxyServer: 네트워크가 프록시 서버를 사용할 때 이 등록 정보 아래 값을 구성하십시오. 이것은 계층 구조 등록 정보이며 값이 없습니다. 이 등록 정보 아래 지정된 값은 HTTP-HTTPS 프로토콜 핸들러에 의해 사용됩니다.

그림 19에서는 Connector Configurator에 표시된 대로 ProxyServer 등록 정보를 표시합니다.

Standard Properties		Connector-Specific Properties	Supported Business Objects	Associated Maps	Resources
	Property	Value	Encrypt	Update Method	Description
1	<input type="checkbox"/> ProtocolHandlerFramework		<input type="checkbox"/>	agent restart	
2	DataHandlerMetaObjectName	MO_DataHandler_Default	<input type="checkbox"/>	agent restart	
3	<input type="checkbox"/> ProtocolListenerFramework		<input type="checkbox"/>	agent restart	
4	<input type="checkbox"/> ProxyServer		<input type="checkbox"/>	agent restart	
5	HttpProxyHost	proxyHostHttp	<input type="checkbox"/>	agent restart	
6	HttpProxyPort	80	<input type="checkbox"/>	agent restart	
7	HttpNonProxyHosts		<input type="checkbox"/>	agent restart	
8	HttpsNonProxyHosts		<input type="checkbox"/>	agent restart	
9	HttpsProxyHost	proxyHostHttps	<input type="checkbox"/>	agent restart	
10	HttpsProxyPort	443	<input type="checkbox"/>	agent restart	
11	SocksProxyHost		<input type="checkbox"/>	agent restart	
12	SocksProxyPort		<input type="checkbox"/>	agent restart	
13	HttpProxyUsername	httpProxyUsername	<input type="checkbox"/>	agent restart	
14	HttpProxyPassword	*****	<input checked="" type="checkbox"/>	agent restart	
15	HttpsProxyUsername	httpsProxyUsername	<input type="checkbox"/>	agent restart	
16	HttpsProxyPassword	*****	<input checked="" type="checkbox"/>	agent restart	
17	<input type="checkbox"/> SSL		<input type="checkbox"/>	agent restart	

그림 19. ProxyServer 등록 정보

HttpProxyHost: HTTP 프록시 서버의 호스트 이름. 네트워크가 HTTP 등록 정보의 프록시 서버를 사용하는 경우 이 등록 정보를 지정하십시오.

기본값 = 없음

HttpProxyPort: 커넥터가 HTTP 프록시 서버에 연결하기 위해 사용하는 포트 번호.

기본값 = 80

HttpNonProxyHosts: 이 등록 정보의 값은 프록시 서버를 통해서가 아니라 직접 연결되어야 하는 하나 이상의 호스트(HTTP의 경우)를 제공합니다. 값은 호스트의 목록일 수 있으며 각각 "|"로 구분됩니다.

기본값 = 없음

HttpsProxyHost: HTTPS 프록시 서버의 호스트 이름.

기본값 = 없음

HttpsProxyPort: 커넥터가 HTTPS 프록시 서버에 연결하기 위해 사용하는 포트 번호.

기본값 = 443

HttpsNonProxyHosts: 이 등록 정보의 값은 프록시 서버를 통해서가 아니라 직접 연결되어야 하는 하나 이상의 호스트(HTTPS의 경우)를 제공합니다. 값은 호스트의 목록일 수 있으며 각각 "|"로 구분됩니다.

기본값 = 없음

SocksProxyHost: Socks 프록시 서버의 호스트 이름. 네트워크가 socks 프록시 사용 시 이 등록 정보를 지정하십시오.

주: 기본적인 JDK가 socks를 지원해야 합니다.

기본값 = 없음

SocksProxyPort: Socks 프록시 서버에 연결할 포트 번호. 네트워크가 socks 프록시 사용 시 이 등록 정보를 지정하십시오.

기본값 = 없음

HttpProxyUsername: HTTP 프록시 서버의 사용자 이름. 요청의 목적지가 HTTP URL이며 ProxyServer -> HttpProxyUsername을 지정하는 경우 프록시 인증 시 HTTP-HTTPS 프로토콜 핸들러는 프록시 인증 머리글을 작성합니다. 핸들러는 인증을 위한 CONNECT 메소드를 사용합니다.

프록시 인증 머리글은 base64 인코딩되며 다음 구조를 갖습니다.

프록시 인증: 기본

Base64EncodedString

핸들러는 base64 인코딩 문자열을 작성하기 위해 콜론(:)으로 구분된 암호 등록 정보와 사용자 이름을 연결합니다.

기본값 = 없음

HttpProxyPassword: HTTP 프록시 서버의 암호. 이 값이 사용되는 방식에 대한 자세한 정보는 『HttpProxyUsername』에서 참조하십시오.

기본값 = 없음

HttpsProxyUsername: HTTPS 프록시 서버의 사용자 이름. 요청의 목적지가 HTTPS URL이며 ProxyServer -> HttpsProxyUsername을 지정하는 경우, HTTP-HTTPS 프로토콜 핸들러는 프록시 인증할 프록시 인증 머리글을 작성합니다. 핸들러는 base64 인코딩 문자열을 작성하기 위해 콜론(:)으로 구분된 HttpsProxyUsername 및 HttpsProxyPassword 구성 등록 정보 값을 연결합니다.

기본값 = 없음

HttpsProxyPassword: HTTPS 프록시 서버의 암호. 이 값이 사용되는 방식에 대한 자세한 정보는 71 페이지의 『HttpsProxyUsername』에서 참조하십시오.

Default = none

SSL: 커넥터의 SSL을 구성하려면 이 등록 정보 아래에서 값을 지정하십시오. 이것은 계층 구조 등록 정보이며 값이 없습니다.

그림 20에서는 Connector Configurator에 표시된 대로 SSL 등록 정보를 표시합니다.

Standard Properties		Connector-Specific Properties		Supported Business Objects		Associated Maps	Resources
	Property	Value	Encrypt	Update Method	Description		
1	ProtocolHandlerFramework		<input type="checkbox"/>	agent restart			
2	DataHandlerMetaObjectName	MO_DataHandler_Default	<input type="checkbox"/>	agent restart			
3	ProtocolListenerFramework		<input type="checkbox"/>	agent restart			
4	ProxyServer		<input type="checkbox"/>	agent restart			
5	SSL		<input type="checkbox"/>	agent restart			
6	SSLVersion	SSL	<input type="checkbox"/>	agent restart			
7	SSLDebug	False	<input type="checkbox"/>	agent restart			
8	KeyStoreType	JKS	<input type="checkbox"/>	agent restart			
9	KeyStore		<input type="checkbox"/>	agent restart			
10	KeyStorePassword		<input type="checkbox"/>	agent restart			
11	KeyStoreAlias		<input type="checkbox"/>	agent restart			
12	TrustStore		<input type="checkbox"/>	agent restart			
13	TrustStorePassword		<input type="checkbox"/>	agent restart			
14	UseClientAuth	False	<input type="checkbox"/>	agent restart			

그림 20. SSL 등록 정보

SSLVersion: 커넥터가 사용하는 SSL 버전. 자세한 정보는 지원되는 SSL 버전의 IBM JSSE 문서를 참조하십시오.

기본값 = SSL

SSLDebug: 이 등록 정보의 값이 true로 설정되면, 커넥터는 javax.net.debug 시스템 등록 정보의 값을 true로 설정합니다. IBM JSSE는 추적 기능을 설정하기 위해 이 등록 정보를 사용합니다. 자세한 정보는 IBM JSSE 문서를 참조하십시오.

기본값 = false

KeyStoreType: 이 등록 정보 값은 KeyStore 및 TrustStore의 유형을 제공합니다. 자세한 정보는 올바른 keystore 유형의 경우 IBM JSSE 문서를 참조하십시오.

기본값 = JKS

KeyStore: 이 등록 정보는 keystore 파일에 대한 전체 경로를 제공합니다. KeyStore 및/또는 KeyStoreAlias 등록 정보를 지정하지 않은 경우, KeyStorePassword, KeyStoreAlias, TrustStore, TrustStorePassword 등록 정보는 무시됩니다. 이 등록 정

보에 지정된 경로를 사용하여 keystore를 로드할 수 없는 경우 커넥터는 시작에 실패합니다. 경로는 keystore 파일에 대한 전체 경로이어야 합니다.

기본값 = 없음

KeyStorePassword: 이 등록 정보는 Keystore에서 개인용 키의 암호를 제공합니다.

기본값 = 없음

KeyStoreAlias: 이 등록 정보는 KeyStore에서 키 쌍의 별명을 제공합니다. HTTPS 리스너는 KeyStore에서 이 개인용 키를 사용합니다. 또한 HTTP-HTTPS 프로토콜 핸들러는 클라이언트 인증을 필요로 하는 HTTPS 서비스 호출 시 KeyStore에서 이 별명을 사용합니다. 등록 정보는 올바른 JSSE 별명으로 설정되어야 합니다.

기본값 = 없음

TrustStore: 이 등록 정보는 TrustStore에 대한 전체 경로를 제공합니다. TrustStore는 커넥터가 신뢰하는 인증서를 저장하는 데 사용됩니다. TrustStore는 KeyStore와 동일한 유형이어야 합니다. TrustStore 파일에 대한 전체 경로를 지정해야 합니다.

기본값 = 없음

TrustStorePassword: 이 등록 정보는 Truststore의 암호를 제공합니다.

기본값 = 없음

UseClientAuth: 이 등록 정보는 SSL 클라이언트 인증이 사용되는지 여부를 지정합니다. true로 설정되면, HTTPS 리스너는 클라이언트 인증을 사용합니다.

기본값 = false

다중 프로토콜 리스너 작성

프로토콜 리스너의 다중 인스턴스를 작성할 수 있습니다. 프로토콜 리스너는 ProtocolListenerFramework -> ProtocolListeners 커넥터 등록 정보의 하위 등록 정보로 구성됩니다. 각 하위(ProtocolListenerFramework -> ProtocolListeners의)는 커넥터의 별개 프로토콜 리스너를 식별합니다. 이에 따라 ProtocolListeners 등록 정보 아래에 새 하위 등록 정보를 구성하여 쓰기 프로토콜 리스너를 작성할 수 있습니다. 새로 작성된 리스너 등록 정보의 모든 하위 등록 정보를 지정하도록 확인하십시오. 각 리스너는 고유하게 이름 지정되어야 합니다. 그러나 리스너의 여러 인스턴스에 대해 동일하게 남는 리스너 프로토콜 등록 정보(http 또는 https)를 변경하지 않습니다.

주: 프로토콜 등록 정보는 스위치로 제공되므로 매우 중요합니다. 리스너나 핸들러를 사용하지 않으려는 경우 이 등록 정보를 비워 두십시오.

HTTP 또는 HTTPS 리스너의 다중 인스턴스를 작성하려는 경우, 각 인스턴스의 여러 포트 및 호스트 등록 정보를 지정하십시오.

핸들러의 다중 인스턴스를 작성할 수 없습니다. 각 프로토콜의 경우 오직 하나의 핸들러일 수 있습니다.

시동 시 커넥터

커넥터를 시작할 때, `init()` 메소드는 System Manager의 Connector Configurator를 사용하여 설정된 구성 등록 정보를 읽습니다. 적절한 기능을 위해서는 커넥터 풀링을 사용 불가능하지 않게 하십시오(커넥터 풀링은 기본값으로 사용 가능합니다). 아래 섹션은 발생 사항을 설명합니다.

프록시 설정

ProxyServer 커넥터 특정 등록 정보를 지정하는 경우, 커넥터는 프록시 시스템 등록 정보를 설정합니다. 프록시 서버는 요청 처리 전용으로 HTTP-HTTPS 프로토콜 핸들러에서 사용됩니다. 또한 커넥터는 설정하는 각 시스템 등록 정보를 추적합니다. ProxyServer 등록 정보에 대한 자세한 정보는 64 페이지의 『커넥터 특정 구성 등록 정보』에서 참조하십시오.

프로토콜 리스너 프레임워크 초기화

시작 중 커넥터는 프로토콜 리스너 프레임워크를 예로 들어 설명하며 초기화합니다. 이 프레임워크는 커넥터 특정 등록 정보 ProtocolListenerFramework를 읽으며, 커넥터는 WorkerThreads 및 RequestPoolSize 커넥터 등록 정보의 값을 읽습니다.

ProtocolListenerFramework 등록 정보가 지정되지 않았거나 누락된 경우, 커넥터는 클라이언트에서 요청을 수신할 수 없으며 경고를 기록합니다.

커넥터는 다음에 ProtocolListenerFramework -> ProtocolListeners 등록 정보를 읽습니다. ProtocolListeners 등록 정보의 모든 첫 번째 레벨 등록 정보는 프로토콜 리스너를 표시합니다. 프로토콜 리스너 프레임워크는 각 리스너를 로드하고 초기화하며 추적합니다. 지속적 이벤트가 사용 가능한 경우, 리스너는 이벤트 복구를 시도합니다.

프로토콜 핸들러 프레임워크 초기화

커넥터는 커넥터 특정 등록 정보 ProtocolHandlerFramework를 읽으며 프로토콜 핸들러 프레임워크를 예로 들어 설명하고 초기화합니다. 이 등록 정보가 누락되었거나 적절하게 설정되지 않은 경우, 커넥터는 요청 처리를 수행할 수 없으며 경고를 기록합니다. 다음으로 커넥터는 프로토콜 핸들러에 해당하는 모든 ProtocolHandlerFramework -> ProtocolHandlers 등록 정보를 읽어 로드, 초기화 및 추적하려고 시도합니다. 커넥터 초기화 중 프로토콜 핸들러가 로드되며 협업에서 서비스 요청 작성 시 초기화되지 않음을 참고하십시오. 프로토콜 핸들러는 다중 스레드가 안전합니다.

로깅

커넥터는 다음 경우에 경고를 기록합니다.

- `ProtocolListenerFramework` 등록 정보가 지정되지 않습니다. 커넥터는 이벤트 공고를 수행할 수 없음을 경고합니다.
- `ProtocolHandlerFramework` 등록 정보가 지정되지 않습니다. 커넥터는 (협업) 요청 처리를 수행할 수 없음을 경고합니다.

추적

추적은 커넥터 작동을 그대로 따르기 위해 설정할 수 있는 선택적 디버깅입니다. 추적 메시지는 `STDOUT`에 기본적으로 쓰여집니다. 추적 메시지 구성에 대한 자세한 정보는 커넥터 구성 등록 정보를 참조하십시오. 사용 가능하게 하고 설정하는 방법을 포함한 추적에 대한 자세한 정보는 *Java용 Connector Development Guide*를 참조하십시오.

커넥터 추적 레벨은 다음과 같습니다.

- | | |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 레벨 0 | 이 레벨은 커넥터 버전을 식별하는 추적 메시지에 사용됩니다. |
| 레벨 1 | <code>pollForEvents</code> 메소드가 호출될 때마다 추적하십시오. ICS로의 전달을 위해 리스너가 작성하는 TLO 이름을 추적하십시오. 요청 Business Object 이름 및 해당 속성을 TLO에서 추적하십시오. |
| 레벨 2 | Business Object가 <code>gotAppEvent()</code> 또는 <code>executeCollaboration()</code> 에서 InterChange Server로 게시될 때마다 기록하는 추적 메시지에 이 레벨을 사용하십시오. 또한 어떤 프로토콜 핸들러가 요청을 처리 중인 지 추적하십시오. |
| 레벨 3 | 처리 중인 Business Object의 ASI를 추적하십시오. 처리 중인 Business Object의 속성을 추적하십시오. 이벤트 공고 중 요청 Business Object의 TLO를 추적하십시오. Data Handler가 리턴한 Business Object를 추적하십시오. |
| 레벨 4 | 다음과 연관된 전송 머리글을 추적하십시오. <ul style="list-style-type: none">• 전송에서 프로토콜 리스너가 수신한 요청 메시지• 프로토콜 리스너가 클라이언트로 전송한 응답 메시지. 스레드의 결과, 처리된 모든 ASI 및 중요한 기능의 모든 항목과 종료 를 추적하십시오. |
| 레벨 5 | 다음을 추적하십시오. <ul style="list-style-type: none">• 각 중요한 메소드의 입력 및 종료• 모든 구성 특정 등록 정보• 각 프로토콜 리스너의 로딩 |

- 전송에서 프로토콜 리스너가 수신한 요청 메시지
- 프로토콜 리스너가 클라이언트로 전송 시 전송되는 응답 메시지.
- 각 프로토콜 핸들러의 로딩
- Data Handler가 리턴한 메시지
- 협업으로 전송된 TLO의 Business Object 덤프
- Data Handler가 리턴한 Business Object의 덤프.

제 5 장 문제점 해결

이 장에서는 커넥터 시작이나 실행 시 발생할 수 있는 문제점에 대해 설명합니다.

시작 문제점

문제점

알고리즘이 지원되지 않음/알고리즘 'SSL'을 사용할 수 없음

잠재적인 솔루션/설명

Connector Configurator에서 지정한 SSL 버전을 JSSE 프로바이더가 지원하지 않으면 이 오류가 발생합니다. 솔루션: 지원되는 SSL 버전의 경우 JSSE 프로바이더의 문서를 확인하십시오. IBM JSSE의 경우 `ProductDir/lib/security` 디렉토리의 `java.security` 파일이 다음 항목을 가지는지 확인하십시오.

```
security.provider.<number>=com.ibm.jsse.  
IBMJSSEProvider
```

keystore 로드 오류:Keystore 파일 경로:"<path>"가 잘못 지정되었음:KeyStore가 없음

여기서 <number>는 보안 프로바이더를 로드하는 우선 순위입니다. 이 오류는 keystore 및/또는 truststore 파일의 잘못된 경로를 지정한 경우 발생합니다. 솔루션: Connector Configurator의 SSL -> KeyStore 등록 정보에 지정된 keystore 파일 경로를 확인하십시오. 또한 truststore를 사용하려는 경우, Connector Configurator의 SSL -> TrustStore 등록 정보에 지정된 truststore 파일 경로를 확인하십시오.

KeyManagementError: KeyStore가 함부로 변경되었습니다.
KeyManagement 오류.

이 오류는 keystore 및/또는 truststore가 함부로 변경되거나 손상된 경우 발생합니다. 이 오류는 암호에 대해 잘못된 값을 지정한 경우에도 발생할 수 있습니다. 솔루션: keystore가 함부로 변경되지 않았는지 확인하십시오. keystore를 재작성하십시오. 또한 SSL -> KeyStorePassword 및 SSL -> TrustStorePassword 커넥터 등록 정보에서 올바른 암호를 입력했는지 확인하십시오.

keystore에서 인증 로드 오류

이 오류는 인증 및/또는 keystore, truststore가 함부로 변경된 경우 발생합니다. 이 오류는 암호에 대해 잘못된 값을 지정한 경우에도 발생할 수 있습니다. 솔루션: 인증, keystore 또는 truststore가 함부로 변경되었는지 알아보십시오. 또한 SSL -> KeyStorePassword 및 SSL -> TruststorePassword 커넥터 등록 정보에서 올바른 암호를 지정했는지 확인하십시오.

서버 소켓 작성 오류, 종결: 오류

이 오류는 HTTP 또는 HTTPS 프로토콜 리스너가 커넥터 등록 정보에서 지정한 포트에 바인드할 수 없는 경우 발생합니다. 솔루션: 모든 HTTP 및 HTTPS 프로토콜 리스너에 지정된 포트를 확인하십시오. 둘 이상의 리스너에 동일한 포트가 지정된 경우, 오직 하나의 리스너가 시작할 수 있습니다. 그리고 해당 포트에서 다른 서비스가 실행 중인지 확인하십시오. 그렇게 하면 프로토콜 리스너의 다른 포트를 선택할 수도 있습니다.

KeyManagementError:UnrecoverableKeyException, 키는 복구될 수 없습니다.

이 오류는 keystore나 truststore를 사용할 수 없는 경우 발생합니다. 솔루션: 새 keystore를 작성하십시오.

SSL 데이터 교환 예외: 알 수 없는 CA

truststore에 CA 인증이 없는 경우 발생합니다. 솔루션: 자체 서명 인증은 물론 CA의 인증이 truststore에 상주하는지 여부를 확인하십시오. 또한 인증의 DN이 호스트 이름(IP 주소 번호)을 갖는지 확인하십시오.

문제점

로그 파일에서 과도한 JSSE 로깅을 주의합니다.

프로토콜 리스너를 지정했으나 리스너가 초기화되지 않은 경우, 커넥터에 다음 경고 메시지가 표시됩니다.

```
Skipping Protocol Listener Property Set  
"SOME_LISTENER_NAME" with protocol property "":  
unable to determine the protocol listener  
class.]
```

프로토콜 핸들러를 지정했지만

초기화되지 않은 경우, 커넥터에 다음 경고 메시지가 표시됩니다.

```
Unable to determine the type of the  
handler; skipping initializing of current  
handler. Handler property details:  
Name: <Handler Name>;  
Value:  
  Name: Protocol; Value:  
  Name: ResponseWaitTimeout; Value:  
  Name: ReplyToQueue; Value: .]
```

잠재적인 솔루션/설명

사용자 콘솔에서 기본적인 모든 JSSE 세부사항을 보지 않으려는 경우, Connector Configurator의 SSL -> SSLDebug 등록 정보의 값을 false로 설정하십시오.

커넥터가 프로토콜 리스너의 프로토콜 등록 정보에 대한 올바른 값을 추출할 수 없습니다. 올바른 값은 http 또는 https입니다. 솔루션: 이것은 오류 조건이 아닙니다. 그러나 커넥터가 이 리스너를 사용하게 하려는 경우, 올바른 프로토콜 등록 정보 값을 지정하십시오.

커넥터가 핸들러의 프로토콜 등록 정보에 대한 올바른 값을 추출할 수 없습니다. 올바른 값은 http 및 https입니다. 솔루션: 이것은 오류 조건이 아닙니다. 그러나 커넥터가 이 핸들러를 사용하도록 하려는 경우, 올바른 프로토콜 등록 정보 값을 지정하십시오.

런타임 오류

문제점

HTTP 응답 구문 분석 오류: HTTP 응답 머리글을 읽는 동안 스트림의 끝에 도달했습니다

언급한 url에 오류가 있으며, 호스트 및 포트 세부사항을 추출할 수 없으며, 목적지가 잘못된 <목적지 URL>을 갖습니다

verb가 <Verb>인 이벤트 Business Object <BO 이름> 전송에 실패했습니다. 실행 상태 "-1"과 다음 오류 메시지를 수신했습니다.

```
MapException: Unable to find the map to map  
business objects <BO Name> for the connector  
controller HTTPConnector
```

요청을 요청 Business Object로 변환하는 데 실패했습니다.

결합:

```
Failure in generating request object -  
no verb could be set on the request bo
```

잠재적인 솔루션/설명

이 오류는 커넥터가 HTTP 서비스를 호출할 때 발생합니다. 대상 HTTP 서비스가 잘못된 HTTP 응답을 전송했으므로 발생합니다. 솔루션: 대상 HTTP 서비스 주소가 올바른지 확인하십시오.

이 오류는 커넥터가 HTTP 서비스를 호출할 때 발생합니다. HTTP 서비스의 잘못된 끝점 주소를 지정했기 때문에 발생합니다.

솔루션: HTTP 서비스의 올바른 주소를 지정했는지 확인하십시오. 이 오류는 커넥터가 동기적으로 이벤트를 전송 중인 협업이 존재하지 않거나 Business Object verb를 승인하지 않기 때문에 통합 브로커가 이벤트 처리에 실패할 경우 발생합니다. 솔루션: 이벤트 공고를 위해 TLO를 사용하는 경우, TLO의 ws_collab 오브젝트 레벨 ASI를 조사하십시오. (TLO의 이름이 오류 메시지에 제공됩니다.) ws_collab ASI의 값을 검사하십시오. 이 협업이 존재하고 실행 중인지 확인하십시오. ws_mode BO 레벨 ASI가 synchro 설정된 경우, ws_collab ASI가 필요합니다. ws_verb 오브젝트 레벨 ASI의 값을 검사하십시오. ws_collab ASI가 지정한 협업이 ws_verb ASI에서 지정한 verb로 트리거될 수 있는지 확인하십시오. 이 협업이 존재하고 실행 중인지 확인하십시오.

이 오류는 커넥터가 통합 브로커에 전송하려고 시도 중인 Business Object의 verb를 커넥터가 판별할 수 없을 때 이벤트 공고 중 발생됩니다.

솔루션: 이 TLO의 ws_verb 오브젝트 레벨 ASI를 지정했는지 확인하십시오. 이 ASI의 값으로 verb를 지정하십시오.

부록 A. 커넥터의 표준 구성 등록 정보

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

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

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

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

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

신규 및 삭제된 등록 정보

다음 표준 등록 정보가 이 릴리스에 추가되었습니다.

새 등록 정보

- XMLNameSpaceFormat

삭제된 등록 정보

- RestartCount
- RHF2MessageDomain

표준 커넥터 등록 정보 구성

어댑터 커넥터에는 두 가지 유형의 구성 등록 정보가 있습니다.

- 표준 구성 등록 정보
- 커넥터 특정 구성 등록 정보

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

Connector Configurator 사용

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

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

등록 정보 값 설정 및 갱신

등록 정보 필드의 기본 길이는 255자입니다.

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

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

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

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

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

표준 등록 정보 요약

표 33은 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 33. 표준 구성 등록 정보 요약

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

표 33. 표준 구성 등록 정보 요약 (계속)

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

표 33. 표준 구성 등록 정보 요약 (계속)

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

표 33. 표준 구성 등록 정보 요약 (계속)

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

표준 구성 등록 정보

이 절에서는 표준 커넥터 구성 등록 정보를 나열하고 각각을 정의합니다.

AdminInQueue

관리 메시지를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMININQUEUE입니다.

AdminOutQueue

관리 메시지를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMINOUTQUEUE입니다.

AgentConnections

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

AgentTraceLevel

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

ApplicationName

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

BrokerType

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

CharacterEncoding

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

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

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

ConcurrentEventTriggeredFlows

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

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

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

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

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

ContainerManagedEvents

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

기본값은 값 없음입니다.

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

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

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

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

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

ControllerStoreAndForwardMode

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

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

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

기본값은 true입니다.

ControllerTraceLevel

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

커넥터 제어기의 추적 메시지 레벨입니다. 기본값은 0입니다.

DeliveryQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

Business Object를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/DELIVERYQUEUE입니다.

DeliveryTransport

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

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

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

WebSphere MQ 및 IDL

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

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

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

JMS

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

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

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

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

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

- CWSHaredEnv.sh 스크립트의 LDR_CNTRL 환경 변수.

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

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

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

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

DuplicateEventElimination

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

또한 이 등록 정보는 false로 설정될 수 있습니다.

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

FaultQueue

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

기본값은 CONNECTORNAME/FAULTQUEUE입니다.

JvmMaxHeapSize

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

기본값은 128m입니다.

JvmMaxNativeStackSize

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

기본값은 128k입니다.

JvmMinHeapSize

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

기본값은 1m입니다.

jms.FactoryClassName

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

기본값은 CxCommon.Messaging.jms.IBMMQSeriesFactory입니다.

jms.MessageBrokerName

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

기본값은 crossworlds.queue.manager입니다.

jms.NumConcurrentRequests

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

기본값은 10입니다.

jms.Password

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

기본값은 없습니다.

jms.UserName

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

기본값은 없습니다.

ListenerConcurrency

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

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

Locale

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

로케일 이름의 형식은 다음과 같습니다.

ll_TT.codeset

여기서

ll

2문자 언어 코드(보통 소문자)

TT

2문자 국가 또는 지역 코드(보통 대문자)

codeset

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

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

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

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

LogAtInterchangeEnd

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

MaxEventCapacity

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

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

MessageFileName

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

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

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

MonitorQueue

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

기본값은 CONNECTORNAME/MONITORQUEUE입니다.

OADAutoRestartAgent

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

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

기본값은 false입니다.

OADMaxNumRetry

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 1000입니다.

OADRetryTimeInterval

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 10입니다.

PollEndTime

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

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

PollFrequency

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

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

기본값은 10000입니다.

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

PollQuantity

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

PollStartTime

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

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

RequestQueue

Business Object를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTOR/REQUESTQUEUE입니다.

RepositoryDirectory

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

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

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

ResponseQueue

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

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

RestartRetryCount

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

기본값은 3입니다.

RestartRetryInterval

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

기본값은 1입니다.

SourceQueue

DeliveryTransport가 JMS이고 ContainerManagedEvents가 지정된 경우에만 적용 가능합니다.

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

기본값은 CONNECTOR/SOURCEQUEUE입니다.

SynchronousRequestQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

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

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

기본값은 CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE입니다.

SynchronousResponseQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

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

기본값은 CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE입니다.

SynchronousRequestTimeout

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

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

기본값은 0입니다.

WireFormat

전송 시 메시지 형식입니다.

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

WsifSynchronousRequest 제한시간

WAS 통합 브로커에만 해당됩니다.

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

기본값은 0입니다.

XMLNamespaceFormat

WebSphere Message Broker 및 WAS 통합 브로커에만 해당됩니다.

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

기본값은 short입니다.

부록 B. Connector Configurator

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

Connector Configurator를 사용하여 다음을 수행할 수 있습니다.

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

주:

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

이 부록에서 다루는 주제는 다음과 같습니다.

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

Connector Configurator 개요

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

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

Connector Configurator를 사용하여 다음을 수행할 수 있습니다.

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

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

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

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

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

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

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

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

Connector Configurator 시작

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

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

독립형 모드로 Configurator 실행

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

이를 수행하려면 다음 단계를 따르십시오.

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

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

System Manager에서 Configurator 실행

System Manager에서 Connector Configurator를 실행할 수 있습니다.

Connector Configurator를 실행하려면 다음을 수행하십시오.

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

기존 구성 파일을 편집하려면 다음을 수행하십시오.

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

커넥터 특정 등록 정보 템플릿 작성

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

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

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

새 템플리트 작성

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

템플리트를 작성하려면 다음을 수행하십시오.

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

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

일반 특성 지정

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

- 일반
 - 등록 정보 유형
 - 갱신 메소드
 - 설명

- 플래그
 - 표준 플래그
- 사용자 정의 플래그
 - 플래그

등록 정보의 일반 특성을 선택한 후, **값** 탭을 누르십시오.

값 지정

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

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

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

기본값 열을 사용하면 임의의 값을 기본값으로 지정할 수 있습니다.

값 범위는 등록 정보 값 대화 상자에서 입력한 범위를 표시합니다.

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

종속성 설정

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

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

등록 정보가 종속되도록 지정하고 종속 조건을 설정하려면, 다음을 수행하십시오.

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

새 구성 파일 작성

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

브로커를 선택하려면 다음을 수행하십시오.

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

다음 작업도 수행할 수 있습니다.

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

커넥터 특정 템플릿에서 구성 파일 작성

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

1. 파일 > 새로 작성 > 커넥터 구성을 누르십시오.

2. 다음 필드가 있는 새 커넥터 대화 상자가 나타납니다.

- 이름

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

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

- 시스템 연결

ICS, WebSphere Message Brokers 또는 WAS를 누르십시오.

- 커넥터 특정 등록 정보 템플릿

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

사용하려는 템플릿을 선택하고 확인을 누르십시오.

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

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

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

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

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

기존 파일 사용

다음 중 하나 이상의 형식으로 기존 파일을 사용할 수 있습니다.

- 커넥터 정의 파일.

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

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

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

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

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

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

System Manager 프로젝트에서 커넥터 구성을 열려면 다음 단계를 따르십시오.

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

구성 파일 완료

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

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

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

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

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

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

구성 파일 등록 정보 설정

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

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

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

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

ICS에서 실행되는 커넥터의 경우, 다음 등록 정보의 값도 필요합니다.

- 연관된 맵
- 자원
- 메시징(적용 가능할 경우)

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

표준 등록 정보는 다음과 같이 커넥터 특정 등록 정보와 다릅니다.

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

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

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

표준 커넥터 등록 정보 설정

표준 등록 정보의 값을 변경하려면 다음을 수행하십시오.

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

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

응용프로그램 특정 구성 등록 정보 설정

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

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

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

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

커넥터 등록 정보 암호화

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

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

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

갱신 메소드

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

지원되는 Business Object 정의 지정

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

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

ICS가 브로커인 경우

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

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

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

지원되는 목록에서 Business Object를 삭제하려면 다음을 수행하십시오.

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

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

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

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

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

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

대부분의 커넥터의 경우 최상의 노력만이 가능한 선택사항입니다.

트랜잭션 레벨 변경 사항을 적용하려면 서버를 다시 시작해야 합니다.

WebSphere Message Broker가 브로커인 경우

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

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

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

WAS가 브로커인 경우

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

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

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

연관된 맵(ICS에만 해당)

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

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

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

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

연관된 맵 탭은 다음 필드를 표시합니다.

- **Business Object** 이름

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

- 연관된 맵

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

- 명시

어떤 경우에는 연관된 맵을 명시적으로 바인드해야 할 수도 있습니다.

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

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

맵을 명시적으로 바인드하려면 다음을 수행하십시오.

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

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

자원(ICS)

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

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

메시징(ICS)

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

추적/로그 파일 값 설정

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

로깅 및 추적 값을 변경하려면 다음을 수행하십시오.

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

- 콘솔에(STDOUT):

로깅 또는 추적 메시지를 STDOUT 화면에 기록합니다.

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

- 파일에:

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

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

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

Data Handler

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

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

구성 파일 저장

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

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

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

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

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

구성 파일 변경

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

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

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

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

구성 완료

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

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

국제화된 환경에서 Connector Configurator 사용

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

Connector Configurator는 다음 위치에서 영어 이외의 문자를 지원합니다.

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

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

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

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

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

부록 C. Adapter for HTTP 학습서

- 『학습서 정보』
- 116 페이지의 『시작하기 전에』
- 117 페이지의 『설치 및 구성』
- 121 페이지의 『비동기 시나리오 실행』
- 123 페이지의 『동기 시나리오 실행』

이 부록에는 다음에 대한 단계별 프로시저가 있습니다.

- 요청 및 이벤트 처리를 위한 비동기 및 동기 이벤트 전송을 설명합니다.
- HTTP 샘플용의 HTTP 어댑터를 구성하는 방법을 예시합니다.
- HTTPS 샘플용의 HTTP 어댑터를 구성하는 방법을 예시합니다.

학습서 정보

이 학습서는 지원되는 프로토콜 HTTP 및 HTTPS에서 Adapter for HTTP의 요청 및 이벤트 처리 국면 모두에 대한 비동기 및 동기 이벤트 전송에 대해 설명하기 위한 것입니다. 각 시나리오에서 어댑터는 다음과 같이 역할합니다.

- 외부 URL을 호출하는 HTTP 클라이언트
- URL에서 HTTP 요청을 청취하고 WebSphere ICS 협업으로 라우트하는 프록시.

학습서는 샘플 시나리오에서 어댑터의 기본 기능을 보여주도록 구성되었습니다.

- 비동기 (요청만) HTTP POST를 예시하는 비동기 시나리오. 이 시나리오에는 두 가지 샘플이 있습니다. 구성 간편성을 위해 동일한 HTTP 어댑터가 HTTP 요청의 청취에 사용되며 HTTP 클라이언트로 URL을 호출하기 위해 사용됩니다.

- **URL에서 HTTP 요청을 청취하는 프록시** 이 샘플에서, 수신 요청은 WebSphere ICS에서 협업 SERVICE_ASYNC_Order_Collab로 라우트됩니다. 협업은 비동기 순서로 나타냅니다. 어댑터가 적절하게 구성되면, 이 협업은 HTTP 또는 HTTPS 프로토콜 중 하나를 사용하여 호출될 수 있습니다.

SERVICE_ASYNC_Order_Collab는 SERVICE_ASYNC_TLO_Order를 갖는 단순한 통과 협업입니다. 이 협업의 트리거링 포트(시작)는 HTTPConnector로 바인드됩니다. 서비스 포트(끝)는 SampleSiebelConnector로 바인드됩니다.

- **외부 URL을 호출하는 HTTP 클라이언트** 이 샘플에서, HTTP 클라이언트는 HTTP 어댑터를 사용하여 비동기적으로 외부 URL을 호출하는 WebSphere ICS 내의 다른 협업 CLIENT_ASYNC_Order_Collab입니다. 어댑터가 적절하게 구성되면 이 HTTP 클라이언트는 프로토콜 중 하나에서 외부 URL을 호출할 수 있습니다. CLIENT_ASYNC_Order_Collab는 CLIENT_ASYNC_TLO_Order를 갖는

단순 통과 협업입니다. 이 협업의 트리거링 포트(시작)는 SampleSAPConnector로 바인드됩니다. 서비스 포트(끝)는 HTTPConnector로 바인드됩니다.

비동기 시나리오의 샘플 두가지 모두 다음 두 개의 응용프로그램과 관련됩니다.

- SampleSiebel: 클라이언트의 순서를 작성합니다.
- SampleSAP: 순서를 작성합니다.
- 비동기 (요청-응답) HTTP 호스트를 예시하는 동기 시나리오. 이 시나리오에는 두 가지 샘플이 있습니다. 구성 간편성을 위해 동일한 HTTP 어댑터가 HTTP 요청의 청구에 사용되며 HTTP 클라이언트로 URL을 호출하기 위해 사용됩니다.
 - URL에서 HTTP 요청을 청구하는 프록시 이 샘플에서, 수신 요청은 WebSphere ICS에서 협업 SERVICE_SYNCH_OrderStatus_Collab로 라우트됩니다. 협업은 동기 OrderStatus로 나타냅니다. 어댑터가 적절하게 구성되면, 이 협업은 HTTP 또는 HTTPS 프로토콜 중 하나를 사용하여 호출될 수 있습니다. SERVICE_SYNCH_OrderStatus_Collab는 SERVICE_SYNCH_TLO_OrderStatus를 갖는 단순한 통과 협업입니다. 이 협업의 트리거링 포트(시작)는 HTTPConnector로 바인드됩니다. 서비스 포트(끝)는 SampleSiebelConnector로 바인드됩니다.
 - 외부 URL을 호출하는 HTTP 클라이언트 이 샘플에서, HTTP 클라이언트는 HTTP 어댑터를 사용하여 외부 URL을 호출하는 WebSphere ICS 내의 다른 협업 CLIENT_SYNCH_OrderStatus_Collab입니다. 어댑터가 적절하게 구성되면 이 HTTP 클라이언트는 프로토콜 중 하나에서 외부 URL을 호출할 수 있습니다. CLIENT_SYNCH_OrderStatus_Collab는 CLIENT_SYNCH_TLO_OrderStatus를 갖는 단순한 통과 협업입니다. 이 협업의 트리거링 포트(시작)는 SampleSAPConnector로 바인드됩니다. 서비스 포트(끝)는 HTTPConnector로 바인드됩니다.

동기 시나리오의 샘플 두가지 모두 다음 두 개의 응용프로그램과 관련됩니다.

- SampleSiebel: 클라이언트의 순서 상태를 검색합니다.
- SampleSAP: 순서 상태를 요청합니다.

양 시나리오는 두 개의 테스트 커넥터를 사용하여 SampleSiebelConnector 및 SampleSAPConnector를 시뮬레이트하는 것과 관련됩니다.

시작하기 전에

학습서를 시작하기 전에 다음을 확인하십시오.

- WebSphere ICS 4.2.2 이상을 설치하여 경험했습니다.
- WebSphere ICS 홈 디렉토리에서 WebSphere Business Integration Adapter for HTTP를 설치했습니다.
- HTTP 기술에 경험이 있습니다.
- XML 기술에 경험이 있습니다.

설치 및 구성

다음 섹션에서, *WBI_folder*는 현재 WebSphere ICS 설치를 포함하는 폴더를 나타냅니다. 모든 환경 변수와 파일 분리가 Windows 2000 형식으로 지정됩니다. AIX 또는 Solaris에서 실행 중인 경우 적절한 변경을 작성하십시오. (예를 들어, *WBI_folder\connectors*는 *WBI_folder/connectors*입니다).

서버 및 도구 시작

1. 단축 아이콘에서 ICS(WebSphere InterChange Server)를 시작하십시오.
2. WebSphere Business Integration System Manager를 시작하고 Component Navigator Perspective를 여십시오.
3. InterChange Server 보기에서 서버 인스턴스로 사용자 서버를 등록하고 연결하십시오.

샘플 내용 로드

Component Navigator Perspective에서

1. 새 통합 구성요소 라이브러리를 작성하십시오.
2. *WBI_folder\connectors\HTTP\samples\WebSphereICS*에 있는 *HTTPSample.jar* 라는 repos 파일을 가져오십시오.

협업 템플릿 컴파일

WebSphere Business Integration System Manager를 사용하여

- *HTTPSample.jar* repos 파일에서 가져온 협업 템플릿을 모두 컴파일하십시오.

커넥터 구성

1. 아직 그렇게 수행하지 않은 경우, 이 안내서에서 설명한 대로 사용자 시스템에 따라 커넥터를 구성하십시오.
2. WebSphere Business Integration System Manager를 사용하여 Connector Configurator에서 HTTPConnector를 여십시오.
3. 또한 샘플에서 사용하려는 프로토콜의 HTTPConnector를 구성해야 합니다.
 - HTTP를 사용하려는 경우 Connector for HTTP를 구성하려면 『HTTP 프로토콜 시나리오의 구성』에서 참조하십시오.
 - HTTPS를 사용하려는 경우 Connector for HTTPS를 구성하려면 118 페이지의 『HTTPS 프로토콜 시나리오의 구성』에서 참조하십시오.

HTTP 프로토콜 시나리오의 구성

이 섹션에서는 HTTP 샘플 시나리오의 커넥터를 구성하는 방법을 표시합니다. 이 문서의 본문에서 설명한 대로, 커넥터는 HTTP 프로토콜 리스너 및 HTTP-HTTPS 프로토콜 핸들러를 포함합니다.

다음에 오는 단계와 설명에서, 계층 구조 커넥터 구성 등록 정보는 -> 기호로 표시됩니다. 예를 들어, A -> B는 A가 계층 구조 등록 정보이며 B가 A의 하위 등록 정보임을 나타냅니다.

이 샘플의 HTTP 프로토콜 리스너를 구성하려면 다음과 같이 하십시오.

1. Connector Configurator에서 HTTPConnector의 커넥터 특정 등록 정보를 클릭하십시오.
2. ProtocolListeners 하위 등록 정보를 표시하려면 **ProtocolListenerFramework** 등록 정보를 펼치십시오.
3. **HTTPListener1** 하위 등록 정보를 표시하려면 **ProtocolListeners** 하위 등록 정보를 펼치십시오.
4. **HTTPListener1 -> Host** 및 **HTTPListener1 -> Port** 등록 정보의 값을 검사하십시오. 호스트에서 실행 중이며 이 TCP/IP 포트에서 청취 중인 다른 프로세스가 없는지 확인하십시오. 선택적으로 **HTTPListener1 -> Host**의 값을 커넥터를 실행할 시스템 이름으로 설정하려고 할 수 있습니다.

샘플의 경우 HTTP-HTTPS 프로토콜 핸들러를 구성할 필요가 있습니다.

HTTPS 프로토콜 시나리오의 구성

이 섹션에서는 HTTPS 샘플 시나리오의 커넥터를 구성하는 방법을 표시합니다. 커넥터는 HTTPS 프로토콜 리스너 및 HTTP-HTTPS 프로토콜 핸들러를 포함합니다.

다음에 오는 단계와 설명에서, 계층 구조 커넥터 구성 등록 정보는 -> 기호로 표시됩니다. 예를 들어, A -> B는 A가 계층 구조 등록 정보이며 B가 A의 하위 등록 정보임을 나타냅니다.

주: 116 페이지의 『시작하기 전에』에서 위에 나열된 사전설치 항목 이외에, 키 및 인증 관리 소프트웨어를 사용하여 keystore 및 truststore를 작성하고 테스트해야 합니다.

SSL 커넥터 특정 등록 정보 구성: HTTPS의 경우, 커넥터는 SSL 커넥터 특정 계층 구조 등록 정보를 구성하는 것이 필요합니다.

1. Connector Configurator에서 HTTPConnector의 커넥터 특정 등록 정보 탭을 클릭하십시오.
2. 모든 하위 등록 정보를 보려면 **SSL** 계층 구조 등록 정보를 펼치십시오. 그리고 계층 구조 SSL 커넥터 특정 등록 정보의 다음 하위 등록 정보를 검사하거나 변경하십시오.
 - **SSL -> KeyStore** 키 및 인증 관리 소프트웨어를 사용하여 작성해야 하는 keystore 파일에 대한 전체 경로로 설정하십시오.
 - **SSL -> KeyStorePassword** KeyStore를 액세스하는 데 필요한 암호로 설정하십시오.

- **SSL -> KeyStoreAlias** KeyStore에서 개인용 키의 별명으로 설정하십시오.
- **SSL -> TrustStore** 키 및 인증 관리 소프트웨어를 사용하여 작성한 truststore 파일의 전체 경로로 설정하십시오.
- **SSL -> TrustStorePassword** TrustStore를 액세스하는 데 필요한 암호로 설정하십시오.

주: Connector Configurator에서 변경사항을 저장하는 것을 잊지 마십시오.

HTTPS 프로토콜 리스너 구성:

1. Connector Configurator에서 HTTPConnector의 커넥터 특정 등록 정보를 클릭하십시오.
2. **ProtocolListeners** 하위 등록 정보를 표시하려면 **ProtocolListenerFramework** 등록 정보를 펼치십시오.
3. **HTTPSListener1** 하위 등록 정보를 표시하려면 **ProtocolListeners** 하위 등록 정보를 펼치십시오. **HTTPSListener1 -> Host** 및 **HTTPSListener1 -> Port** 등록 정보의 값을 검사하십시오. 호스트에서 실행 중이며 이 TCP/IP 포트에서 청취 중인 다른 프로세스가 없는지 확인하십시오. 선택적으로 **HTTPSListener1 -> Host**의 값을 커넥터를 실행 중인 시스템 이름으로 설정하려고 할 수 있습니다.

샘플의 경우 HTTP-HTTPS 프로토콜 핸들러를 구성할 필요가 있습니다.

KeyStore 및 TrustStore 설정: 샘플 시나리오에서 사용하기 위해 KeyStore 및 TrustStore를 빠르게 설정할 수 있습니다. 프로덕션 시스템의 경우, 타사 소프트웨어를 사용하여 인증 및 키 생성은 물론 keystore를 설정하고 관리해야 합니다. 이 자원을 설정하고 관리하기 위한 Adapter for HTTP의 파트로 어떤 도구도 제공되지 않습니다.

이 섹션에서는 Java Virtual Machine이 사용자 시스템에 설치되어 있으며 JVM(Java Virtual Machine)에 제공된 keytool과 친숙하다고 가정합니다. keytool에서의 문제점 해결에 대한 자세한 정보는 JVM을 수반한 문서를 참조하십시오.

KeyStore를 설정하려면 다음과 같이 하십시오.

1. keytool을 사용하여 KeyStore를 작성합니다. KeyStore에서 키 쌍을 작성해야 합니다. 이를 수행하려면 명령행에서 다음을 입력하십시오.

```
keytool -genkey -alias httpadapter -keystore c:\security\keystore
```

2. keytool은 즉시 암호를 프롬프트합니다. **SSL -> KeyStorePassword** 커넥터 등록 정보의 값에 입력한 암호를 지정하십시오.

위의 예에서 명령행에 **-keystore c:\security\keystore**를 지정한 경우 **SSL -> KeyStore** 등록 정보의 값으로 **c:\security\keystore**를 입력하는 것을 참조하십시오. 또한 명령행에 **-alias httpadapter**를 지정한 경우 **SSL -> KeyStoreAlias** 커넥터 등록 정보의 값으로 **httpadapter**를 입력합니다. 그러면 keytool은 인증의

세부사항을 프롬프트합니다. 다음에서는 각 프롬프트에서 입력할 수 있는 내용을 예시하지만 단지 예이며, 항상 keytool 문서를 나타내고 지연시킵니다.

```
What is your first and last name?  
[Unknown]: HostName  
What is the name of your organizational unit?  
[Unknown]: myunit  
What is the name of your organization?  
[Unknown]: myorganization  
What is the name of your City or Locality?  
[Unknown]: mycity  
What is the name of your State or Province?  
[Unknown]: mystate  
What is the two-letter country code for this unit?  
[Unknown]: mycountryIs <CN=HostName, OU=myunit, O=myorganization, L=mycity, ST=mystate, C=mycountry> correct?  
[no]: yes
```

3. What is your first and last name?의 경우 커넥터를 실행 중인 시스템의 이름을 입력해야 함을 참고하십시오. 그러면 keytool은 다음을 프롬프트합니다.

<httpadapter>의 키 암호를 입력하십시오(keystore 암호와 동일한 경우 RETURN).

4. 동일한 암호를 사용하려면 리턴을 누르십시오. 자체 서명 인증을 사용하려는 경우 위에서 작성한 인증을 내보내려고 할 수 있습니다. 그렇게 하려면 명령행에서 다음을 입력하십시오.

```
C:\security>keytool -export -alias httpadapter -keystore c:\security\keystore  
-file c:\security\httpadapter.cer
```

5. keytool은 이제 keystore 암호를 프롬프트합니다. 위에 입력한 암호를 입력하십시오.

TrustStore를 설정하려면 다음과 같이 하십시오.

1. 신뢰 인증을 TrustStore로 가져오려면, 다음 명령을 입력하십시오.

```
keytool -import -alias trusted1 -keystore c:\security\truststore  
-file c:\security\httpadapter.cer
```

2. keytool은 이제 keystore 암호를 프롬프트합니다.

-keystore c:\security\truststore를 입력한 경우, SSL -> TrustStore 등록 정보가 c:\security\truststore로 설정되었는지 확인하십시오. 또한 SSL -> TrustStorePassword 등록 정보의 값을 위에서 입력한 암호로 설정하십시오.

사용자 프로젝트 작성

- WebSphere Business Integration System Manager를 사용하여, 새 사용자 프로젝트를 작성하십시오. 117 페이지의 『샘플 내용 로드』에서 작성한 통합 구성요소 라이브러리에서 모든 구성요소를 선택하십시오.

프로젝트 추가 및 전개

1. 서버 인스턴스 보기에서 『사용자 프로젝트 작성』에서 작성한 사용자 프로젝트를 WebSphere ICS에 추가하십시오.

2. 이 사용자 프로젝트의 모든 구성요소를 ICS로 전개하십시오.

ICS 다시 시동

1. 모든 변경사항이 유효한지 확인하려면 ICS를 다시 시동하십시오.
2. 모든 협업 오브젝트, 커넥터 제어기 및 맵이 녹색 상태에 있는지 있는지 확인하려면 System Monitor 도구를 사용하십시오.

비동기 시나리오 실행

이 시나리오는 Asynch Order Service HTTP 서비스를 호출합니다. 시나리오를 실행하기 전에, 데이터 플로우의 이 단계별 개요를 검토하십시오.

1. CLIENT_ASYNC_TLO_Order.Create 이벤트는 Test Connector의 한 인스턴스에서 실행 중인 응용프로그램 SampleSAP에서 발신됩니다.
2. 이벤트는 협업 CLIENT_ASYNC_Order_Collab에 대한 SampleSAP에서 전송됩니다.
3. 이벤트는 협업에서 HTTPConnector로 전송됩니다.
4. 그러면 HTTPconnector는 CLIENT_ASYNC_TLO_Order 오브젝트의 하위인 XML_Order 오브젝트를 찾습니다.
5. 요청 Business Object는 XML Data Handler를 사용하여 XML 메시지로 변환됩니다. HTTPconnector는 프로토콜 Config MO(Meta-Object)의 목적지 속성으로 제공하는 URL로 XML 메시지를 전송합니다. 커넥터가 사용하는 프로토콜 Config MO는 CLIENT_ASYNC_TLO_Order의 핸들러 속성 값에 의해 좌우됩니다. 이 값은 http 또는 https로 설정되어야 합니다.
6. XML 요청은 URL로 게시됩니다. 이전에 언급한 대로 동일한 HTTPConnector는 동일한 URL에서 XML 요청을 청취합니다. 커넥터의 프로토콜 리스너는 XML 메시지를 수신합니다.
7. 커넥터는 XML 메시지를 XML_Order로 변환한 후 SERVICE_ASYNC_TLO_Order 오브젝트를 작성합니다. XML_Order 오브젝트는 SERVICE_ASYNC_TLO_Order 오브젝트의 하위로 설정됩니다.
8. HTTPConnector는 이제 비동기적으로 SERVICE_TLO_Order 오브젝트를 ICS로 게시합니다. 이것은 비동기 URL 호출을 완료합니다.

이것은 비동기 호출(요청만)이므로 응답이 HTTP 클라이언트로 다시 전송되지 않습니다. SERVICE_ASYNC_Order_Collab이 이 오브젝트를 수신하면 협업은 Test Connector의 두 번째 인스턴스로서 실행하는 SampleSiebel이라는 응용프로그램으로 Business Object를 전송합니다. 오브젝트가 Test Connector에 표시됩니다. 응답 성공이 SampleSiebel 응용프로그램에서 선택되면 이벤트는 다시 SERVICE_ASYNC_Order_Collab으로 전송됩니다.

비동기 시나리오를 실행하려면 다음과 같이 하십시오.

1. 아직 실행 중이 아니라면 ICS 통합 브로커를 시작하십시오.
2. Connector for HTTP를 시작하십시오.
3. Test Connector의 두 개의 인스턴스를 시작하십시오.
4. Test Connector를 사용하여 SampleSAPConnector 및 SampleSiebelConnector의 프로파일을 정의하십시오.
5. 각 Test Connector에서 **FILE -> CONNECT AGENT**를 시작하여 에이전트 시뮬레이터를 시작하십시오.
6. Test Connector를 사용하여 SampleSAPConnector를 시뮬레이션하는 동안 메뉴에서 **EDIT -> LOAD BO**를 선택하십시오. 다음 파일을 로드하십시오.

```
WBI_folder\connectors\HTTP\samples\WebSphereICS\OrderStatus  
\CLIENT_ASYNC_TLO_Order.bo
```

Test Connector는 CLIENT_ASYNC_TLO_Order가 로드되었음을 표시합니다.

7. HTTP URL 주소를 확인하십시오.
 - HTTP 샘플을 실행하려면 다음과 같이 하십시오.
 - a. Test Connector에서 CLIENT_ASYNC_TLO_Order Business Object의 핸들러 값을 http로 설정하십시오.
 - b. CLIENT_ASYNC_TLO_Order의 요청 속성을 펼치십시오. 이 속성은 유형 CLIENT_ASYNC_Order Business Object에 대한 것입니다.
 - c. XML_ASYNC_TLO_Order의 HTTPCfgMO 속성을 펼치십시오. 이 속성은 유형 XML_Order_HTTP_CfgMO에 대한 것입니다.
 - d. XML_Order_HTTP_CfgMO의 목적지 속성이 http://localhost:8080/wbia/http/samples로 설정되는지 확인하십시오.
 - HTTPS 샘플을 실행하려면 다음과 같이 하십시오.
 - a. 이것이 HTTPS 호출임에도 불구하고 CLIENT_ASYNC_TLO_Order Business Object의 핸들러 속성 값이 http로 설정되었는지 확인하십시오.
 - b. CLIENT_ASYNC_TLO_Order의 요청 속성을 펼치십시오. 이 속성은 유형 XML_Order Business Object에 대한 것입니다.
 - c. XML_Order의 HTTPCfgMO 속성을 펼치십시오. 이 속성은 유형 XML_Order_HTTP_CfgMO에 대한 것입니다.
 - d. XML_Order_HTTP_CfgMO의 목적지 속성 값이 https://localhost:443/wbia/http/samples로 설정되었는지 확인하십시오.
8. Test Connector로 SampleSAPConnector를 시뮬레이션하는 동안 로드된 테스트 **BO**를 클릭하십시오. 메뉴에서 **REQUEST->SEND**를 선택하십시오. 이벤트 플로우에 대해 자세히 알려면 이 섹션의 앞 부분에서 단계별 개요를 참조하십시오.

9. Test Connector로 SampleSiebelConnector를 시뮬레이트하는 동안 **REQUEST->ACCEPT REQUEST**를 선택하십시오. 이벤트 로드된 SERVICE_ASYNC_TLO_Order.Create가 Test Connector의 오른쪽 패널에 표시됩니다.
10. Business Object를 두 번 클릭하십시오. Business Object가 창에서 열립니다.
11. Business Object의 요청 속성을 펼치십시오. 요청 속성은 유형 SERVICE_ASYNC_Order에 대한 것입니다. OrderId, CustomarId 및 기타 SERVICE_ASYNC_Order 속성을 검사하여 수신한 순서를 확인하십시오. 이것은 비동기 시나리오의 실행을 완료합니다.
12. Business Object를 검사하고 나면 창을 닫으십시오. **REQUEST ->REPLY->SUCCESS**를 선택하십시오.

동기 시나리오 실행

이 시나리오는 Synch OrderStatus Service HTTP 서비스를 호출합니다. 시나리오를 실행하기 전에, 데이터 플로우의 이 단계별 개요를 검토하십시오.

1. CLIENT_SYNCH_TLO_OrderStatus.Retrieve 이벤트는 Test Connector의 한 인스턴스에서 실행 중인 응용프로그램 SampleSAP에서 기원합니다.
2. 이벤트는 SampleSAP에서 CLIENT_SYNCH_OrderStatus_Collab라는 협업으로 전송됩니다.
3. 이벤트는 협업에서 Connector for HTTP로 전송됩니다.
4. Connector for HTTP는 CLIENT_SYNCH_TLO_OrderStatus 오브젝트의 요청 하위인 XML_OrderStatus 오브젝트를 찾습니다.
5. Connector for HTTP는 XML Data Handler를 호출하여 XML_OrderStatus Business Object를 XML 메시지로 변환합니다.
6. XML 요청은 URL로 게시됩니다. 이전에 언급한 대로 동일한 HTTPConnector는 동일한 URL에서 XML 요청을 청취합니다. 커넥터의 프로토콜 리스너는 XML 메시지를 수신합니다.
7. 커넥터의 프로토콜 리스너는 XML 메시지를 XML_OrderStatus로 변환한 후 SERVICE_SYNCH_TLO_Order 오브젝트를 작성합니다. XML_OrderStatus 오브젝트는 SERVICE_SYNCH_TLO_Order 오브젝트의 하위로 설정됩니다.
8. Connector for HTTP는 이제 비동기적으로 SERVICE_SYNCH_TLO_OrderStatus 오브젝트를 WebSphere ICS에서 실행 중인 SERVICE_SYNCH_OrderStatus_Collab 협업으로 게시합니다. 이것은 동기 실행이므로 Connector for HTTP는 협업이 실행하고 응답을 리턴할 때까지 차단된 채로 있습니다.
9. Connector for HTTP는 이제 비동기적으로 SERVICE_TLO_OrderStatus 오브젝트를 WebSphere ICS에서 실행 중인 SERVICE_SYNCH_OrderStatus_Collab 협

업으로 게시합니다. 이것은 동기 실행이므로 Connector for HTTP는 협업이 실행하고 응답을 리턴할 때까지 차단된 채로 있습니다.

10. 값을 편집하고 SampleSiebel 응용프로그램에서 응답 성공을 선택한 후에 이벤트는 다시 SERVICE_SYNCH_OrderStatus_Collab 협업으로 전송됩니다.
11. SERVICE_SYNCH_OrderStatus_Collab은 SERVICE_SYNCH_TLO_OrderStatus 오브젝트를 수신합니다. 그런 다음 협업은 Business Object를 HTTPConnector로 전송합니다.
12. HTTPConnector는 SERVICE_SYNCH_OrderStatus_TLO의 하위인 XML_OrderStatus Business Object를 찾습니다. 이 Business Object는 XML Data Handler에 의해 XML 응답 메시지로 변환됩니다.
13. XML 응답은 다시 HTTP 클라이언트로 전송됩니다.
14. 이 경우 Connector for HTTP의 프로토콜 핸들러인 HTTP 클라이언트가 응답을 수신합니다. 커넥터는 응답 메시지로 XML Data Handler를 호출합니다. XML Data Handler는 응답 메시지를 XML_OrderStatus Business Object로 변환합니다. HTTPConnector는 이 오브젝트를 CLIENT_SYNCH_OrderStatus_TLO의 하위로 설정합니다.
15. CLIENT_SYNCH_OrderStatus_TLO는 CLIENT_SYNCH_OrderStatus_Collab 협업으로 리턴됩니다.
16. CLIENT_SYNCH_OrderStatus_Collab은 CLIENT_SYNCH_OrderStatus_TLO를 Test Connector의 첫 번째 인스턴스로 실행하는 SampleSAP 응용프로그램으로 전송합니다. Test Connector는 이 오브젝트를 표시합니다.

동기 시나리오를 실행하려면 다음과 같이 하십시오.

1. 아직 실행 중이 아니라면 ICS 통합 브로커를 시작하십시오.
2. Connector for HTTP를 시작하십시오.
3. Test Connector의 두 인스턴스를 시작하십시오.
4. Test Connector를 사용하여 SampleSAPConnector 및 SampleSiebelConnector의 프로파일을 정의하십시오.
5. 각 Test Connector에서 **FILE -> CONNECT AGENT**를 선택하여 에이전트 시뮬레이터를 시작하십시오.
6. Test Connector를 사용하여 SampleSAPConnector를 시뮬레이팅하는 동안 메뉴에서 **EDIT -> LOAD BO**를 선택하십시오. 다음 파일을 로드하십시오.

```
WBI_folder\connectors\HTTP\samples\WebSphereICS\OrderStatus
\CLIENT_SYNCH_TLO_OrderStatus.bo
```

Test Connector는 CLIENT_SYNCH_TLO_OrderStatus가 로드되었음을 표시합니다.

7. HTTP URL 주소를 확인하십시오.
 - **HTTP** 샘플을 실행하려면 다음과 같이 하십시오.

- a. Test Connector에서 CLIENT_SYNCH_TLO_OrderStatus Business Object의 핸들러 속성 값이 http로 설정되었는지 확인하십시오.
 - b. CLIENT_SYNCH_TLO_OrderStatus의 요청 속성을 펼치십시오. 이 속성은 유형 XML_OrderStatus Business Object에 대한 것입니다.
 - c. XML_OrderStatus의 HTTPCfgMO 속성을 펼치십시오. 이 속성은 유형 XML_Order_HTTP_CfgMO에 대한 것입니다.
 - d. XML_Order_HTTP_CfgMO의 목적지 속성이 http://localhost:8080/wbia/http/samples로 설정되는지 확인하십시오.
- **HTTPS 샘플:**
 - a. Test Connector에서 CLIENT_SYNCH_TLO_OrderStatus Business Object의 핸들러 속성 값이 https 호출이라고 할지라도 http로 설정되었는지 확인하십시오.
 - b. CLIENT_SYNCH_TLO_OrderStatus의 요청 속성을 펼치십시오. 이 속성은 유형 XML_OrderStatus Business Object에 대한 것입니다.
 - c. XML_OrderStatus의 HTTPCfgMO 속성을 펼치십시오. 이 속성은 유형 XML_Order_HTTP_CfgMO에 대한 것입니다.
 - d. XML_Order_HTTP_CfgMO의 목적지 속성 값이 https://localhost:443/wbia/http/samples로 설정되었는지 확인하십시오.
8. Test Connector로 SampleSAPConnector를 시뮬레이트하는 동안 로드된 테스트 **BO**를 클릭하십시오. 메뉴에서 **REQUEST -> SEND**를 선택하십시오. 데이터 플로우에 대해 자세히 알려면 이 섹션의 앞 부분에서 단계별 개요를 참조하십시오.
 9. 이벤트 로드된 SERVICE_SYNCH_TLO_OrderStatus.Retrieve가 SampleSiebelConnector를 시뮬레이트하는 Test Connector 인스턴스의 오른쪽 패널에 표시됩니다. Business Object를 두 번 클릭하여 창에 표시하십시오.
 10. Business Object의 요청 속성을 펼치십시오. 요청 값을 검사하여 SampleSAPConnector에서 전송된 값이 변하지 않고 그대로인지 확인하십시오.
 11. **LOAD BO**를 선택하여 이 Business Object의 응답 속성을 채우십시오. 다음 과일을 로드하십시오.
 - WBI_folder\connectors\HTTP\samples\WebSphereICS\SERVICE_SYNCH_TLO_OrderStatus.bo
 Test Connector는 SERVICE_SYNCH_TLO_OrderStatus가 로드되었음을 표시해야 합니다.
 12. **REQUEST -> REPLY -> SUCCESS**를 선택하십시오.
 13. 이벤트 로드된 SERVICE_SYNCH_TLO_OrderStatus.Retrieve가 SampleSAPConnector를 시뮬레이트하는 Test Connector의 오른쪽 패널에 표시됩니다.

14. 창에 표시되는 **SERVICE_SYNCH_TLO_OrderStatus.Retrieve** Business Object 를 두 번 클릭하십시오. SampleSiebelConnector가 순서 상태를 리턴한 경우, 채 워진 Business Object의 응답 속성을 보아야 합니다. 순서 상태를 확인하려면 응답 속성을 펼치십시오.
 15. Business Object를 검사하고 나면 창을 닫으십시오. **REQUEST -> REPLY -> SUCCESS**를 선택하십시오.
- 이것은 동기 시나리오의 실행을 완료합니다.

부록 D. HTTPS/SSL 구성

- 『Keystore 설정』
- 128 페이지의 『TrustStore 설정』
- 129 페이지의 『공용 키 인증을 위한 CSR(certification request) 생성』

SSL을 사용하려는 경우 keystore, 인증 및 키 생성을 관리하려면 타사 소프트웨어를 사용해야 합니다. Connector for HTTP는 이 작업의 도구화와 함께 제공되지 않습니다. 그러나 자체 서명 인증서를 작성하고 keystore를 관리하기 위해 IBM JRE에서 제공하는 keytool을 사용하도록 선택할 수 있습니다.

키 및 인증 관리 유틸리티, keytool은 자체 공용/개인용 키 쌍 및 연관된 인증을 관리할 수 있게 합니다. 이것은 자체 인증(다른 사용자나 서비스에 사용자 자신을 인증)이나 디지털 서명을 사용하는 데이터 무결성 및 인증 서비스에서 사용하기 위한 것입니다. 또한 keytool 유틸리티는 통신하는 피어의 공용 키(인증 양식으로)를 저장하도록 허용합니다.

이 부록에서는 keytool을 사용하여 keystore를 설정하는 방법에 대해 설명합니다. 이 부록은 예시 목적만을 위한 것으로 keytool이나 관련 제품의 문서의 대체를 위한 것이 아닙니다. 주의하십시오. keystore를 설정하기 위해 사용하는 도구의 소스 문서를 항상 참조하십시오. keytool에서 자세한 정보는 다음을 참조하십시오.

- <http://java.sun.com/j2se/1.3/docs/tooldocs/tools.html#security>

Keystore 설정

keytool을 사용하여 KeyStore를 작성하려면 우선 KeyStore에서 키 쌍을 작성해야 합니다. 예를 들어, 다음 명령 행을 입력하는 경우

```
keytool -genkey -alias httpadapter -keystore c:\security\keystore
```

keytool은 즉시 암호를 프롬프트합니다. 사용자 선택 암호(keytool 암호 이내)를 입력할 수 있지만, SSL ” KeyStorePassword 커넥터 등록 정보의 값으로 keytool에 입력된 암호를 지정해야 합니다. 자세한 정보는 73 페이지의 『KeyStorePassword』에서 참조하십시오.

샘플 명령은 keystore로 명명된 keystore를 c:\security\keystore 디렉토리에 작성합니다. 이에 따라 c:\security\keystore를 SSL ” KeyStore 커넥터 계층 구조 등록 정보의 값으로 입력합니다. 또한 위의 명령행 예에서, -alias httpadapter를 SSL ” KeyStoreAlias 커넥터 계층 구조 등록 정보의 값으로 입력합니다. 그러면 keytool 유

틸리티는 인증의 세부사항을 프롬프트합니다. 다음에서는 각 프롬프트에서 입력할 수 있는 내용을 예시합니다. (keytool 문서 참조.)

```
What is your first and last name?  
[Unknown]: HostName  
What is the name of your organizational unit?  
[Unknown]: wbi  
What is the name of your organization?  
[Unknown]: IBM  
What is the name of your City or Locality?  
[Unknown]: Burlingame  
What is the name of your State or Province?  
[Unknown]: CA  
What is the two-letter country code for this unit?  
[Unknown]: US  
Is <CN=HostName, OU=wbi, O=IBM, L=Burlingame,  
ST=CA, C=US> correct?  
[no]: yes
```

그러면 keytool은 암호를 프롬프트합니다:

<httpadapter>의 키 암호를 입력하십시오(keystore 암호와 동일한 경우 RETURN).

동일한 암호를 사용하려면 리턴을 누르십시오. 자체 서명 인증을 사용하려는 경우 위에서 작성한 인증을 내보내려고 할 수 있습니다. 이 경우 명령행에서 다음을 입력하십시오.

```
keytool -export -alias httpadapter -keystore c:\security\keystore -file  
wsadapter.cer
```

keytool은 이제 keystore 암호를 프롬프트합니다. 위에 입력한 암호를 입력하십시오.

TrustStore 설정

다음의 경우 TrustStore를 설정할 수도 있습니다.

HTTPS 프로토콜 리스너가 클라이언트를 인증하게 하려면, SSL " UseClientAuth 커넥터 구성 등록 정보를 true로 설정하십시오. 이 경우 HTTPS 프로토콜 리스너는 TrustStore가 모든 신뢰 클라이언트의 인증을 포함할 것으로 예상합니다. 커넥터는 JSSE 기본값 메커니즘을 사용하여 클라이언트를 신뢰함을 참고하십시오.

HTTPS 서비스를 호출하려는 경우 HTTP-HTTPS 프로토콜 핸들러는 TrustStore가 서비스를 신뢰하도록 요구합니다. 이것은 TrustStore가 모든 신뢰 HTTP 서비스의 인증을 포함해야 함을 의미합니다. 커넥터는 JSSE 기본값 메커니즘을 사용하여 클라이언트를 신뢰함을 참고하십시오. 신뢰 인증을 TrustStore로 가져오려면, 다음 명령을 입력하십시오.

```
keytool -import -alias trusted1 -keystore c:\security\truststore -file  
c:\security\trusted1.cer
```

keytool은 이제 keystore 암호를 프롬프트합니다. -keystore c:\security\truststore 를 입력한 경우, SSL -> TrustStore 계층 구조 등록 정보가 c:\security\truststore 로 설정되었는지 확인하십시오. 또한 SSL -> TrustStorePassword 계층 구조 등록 정보의 값을 이전에 입력한 암호로 설정해야 합니다.

공용 키 인증을 위한 CSR(certification request) 생성

SSL 데이터 교환이 ID를 신뢰하는 기존의 신뢰 파트너 간에 이루어지면 자체 서명 인증이 적합할 수 있습니다. 그러나 인증 기관(CA)이 서명하면 다른 사람이 인증서를 더 신뢰하기 쉽습니다.

keytool 유틸리티를 사용하여 CA에서 서명한 인증을 얻으려면 우선 CSR(Certificate Signing Request)을 생성한 후 CSR을 CA에 제공하십시오. 그러면 CA는 인증에 서명하여 리턴합니다.

다음 명령을 입력하여 CSR을 생성합니다.

```
keytool -certreq -alias wsadapter -file httpadapter.csr  
-keystore c:\security\keystore
```

명령에서, alias는 개인용 키에 작성한 keystore 별명입니다. keytool 유틸리티는 사용자 CA에 제공한 CSR 파일을 생성합니다. 그러면 CA는 서명 인증을 사용자에게 제공합니다. 사용자는 이 인증을 keystore로 가져와야 합니다. 그렇게 수행하기 위해 다음 명령을 입력합니다.

```
keytool -import -alias wsadapter -keystore c:\security\keystore -trustcacerts  
-file casignedcertificate.cer
```

사용자가 가져오면 keystore의 자체 서명 인증이 CA 서명 인증으로 바뀝니다.

주의사항

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

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

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.

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

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

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

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

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

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

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

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

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

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

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

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

프로그래밍 인터페이스 정보

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

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

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

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

상표 및 서비스표

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

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

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

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

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

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



IBM WebSphere Business Integration Adapter Framework, V2.4.0

IBM