

IBM WebSphere Business Integration
Adapters



Adapter for Web Services 사용자 안내서

버전 3.2.x

IBM WebSphere Business Integration
Adapters



Adapter for Web Services 사용자 안내서

버전 3.2.x

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 247 페이지의 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

새 개정판에서 별도로 명시하지 않는 한 이 문서의 이 개정판은 IBM WebSphere Business Integration Adapters for Web Services 버전 3.2.x 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견을 보내시려면 ibmkspe@kr.ibm.com으로 전자 우편을 보내십시오. 고객의 의견을 기대합니다.

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

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

이 책의 정보

IBM^(R) WebSphere^(R) Business Integration Adapter 포트폴리오는 앞선 e-business 기술과 엔터프라이즈 응용프로그램에 대한 통합 연결성을 제공합니다. 이 책에서는 Adapter for Web Services의 설치, 구성 및 Business Object 개발에 대해 설명합니다.

이 책의 사용자

이 책은 IBM WebSphere 고객, 컨설턴트, 개발자 및 WebSphere Business Integration Adapter for Web Service를 구현하고 있는 사람을 위한 것입니다.

이 책을 사용하기 전에

다양한 전제조건이 이 책의 전체에서 인용됩니다. 이 중에서 많은 부분이 웹 서비스에 대한 정보 또는 자원을 포함하는 웹 사이트에 대한 참조로 구성됩니다. 사용자는 또한 WebSphere Business Integration 시스템 구현에도 익숙해야 합니다. 우선 *Technical Introduction to IBM WebSphere InterChange Server*를 보는 것이 좋으며, 여기에는 보다 상세한 문서에 대한 상호 참조가 들어 있습니다.

관련 서적

이 제품과 함께 제공되는 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 정보의 경우:
 - <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 사이트에는 문서를 다운로드, 설치 및 보는 데 필요한 간단한 지침이 포함되어 있습니다.

인쇄 규칙

이 책은 다음 규칙을 사용합니다.

<code>courier</code> 글꼴	명령어, 파일 이름, 사용자가 입력하는 정보 또는 시스템이 화면에 인쇄하는 정보 등의 리터럴 값을 표시합니다.
굵은체	새로운 용어가 처음으로 나타날 때 표시합니다.
기울임	변수 이름 또는 상호 참조를 표시합니다.
파란색 윤곽선	온라인 메뉴얼을 볼 때만 볼 수 있는 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조의 오브젝트로 건너뛰려면 윤곽선 내부를 클릭하십시오.
{ }	구문 행에서 사용자가 반드시 하나만을 선택해야 하는 옵션 세트는 중괄호 안에 표시됩니다.
[]	구문 행에서 선택적 매개변수는 대괄호 안에 표시됩니다.
...	구문 행에서 말줄임표(...)는 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 서로 구분할 이름의 개별 요소를 포함합니다.
/, \	이 책에서 백슬래시(\)는 디렉토리 경로의 규칙으로 사용됩니다. UNIX 설치의 경우 백슬래시를 슬래시(/)로 대체합니다. 모든 IBM 제품 경로 이름은 제품이 시스템에 설치되는 디렉토리와 관련되어 있습니다.
<code>%text%</code> 및 <code>\$text</code>	퍼센트(%) 부호 내의 텍스트는 Windows <code>text</code> 시스템 변수 또는 사용자 변수의 값을 표시합니다. UNIX 환경에서의 동등한 표기법은 <code>\$text</code> 로서, <code>text</code> UNIX 환경 변수의 값을 표시합니다.
<i>ProductDir</i>	IBM WebSphere Business Integration Adapter 제품이 설치된 디렉토리를 표시합니다. CROSSWORLDS 환경 변수에는 <i>ProductDir</i> 디렉토리 경로가 있습니다. 이 경로의 기본값은 <code>IBM\WebSphereAdapters</code> 입니다.
”	파일 선택 “갱신” SGML 참조와 같이 메뉴의 선택항목 표시

목차

이 책의 정보	iii	추적	116
이 책의 사용자	iii	제 5 장 SOAP Data Handler	119
이 책을 사용하기 전에.	iii	SOAP Data Handler 구성.	119
관련 서적	iii	SOAP Data Handler 처리.	127
인쇄 규칙	iv	응용프로그램 특정 정보 기능 사용	133
이 릴리스의 새로운 기능.	vii	플러그 가능 이름 핸들러 지정.	153
릴리스 3.2.x의 새로운 기능.	vii	제한사항	154
릴리스 3.1.x의 새로운 기능.	vii	제 6 장 요청 처리용 협업 사용 가능	157
릴리스 3.0.x의 새로운 기능.	viii	요청 처리 협업 선택 목록	157
제 1 장 커넥터의 개요.	1	제 7 장 협업을 웹 서비스로 표시	159
Adapter for Web Services 환경	1	프로시저 체크리스트	159
용어	6	Business Object 식별 또는 개발.	160
Connector for Web Services 구성요소	8	협업 템플릿 선택 또는 개발.	160
Connector for Web Services의 구조	12	새 협업 오브젝트의 포트 바인딩	160
설치, 구성 및 설계 체크리스트	14	WSDL 구성 마법사	163
제한사항	15	제 8 장 WSDL ODA 사용	171
제 2 장 설치 및 시작.	17	WSDL ODA 시작	171
설치 작업의 개요	17	WSDL ODA 실행	172
커넥터 및 관련 파일 설치	17	에이전트 구성	173
설치된 파일 구조	18	WSDL 문서 지정.	175
구성 작업의 개요	19	선택사항 확인	177
어댑터의 다중 인스턴스 실행	20	오브젝트 생성	178
커넥터 시작 및 중지	22	제한사항	179
제 3 장 Business Object 요구사항	25	제 9 장 문제점 해결.	181
Business Object 메타 데이터	25	시작 문제점.	181
커넥터 Business Object 구조	26	런타임 오류.	183
Business Object 개발.	61	부록 A. 커넥터의 표준 구성 등록 정보.	185
제 4 장 웹 서비스 커넥터	63	신규 및 삭제된 등록 정보	185
커넥터 처리	64	표준 커넥터 등록 정보 구성	185
SOAP/HTTP(S) 웹 서비스	66	표준 등록 정보 요약.	187
SOAP/JMS 웹 서비스.	67	표준 구성 등록 정보.	190
이벤트 처리	68	부록 B. Connector Configurator	203
요청 처리	80	Connector Configurator 개요.	203
커넥터 및 JMS	89	Connector Configurator 시작.	204
SSL.	91	System Manager에서 Configurator 실행	205
커넥터 구성	94	커넥터 특정 등록 정보 템플릿 작성	205
시작 시의 커넥터	115	새 구성 파일 작성	208
로깅	116		

기존 파일 사용	209	부록 D. 3.0.x로 이주	239
구성 파일 완료	210	역호환성	239
구성 파일 등록 정보 설정	211	업그레이드 작업	239
구성 파일 저장	218	부록 E. HTTPS/SSL 구성	243
구성 파일 변경	218	키스토어 설정	243
구성 완료	219	TrustStore 설정	244
국제화된 환경에서 Connector Configurator 사용	219	공용 키 인증을 위해 CSR(Certificate Signing Request) 생성	245
부록 C. Adapter for Web Services 자습서	221	주의사항	247
자습서 정보	221	프로그래밍 인터페이스 정보	248
시작하기 전에	223	상표 및 서비스표	249
설치 및 구성	223		
비동기 시나리오 실행	230		
동기 시나리오 실행	233		

이 릴리스의 새로운 기능

릴리스 3.2.x의 새로운 기능

이 릴리스에는 다음과 같은 개선사항이 포함되어 있습니다.

- 모든 리스너 및 프로토콜 핸들러에서 사용자 정의 등록 정보에 대한 지원
- SOAP/HTTP(S) 리스너 및 핸들러에서 표준 머리글의 전파 및 읽기에 대한 지원
- SOAP/JMS 리스너 및 핸들러에서 주요 JMS 표준 머리글의 읽기에 대한 지원
- SOAP/HTTP 및 SOAP/HTTPS 리스너에서 비동기 HTTP 처리의 성공 리턴 코드 사용자 정의에 대한 지원
- SOAP/HTTP-HTTPS 핸들러에서 비동기 HTTP 처리의 승인 가능한 다중 성공 코드에 대한 지원
- SOAP/JMS 리스너에서 바이트 메시지에 대한 지원
- 요청 Content-Type 및 요청 URL을 기반으로 한 SOAP/HTTP 및 SOAP/HTTPS 리스너에서 MIME 유형 및 메시지 charset 사용자 정의에 대한 지원
- 응답 ContentType을 기반으로 한 SOAP/HTTP-HTTPS 핸들러에서 런타임 MIME 유형 및 메시지 charset 사용자 정의에 대한 지원
- SOAP/HTTP-HTTPS 핸들러는 물론 SOAP/HTTP 및 SOAP/HTTPS 리스너에서 임의의 ContentType 메시지를 보내고 받을 수 있는 기능
- SOAP/HTTP-HTTPS 핸들러는 물론 SOAP/HTTP 및 SOAP/HTTPS 리스너에서 전송 메시지의 MIME 유형 및 charset을 런타임 시 지정하는 기능

3.2.x 버전 이후로 Microsoft Windows NT에서는 웹 서비스에 대한 어댑터가 실행되지 않습니다.

WSDL ODA가 향상되었습니다. 이제 WSDL ODA는 인터페이스 파일을 읽어 이벤트 처리 TLO를 생성할 수 있습니다.

어댑터 설치 정보는 이 안내서에서 이동되었습니다. 자세한 정보는 17 페이지의 『커넥터 및 관련 파일 설치』를 참조하십시오.

릴리스 3.1.x의 새로운 기능

웹 서비스 커넥터가 향상되었습니다. 새 기능 및 구성요소는 다음과 같습니다.

- HTTP 신임 전파의 지원. 자세한 정보는 37 페이지의 『이벤트 처리를 위한 HTTP 신임 전파』, 55 페이지의 『요청 처리를 위한 HTTP 신임 전파』 및 106 페이지의 『ProxyServer』를 참조하십시오.

- WSDL ODA 및 SOAP Data Handler는 rpc/literal, rpc/encoded 및 document/literal의 Style/Use 등록 정보를 지원합니다. 자세한 정보는 123 페이지의 『SOAP 메시지의 Style 및 Use 영향』을 참조하십시오.
- WSDL 구성 마법사는 SOAP 구성 MO에서 Use 속성을 지원합니다. Use 속성에 대한 자세한 정보는 121 페이지의 『SOAP 구성 Meta Object: 모든 SOAP Business Object의 하위 오브젝트』를 참조하십시오. 또한 WSDL 구성 마법사는 일부 요소의 국제화 문자를 지원합니다. 자세한 정보는 168 페이지의 『처리 요구사항 및 예외』를 참조하십시오.
- JMS 프로토콜 리스너는 새 구성 등록 정보(SessionPoolSize)를 갖습니다. 자세한 정보는 103 페이지의 표 41을 참조하십시오.
- 커넥터는 다음 플랫폼에서 실행합니다.
 - Microsoft Windows NT 4.0 Service Pack 6A 또는 Windows 2000
 - Solaris 7, 8 또는 AIX 5.1, 5.2 또는 HP UX 11.i

릴리스 3.0.x의 새로운 기능

웹 서비스 커넥터가 다시 설계되었습니다. 새 기능 및 구성요소는 다음과 같습니다.

- SOAP/HTTP, SOAP/HTTPS 및 SOAP/JMS 웹 서비스를 위한 양방향 지원을 갖는 웹 서비스 커넥터(이전 릴리스의 SOAP 커넥터와 프록시 클래스 대체)
- 협업을 웹 서비스로서 표시하기 위한 향상된 WebSphere Business Integration Toolset 성능
- WSDL(Web Services Description Language) ODA(Object Discovery Agent)
- 응용프로그램 특정 정보(ASI) 성능을 이용하는 새 Business Object 구조
- 새로운 SOAP Data Handler

1장에는 새 어댑터의 개요가 있습니다. 이 제품의 이전 릴리스와의 역호환성에 대해서는 239 페이지의 부록 D 『3.0.x로 이주』를 참조하십시오.

“CrossWorlds” 이름은 더 이상 전체 시스템을 설명하거나 구성요소 또는 도구 이름을 수정하는 데 사용되지 않습니다. 다른 점에서는 이전과 거의 동일합니다. 예를 들어, “CrossWorlds System Manager”는 이제 “System Manager”이고 “CrossWorlds InterChange Server”는 “WebSphere InterChange Server”입니다.

커넥터가 국제화되었습니다. 자세한 정보는 4 페이지의 『로케일 종속 데이터』 및 185 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

이 책은 WebSphere ICS(Interchange Server)와 함께 이 어댑터 사용에 대한 정보를 제공합니다.

제 1 장 커넥터의 개요

- 『Adapter for Web Services 환경』
- 6 페이지의 『용어』
- 8 페이지의 『Connector for Web Services 구성요소』
- 12 페이지의 『Connector for Web Services의 구조』
- 14 페이지의 『설치, 구성 및 설계 체크리스트』
- 15 페이지의 『제한사항』

커넥터는 WebSphere Business Integration Adapter for Web Service의 런타임 구성 요소입니다. 비즈니스는 커넥터를 사용하여 해당 조직 내부 또는 거래 업체가 사용하는 웹 서비스를 모두 모아서 공개하고 사용할 수 있습니다. 커넥터 및 이 문서에서 설명하는 다른 구성요소는 SOAP(Simple Object Access Protocol) 메시지의 본문에서 Business Object 정보 교환에 필요한 기능을 제공합니다.

이 장은 WebSphere Business Integration Adapter for Web Service를 구현하는 데 사용되는 범위, 구성요소, 설계 도구 및 구조에 대해 설명합니다. 이 책에 설명된 웹 서비스 구성요소를 설치 및 구성하기 위해 완료해야 하는 작업의 개요도 소개합니다. 구성요소 설치 및 구성에 대한 정보는 14 페이지의 『설치, 구성 및 설계 체크리스트』을 참조하십시오.

주: Adapter for Web Services는 표준 Adapter Framework API를 구현합니다. 이러한 이유로, 어댑터는 프레임워크가 지원하는 통합 브로커에서 작동될 수 있습니다. 그러나 어댑터에서 제공하는 기능은 IBM WebSphere InterChange Server(ICS) 통합 브로커를 지원하기 위해 특별히 설계된 것입니다. 이에 따라 System Manager에 웹 서비스로 나타내기 옵션을 선택하면, ICS를 나타내며 다른 통합 브로커는 나타내지 않습니다.

Adapter for Web Services 환경

어댑터를 설치, 구성 및 사용하기 전에 환경 요구사항을 이해해야 합니다.

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

브로커 호환성

어댑터가 사용하는 어댑터 프레임워크는 어댑터가 통신하는 통합 브로커(또는 브로커)의 버전과 호환 가능해야 합니다. Adapter for Web Services의 3.2 버전은 다음 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:
 - 4.1.1
 - 4.2
 - 4.2.1
 - 4.2.2

예외사항은 릴리스 정보를 참조하십시오.

주: 어댑터의 3.2 버전에 문서화된 모든 기능을 활용하려면 WebSphere InterChange Server 버전 4.2.2를 설치해야 합니다.

소프트웨어 전제조건

Connector for Web Service를 설치하기 전에 다음 가정 및 소프트웨어 요구사항을 검토하십시오.

- 커넥터 및 기타 구성요소의 설계는 SOAP 1.1에 공개된 스펙을 기반으로 합니다.
- SOAP/JMS 웹 서비스를 사용하려면 고유한 JMS 및 JNDI 구현을 설치해야 합니다.
- HTTPS/SSL을 사용하는 경우, 키스토어 및 트러스트스토어 작성을 위한 고유한 타사 소프트웨어가 필요합니다.

어댑터 플랫폼

어댑터가 다음 플랫폼(운영 체제)에서 실행됩니다.

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

표준 및 API

다양한 표준과 기술이 웹 서비스에 네트워크를 통한 기능 액세스를 제공합니다.

어댑터가 사용하는 표준은 다음과 같습니다.

- SOAP 버전 1.1. SOAP 스펙 버전 1.1을 참조하십시오.
- WSDL 1.1 SOAP 바인딩
- HTTP 1.0
- JMS 1.0.2

어댑터가 사용하는 API는 다음과 같습니다.

- Apache SOAP 2.3.1 API: 커넥터는 Apache Foundation의 SOAP API를 통합합니다. Apache SOAP API는 SOAP 버전 1.1의 개방형 소스 구현입니다. Apache SOAP API에는 다음 요구사항이 있습니다.
 - Java Activation Framework 1.0.1(activation.jar)
 - JavaMail(TM) API 1.2(mail.jar)
- Xerces Java 구문 분석기 1.4.3 이상 Xerces2는 완전히 호환되는 XML 스키마 프로세서입니다.
- JMS API 버전 1.0.2
- WSDL4J 1.2.1 - WSDL4J(Web Service Description Language for Java API)는 WSDL 문서에 대한 오브젝트 모델을 제공합니다.
- UDDI4J-WSDL 2.1.0 - UDDI4J-WSDL API는 WSDL4J API에 의해 정의된 일부 클래스는 물론 UDDI4J API에 있는 클래스를 캡슐화합니다.
- JNDI 1.2.1
- WSDL4J 1.0
- IBM JSSE 1.0.2

구성에 따라서는 추가 소프트웨어를 설치해야 할 수 있습니다. 아래 섹션에서 이들 추가 소프트웨어에 대해 설명합니다.

JMS 프로토콜

JMS 프로토콜을 사용할 경우, JMS 제공자를 설치하고 대기열을 작성해야 합니다. 대기열 작성은 실제로는 사용자 요구사항에 의존합니다. 협업을 웹 서비스로 공개하고 외부 웹 서비스 호출을 위해 JMS 프로토콜을 사용할 수 있습니다. 자세한 정보는 89 페이지의 『커넥터 및 JMS』를 참조하십시오.

JNDI: JNDI를 구성한 후 커넥터에 대한 JNDI 구성 등록 정보에 적당한 매개변수를 입력해야 합니다. 또한 연결 팩토리와 JMS 목적지(대기열) 오브젝트가 JNDI에서 사용 가능한지 확인해야 합니다. JNDI를 사용하기 원하고 JNDI 구현이 없는 경우, Sun Microsystems에서 파일 시스템 JNDI의 참조 구현을 다운로드할 수 있습니다. 자세한 정보는 89 페이지의 『커넥터 및 JMS』를 참조하십시오.

SSL

SSL을 사용할 계획인 경우, 키스토어, 인증 및 키 생성 관리를 위해 타사 소프트웨어를 사용해야 합니다. 키스토어, 인증을 설정하거나 키 생성을 위한 도구가 제공되지 않습니다. keytool(IBM JRE와 함께 제공됨)를 사용하여 자체 서명한 인증을 작성하고 키스토어를 관리할 수 있습니다. 자세한 정보는 91 페이지의 『SSL』을 참조하십시오.

로케일 종속 데이터

커넥터가 2바이트 문자 세트를 지원할 수 있도록 국제화되었습니다. 커넥터가 한 문자 코드를 사용하는 위치에서 다른 코드 세트를 사용하는 위치로 데이터를 전송하는 경우, 데이터 의미를 보존하기 위해 문자 변환을 수행합니다.

JVM(Java Virtual Machine) 내의 Java 런타임 환경은 데이터를 Unicode 문자 코드 세트로 표시합니다. Unicode는 가장 잘 알려져 있는 문자 코드 세트(1바이트 및 여러 바이트 모두)로 문자에 대한 인코딩을 포함합니다. WebSphere Business Integration System에서 대부분의 구성요소는 Java로 작성됩니다. 따라서 데이터가 대부분의 통합 구성요소 간에 전송될 때, 문자를 변환할 필요가 없습니다.

주: 커넥터는 국제화되지 않았습니다. 이것은 추적 및 로그 메시지가 변환되지 않음을 의미합니다.

웹 서비스 커넥터

이 섹션은 로컬화 및 커넥터에 대해 설명합니다.

이벤트 공고: 커넥터는 이벤트 공고용 플러그 사용 가능한 프로토콜 리스너를 사용합니다. 프로토콜 리스너는 전송에서 SOAP 메시지를 추출하고 SOAP Data Handler를 호출합니다. 이 섹션은 각 리스너가 전송을 통해 SOAP 메시지를 인코딩하는 방법에 대해 설명합니다.

- **SOAP/HTTP 및 SOAP/HTTPS 리스너** 이들 리스너는 HTTP 요청 메시지의 본문을 바이트 단위로 읽습니다. 본문의 인코딩은 HTTP Content-Type 머릿글의 charset 매개변수에서 제공됩니다. charset 매개변수가 없으면 ISO-8859-1(ISO Latin 1)로 간주됩니다. 리스너는 이 인코딩을 사용하여 요청 메시지의 본문을 Java 문자열로 변환합니다. 이 Java 문자열은 SOAP Data Handler를 호출하는 데 사용됩니다. 동기(요청-응답) 웹 서비스의 경우, SOAP Data Handler는 협업에서 리턴하는 Business Object를 사용하여 호출됩니다. SOAP Data Handler에서 리턴하는 Java 문자열은 HTTP 요청 메시지의 인코딩을 사용하여 바이트로 변환됩니다.
- **SOAP/JMS 리스너** 이 리스너는 JMS 바이트 메시지는 물론 JMS 텍스트 메시지도 지원합니다.

요청 처리: 커넥터는 요청 처리를 위해 플러그 가능한 프로토콜 핸들러를 사용합니다. 프로토콜 핸들러는 SOAP Data Handler를 호출합니다. 이 섹션은 각 핸들러가 전송을 통해 SOAP 메시지를 인코딩하는 방법에 대해 설명합니다.

- **SOAP/HTTP-HTTPS 핸들러** 이러한 핸들러는 SOAP Data Handler를 호출합니다. 웹 서비스 요청을 구성하기 위해, Data Handler가 리턴하는 문자열이 UTF 8 인코딩을 사용하는 바이트로 변환됩니다. 동기(요청-응답) 웹 서비스의 경우 프로토콜 핸들러는 HTTP 응답 메시지의 본문을 읽습니다. 본문의 인코딩은 HTTP Content-Type 머릿글의 charset 매개변수에서 제공됩니다. charset 매개변수가 없으면 ISO-8859-1로 간주합니다. 핸들러는 이 인코딩을 사용하여 응답 메시지의 본문을 Java 문자열로 변환합니다. SOAP Data Handler는 이 문자열을 사용하여 호출됩니다.
- **SOAP/JMS 핸들러** 이 리스너는 JMS 텍스트 메시지를 지원합니다.

SOAP Data Handler

이 섹션은 로컬화 및 SOAP Data Handler에 대해 설명합니다.

SOAP 문자 제한사항: XML 요소 이름과 속성 이름은 Business Object 이름, Business Object 속성 이름 또는 Business Object 응용프로그램 특정 정보에서 사용할 수 있는 적합한 ascii 문자여야 합니다.

국제화된 문자는 Business Object 이름 또는 Business Object 속성 이름에서 지원되지 않습니다. 속성 값만 국제화할 수 있습니다.

SOAP Data Handler 처리: SOAP 메시지를 Business Object로 변환할 때, Data Handler는 문자열만을 수신할 수 있습니다. Data Handler는 단순히 문자열 값으로 Business Object를 채우고 Business Object를 리턴합니다. Java 문자열은 UCS2이므로 2바이트 가능 문자는 문제점 없이 변환됩니다. XML 요소와 속성 값만 비ASCII 문자가 될 수 있습니다(문자 제한사항 참조). Business Object를 SOAP 메시지로 변환할 때, Data Handler는 Xerces 구문 분석기를 사용하여 Business Object를 문자열로 변환합니다. Java 문자열은 UCS2이므로 2바이트 가능 문자는 문제점 없이 변환됩니다. XML 요소와 속성 값만 비ASCII 문자가 될 수 있습니다(문자 제한사항 참조).

WSDL ODA

이 섹션은 로컬화 및 WSDL ODA에 대해 설명합니다.

WSDL ODA는 WSDL 파일에서 합법적인 ASCII 외의 문자를 지원하지 않습니다. WSDL ODA는 다른 문자 세트로 된 파일 이름과 URL을 지원할 수 있습니다. 그러나 이들 파일의 내용은 반드시 합법적인 ASCII로 되어 있어야 합니다.

WSDL ODA의 구성 에이전트 테이블에 있는 등록 정보는 다음과 같이 국제화됩니다.

- **WSDL_URL** URL은 자국어일 수 있습니다.
- **UDDI_InquiryAPI_URL** UDDI 레지스트리 지원을 확인하십시오.
- **WebServiceProvider** 합법적인 ASCII 문자만 가능
- **WebService** 합법적인 ASCII 문자만 가능

- **MimeType** 합법적인 ASCII 문자만 가능
- **BOPrefix** 합법적인 ASCII 문자만 가능
- **BOVerb** 합법적인 ASCII 문자만 가능
- **Collaboration** 합법적인 ASCII 문자만 가능

용어

다음 용어가 이 안내서에서 사용됩니다.

- **ASI(Application-Specific Information)**는 특정 응용프로그램 또는 기술에 맞게 조정된 코드입니다. ASI는 Business Object 정의의 Business Object 레벨과 속성 레벨에 모두 존재합니다.
- **ASBO(Application-Specific Business Object)** ASI를 포함할 수 있는 Business Object
- **BO(Business Object)** 비즈니스 엔티티(예: 고객)와 데이터에 대한 조치(예: 작성 또는 갱신 조작)를 나타내는 속성 세트. IBM WebSphere 시스템의 구성요소는 Business Object를 사용하여 정보를 교환하고 조치를 트리거합니다.
- **Content-Type** *type/subtype* 및 선택적 매개변수를 포함하는 HTTP 프로토콜 머리글. 예를 들어, Content-Type 값에서 `text/xml; charset=ISO-8859-1`, `text/xml` 는 유형/부속 유형이며 `charset=ISO-8859-1`은 선택적 Charset 매개변수입니다.
- **ContentType**은 Content-Type 머리글 값 중에서 *type/subtype* 부분만을 참조합니다. 예를 들어, Content-Type 값에서 `text/xml; charset=ISO-8859-1`, `text/xml` 은 이 문서에서는 ContentType으로 참조됩니다.
- **MO_DataHandler_DefaultSOAPConfig** 특별히 SOAP Data Handler를 위한 하위 Data Handler Meta Object
- 일반 **Business Object(GBO)** ASI가 없고 응용프로그램과 연결되어 있지 않은 Business Object
- **MO_DataHandler_Default** 인스턴스화하려는 Data Handler를 판별하기 위해 커넥터 에이전트가 사용하는 Data Handler Meta Object. 이것은 커넥터의 `DataHandlerMetaObjectName` 구성 등록 정보에 지정됩니다.
- 비최상위 레벨 **Business Object(비TLO)** 비TLO는 웹 서비스 TLO 구조를 고수하지 않는 모든 Business Object입니다.
- 프로토콜 구성 **MO** 요청 처리 중에 SOAP/JMS, SOAP/HTTP-HTTPS 프로토콜 핸들러는 프로토콜 구성 MO를 사용하여 대상 웹 서비스의 목적지를 판별합니다. 이벤트 처리 중에 협업을 SOAP/JMS 웹 서비스로 표시하려는 경우, 커넥터는 프로토콜 구성 MO를 사용하여 SOAP/JMS 프로토콜 리스너에서 협업으로 JMS 메시지 머리글 정보를 전달합니다.

- **SOAP(Simple Object Access Protocol)**는 전자 통신에 대한 기본 프로토콜로서 XML로 작성된 간단한 요청 및 응답 메시지를 사용하는 모델을 정의합니다. SOAP 메시지 전달은 플랫폼에 독립적인 원격 프로시저 호출(RPC) 메커니즘이지만, 모든 종류의 XML 정보의 교환(문서 교환)에 사용할 수 있습니다.
- **SOAP Business Object** SOAP Business Object는 TLO의 하위이며 SOAP 요청, SOAP 응답 또는 SOAP 결합 Business Object가 될 수 있습니다. SOAP Business Object에는 SOAP Business Object의 하위인 SOAP 구성 MO를 포함하여 SOAP Data Handler가 처리하는 데 필요한 정보가 들어 있으며 또한 SOAP 머리글 컨테이너 Business Object도 들어 있습니다.
- **SOAP 구성 MO(구성 Meta Object) Data Handler**는 단일 변환(예: SOAP 메시지에서 SOAP Business Object로)에 대한 구성 정보가 있는 오브젝트를 필요로 합니다. 이 정보는 SOAP Business Object의 하위 오브젝트에 메타 데이터로 저장됩니다. 이 하위 오브젝트는 SOAP 구성 MO입니다.
- **SOAP 머리글 하위 Business Object** Business Object는 SOAP 메시지의 단일 머리글 요소를 표시합니다. 머리글 요소는 SOAP 메시지의 SOAP-Env:Header 요소의 바로 아래 하위입니다. 머리글 컨테이너 Business Object의 모든 속성은 이러한 유형이어야 합니다. 이 Business Object에는 actor 및 mustUnderstand 속성이 있을 수 있습니다. 이러한 속성은 SOAP 머리글 요소의 actor 및 mustUnderstand 속성에 해당합니다.
- **SOAP 머리글 컨테이너 Business Object** SOAP 메시지의 머리글에 대한 정보를 포함하는 Business Object. 이 Business Object에는 하나 이상의 하위 Business Object가 들어 있습니다. 각 하위 Business Object는 SOAP 메시지의 머리글 항목을 표시합니다. SOAP Data Handler Business Object에는 SOAP 머리글 컨테이너 Business Object 유형의 속성이 있을 수 있습니다. 이 속성을 SOAP 머리글 속성이라고도 합니다. 이러한 속성에는 119 페이지의 제 5 장 『SOAP Data Handler』에 설명된 대로 특수한 응용프로그램 특정 정보 요구사항이 있습니다. 이 속성은 SOAP Business Object의 바로 아래의 하위 오브젝트여야 합니다.
- **최상위 레벨 Business Object** 웹 서비스 최상위 레벨 Business Object에는 SOAP 요청, SOAP 응답(선택적) 및 하나 이상의 SOAP 결합(선택적) Business Object가 있습니다. TLO는 커넥터가 이벤트 처리와 요청 처리 모두를 위해 사용합니다.
- **웹 서비스**는 제품 작성, 처리 및 연속적인 공급을 목적으로 네트워크를 통해 설명, 공개, 탐색 또는 호출이 가능한 자급식 모듈형의 동적 분산 응용프로그램입니다. 웹 서비스는 로컬, 분산 또는 웹 기반일 수 있습니다. 웹 서비스는 개방형 표준(예: TCP/IP, HTTP, Java, HTML 및 XML) 위쪽에 빌드됩니다. 웹 서비스는 새로운 표준 기술(예: 메시지 전달을 위한 SOAP(Simple Object Access Protocol) 및 공개를 위한 UDDI(Universal Description, Discovery and Integration) 및 WSDL(Web Service Description Language))을 사용합니다.

- **UDDI(Universal Description, Discovery and Integration)**는 웹 서비스에 대한 정보를 공개하고 발견하는 방법을 정의하는 스펙입니다. UDDI 스펙은 UDDI 레지스트리 정보에 대한 프로그램적 액세스를 허용하는 XML 기반 인터페이스(API)를 제공합니다. SOAP는 이들 API에 대한 기본 RPC 메커니즘입니다.
- **WSDL(Web Services Description Language)**은 웹 서비스에 대한 소프트웨어 인터페이스를 정의하는 XML 용어 사전입니다. 프로그래밍 레벨에서 자동 통합에 필요한 모든 웹 서비스 기술 세부사항을 구성하며 IBM WebSphere 협업을 웹 서비스로서 공개하는 데 사용됩니다. WSDL 대 웹 서비스는 IDL 대 CORBA 오브젝트의 관계와 동일합니다.

WSDL에 대한 자세한 정보는 다음 웹 사이트를 참조하십시오.

<http://www.w3.org/TR/wsdl>

Connector for Web Services 구성요소

그림 1은 프로토콜 핸들러와 리스너 프레임워크 및 SOAP Data Handler를 포함한 Connector for Web Services에 대해 설명합니다.

주: 웹 서비스 어댑터에는 XML Data Handler를 제한적으로 사용할 수 있는 라이선스가 포함되어 있습니다. 그러나 어댑터에서는 XML Data Handler를 작동할 필요가 없습니다.

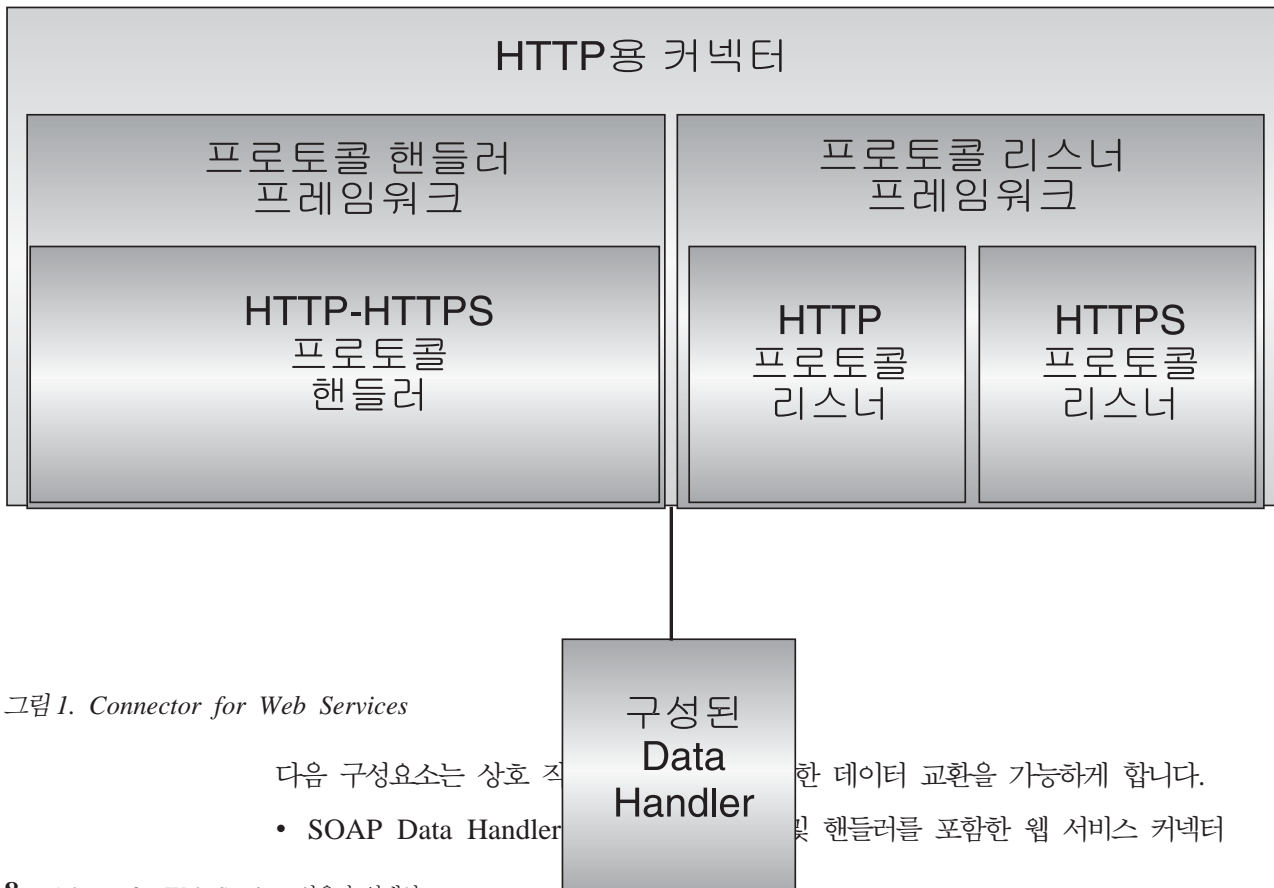


그림 1. Connector for Web Services

다음 구성요소는 상호 작용하는 데이터 교환을 가능하게 합니다.

- SOAP Data Handler 및 핸들러를 포함한 웹 서비스 커넥터

- 웹 서비스 사용 가능 협업
- Business Object 및 SOAP 메시지
- WebSphere Business Integration InterChange Server

웹 서비스 커넥터

요청 처리 중에 웹 서비스 커넥터는 Business Object를 SOAP 요청 메시지로 변환하여 목적지 웹 서비스에 전달하여 협업 서비스 호출에 응답합니다. 선택적으로(동기 요청 처리의 경우) 커넥터는 SOAP 응답 메시지를 SOAP 응답 Business Object로 변환하고 이들을 협업으로 리턴합니다.

이벤트 처리 중에, 커넥터는 클라이언트 웹 서비스의 SOAP 요청 메시지를 SOAP 요청 Business Object로 변환하고 처리를 위해 협업(웹 서비스로 표시된)으로 전달하여 처리합니다. 커넥터는 선택적으로 협업에서 SOAP 응답 Business Object를 수신하는데, 이것은 SOAP 응답 메시지로 변환된 후 클라이언트 웹 서비스로 리턴됩니다.

자세한 정보는 63 페이지의 제 4 장 『웹 서비스 커넥터』를 참조하십시오.

주: 이 문서에서, 별도로 언급되지 않는 한 커넥터의 모든 언급은 웹 서비스 커넥터에 대한 참조입니다.

프로토콜 리스너 및 핸들러

커넥터는 다음 프로토콜 리스너와 핸들러를 포함합니다.

- SOAP/HTTP 프로토콜 리스너
- SOAP/HTTPS 프로토콜 리스너
- SOAP/JMS 프로토콜 리스너
- SOAP/HTTP-HTTPS 프로토콜 핸들러
- SOAP/JMS 프로토콜 핸들러

프로토콜 리스너는 SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS 형식으로 내부 또는 외부 웹 서비스 클라이언트로부터 이벤트를 발견합니다. 프로토콜 리스너는 웹 서비스로서 표시된 협업에서 처리해야 하는 이벤트를 커넥터에 알립니다. 그러면 프로토콜 리스너는 Business Object 레벨 및 속성 레벨 ASI, 커넥터 등록 정보 및 프로토콜 구성 오브젝트에 임베드된 변환 규칙을 읽어서 협업, Data Handler, 처리 모드(동기/비동기) 및 웹 서비스 트랜잭션의 변환 특정 형태를 판별합니다. 프로토콜 리스너 처리의 상세한 설명은 68 페이지의 『프로토콜 리스너』를 참조하십시오.

프로토콜 핸들러는 협업을 위해 SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS 형식으로 웹 서비스를 호출합니다. 프로토콜 핸들러는 TLO ASI 및 프로토콜 구성 오브젝트에 임베드된 변환 규칙을 읽어서 요청을 처리하는 방법(동기적으로 또는 비동기적으로), SOAP 메시지를 SOAP Business Object로 변환하거나 그 반대로 변환하기 위해 사용할 데이터 핸들러를 판별하며 웹 서비스의 대상 주소를 판별합니다(SOAP 요청

Business Object 프로토콜 구성 MO의 목적지 속성으로부터). 동기 트랜잭션의 경우, 프로토콜 핸들러는 SOAP 응답 메시지를 SOAP 응답 Business Object로 변환하고 협업으로 다시 전달하여 처리합니다.

프로토콜 핸들러에 대한 자세한 정보는 81 페이지의 『프로토콜 핸들러』를 참조하십시오.

SOAP Data Handler

SOAP Data Handler는 SOAP Business Object를 SOAP 메시지로 및 그 반대로 변환합니다. SOAP Data Handler에 대한 자세한 정보는 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

자세한 내용은 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

웹 서비스 구성 도구

웹 서비스를 호출하거나 웹 서비스로서 표시되는 협업으로 웹 서비스 솔루션을 전개할 수 있습니다.

요청 처리를 위해 협업을 사용 가능하게 할 때, WSDL ODA(Object Discovery Agent)를 사용하여 웹 서비스 TLO를 생성합니다. 요청 처리 및 WSDL ODA에 대한 자세한 정보는 157 페이지의 제 6 장 『요청 처리용 협업 사용 가능』을 참조하십시오.

협업을 웹 서비스로 표시할 때, WSDL 구성 마법사를 사용하는데, 이것은 나중에 예를 들어 UDDI 레지스트리를 통해 표시하는 협업에 대한 WSDL 문서를 생성하는 데 도움이 됩니다. 커넥터는 이 정보를 공개하는 데 대한 도구를 제공하지 않습니다. 협업을 웹 서비스로 표시에 대한 정보는 159 페이지의 제 7 장 『협업을 웹 서비스로 표시』를 참조하십시오.

커넥터 전개

웹 서비스 커넥터를 전개하는 두 가지 방법이 있습니다.

- 해당 비즈니스 프로세스가 SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS 웹 서비스 형식으로 통신하는 엔터프라이즈 내의 인트라넷 기본 솔루션으로서 방화벽 뒤(그림 2 참조)

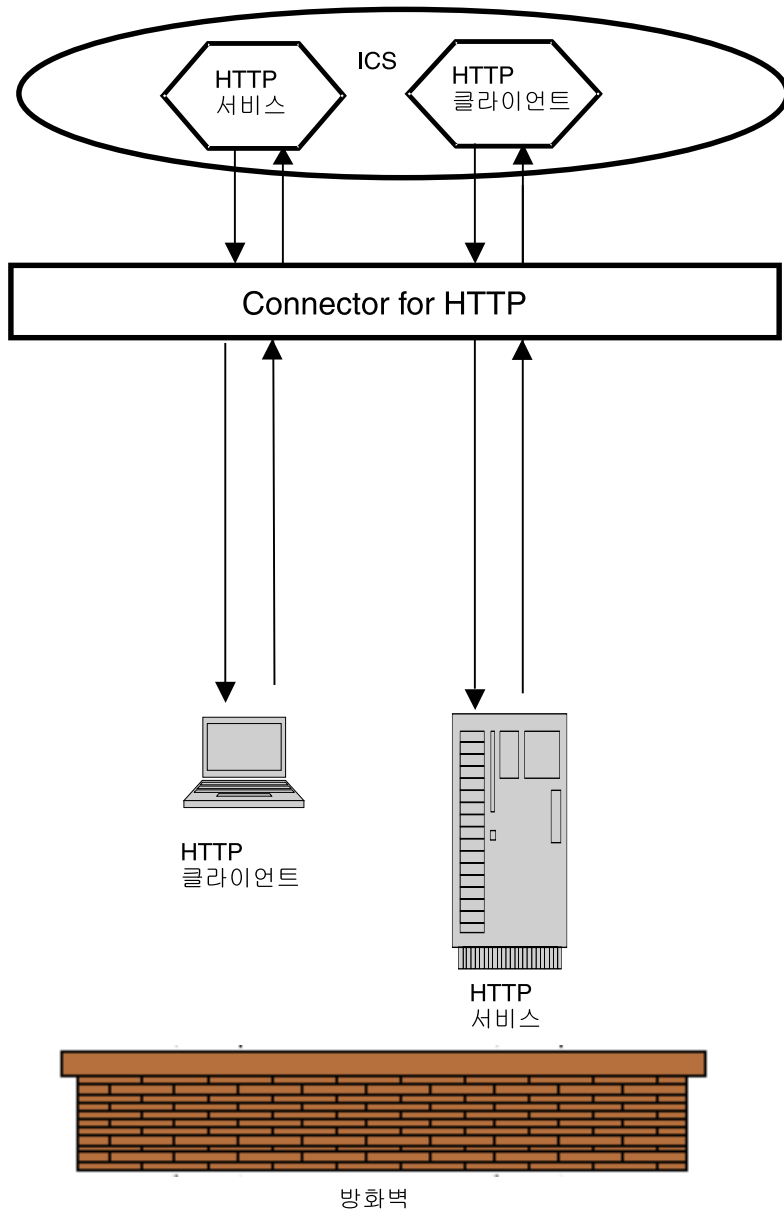


그림2. 인트라넷 솔루션으로서 웹 서비스 어댑터

- 엔터프라이즈에 외부인 웹 서비스를 처리, 필터링 및 기타 웹 서비스와의 통신을 관리하기 위해 프론트엔드 또는 게이트웨이 서버가 있는 방화벽 뒤

주: 웹 서비스 커넥터는 외부 웹 서비스(부터의) 수신 또는 전송 메시지 관리를 위한 게이트웨이 또는 프론트엔드를 포함하지 않습니다. 사용자의 게이트웨이를 구성하고 전개해야 합니다. 커넥터는 **DMZ**에 또는 방화벽의 **밖**이 아니라 엔터프라이즈 **내**에만 전개되어야 합니다.

Connector for Web Services의 구조

이 섹션은 상위 레벨의 구성요소 구조에 대해할 수 있도록 두 가지 데이터 플로우에 대해합니다. 그림 3은 이 두 가지 시나리오에 대해합니다. 이들 두 가지 시나리오는 아래 그림에서 설명됩니다.

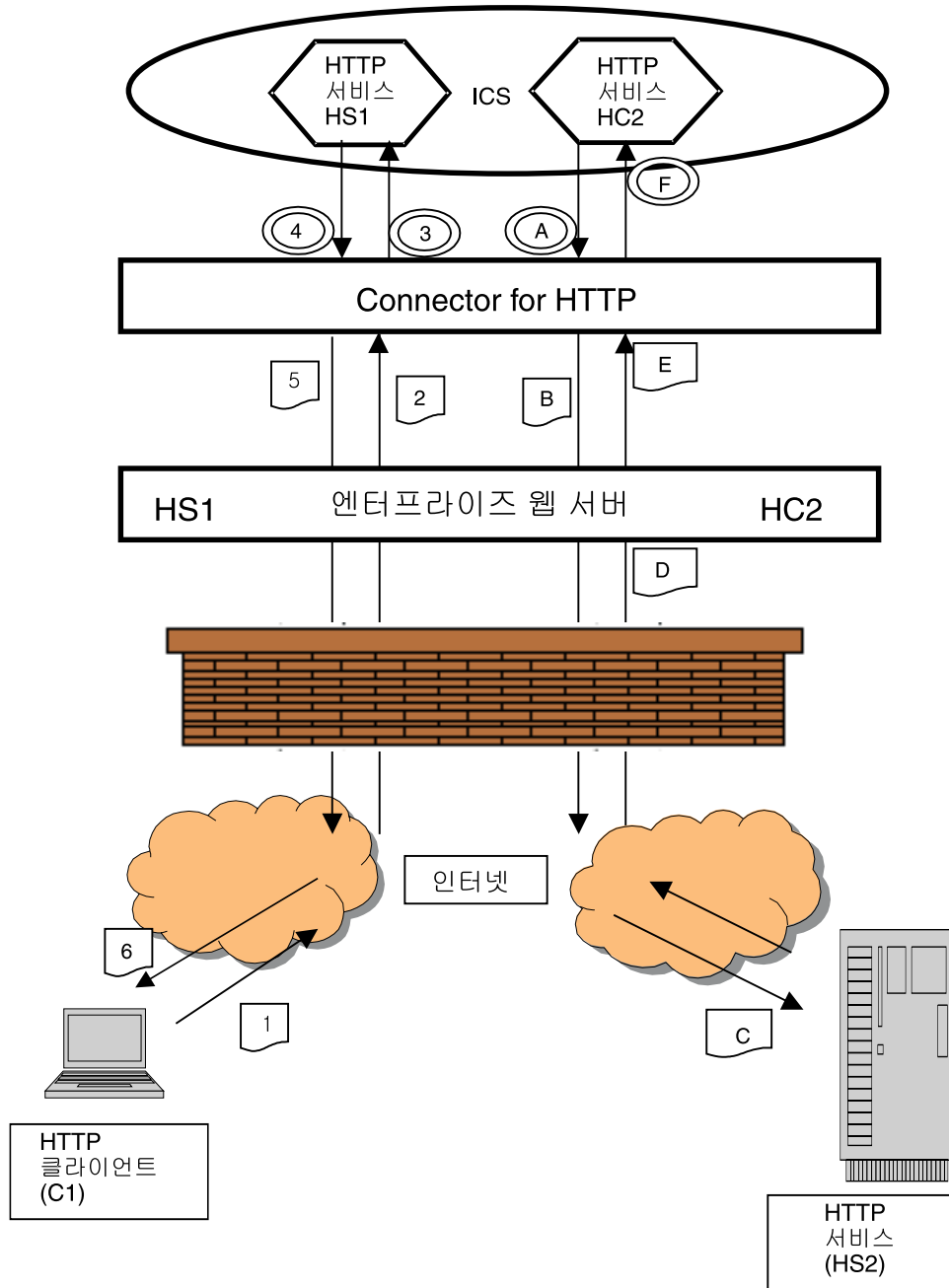


그림 3. 웹 서비스 메시지의 플로우

요청 처리는 협업이 웹 서비스를 호출하기 위해 커넥터에 대한 서비스 호출을 요청할 때 발생하는 이벤트의 순서에 대해합니다. 이 시나리오에서는 협업이 클라이언트 역할을 수행하여 서버에 요청을 전송합니다.

- A 협업이 커넥터에 서비스 호출 요청을 전송하고, 커넥터가 SOAP Data Handler를 호출하여 Business Object를 SOAP 요청 메시지로 변환합니다.
- B 커넥터가 SOAP 메시지를 전송하여 웹 서비스 WS2를 호출합니다. 목적지가 외부 웹 서비스인 경우, 커넥터는 SOAP 메시지를 게이트웨이로 전송합니다. 게이트웨이가 SOAP 메시지를 목적지 웹 서비스에 해당하는 엔드포인트로 전송합니다. 그러면 웹 서비스가 호출됩니다.
- C 호출된 웹 서비스가 SOAP 요청 메시지를 수신하고 요청된 처리를 수행합니다.
- D 호출된 웹 서비스가 SOAP 응답(또는 결함) 메시지를 전송합니다. 웹 서비스가 엔터프라이즈의 외부에 있는 경우, 게이트웨이가 SOAP 응답 메시지를 수신하고 라우트합니다.
- E SOAP 응답(또는 결함) 메시지가 다시 커넥터로 라우트되고, 커넥터는 Data Handler를 호출하여 해당 메시지를 응답 또는 결함 Business Object로 변환합니다.
- F 커넥터가 SOAP 응답 또는 결함 Business Object를 협업으로 리턴합니다.

이벤트 처리는 협업이 웹 서비스로 호출될 때 발생하는 이벤트의 순서에 대해합니다. 이 시나리오에서 웹 서비스로 표시되는 협업은 서버 역할을 수행하여 클라이언트(외부 또는 내부)의 요청을 승인하고 필요에 따라 응답합니다.

- 1 클라이언트 웹 서비스(WS1)가 협업에 대해 생성된 WSDL 문서에 지정된 목적지로 SOAP 요청 메시지를 전송합니다.
- 2 클라이언트 웹 서비스가 외부인 경우, 게이트웨이가 메시지를 수신하고 커넥터로 라우트합니다.
- 3 커넥터가 SOAP 메시지를 SOAP Data Handler로 전송하여 SOAP 메시지를 Business Object로 변환합니다. 커넥터는 웹 서비스로서 표시된 협업을 호출합니다.
- 4 협업이 SOAP 응답(또는 결함) Business Object를 리턴합니다.
- 5 커넥터가 SOAP Data Handler를 호출하여 SOAP 응답(또는 결함) Business Object를 SOAP 응답 메시지로 변환합니다. 커넥터는 응답을 게이트웨이로 리턴합니다.
- 6 클라이언트 웹 서비스가 외부인 경우, 게이트웨이가 SOAP 응답 메시지를 클라이언트 웹 서비스(WS1)로 라우트합니다.

설치, 구성 및 설계 체크리스트

이 섹션은 웹 서비스 솔루션을 설치, 구성 및 설계할 때 수행해야 하는 작업을 요약합니다. 각 섹션에서는 작업을 간략하게 설명하고 이 책에서 작업 수행 방법을 설명하거나 배경 정보를 제공하는 관련 섹션의 링크를 제공하며, 관련 문서를 상호 참조합니다.

어댑터 설치

설치 사항 및 위치에 대한 설명은 17 페이지의 제 2 장 『설치 및 시작』을 참조하십시오.

커넥터 등록 정보 구성

커넥터에는 두 가지 유형의 구성 등록 정보(표준 구성 등록 정보 및 커넥터 특정 구성 등록 정보)가 있습니다. 이러한 등록 정보의 일부에는 변경해서는 안 될 기본값이 있습니다. 커넥터를 실행하기 전에 이 등록 정보의 일부 값을 설정해야 할 수 있습니다. 자세한 정보는 63 페이지의 제 4 장 『웹 서비스 커넥터』를 참조하십시오.

프로토콜 핸들러 및 리스너 구성

이 구성요소의 작동을 관리하는 커넥터 구성 등록 정보에 값을 지정할 때 프로토콜 핸들러 및 리스너를 구성합니다. 자세한 정보는 63 페이지의 제 4 장 『웹 서비스 커넥터』를 참조하십시오.

웹 서비스용 협업 사용 가능

웹 서비스용 협업을 사용할 수 있을 경우, 호출하거나 웹 서비스로 표시할 수 있는 협업을 작성합니다. 또한 Business Object를 작성하거나 조정합니다. 관련된 작업의 개요는 10 페이지의 『웹 서비스 구성 도구』를 참조하십시오.

협업을 웹 서비스로 표시

단계별 설명에 대해서는 159 페이지의 제 7 장 『협업을 웹 서비스로 표시』를 참조하십시오.

웹 서비스 호출을 위한 협업 사용 가능

단계별 설명에 대해서는 157 페이지의 제 6 장 『요청 처리용 협업 사용 가능』을 참조하십시오.

SOAP Data Handler 구성

제품 파일 설치 후 시작하기 전에 Data Handler Meta Object에서 정보를 구성합니다. 사용자 정의 이름 핸들러를 추가하지 않는 경우, 기본 SOAP Data Handler 구성을 사용하여 시간을 절약할 수 있습니다. 그러나 각 Data Handler 변환 시 특정 Meta Object 정보를 구성해야 합니다. 이 정보는 SOAP 구성 MO에 포함됩니다. Business Object를 작성할 때 SOAP 구성 MO를 지정합니다. 웹 서비스(요청 처리)를 호출하는 협업

개발 시 이러한 작업 중 대부분은 자동화됩니다. WSDL ODA를 사용하여 SOAP 메시지의 Business Object를 생성할 때 SOAP 구성 MO는 자동으로 생성됩니다.

Data Handler 구성에 대한 자세한 정보는 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

제한사항

- WSDL ODA가 자동으로 Business Object를 생성합니다. 결과가 사용자 요구사항을 충족시키지 않는 경우, Business Object Designer를 사용하여 Business Object를 수동으로 작성해야 합니다.
속성 양식, 사용 및 유형과 요소를 사용한 파트 정의의 다양한 조합에 대한 WSDL ODA 지원을 참조하십시오.
- 양식(rpc, 문서), 사용(리터럴, 인코드) 및 파트 정의 방법에 대한 XML 제한사항에 대해서는 119 페이지의 제 5 장 『SOAP Data Handler』 및 157 페이지의 제 6 장 『요청 처리용 협업 사용 가능』을 참조하십시오.
- 커넥터는 SOAP/HTTP 및 SOAP/JMS 바인딩만을 지원합니다.
- 커넥터의 SOAP/JMS 프로토콜 리스너는 대기열 목적지만을 지원합니다. 주제는 지원되지 않습니다. JMS 텍스트 및 바이트 메시지가 지원됩니다.
- HTTP POST 요청 및 응답은 지원됩니다. 다른 HTTP 메소드는 지원되지 않습니다. HTTP 1.1 지속적 연결은 지원되지 않습니다.

제 2 장 설치 및 시작

- 『설치 작업의 개요』
- 『커넥터 및 관련 파일 설치』
- 19 페이지의 『구성 작업의 개요』
- 20 페이지의 『어댑터의 다중 인스턴스 실행』
- 22 페이지의 『커넥터 시작 및 중지』

이 장은 Connector for Web Services 구현을 위한 구성요소 설치 방법에 대해 설명합니다. 일반적으로 ICS 시스템의 설치에 대한 정보는 사용자 플랫폼에 적합한 시스템 설치 안내서를 참조하십시오.

설치 작업의 개요

브로커 호환성, 어댑터 프레임워크, 소프트웨어 전제조건, 종속성, 표준 및 API에 대한 정보는 1 페이지의 『Adapter for Web Services 환경』의 내용을 참조하십시오.

Connector for Web Service를 설치하려면 다음 작업을 수행해야 합니다.

ICS 설치

시스템 설치 및 ICS 시작을 포함하는 이 작업은 시스템 설치 안내서에 설명되어 있습니다. ICS 버전 4.2를 설치해야 합니다.

저장소로 파일을 로드하려면 *Implementation Guide for WebSphere InterChange Server*를 참조하십시오.

커넥터 및 관련 파일 설치

이 작업에는 소프트웨어 패키지의 커넥터(및 관련 구성요소)용 파일을 사용자 시스템에 설치하는 것이 포함됩니다. 『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 테이블에서는 설치된 파일 구조를 표시합니다.

Windows 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

유틸리티가 커넥터를 설치하고 시작 메뉴에 커넥터 에이전트의 바로 가기를 추가합니다.

표 1은 커넥터가 사용하는 Windows 파일 구조에 대해하며, 설치 프로그램을 통한 커넥터 설치를 선택할 때 자동으로 설치되는 파일을 보여줍니다.

표 1. 어댑터용으로 설치된 Windows 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
\lib\WBIA.jar	WebSphere Business Integration Adapter jar 파일
\bin\CWConnEnv.bat	일반 커넥터 시작 파일
\bin\ODAEnv.bat	일반 ODA 시작 파일
connectors\WebServices\CWebServices.jar	웹 서비스 커넥터
connectors\WebServices\start_WebServices.bat	커넥터용 시작 파일
DataHandlers\CwSOAPDataHandler.jar	Data Handler
DataHandlers\CwSOAPNameHandler.jar	SOAP 이름 핸들러
repository\DataHandlers\MO_DataHandler_SOAP.txt	SOAP Data Handler 관련 파일
bin\Data\App\WebServicesConnectorTemplate	웹 서비스 커넥터 템플릿
ODA\WSDL\WSDLODA.jar	WSDL ODA
ODA\WSDL\start_WSDLODA.bat	WSDL ODA 시작 파일
connectors\WebServices\dependencies\soap.jar	SOAP Connector, SOAP Data Handler, WSDL 구성 마 법사 및 WSDL ODA에 필수 Apache SOAP API
connectors\WebServices\dependencies\LICENSE	Apache 라이선스 파일
connectors\WebServices\dependencies\mail.jar	JavaMail API
connectors\WebServices\dependencies\activation.jar	Java Activation Framework
connectors\WebServices\dependencies\ibmjse.jar	IBM의 JSSE(Java Secure Socket Extension) API
connectors\WebServices\dependencies\jms.jar	Java 메시징 서비스
connectors\WebServices\dependencies\uddi4j-wsd1.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\uddi4jv2.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\IPL10.txt	WSDL ODA에 필수 라이선스 파일
connectors\WebServices\dependencies\wsd14j.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\CPL10.txt	WSDL ODA에 필수 라이선스 파일
connectors\WebServices\dependencies\qname.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\j2ee.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\wsbw2.1.1\common.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\wsbw2.1.1\core.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\wsbw2.1.1\xercesImpl.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\wsbw2.1.1\xmlParserAPIs.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\wsbw2.1.1\xsd.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\wsbw2.1.1\xsd.resources.jar	WSDL ODA에 필수
connectors\WebServices\dependencies\IBMReadme.txt	라이선스
connectors\WebServices\samples\WebSphereICS\WebServicesSample.jar	샘플용 저장소 파일
connectors\WebServices\samples\WebSphereICS\CLIENT_SYNCH_TLO_OrderStatus.bo	테스트 커넥터용 샘플 (동기) Business Object
connectors\WebServices\samples\WebSphereICS\CLIENT_ASYNC_TLO_Order.bo	테스트 커넥터용 샘플 (비동기) Business Object
connectors\messages\WebServicesConnector.txt	커넥터 메시지 파일
ODA\messages\WSDLODAAgent.txt	WSDL ODA용 메시지 파일

주: 모든 제품 경로 이름은 제품이 시스템에 설치된 디렉토리과 관련됩니다.

UNIX 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

표 2는 커넥터가 사용하는 UNIX 파일 구조에 대해하며, 설치 프로그램을 통해 커넥터를 설치할 것을 선택할 때 자동으로 설치되는 파일을 보여줍니다.

표 2. 어댑터용으로 설치된 UNIX 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
/lib/WBIA.jar	WebSphere Business Integration Adapter jar 파일
/bin/CWConnEnv.bat	일반 커넥터 시작 파일
/bin/ODAEEnv.bat	일반 ODA 시작 파일
connectors/WebServices/CWWebServices.jar	웹 서비스 커넥터
connectors/WebServices/start_WebServices.sh	커넥터용 시작 파일
DataHandlers/CwSOAPDataHandler.jar	Data Handler
DataHandlers/CwSOAPNameHandler.jar	SOAP 이름 핸들러
repository/DataHandlers/MO_DataHandler_SOAP.txt	SOAP Data Handler 관련 파일
bin/Data/App/WebServicesConnectorTemplate	웹 서비스 커넥터 템플릿
ODA/WSDL/WSDLODA.jar	WSDL ODA
ODA/WSDL/start_WSDLODA.sh	WSDL ODA 시작 파일
connectors/WebServices/dependencies/soap.jar	SOAP Connector, SOAP Data Handler, WSDL 구성 마법사 및 WSDL ODA에 필수 Apache SOAP API
connectors/WebServices/dependencies/LICENSE	Apache 라이선스 파일
connectors/WebServices/dependencies/mail.jar	JavaMail API
connectors/WebServices/dependencies/activation.jar	Java Activation Framework
connectors/WebServices/dependencies/ibmjss.jar	IBM의 JSSE(Java Secure Socket Extension) API
connectors/WebServices/dependencies/jms.jar	Java 메시징 서비스
connectors/WebServices/dependencies/uddi4j-wsdl.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/uddi4jv2.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/IPL10.txt	WSDL ODA에 필수 라이선스 파일
connectors/WebServices/dependencies/wsd14j.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/CPL10.txt	WSDL ODA에 필수 라이선스 파일
connectors/WebServices/dependencies/qname.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/j2ee.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/wswb2.1.1/common.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/wswb2.1.1/core.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/wswb2.1.1/xercesImpl.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/wswb2.1.1/xmlParserAPIs.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/wswb2.1.1/xsd.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/wswb2.1.1/xsd.resources.jar	WSDL ODA에 필수
connectors/WebServices/dependencies/IBMReadme.txt	라이선스
connectors/WebServices/samples/WebSphereICS/WebServicesSample.jar	샘플용 저장소 파일
connectors/WebServices/samples/WebSphereICS/CLIENT_SYNC_TLO_OrderStatus.bo	테스트 커넥터용 샘플 (동기) Business Object
connectors/WebServices/samples/WebSphereICS/CLIENT_ASYNC_TLO_Order.bo	테스트 커넥터용 샘플 (비동기) Business Object
connectors/messages/WebServicesConnector.txt	커넥터 메시지 파일
ODA/messages/WSDLODAAgent.txt	WSDL ODA용 메시지 파일

주: 모든 제품 경로 이름은 제품이 시스템에 설치된 디렉토리와 관련됩니다.

구성 작업의 개요

설치 후 및 시작 전에 다음과 같이 구성요소를 구성해야 합니다.

커넥터 구성

이 작업은 커넥터 설정 및 구성을 포함합니다. 94 페이지의 『커넥터 구성』을 참조하십시오.

Business Object 구성

Business Object 구성 단계는 제품 모음을 구현하기 위해 사용자가 선택하는 방법에 따라 다릅니다.

- 요청 처리 다음에 대응하는 Business Object를 작성해야 합니다.

- 각 웹 서비스에 전송할 요청 메시지
- 결함을 포함한 가능한 각 응답

자세한 정보는 25 페이지의 제 3 장 『Business Object 요구사항』을 검토한 후 157 페이지의 제 6 장 『요청 처리용 협업 사용 가능』을 참조하십시오.

- 이벤트 처리 TLO 또는 비TLO Business Object를 사용할 수 있습니다.

자세한 정보는 25 페이지의 제 3 장 『Business Object 요구사항』을 검토한 후 159 페이지의 제 7 장 『협업을 웹 서비스로 표시』를 참조하십시오.

Data Handler 구성

설치 후에 Data Handler Meta Object를 구성해야 합니다. 또한, 각 SOAP Business Object에 대해 SOAP 구성 MO도 구성해야 합니다. Data Handler를 구성하려면 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

협업 구성

- 요청 처리 처리의 일부로 웹 서비스를 호출하는 협업의 경우, WSDL ODA를 사용하여 Business Object를 생성한 후 협업 오브젝트 포트를 커넥터에 바인드합니다. 단계별 절차를 포함한 자세한 정보는 157 페이지의 제 6 장 『요청 처리용 협업 사용 가능』을 참조하십시오.

- 이벤트 처리 웹 서비스로 표시되는 협업의 경우, WSDL 구성 마법사를 사용하여 WSDL 문서를 생성하고 잠재적 클라이언트가 해당 문서를 사용할 수 있도록 한 후 클라이언트가 협업을 호출할 수 있도록 협업 오브젝트의 포트를 구성해야 합니다. 단계별 절차를 포함한 자세한 정보는 159 페이지의 제 7 장 『협업을 웹 서비스로 표시』를 참조하십시오.

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 21 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 및 중지

중요: 이 장의 앞에서 언급한 것처럼, 적절한 조작을 보장하기 위해 설치한 후 커넥터를 시작하기 전에 커넥터, Business Object, SOAP Data Handler Meta Object 및 협업을 구성해야 합니다. 이들 작업의 요약은 19 페이지의 『구성 작업의 개요』를 참조하십시오. 그리고 커넥터 폴링은 사용 불가능하지 않아야 합니다(커넥터 폴링은 기본적으로 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	<code>connector_manager_connName</code>
Windows	<code>start_connName.bat</code>

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

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

- Windows 시스템의 경우:
`start_connName connName brokerName [-cconfigFile]`

- UNIX 기반 시스템의 경우:
`connector_manager_connName -start`

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

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

제 3 장 Business Object 요구사항

- 『Business Object 메타 데이터』
- 26 페이지의 『커넥터 Business Object 구조』
- 26 페이지의 『동기 이벤트 처리 TLO』
- 39 페이지의 『비동기 이벤트 처리 TLO』
- 43 페이지의 『이벤트 처리 비TLO』
- 44 페이지의 『동기 요청 처리 TLO』
- 44 페이지의 『동기 요청 처리 TLO』
- 55 페이지의 『비동기 요청 처리 TLO』
- 61 페이지의 『Business Object 개발』

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

Business Object 메타 데이터

Connector for Web Service는 메타 데이터 구동 커넥터입니다. Business Object에서 메타 데이터는 Business Object 정의에 저장되고 커넥터가 응용프로그램과 상호작용하는 것을 돕는 응용프로그램에 대한 데이터입니다. 메타 데이터 구동 커넥터는 커넥터에 하드 코드되는 지시사항이 아니라 Business Object 정의에 코드화되는 메타 데이터를 바탕으로 커넥터가 지원하는 각 Business Object를 처리합니다.

Business Object 메타 데이터에는 Business Object의 구조, 해당 속성 등록 정보의 설정 및 해당 응용프로그램 특정 정보의 내용이 있습니다. 커넥터가 메타 데이터 구동이 기 때문에 커넥터 코드에 대한 수정이 필요없이 새로운 또는 수정된 Business Object를 처리할 수 있습니다. 그러나 커넥터의 구성된 Dats Handler가 해당 Business Object의 구조, 오브젝트 카디널리티, 응용프로그램 특정 텍스트의 형식 및 Business Object의 데이터베이스 표시에 대한 가정을 작성합니다. 따라서 웹 서비스용 Business Object를 작성 또는 수정할 때 사용자 수정은 커넥터가 따르도록 설계된 규칙을 준수해야 하며, 그렇지 않으면 커넥터는 새로운 또는 수정된 Business Object를 올바르게 처리할 수 없게 됩니다.

메타 데이터, Meta Object 및 구성과 Business Object 및 SOAP 메시지와 상호작용에 대한 자세한 정보는 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

커넥터 Business Object 구조

커넥터는 다음 두 유형의 Business Object를 처리합니다.

- **TLO** 웹 서비스 최상위 레벨 Business Object(TLO)는 요청 Business Object를 포함하며, 선택적으로 요청 및 결함 Business Object를 포함합니다. 이들 하위 오브젝트에는 내용 데이터 뿐만 아니라 SOAP 구성 MO 및 선택적으로 프로토콜 구성 MO를 포함합니다. 아래의 섹션에는 응용프로그램 특정 정보, 속성 및 요청 대 이벤트 처리와 관련된 요구사항 뿐만 아니라 TLO, 요청, 응답 및 결함 오브젝트도 설명되어 있습니다.

주: TLO는 요청 처리와 이벤트 처리에 사용됩니다.

- **비TLO** 이들은 TLO가 아니지만 WSDL 생성에서 WSDL 구성 마법사에 의해 사용된 일반 Business Object(GB) 및 응용프로그램 특정 Business Object(ASBO)입니다. 커넥터는 이벤트 처리 중에 비TLO를 처리할 수 있습니다. 이들 오브젝트는 43 페이지의 『이벤트 처리 비TLO』 아래에 설명되어 있습니다. 자세한 정보는 163 페이지의 『WSDL 구성 마법사』를 참조하십시오.

주: 비TLO는 이벤트 처리에만 사용됩니다.

주: 요청, 응답 및 결함 Business Object에 포함되는 SOAP 머리글 컨테이너와 머리글 Business Object는 이 장에서 다루지 않습니다. SOAP 머리글 컨테이너와 머리글 Business Object에 대한 정보는 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

동기 이벤트 처리 TLO

이벤트 처리의 경우, 커넥터는 두 종류(동기 및 비동기)의 TLO를 허용합니다. 이 섹션은 동기 이벤트 처리 TLO에 대해 설명합니다.

27 페이지의 그림 4에서는 동기 이벤트 처리용 Business Object 계층 구조를 보여줍니다. 요청 및 응답 오브젝트는 필수이지만, 결함 오브젝트는 선택적입니다.

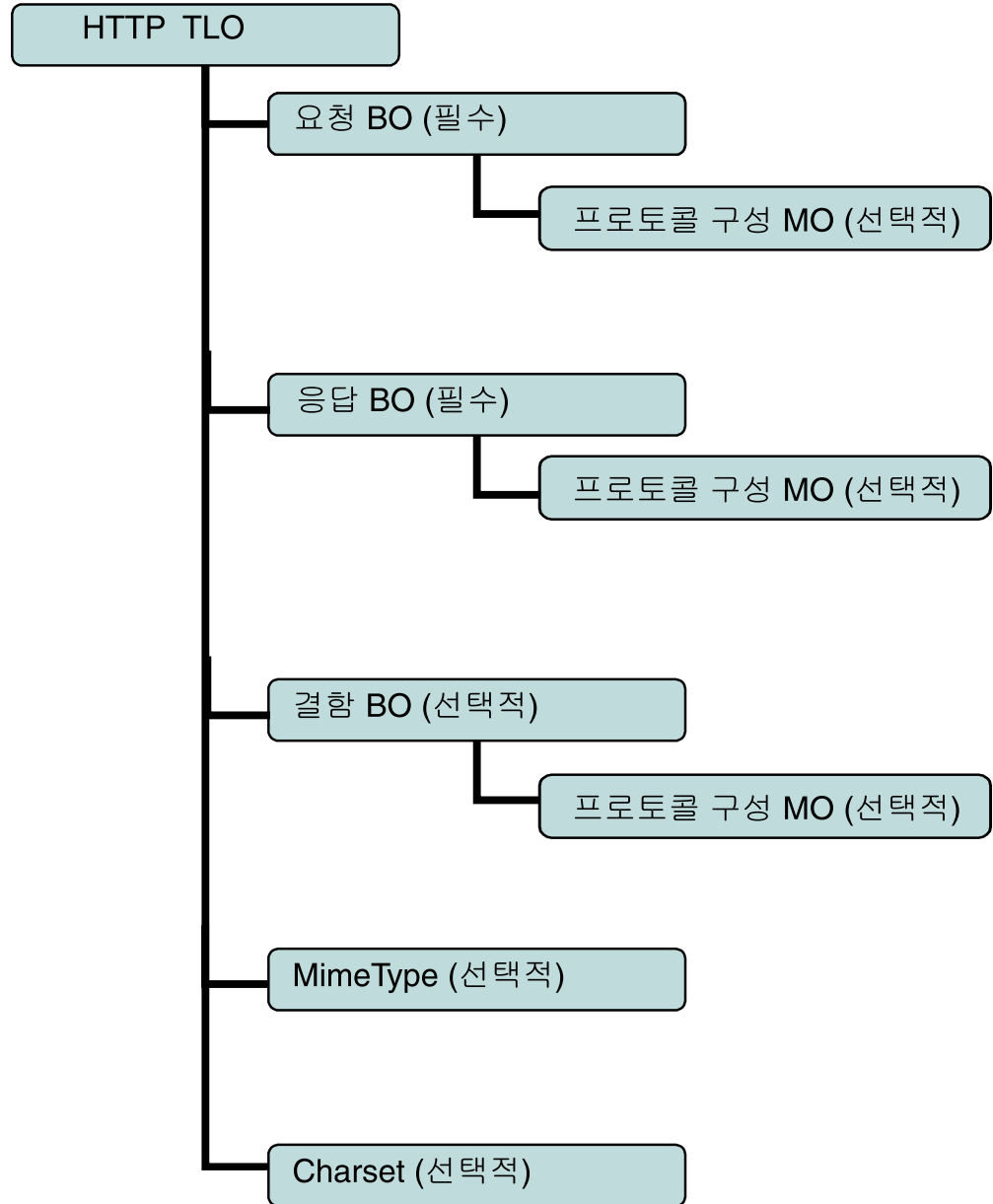


그림 4. 동기 이벤트 처리용 Business Object 계층 구조

TLO는 속성 레벨 ASI가 있는 속성뿐만 아니라 오브젝트 레벨 ASI도 포함합니다. 두 종류의 ASI 모두가 아래에 설명되어 있습니다.

동기 이벤트 처리 TLO에 대한 오브젝트 레벨 ASI

오브젝트 레벨 ASI는 TLO의 특성 및 그것이 포함하는 오브젝트에 대한 기본 정보를 제공합니다. 그림 5에서는 동기 이벤트를 처리를 위한 샘플 TLO인 SERVICE_SYNCH_OrderStatus에 대한 오브젝트 레벨 ASI를 보여줍니다.

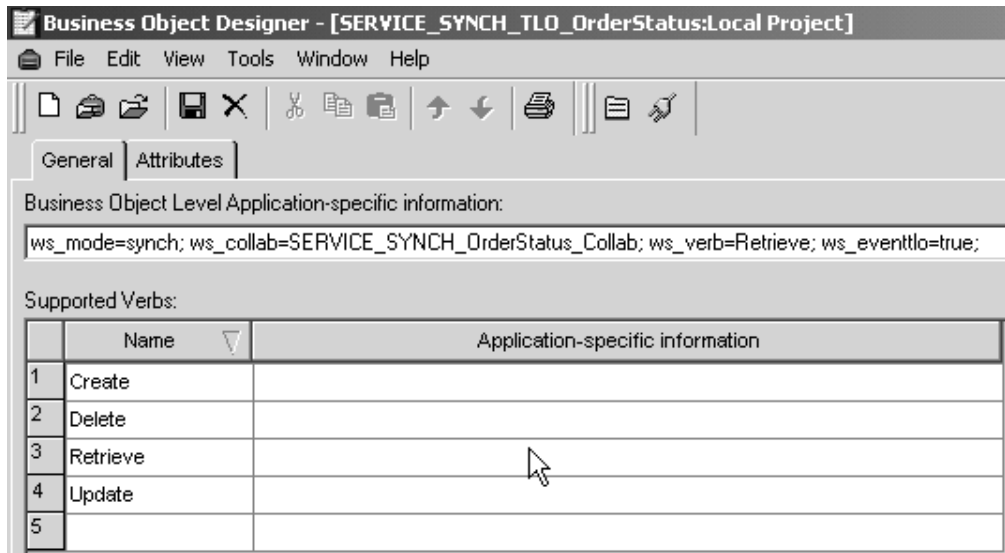


그림 5. 동기 이벤트 처리용 최상위 레벨 Business Object

아래의 표 4에서 동기 이벤트 처리 TLO에 대한 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 설정되는 경우, 커넥터는 이 오브젝트를 이벤트 처리만을 위한 TLO로 취급합니다. WSDL 구성 마법사가 이 ASI를 사용하여 Business Object가 TLO인지 여부를 판별합니다. 이에 대한 자세한 정보는 163 페이지의 『WSDL 구성 마법사』를 참조하십시오.
ws_collab=collabname	이 ASI는 커넥터에 호출할 협업을 알립니다. 값은 협업의 이름입니다. (이 ASI는 협업의 TLO를 판별하기 위한 WSDL 생성 중에도 사용됩니다. 이에 대한 자세한 정보는 163 페이지의 『WSDL 구성 마법사』를 참조하십시오.) 그림 5에 표시된 샘플에서, 협업 이름은 SERVICE_SYNCH_OrderStatus_Collab입니다.
ws_verb=verb	TLO를 협업에 전달하기 전에 커넥터가 이 ASI를 사용하여 TLO에 Verb를 설정합니다. 그림 5에 표시된 샘플에서, Verb는 Retrieve입니다.
ws_mode=synch	이벤트 공고 중에 커넥터가 이 ASI 등록 정보를 사용하여 협업을 동기식으로(synch) 또는 비동기식으로(asynch) 호출할지 여부를 판별합니다. 동기 처리의 경우, 이 ASI가 synch로 설정되어야 합니다. 기본값은 asynch입니다.

동기 이벤트 처리 TLO에 대한 속성 레벨 ASI

각 동기 이벤트 처리 TLO에는 속성과 속성 레벨 ASI가 있습니다. 그림 6에서는 샘플 TLO인 SERVICE_SYNCH_OrderStatus의 속성을 보여줍니다. 또한 응용프로그램 특정 정보 열에 있는 속성 레벨 ASI도 보여줍니다.

	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 처리용 TLO 속성

표 5에는 동기 이벤트 처리 TLO의 요청, 응답, 결합, MimeType 및 Charset 속성에 대한 속성 레벨 ASI가 요약되어 있습니다.

표 5. 동기 이벤트 처리 TLO 속성 ASI

TLO 속성	속성 레벨 ASI	설명
MimeType		선택적 속성; 지정할 경우 해당 값은 동기 응답을 호출하기 위한 Data Handler의 MIME 유형으로 사용됩니다. 유형은 String이며 기본값은 xml/soap입니다.
Charset		문자열 유형의 이 선택적 매개변수는 전송되는 Business Object를 메시지로 변환할 때 Data Handler에서 설정되는 charset을 지정합니다. 참고: 이 속성에 지정된 Charset 값은 응답 메시지의 Content-Type 프로토콜 머릿글에 전달되지 않습니다.
Request	ws_botype=request	이 속성은 웹 서비스 요청에 대응합니다. 커넥터는 해당 ASI를 사용하여 TLO 속성이 SOAP 요청 BO 유형인지 판별합니다. 속성 이름이 아닌 이 ASI가 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 속성의 ASI를 사용합니다. 이 속성은 동기 이벤트 처리 TLO에 필수입니다.

표 5. 동기 이벤트 처리 TLO 속성 ASI (계속)

TLO 속성	속성 레벨 ASI	설명
Response	ws_botype=response	이 속성은 웹 서비스가 리턴하는 응답에 대응합니다. 커넥터는 이 ASI를 사용하여 TLO 속성이 SOAP 응답 BO 유형인지 판별합니다. 속성 이름이 아닌 이 ASI가 속성 유형을 판별합니다. 둘 이상의 응답 속성이 있는 경우, 커넥터는 첫 번째 속성의 ASI를 사용합니다. 이 속성은 동기 이벤트 처리 TLO에 필수입니다.
Fault	ws_botype=fault ws_botype=defaultfault	동기 이벤트 처리에 대해 선택적인 이 속성은 협업이 응답을 채울 수 없을 때 리턴하는 결함 메시지에 해당합니다. 커넥터는 속성 이름이 아닌 이 ASI를 사용하여 속성이 SOAP 결함 BO 유형인지를 판별합니다. ws_botype=defaultfault인 경우, WSDL 구성 마법사가 머릿글 처리에 이 결함 Business Object를 사용합니다. 자세한 정보는 133 페이지의 『머릿글 결함 처리』를 참조하십시오.

동기 이벤트 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 동기 이벤트 처리를 위해 필수입니다. 요청 Business Object에는 오브젝트 레벨 ASI가 있습니다. 예를 들어, Business Object Designer에서 SERVICE_SYNCH_OrderStatus_Request를 열고 일반 탭을 누르면 31 페이지의 그림 7에서처럼 오브젝트 레벨 ASI가 표시됩니다.

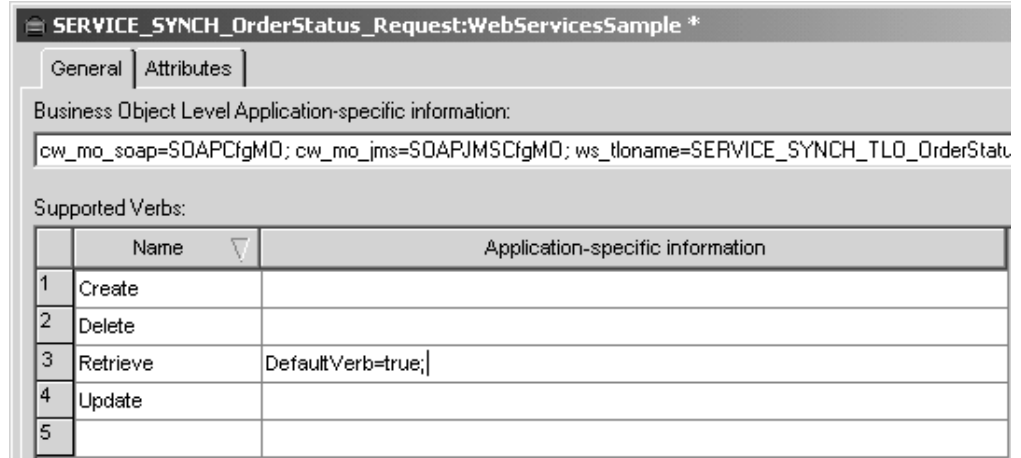


그림 7. 동기 이벤트 처리 요청 오브젝트의 오브젝트 레벨 ASI

동기 이벤트 처리를 위한 요청 Business Object에 대한 오브젝트 레벨 ASI가 표 6에 설명되어 있습니다. 그림 7에서와 같이 요청 Business Object에 대한 기본 Verb를 지정할 수 있습니다. 요청 Business Object의 최상위 레벨에 있는 지원되는 Verb 목록에서 Verb의 ASI 필드에 다음을 지정하여

DefaultVerb=true;

를 지정하여 기본 Verb를 지정할 수 있습니다. DefaultVerb ASI가 지정되지 않고 Data Handler가 Verb가 설정되지 않은 Business Object를 처리하는 경우, Business Object는 Verb 없이 리턴됩니다.

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

오브젝트 레벨 ASI	설명
cw_mo_soap=SOAPCfmgMO	이 ASI 값은 SOAP 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이것은 요청 Business Object에 대한 Data Handler 변환을 정의하는 Meta Object입니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』를 참조하십시오.
cw_mo_jms=SOAPJMSCfmgMO 또는 cw_mo_http=SOAPHTTPCfmgMO	이 ASI 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 첫 번째 ASI는 SOAP/JMS 프로토콜 리스너를 지정하며 두 번째는 SOAP/HTTP 또는 SOAP/HTTPS 프로토콜 리스너를 지정합니다. ASI와 프로토콜 구성 MO는 선택적입니다. 자세한 정보는 33 페이지의 『프로토콜 구성 MO』를 참조하십시오.
ws_tlname=tlname	이 ASI는 이 오브젝트가 속하는 웹 서비스 TLO의 이름을 지정합니다. 이벤트 처리 중에 커넥터가 이 ASI를 사용하여 Data Handler에 의해 전달되는 요청 Business Object가 TLO의 하위인지 여부를 판별합니다. 이 경우, 커넥터는 지정된 TLO를 작성하고, 요청 Business Object를 그의 하위로 설정하고, TLO 오브젝트 레벨 ASI를 사용하여 등록된 협업에 전달합니다.

동기 이벤트 처리를 위한 응답 Business Object

응답 Business Object는 TLO의 하위이며 동기 이벤트 처리를 위해 필수입니다. 동기 이벤트 처리를 위한 응답 Business Object에 대한 오브젝트 레벨 ASI가 표 7에 설명되어 있습니다.

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

오브젝트 레벨 ASI	설명
cw_mo_soap=SOAPCfgMO	이 ASI 값은 SOAP 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 응답 Business Object의 Data Handler 변환을 정의하는 SOAP 구성 MO입니다. 자세한 정보는 『SOAP 구성 MO』를 참조하십시오.

주: 응답 BO에 대한 프로토콜 구성 MO 오브젝트 레벨 ASI를 선택적으로 포함할 수 있습니다.

동기 이벤트 처리를 위한 결합 Business Object

결합 Business Object는 TLO의 하위이며 동기 이벤트 처리의 경우 선택적입니다. 동기 이벤트 처리를 위한 결합 Business Object에 대한 오브젝트 레벨 ASI가 표 8에 설명되어 있습니다.

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

오브젝트 레벨 ASI	설명
cw_mo_soap=SOAPCfgMO	이 ASI 값은 SOAP 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 결합 Business Object의 Data Handler 변환을 정의하는 SOAP 구성 MO입니다. 자세한 정보는 『SOAP 구성 MO』를 참조하십시오.

주: 결합 BO에 대한 프로토콜 구성 MO 오브젝트 레벨 ASI를 선택적으로 포함할 수 있습니다.

SOAP 구성 MO

그림 8에서는 Business Object Designer에서 펼쳐진 샘플 SOAP 구성 MO를 보여줍니다.

Business Object Designer - SERVICE_SYNCH_OrderStatus_Request:Local Project

File Edit View Tools Window Help

SERVICE_SYNCH_OrderStatus_Request:Local Project

General Attributes

	Pos	Name	Type	Key	Foreign	Required	Card	Maximum Length	Default
1	1	OrderId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	
2	4	SOAPJMSCfgMO	SERVICE_SYNCH_SOAP_JMS_OrderStatus_Req	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
3	5	ObjectEventId	String						
4	2	OrderHeader	SERVICE_SYNCH_OrderStatus_Request_Header	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
5	3	SOAPCfgMO	SERVICE_SYNCH_OrderStatus_Request_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
5.1	3.6	BodyName	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	getOrderStatus
5.2	3.3	Style	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	rpc
5.3	3.4	TypeInfo	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	true
5.4	3.5	TypeCheck	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	none
5.5	3.2	BOVerb	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	Retrieve
5.6	3.7	ObjectEventId	String						
5.7	3.1	BodyNS	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	http://www.mycompany.com/samples/orderstatus

그림 8. 동기 이벤트 처리용 SOAP 구성 MO 속성

SOAP 구성 MO는 하나의 Data Handler 변환(SOAP 메시지를 Business Object로 또는 Business Object를 SOAP 메시지로 변환)을 위한 형식화 작동을 정의합니다. 요청, 응답 및 결합 속성 각각에는 하나의 SOAP 구성 MO가 있습니다. 해당 속성인 BodyName, BodyNS, Style, Use, TypeInfo, TypeCheck 및 BOVerb의 유형은 항상 String입니다. 이들 속성은 SOAP 메시지 요소에 해당하며 값은 SOAP Data Handler가 메시지와 오브젝트를 읽고 유효성을 검증하는 방법을 결정합니다. SOAP 구성 MO 및 속성에 대한 자세한 정보는 121 페이지의 『SOAP 구성 Meta Object: 모든 SOAP Business Object의 하위 오브젝트』를 참조하십시오. 모든 SOAP 구성 MO에는 요청, 응답 또는 결합 오브젝트에 대해 BodyName 및 BodyNS의 기본값에 해당하는 고유 항목이 있어야 합니다.

프로토콜 구성 MO

그림 9에서는 해당 속성이 인바운드 SOAP 메시지의 머리글에 해당하는 JMS 프로토콜 구성 MO를 보여줍니다.

	Pos	Name	Type	Key	Foreign	Required	Card
1	1	Request	SERVICE_SYNCH_OrderStatus_Request	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
1.1	1.1	OrderId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
1.2	1.5	ObjectEventId	String				
1.3	1.3	SOAPCfgMO	SERVICE_SYNCH_OrderStatus_Request_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
1.4	1.4	SOAPJMSCfgMO	SERVICE_SYNCH_SOAP_JMS_OrderStatus_Request_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
1.4.1	1.4.1	MessageId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.2	1.4.2	Priority	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.3	1.4.3	Expiration	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.4	1.4.4	DeliveryMode	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.5	1.4.5	ReplyTo	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.6	1.4.6	ObjectEventId	String				

그림 9. 동기 이벤트 처리용 JMS 프로토콜 구성 MO 속성

이 MO는 선택적으로 요청, 응답의 하위로서 포함되거나 이벤트 처리를 위한 결합 Business Object로서 포함됩니다. 일반적으로 프로토콜 머리글 및 사용자 정의 등록 정보를 요청 메시지로부터 읽거나 응답 또는 결합 메시지로 전달해야 할 경우에 이를 지정합니다. 위에서 설명한 바와 같이 요청 Business Object는 선택적으로 프로토콜 구성 MO의 이름을 Business Object 레벨 ASI로서 선언합니다.

- `cw_mo_jms=JMSProtocolListenerConfigMOAttribute`
- `cw_mo_http=HTTPProtocolListenerConfigMOAttribute`

이벤트 처리 중에 커넥터는 프로토콜 리스너(SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS)를 사용하여 전송에서 이벤트를 검색합니다. 이 이벤트는 웹 서비스로 표시된 협업에서 서비스를 요청하는 내부 또는 외부 웹 서비스 클라이언트의 메시지입니다. 각 전송에는 고유한 머리글 요구사항이 있습니다. 커넥터는 프로토콜 구성 MO를 사용하여 프로토콜 리스너에서 협업으로 프로토콜 특정 머리글 정보를 전달합니다. 프로토콜 구성 MO 속성은 인바운드 SOAP/JMS 메시지의 머리글에 해당합니다. 커넥터는 인바운드 SOAP 메시지 내용을 사용하여 Business Object에 이들 속성의 값을 설정합니다. SOAP/JMS 프로토콜의 경우, 이벤트 및 요청 처리에 필요한 프로토콜 구성 MO 속성은 다음과 같습니다.

표 9. SOAP JMS 프로토콜 구성 MO 속성: 이벤트 및 요청 처리

SOAP/JMS 프로토콜 구성 MO 속성	JMSHeaderName	설명
CorrelationID	JMSCorrelationID	인바운드 메시지: 이 속성은 JMSCorrelationID 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSCorrelationID 머리글로 설정됩니다.

표 9. SOAP JMS 프로토콜 구성 MO 속성: 이벤트 및 요청 처리 (계속)

SOAP/JMS 프로토콜 구성 MO 속성	JMSHeaderName	설명
MessageId	JMSMessageId	인바운드 메시지: 이 속성은 JMSMessageId 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 이 속성은 아웃바운드 메시지에 사용되지 않습니다.
Priority	JMSPriority	인바운드 메시지: 이 속성은 JMSPriority 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSPriority 머리글에 설정됩니다.
Expiration	JMSExpiration	인바운드 메시지: 이 속성은 JMSExpiration 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSExpiration 머리글에 설정됩니다.
DeliveryMode	JMSDeliveryMode	인바운드 메시지: 이 속성은 JMSDeliveryMode 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSDeliveryMode 머리글에 설정됩니다.
Destination	JMSDestination	인바운드 메시지: 이 속성은 JMSDestination 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 요청 처리 이 속성의 값은 목적지 대기열 이름으로 사용되며 간접적으로 전송 메시지의 JMSDestination 머리글에 파생된 목적지 경로로 설정됩니다. 이벤트 공고의 동기 응답: 이 속성은 사용되지 않습니다.
Redelivered	JMSRedelivered	인바운드 메시지: 이 속성은 JMSRedelivered 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSRedelivered 머리글에 설정됩니다.
ReplyTo	JMSReplyTo	인바운드 메시지: 이 속성은 JMSReplyTo 머리글에 있는 값으로 설정됩니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSReplyTo 머리글에 설정됩니다.
TimeStamp	JMSTimeStamp	인바운드 메시지: 이 속성은 JMSTimeStamp 머리글의 값으로 채워집니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSTimeStamp 머리글에 설정됩니다.
유형	JMSType	인바운드 메시지: 이 속성은 JMSType 머리글에 있는 값으로 채워집니다. 아웃바운드 메시지: 이 속성의 값은 전송 메시지의 JMSType 머리글에 설정됩니다.
UserDefinedProperties	36 페이지의 『이벤트 처리를 위한 사용자 정의 등록 정보』를 참조하십시오.	이 선택적 읽기/쓰기 속성은 사용자 정의된 프로토콜 등록 정보 Business Object를 유지합니다. 자세한 정보는 36 페이지의 『이벤트 처리를 위한 사용자 정의 등록 정보』를 참조하십시오.

주: JMS 프로토콜 구성 MO에 전달된 머리글 값이 요청-응답 이벤트의 컨텍스트에서 논리적으로 올바른지를 확인하는 것은 협업의 책임입니다.

SOAP/HTTP(S) 프로토콜의 경우, 프로토콜 구성 MO 속성은 다음과 같습니다.

표 10. 이벤트 처리를 위한 HTTP/HTTPS 프로토콜 구성 MO 속성

속성	필수	유형	설명
Content-Type	아니오	문자열	이 속성의 값은 전송되는 메시지의 Content-Type 머리글을 정의합니다(여기에는 메시지 ContentType 및 전송 메시지에 대한 0개 이상의 매개변수(Charset)가 포함될 수 있음). 구문은 HTTP 프로토콜의 Content-Type 머리글의 경우와 동일합니다. 예제: text/html; charset=ISO-8859-4. Content-Type 속성이 정의되지 않은 경우, 커넥터는 요청의 ContentType을 응답/결합 메시지의 ContentType으로 사용합니다.
UserDefinedProperties	아니오	Business Object	이 속성은 사용자 정의 프로토콜 등록 정보 Business Object를 유지합니다.
하나 이상의 HTTP 머리글	아니오	문자열	이 속성을 사용하여 핸들러는 지정된 HTTP 머리글에 대한 값을 전달하거나 검색할 수 있습니다.
Authorization_UserID	아니오	String	이 속성은 HTTP 기본 인증의 사용자 ID에 해당합니다.
Authorization_Password	아니오	String	이 속성은 HTTP 기본 인증의 암호에 해당합니다.

이러한 속성들은 다음에 설명되어 있습니다.

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

프로토콜 리스너에 대한 자세한 정보를 보려면 68 페이지의 『프로토콜 리스너』의 내용을 참조하십시오. (요청 처리를 위한 프로토콜 구성 MO에 대해하는 정보를 보려면 44 페이지의 『동기 요청 처리 TLO』의 내용을 참조하십시오.)

이벤트 처리를 위한 사용자 정의 등록 정보: HTTP(S) 프로토콜 구성 MO에 선택적으로 사용자 정의 등록 정보를 지정할 수 있습니다. 이 작업은 UserDefinedProperties 속성을 포함함으로써 이루어집니다. 이 속성은 등록 정보 값이 있는 하나 이상의 하위 등록 정보를 지닌 Business Object와 대응됩니다. 이 Business Object의 모든 속성은 다음과 같이 메시지 머리글의 변수 부분에서 읽혀질(또는 동기 응답의 경우에는 쓰여질) 단일 등록 정보를 정의해야 합니다.

- 프로토콜 등록 정보 유형에 상관없이 속성의 유형은 항상 String이어야 합니다. 속성의 응용프로그램 특정 정보에는 속성이 맵핑되는 프로토콜 메시지 등록 정보의 이름 및 형식을 정의하는 두 개의 이름-값 쌍이 포함되어 있습니다.

표 11에서는 이러한 속성에 대한 응용프로그램 특정 정보를 요약합니다.

표 11. 사용자 정의 프로토콜 등록 정보 속성에 대한 응용프로그램 특정 정보: 이름=값 쌍 내용

이름	값	설명
ws_prop_name (대소문자를 구분하지 않으며, 지정하지 않으면 속성 이름이 등록 정보 이름으로 사용됨)	임의의 유효한 프로토콜 등록 정보 이름	프로토콜 등록 정보의 이름입니다. 일부 벤더는 확장 기능을 제공하기 위해 특정 등록 정보를 준비합니다. 일반적으로 이러한 벤더 특정 기능에 대한 액세스가 필요하지 않은 경우에는 JMS로 시작하는 사용자 정의 등록 정보(JMS 프로토콜용)를 정의하지 마십시오.
ws_prop_type (대소문자를 구분하며 JMS에 대해 선택적임. 지정하지 않으면 String으로 가정함, String 유형만 의미가 있으므로 HTTP(S)에 대해서는 관계가 없음)	String, Integer, Boolean, Float, Double, Long, Short	프로토콜 등록 정보의 유형입니다. JMS 프로토콜의 경우, JMS API는 JMS Message에 등록 정보 값을 설정하기 위한 다수의 메소드를 제공합니다. setIntProperty, setLongProperty, setStringProperty 등. 여기에 지정된 JMS 등록 정보의 유형은 메시지에서 등록 정보 값의 설정에 사용되는 메소드를 표시합니다.

주어진 사용자 정의 등록 정보 ASI(ws_prop_name 또는 ws_prop_type)가 올바르지 않으며 이 머리글을 처리하기 위한 논리적인 방법이 없는 경우(예: HTTP 처리에 대해 등록 정보 유형을 무시함), 커넥터는 경고를 로깅하며 이 등록 정보를 무시합니다. ws_prop_name 또는 ws_prop_type에 대한 필수 확인을 수행한 후에 사용자 정의 등록 정보의 값이 설정되거나 검색되지 않으면 커넥터는 오류를 로깅하며 이벤트가 실패합니다.

UserDefinedProperties 속성이 지정되어 있으면 커넥터가 UserDefinedProperties Business Object의 인스턴스를 작성합니다. 그리고 커넥터는 메시지에서 등록 정보 값을 추출하여 이를 Business Object에 저장하려고 시도합니다. 적어도 하나의 등록 정보 값이 정상적으로 검색되면 커넥터는 프로토콜 구성 MO에 수정된 UserDefinedProperties 속성을 설정합니다.

동기 이벤트 처리의 경우, UserDefinedProperties 속성이 지정되어 있고 자체 Business Object가 인스턴트화되어 있으면 커넥터는 이 하위 Business Object의 각 속성을 처리하며 이에 따라 메시지 등록 정보 값을 설정합니다.

이벤트 처리를 위한 HTTP 신임 전파: 신임 전파의 목적을 위해, 커넥터는 HTTP 프로토콜 구성 MO에서 Authorization_UserID 및 Authorization_Password 속성을 지원합니다. 지원은 HTTP 기본 인증 설계의 파트로서 이 신임의 전파로 제한됩니다.

SOAP/HTTP 또는 SOAP/HTTPS 프로토콜 리스너가 인증 머리글을 포함하는 SOAP/HTTP 웹 서비스 요청을 처리하는 경우, 리스너는 HTTP 기본 인증과 일치하는 지 여부를 결정하기 위해 머리글을 구문 분석합니다. 그런 경우, 리스너는 사용자 이름 및 암호를 추출하고 디코드합니다(Base64 사용). 디코드된 이 문자열은 콜론으로 구분된 사용자 이름과 암호로 구성됩니다. 프로토콜 리스너가 프로토콜 구성 MO에서 Authorization_UserId 및 Authorization_Password 속성을 찾은 경우, 리스너는 이벤트 권한 머리글에서 추출한 사용자 이름과 암호로 이 값을 설정합니다.

머리글 컨테이너 Business Object

그림 10에서는 확장 머리글 컨테이너 속성, OrderHeader를 보여줍니다.

	Pos	Name	Type	Key	Required	Card	App Spec Info
1	1	Request	SERVICE_SYNCH_OrderStatus_	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=request
1.1	1.1	OrderId	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
1.2	1.4	SOAPJMSCfgMO	SERVICE_SYNCH_SOAP_JMS_	<input type="checkbox"/>	<input type="checkbox"/>	1	
1.3	1.5	ObjectEventId	String				
1.4	1.2	OrderHeader	SERVICE_SYNCH_OrderStatus_ Request_Header	<input type="checkbox"/>	<input type="checkbox"/>	1	soap_location=SOAPHeader
1.4.1	1.2.1	transaction	SERVICE_SYNCH_OrderStatus_ TransactionHeaderChild	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	headerFault=transactionFault;elem_ns=http://www.mycompany.com/samples/transaction;type_name=Transaction_HeaderChild
1.4.1.1	1.2.1.1	TransactionId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
1.4.1.2	1.2.1.2	ObjectEventId	String				
1.4.1.3	1.2.1.3	actor	String	<input type="checkbox"/>	<input type="checkbox"/>		attr_name=actor
1.4.1.4	1.2.1.4	mustUnderstand	String	<input type="checkbox"/>	<input type="checkbox"/>		attr_name=mustUnderstand
1.4.1.5	1.2.3	ObjectEventId	String				
1.4.3	1.2.2	affiliate	SERVICE_SYNCH_OrderStatus_ TradingPartnerHeaderChild	<input type="checkbox"/>	<input type="checkbox"/>	1	headerFault=affiliateFault;elem_ns=http://www.mycompany.com/samples/partner;type_name=TradingPartner_HeaderChild
1.4.3.1	1.2.2.1	partnerId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
1.4.3.2	1.2.2.2	ObjectEventId	String				
1.4.3.3	1.2.2.3	routingNumber	String	<input type="checkbox"/>	<input type="checkbox"/>		
1.5	1.3	SOAPCfgMO	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	
2	2	Response	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=response
3	3	Fault	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=fault

그림 10. 머리글 컨테이너 및 하위 Business Object

SOAP 머리글 속성이라고도 하는 머리글 컨테이너 속성은 하위 Business Object만 포함하는 Business Object에 해당합니다. 각 하위는 SOAP 메시지의 머리글 항목을 나타냅니다. 그림 10에 표시된 예에서 요청 머리글 컨테이너는 OrderHeader입니다. SOAP 머리글 속성에는 SOAP Data Handler에 필요한 응용프로그램 특정 정보(ASI)가 있습니다. 예를 들어, 머리글 컨테이너 Business Object는 해당 ASI인 soap_location=SOAPHeader에 의해 식별됩니다. 머리글 처리에 대한 정보는 127 페이지의 『SOAP Data Handler 처리』를 참조하십시오.

요청, 응답 또는 결합 오브젝트이든지 간에 모든 SOAP Business Object에는 단 하나의 머리글 컨테이너가 있습니다.

머리글 하위 Business Object

그림 10에 표시된 예에서, 요청 머리글 컨테이너(OrderHeader)의 두 하위 속성은 1) 유형 SERVICE_SYNCH_OrderStatus_TransactionHeaderChild의 트랜잭션과 2) 유형 SERVICE_SYNCH_OrderStatus_TradingPartnerHeaderChild의 계열입니다. 이러한 속성은 머리글 하위 Business Object에 해당합니다. 각각은 SOAP 메시지의 단일 머리글 요소를 표시합니다. 머리글 요소는 SOAP 메시지의 SOAP-Env:Header 요소의 바로 아래 하위입니다. 그림 10에서처럼, 머리글 하위 Business Object에는 actor 및 mustUnderstand 속성이 있을 수 있습니다. 이러한 속성은 SOAP 머리글 요소의 actor 및 mustUnderstand 속성에 해당합니다. 머리글 처리에 대한 정보는 127 페이지의 『SOAP Data Handler 처리』를 참조하십시오.

머리글 하위 오브젝트의 수는 SOAP 머리글 메시지 요소를 나타내는 데 필요한 만큼 있을 수 있습니다.

비동기 이벤트 처리 TLO

그림 11에서는 비동기 이벤트 처리용 Business Object 계층 구조를 보여줍니다. 요청 오브젝트만 필요합니다.

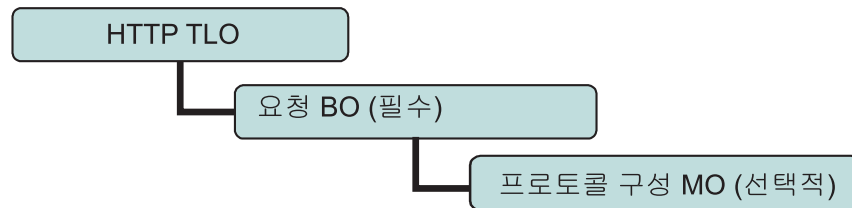


그림 11. 비동기 이벤트 처리용 Business Object 계층 구조

TLO에는 속성 레벨 ASI가 있는 속성뿐만 아니라 오브젝트 레벨 ASI를 포함합니다. 두 종류의 ASI 모두가 아래에 설명되어 있습니다. 머리글 컨테이너 및 머리글 하위 Business Object에 대한 정보는 38 페이지의 『머리글 컨테이너 Business Object』를 참조하십시오.

비동기 이벤트 처리 TLO에 대한 오브젝트 레벨 ASI

오브젝트 레벨 ASI는 TLO의 특성 및 그것이 포함하는 오브젝트에 대한 기본 정보를 제공합니다. 그림 12에서는 비동기 이벤트 처리에 대한 샘플 TLO인 SERVICE_ASYNC_TLO_Order에 대한 오브젝트 레벨 ASI를 보여줍니다.

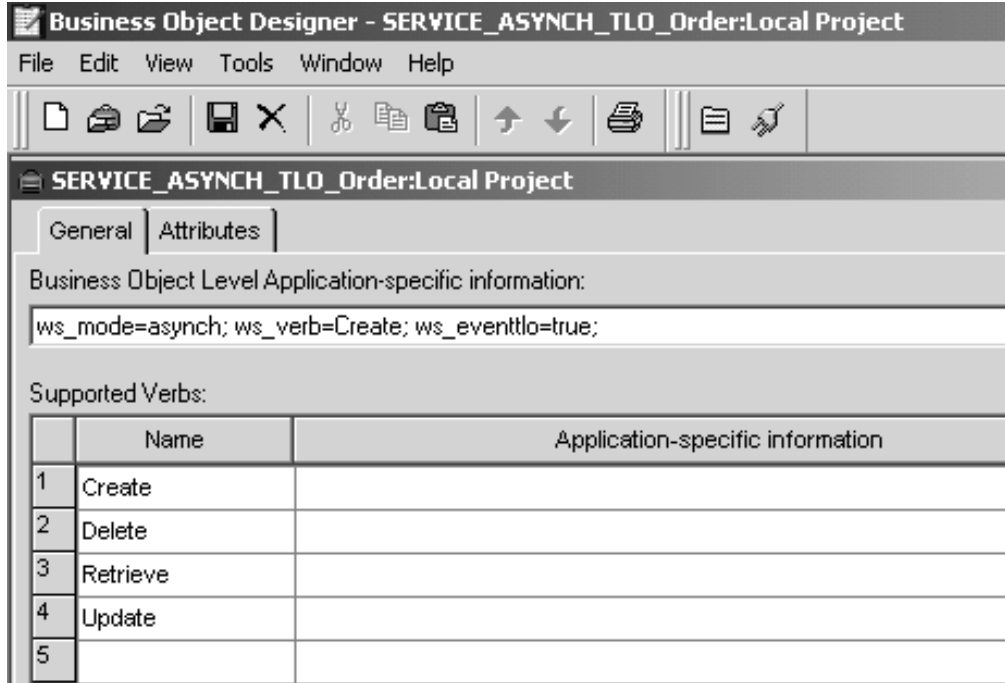


그림 12. 비동기 이벤트 처리용 최상위 레벨 Business Object

아래의 표 4는 비동기 이벤트 처리 TLO에 대한 오브젝트 레벨 ASI에 대해 설명합니다.

표 12. 비동기 이벤트 처리 TLO 오브젝트 ASI

오브젝트 레벨 ASI	설명
ws_eventtlo=true	이 ASI 등록 정보를 true로 설정할 경우, 커넥터는 이벤트를 처리를 위해 이 오브젝트를 TLO로 취급합니다. WSDL 구성 마법사가 이 ASI를 사용하여 Business Object가 TLO인지 여부를 판별합니다. 이에 대한 자세한 정보는 163 페이지의 『WSDL 구성 마법사』를 참조하십시오.
ws_verb=verb	TLO를 협업에 전달하기 전에 커넥터가 이 ASI를 사용하여 TLO에 Verb를 설정합니다. 그림 12에 표시된 샘플에서, Verb는 Create입니다.
ws_mode=asynch	이벤트 공고 중에 커넥터가 이 ASI 등록 정보를 사용하여 협업을 동기식(synch) 또는 비동기식(asynch) 호출할지 여부를 판별합니다. 비동기 처리의 경우, 이 ASI가 asynch로 설정되어야 합니다. 기본값은 asynch입니다.

주: 동기 이벤트 처리와는 달리 비동기 이벤트 처리의 경우에는 TLO 레벨에서 협업 이름 ASI가 필요하지 않습니다. 대신 통합 브로커는 응용프로그램 이벤트가 모든 등록 협업에 도달하도록 합니다.

비동기 이벤트 처리 TLO에 대한 속성 레벨 ASI

각 비동기 이벤트 처리 TLO에는 요청 Business Object에 해당하는 단일 속성이 있습니다. 그림 13에서는 샘플 TLO인 SERVICE_ASYNC_TLO_Order의 요청 속성과 속성의 ASI를 보여줍니다.

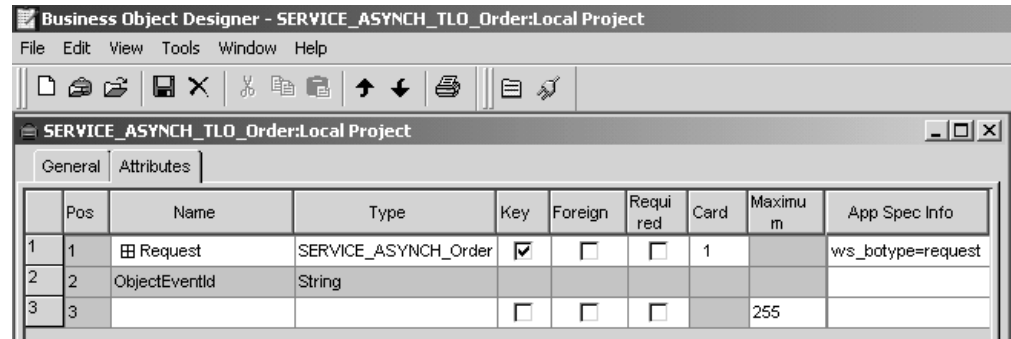


그림 13. 비동기 이벤트 처리용 TLO 속성

표 13은 비동기 이벤트 처리 TLO의 요청 속성에 대한 속성 레벨 ASI를 요약합니다.

표 13. 비동기 이벤트 처리 TLO 속성 ASI

TLO 속성	속성 레벨 ASI	설명
Request	ws_botype=request	이 속성은 웹 서비스 요청에 대응합니다. 커넥터는 해당 ASI를 사용하여 TLO 속성이 SOAP 요청 BO 유형인지 판별합니다. 속성 이름이 아닌 이 ASI가 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 속성의 ASI를 사용합니다. 이 속성은 동기 이벤트 처리 TLO에 필수입니다.

비동기 이벤트 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 비동기 이벤트 처리를 위해 필수입니다. 요청 Business Object에 대한 기본 Verb를 지정할 수 있습니다. 요청 Business Object의 최상위 레벨에 있는 지원되는 Verb 목록에서 Verb의 ASI 필드에 다음을 지정하여 DefaultVerb=true;

를 지정하여 기본 Verb를 지정할 수 있습니다. DefaultVerb ASI가 지정되지 않고 Data Handler가 Verb가 설정되지 않은 Business Object를 처리하는 경우, Business Object는 Verb 없이 리턴됩니다. 비동기 이벤트 처리를 위한 요청 Business Object에 대한 오

브젝트 레벨 ASI가 표 14에 설명되어 있습니다.

표 14. 비동기 이벤트 처리: 요청 Business Object에 대한 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
<code>cw_mo_soap=SOAPCfgMO</code>	이 ASI 값은 SOAP 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 요청 Business Object에 대한 Data Handler 변환을 정의하는 SOAP 구성 MO입니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』를 참조하십시오.
<code>cw_mo_jms=SOAPJMScfgMO</code> 또는 <code>cw_mo_http=SOAPHTTPCfgMO</code>	이 ASI 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 첫 번째 ASI는 SOAP/JMS 프로토콜 리스너를 지정하며 두 번째는 SOAP/HTTP 또는 SOAP/HTTPS 프로토콜 리스너를 지정합니다. ASI와 프로토콜 구성 MO는 선택적입니다. 자세한 정보는 33 페이지의 『프로토콜 구성 MO』를 참조하십시오.
<code>ws_tlname=tlname</code>	이 ASI는 이 오브젝트가 속하는 웹 서비스 TLO의 이름을 지정합니다. 이벤트 처리 중에 커넥터가 이 ASI를 사용하여 Data Handler에 의해 전달되는 요청 Business Object가 TLO의 하위인지 여부를 판별합니다. 이 경우, 커넥터는 지정된 TLO를 작성하고, 요청 Business Object를 그의 하위로 설정하며, TLO 오브젝트 레벨 ASI를 사용하여 등록된 협업에 전달합니다.

그림 14에 표시된 샘플에서, 요청 속성에는 SOAP 구성 MO와 머리글 컨테이너 (OrderHeader)는 물론 내용 관련 속성(OrderLineItems)이 있습니다. SOAP 구성 MO, 프로토콜 구성 MO, SOAP 머리글 컨테이너 및 머리글 하위 Business Object의 요구 사항과 특성은 동기 이벤트 처리에서와 같이 비동기 이벤트 처리에 대해서도 동일합니다. 자세한 정보는 26 페이지의 『동기 이벤트 처리 TLO』의 주제를 참조하십시오.

	Pos	Name	Type	Key	Required	Card	Maximum	App Spec Info
1	1	Request	SERVICE_ASYNC_Order	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1		ws_botype=request
1.1	1.1	OrderId	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		255	
1.2	1.2	OrderDate	Date	<input type="checkbox"/>	<input type="checkbox"/>			
1.3	1.3	CustomerId	String	<input type="checkbox"/>	<input type="checkbox"/>		255	
1.4	1.4	OrderLineItems	SERVICE_ASYNC_Order_LineItem	<input type="checkbox"/>	<input type="checkbox"/>	N		type_name=Order_LineItem
1.5	1.5	OrderHeader	SERVICE_ASYNC_Order_Header	<input type="checkbox"/>	<input type="checkbox"/>	1		soap_location=SOAPHeader
1.6	1.6	SOAPCfgMO	SERVICE_ASYNC_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.7	1.7	SOAPJMSCfgMO	SERVICE_ASYNC_SOAP_JMS_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.8	1.8	ObjectEventId	String					
2	2	ObjectEventId	String					
3	3			<input type="checkbox"/>	<input type="checkbox"/>		255	

그림 14. 비동기 이벤트 처리용 요청 속성

이벤트 처리 비TLO

오브젝트 레벨 ASI ws_eventtlo=true가 Business Object에 존재하지 않는 경우, 커넥터는 오브젝트가 TLO가 아니라고 결론짓습니다. 이벤트 처리 중에, 커넥터는 비TLO(일반 Business Object와 응용프로그램 특정 Business Object)를 처리할 수 있습니다. 비TLO에서는 동일한 Business Object가 요청 및 응답 Business Object를 나타냅니다.

비TLO에는 SOAP 구성 MO가 없습니다. 협업을 웹 서비스로 표시할 때, WSDL 구성 마법사가 커넥터의 WSCollaborations 등록 정보를 구성합니다. 커넥터는 WSCollaborations 등록 정보를 사용하여 요청 메시지의 BodyName 및 BodyNS를 판별합니다. 비TLO의 경우, WSCollaborations 등록 정보가 Business Object 해석에 사용됨을 주의하십시오.

비TLO 사용의 장점은 웹 서비스 솔루션에 사용할 새로운 TLO 구조 Business Object를 개발할 필요가 없다는 것입니다. 그러나 TLO를 사용하면 데이터(고객, 회사 또는 기타)를 더욱 정확하고 경제적으로 표시할 수 있습니다. TLO Business Object는 또한 비TLO보다 더 많은 사용자 정의를 제공합니다.

비TLO를 WSDL 구성 마법사에 대한 입력으로 사용할 때의 요구사항에 대한 자세한 정보는 160 페이지의 『Business Object 식별 또는 개발』을 참조하십시오.

동기 요청 처리 TLO

요청 처리의 경우, 커넥터는 두 종류의 TLO(동기 및 비동기)를 사용할 수 있습니다. 이 섹션은 동기 요청 처리 TLO에 대해 설명합니다.

그림 15에서는 동기 요청 처리를 위한 TLO Business Object 계층 구조를 보여줍니다. 요청 및 응답 오브젝트는 필수이지만, 결함 오브젝트는 선택적입니다. 이벤트 처리와는 달리, 프로토콜 구성 MO는 요청 오브젝트에 대해서는 필수이고, 응답 및 결함 오브젝트에 대해서는 선택적입니다. 머리글 컨테이너 및 머리글 하위 Business Object에 대한 정보는 38 페이지의 『머리글 컨테이너 Business Object』를 참조하십시오.

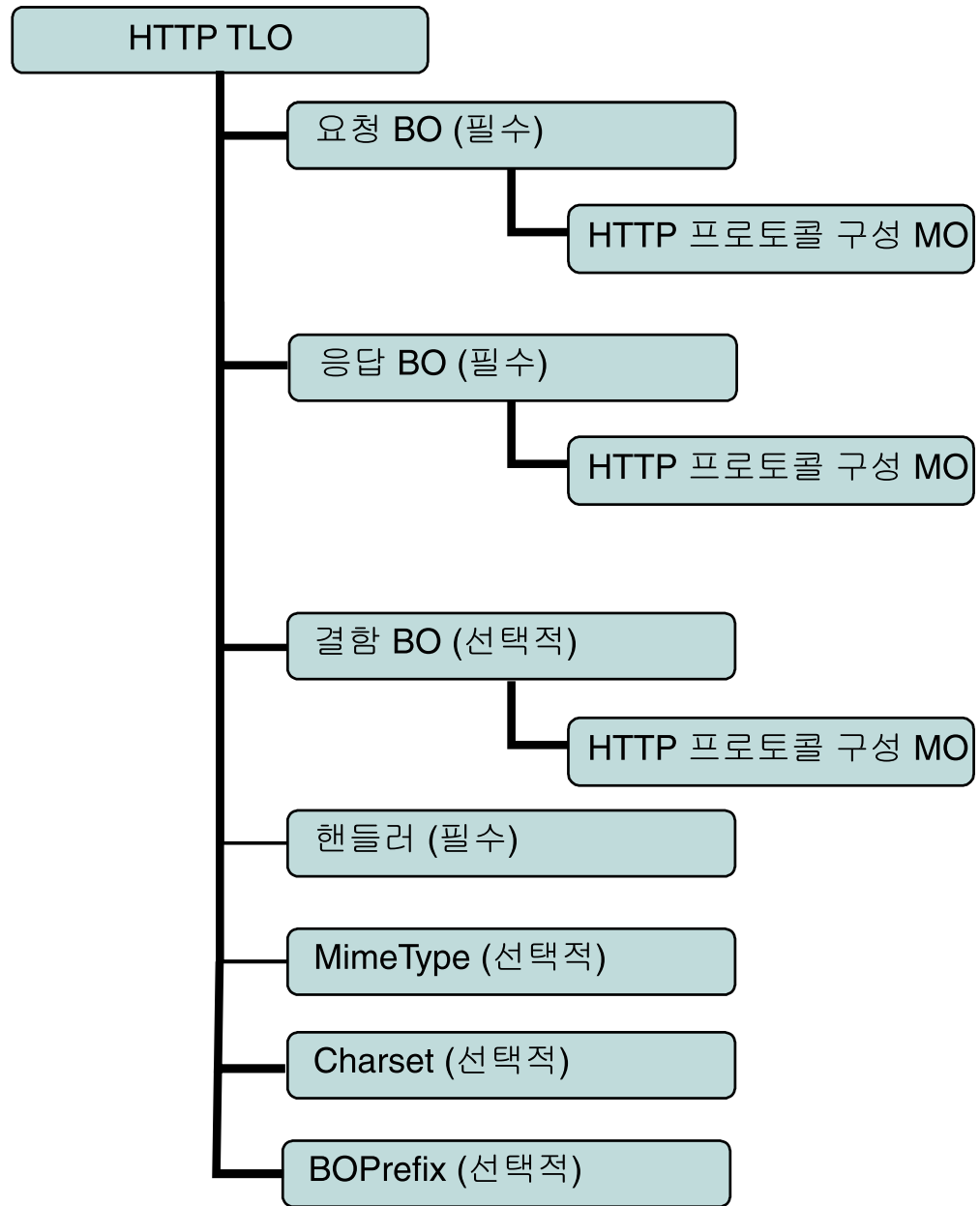


그림 15. 동기 요청 처리용 Business Object 계층 구조

동기 요청 처리 TLO에 대한 오브젝트 레벨 ASI

오브젝트 레벨 ASI는 TLO의 특성 및 그것이 포함하는 오브젝트에 대한 중요한 정보를 제공합니다. 그림 16에서는 동기 요청 처리용 샘플 TLO인 CLIENT_SYNCH_TLO_OrderStatus를 보여줍니다.

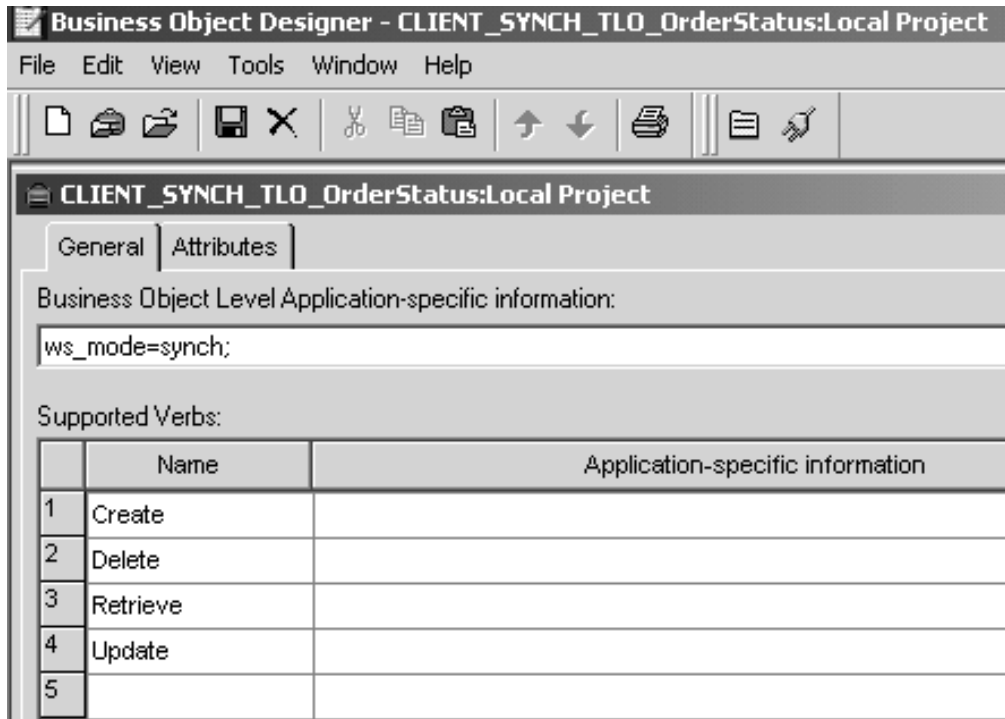


그림 16. 동기 요청 처리용 최상위 레벨 Business Object

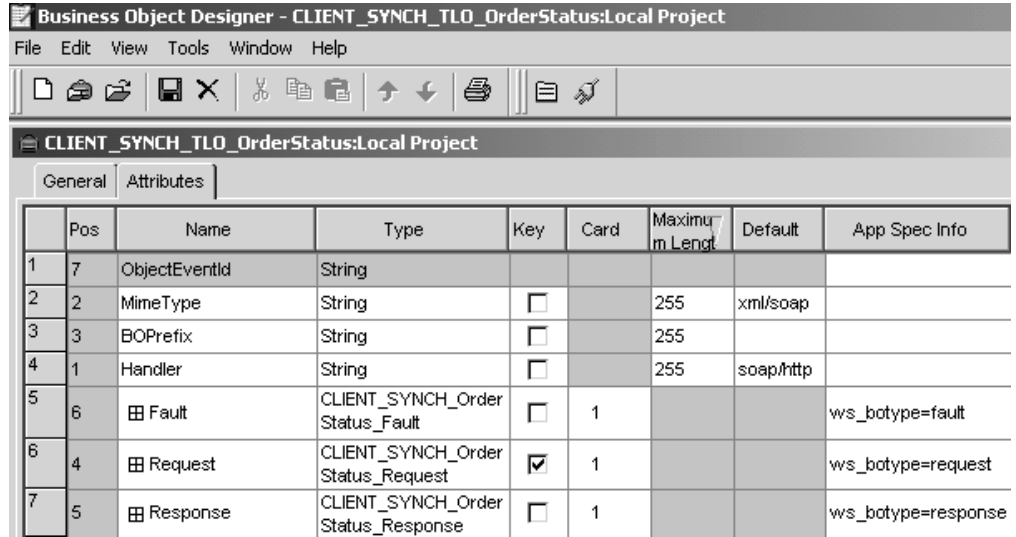
표 15는 동기 요청 처리 TLO에 대한 오브젝트 레벨 ASI에 대해 설명합니다. 동기 이벤트 처리 TLO에 대한 ASI와는 달리, 요청 처리를 위한 이 레벨에서 ws_collab, ws_verb 또는 ws_eventtlo ASI는 필수가 아닙니다.

표 15. 동기 요청 처리 TLO 오브젝트 ASI

오브젝트 레벨 ASI	설명
ws_mode=synch	요청 처리 중에 커넥터가 이 ASI 등록 정보를 사용하여 웹 서비스를 동기식으로(synch) 또는 비동기식으로(asynch) 호출할지 여부를 판별합니다. synch가 표시되면 커넥터는 응답을 기다리므로 TLO에는 요청 및 응답 Business Object 그리고 선택적으로 하나 이상의 결합 오브젝트가 포함되어야 합니다. 기본값은 asynch입니다.

동기 요청 처리 TLO에 대한 속성 레벨 ASI

그림 17에서는 CLIENT_SYNCH_TLO_OrderStatus TLO의 속성은 물론 속성 레벨 ASI를 보여줍니다.



	Pos	Name	Type	Key	Card	Maximum Length	Default	App Spec Info
1	7	ObjectEventId	String					
2	2	MimeType	String	<input type="checkbox"/>		255	xml/soap	
3	3	BOPrefix	String	<input type="checkbox"/>		255		
4	1	Handler	String	<input type="checkbox"/>		255	soap/http	
5	6	Fault	CLIENT_SYNCH_OrderStatus_Fault	<input type="checkbox"/>	1			ws_botype=fault
6	4	Request	CLIENT_SYNCH_OrderStatus_Request	<input checked="" type="checkbox"/>	1			ws_botype=request
7	5	Response	CLIENT_SYNCH_OrderStatus_Response	<input type="checkbox"/>	1			ws_botype=response

그림 17. 동기 요청 처리용 TLO 속성

표 16은 속성과 그림 17에 표시된 ASI에 대해 설명합니다.

표 16. 요청 처리 TLO 속성

TLO 속성	속성 레벨	ASI	설명
MimeType	없음		이 속성은 요청 Business Object를 요청 메시지로 변환하기 위해 커넥터가 호출하는 Data Handler의 MIME 유형을 지정합니다. 이 값은 메시지 변환 규칙 구성에 따라 동기 응답/결함 메시지를 Business Object로 변환하는 데 사용될 수 있습니다.
BOPrefix	없음		String 유형의 이 속성은 Data Handler로 전달됩니다.
Handler	없음		이 속성은 웹 서비스 요청을 처리하는 데 사용할 프로토콜 핸들러를 지정하며 요청 처리용으로만 사용됩니다. 다음 값 중 하나를 사용합니다. <ul style="list-style-type: none"> soap/jms 커넥터는 SOAP/JMS 프로토콜 핸들러를 사용하여 요청을 처리합니다. soap/http 커넥터는 SOAP/HTTP, SOAP/HTTPS 프로토콜 핸들러를 사용하여 이 웹 서비스 요청을 처리합니다. 기본값은 soap/http입니다.
Charset			문자열 유형의 이 선택적 매개변수는 요청 Business Object를 메시지로 변환할 때 Data Handler에 설정되는 Charset을 지정합니다. 참고: 이 속성에 지정된 charset 값은 요청 메시지의 Content-Type 프로토콜 머릿글에 전달되지 않습니다.

표 16. 요청 처리 TLO 속성 (계속)

TLO 속성	속성 레벨 ASI	설명
Request	ws_botype=request	이 속성은 웹 서비스 요청 Business Object에 해당합니다. 커넥터는 이 속성 ASI를 사용하여 TLO 속성이 SOAP 요청 BO 유형인지 판별합니다. 속성 이름이 아닌 이 ASI가 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 채워진 속성의 ASI를 사용합니다.
Response	ws_botype=response	이 속성은 협업으로 리턴된 응답에 해당하며 동기 요청 처리에 필요합니다. 커넥터는 이 속성 ASI를 사용하여 TLO 속성이 SOAP 응답 BO 유형인지 판별합니다. 속성 이름이 아닌 이 ASI가 속성 유형을 판별합니다.
Fault	ws_botype=fault 또는 ws_botype=defaultfault	동기 요청 처리의 경우 선택적인 이 속성은 응답을 채울 수 없을 때 웹 서비스가 리턴하는 결함 메시지에 해당합니다. 커넥터는 이 ASI를 사용하여 TLO 속성이 SOAP 결함 BO 유형인지 판별합니다. 속성 이름이 아닌 이 ASI가 속성 유형을 판별합니다. 결함 메시지가 세부사항 요소인 경우, defaultfault Business Object가 리턴됩니다. defaultfault는 기본 Business Object 해석에서 사용됩니다. 자세한 정보는 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

동기 요청 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 동기 요청 처리를 위해 필수입니다. 요청 Business Object에는 오브젝트 레벨 ASI가 있습니다.

예를 들어, CLIENT_SYNCH_OrderStatus_Request를 열고 일반 탭을 누르면 그림 18에 표시된 대로 오브젝트 레벨 ASI가 표시됩니다.

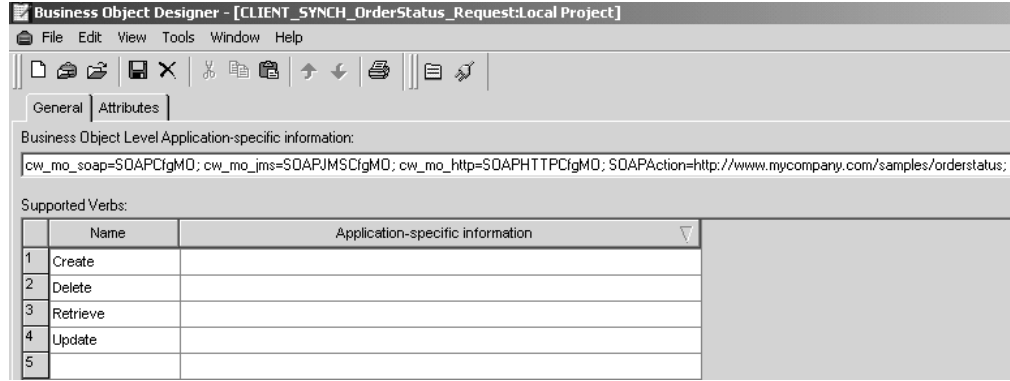


그림 18. 동기 요청 처리용 요청 오브젝트 ASI

표 17은 동기 요청 처리용 요청 Business Object에 대한 오브젝트 레벨 ASI에 대해 설명합니다.

표 17. 동기 요청 처리: 요청 Business Object에 대한 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
<code>cw_mo_soap=SOAPCfmgMO</code>	이 ASI 값은 SOAP 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 요청 Business Object에 대한 Data Handler 변환을 정의하는 SOAP 구성 MO입니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』를 참조하십시오.
<code>cw_mo_jms=SOAPJMScfmgMO</code>	이 ASI 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 JMS 프로토콜 핸들러의 목적지 웹 서비스를 지정하는 프로토콜 구성 MO입니다. 자세한 정보는 51 페이지의 『요청 처리용 요청 Business Object의 JMS 프로토콜 구성 MO』를 참조하십시오.
<code>cw_mo_http=SOAPHTTpcfmgMO</code>	이 선택적 ASI의 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이것은 SOAP/HTTP-HTTPS 프로토콜 핸들러에 대한 목적지를 지정하는 별도의 프로토콜 구성 MO입니다. 이 ASI는 SOAP/HTTP 및 SOAP/HTTPS 프로토콜 핸들러가 사용합니다. TLO 요청 속성에 사용 중인 웹 서비스 프로토콜 유형에 따라 요청 처리를 위한 JMS 또는 HTTP 프로토콜 구성 MO가 있어야 한다는 것에 주의하십시오. 자세한 정보는 51 페이지의 『요청 처리용 HTTP 프로토콜 구성 MO』를 참조하십시오.
<code>SOAPAction=SOAPActionURI</code>	커넥터는 이 ASI를 사용하여 요청 메시지에서 SOAPAction 머리글을 설정할지 여부를 판별합니다. 대상 웹 서비스에 SOAPAction 머리글이 필요한 경우에만 이 ASI를 지정하십시오. 이 ASI는 요청 처리에 사용되지만 이벤트 공고에는 사용되지 않습니다.

동기 요청 처리를 위한 응답 Business Object

응답 Business Object는 TLO의 하위이며 동기 요청 처리를 위해 필수입니다. 동기 요청 처리용 응답 Business Object에 대한 오브젝트 레벨 ASI가 표 18에 설명되어 있음

니다.

표 18. 동기 요청 처리: 응답 Business Object용 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_soap=SOAPCfgMO	이 ASI 값은 SOAP 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 응답 Business Object의 Data Handler 변환을 정의하는 SOAP 구성 MO입니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』를 참조하십시오.
cw_mo_jms=SOAPJMScfg MO 또는 cw_mo_http=SOAPHTTPCfgMO	이 ASI 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 JMS 또는 HTTP(s) 프로토콜 핸들러에 대한 응답 SOAP 메시지에 머릿글을 지정하는 프로토콜 구성 MO(응답 Business Object에 대해 선택적)입니다. 자세한 정보는 33 페이지의 『프로토콜 구성 MO』를 참조하십시오.

응답 Business Object에 대한 기본 Verb를 지정할 수 있습니다. 요청 Business Object의 최상위 레벨에 있는 지원되는 Verb 목록에서 Verb의 ASI 필드에 다음을 지정하여 DefaultVerb=true;

를 지정하여 기본 Verb를 지정할 수 있습니다. DefaultVerb ASI가 지정되지 않고 Data Handler가 Verb가 설정되지 않은 Business Object를 처리하는 경우, 응답 Business Object가 Verb 없이 리턴됩니다.

동기 요청 처리를 위한 결합 Business Object

결합 Business Object는 TLO의 하위이며 동기 요청 처리의 경우 선택적입니다. 동기 요청 처리용 결합 Business Object에 대한 오브젝트 레벨 ASI가 표 8에 설명되어 있습니다.

표 19. 동기 요청 처리: 결합 Business Object에 대한 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
cw_mo_soap=SOAPCfgMO	이 ASI 값은 SOAP 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 결합 Business Object의 Data Handler 변환을 정의하는 SOAP 구성 MO입니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』를 참조하십시오.
cw_mo_jms=SOAPJMScfg MO 또는 cw_mo_http=SOAPHTTPCfgMO	이 ASI 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 JMS 프로토콜 핸들러에 대한 응답 SOAP 메시지에 머릿글을 지정하는 결합 Business Object에 대해 선택적인 프로토콜 구성 MO입니다. 자세한 정보는 33 페이지의 『프로토콜 구성 MO』를 참조하십시오.

SOAP 구성 MO

SOAP 구성 MO(SOAPCfgMO)의 속성은 이벤트 처리 SOAP 구성 MO의 속성과 동일합니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』 및 121 페이지의 『SOAP 구성 Meta Object: 모든 SOAP Business Object의 하위 오브젝트』를 참조하십시오.

요청 처리용 요청 Business Object의 JMS 프로토콜 구성 MO

JMS 프로토콜 구성 MO는 JMS 웹 서비스 사용 시 요청 Business Object에서 필수이며 응답 및 결합 오브젝트에 대해서는 선택적입니다. 표 20에서는 요청 처리 JMS 프로토콜 구성 MO에 대해합니다. 목적지는 가장 중요하며 유일한 필수 속성입니다. JMS 프로토콜 핸들러에는 이 속성을 사용하여 요청한 웹 서비스가 있습니다. 또한 33 페이지의 『프로토콜 구성 MO』의 JMS 구성 MO에 설명된 모든 속성은 선택적입니다.

표 20. 요청 처리용 JMS 프로토콜 구성 MO 속성

속성	필수	유형	설명
Destination	예	String	대상 웹 서비스의 목적지 대기열 이름. JMS 프로토콜 핸들러는 이 속성을 사용하여 웹 서비스의 목적지를 판별합니다. 커넥터 특정 JNDI 등록 정보인 LookupQueuesUsingJNDI를 true로 설정하면 JMS 프로토콜 핸들러는 JNDI를 사용하여 이 대기열을 찾습니다. 이 속성이 목적지 대기열의 JNDI 이름을 제공하는지 확인하십시오.

요청 처리용 HTTP 프로토콜 구성 MO

요청 처리 중에, SOAP/HTTP-HTTPS 프로토콜 핸들러가 HTTP 프로토콜 구성 MO를 사용하여 대상 웹 서비스의 목적지를 판별합니다. 이 프로토콜 구성 MO는 요청 Business Object에 필수입니다. SOAP/HTTP-HTTPS 프로토콜 핸들러는 HTTP 1.0 POST 요청만 지원합니다. 표 21에서처럼, 유일한 필수 속성(Destination)은 대상 웹 서비스의 전체 URL입니다. 선택적 권한 속성은 아래 섹션에서 설명됩니다.

표 21. 요청 처리를 위한 HTTP 프로토콜 구성 MO 속성

속성	필수	유형	설명
Destination	예	String	대상 웹 서비스의 목적지 URL. SOAP/HTTP-HTTPS 프로토콜 핸들러가 이 속성을 사용하여 웹 서비스의 목적지를 판별합니다.
Content-Type	요청 Business Object의 경우에는 필수이며, 그렇지 않으면 선택적입니다.	String	이 속성의 값은 전송되는 메시지의 Content-Type 머리글을 정의합니다(여기에는 메시지 ContentType 및 선택적으로 전송되는 메시지의 charset이 포함될 수 있음). 구문은 HTTP 프로토콜의 Content-Type 머리글의 경우와 동일합니다. 예제: text/html; charset=ISO-8859-4. Content-Type 속성이 정의되지 않은 경우에 커넥터는 text/xml을 메시지의 ContentType으로서 사용합니다.

표 21. 요청 처리를 위한 HTTP 프로토콜 구성 MO 속성 (계속)

속성	필수	유형	설명
Authorization_UserId	아니오	String	이 속성은 HTTP 기본 인증의 사용자 ID에 해당합니다. 자세한 정보는 55 페이지의 『요청 처리를 위한 HTTP 신임 전파』를 참조하십시오.
Authorization_Password	아니오	String	이 속성은 HTTP 기본 인증의 암호와 대응됩니다. 자세한 정보는 55 페이지의 『요청 처리를 위한 HTTP 신임 전파』를 참조하십시오.
하나 이상의 HTTP 머리글	아니오	문자열	이 속성을 사용하여 핸들러는 지정된 HTTP 머리글에 대한 값을 전달하거나 검색할 수 있습니다.
UserDefinedProperties	아니오	Business object	이 속성은 사용자 정의 프로토콜 등록 정보 Business Object를 유지합니다. 자세한 정보는 『요청 처리를 위한 사용자 정의 등록 정보』를 참조하십시오.
MessageTransformationMap	아니오	단일 카디널리티 Business Object	영(0) 또는 그 이상의 메시지 변환 규칙을 유지하는 Business Object를 지시하는 속성입니다. 규칙에는 규칙에 지정된 수신 중인 메시지에 적용할 MIME 유형 및 charset과 관련된 정보가 들어 있습니다. 자세한 정보는 54 페이지의 『메시지 변환 맵』을 참조하십시오.

그림 19에서는 Business Object Designer의 일부 HTTP 프로토콜 구성 MO 속성을 표시합니다.

General		Attributes								
	Pos	Name	Type	Key	Foreign	Required	Card	Maximum	Default	App Spec Info
1	1	OrderId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255		
2	2	OrderHeader	CLIENT_SYNCH_OrderStatus_Request_Header	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3	3	HTTPTCfMO	CLIENT_SYNCH_OrderStatus_HTTPTCfMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3.1	3.1	Date	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.2	3.2	Content-Type	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.3	3.3	MessageTransformationMap	HTTP_CfMO_MsgTrnstfMap	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3.3.1	3.3.1	TransformationRule	HTTP_CfMO_MsgTrnstfRule	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N			
3.3.1.1	3.3.1.1	Content-Type	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	/*	
3.3.1.2	3.3.1.2	MimeType	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.3.1.3	3.3.1.3	Charset	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.3.1.4	3.3.1.4	ObjectEventId	String							
3.3.1.5	3.3.1.5	ObjectEventId	String							
3.4	3.4	UserDefinedProperties	HTTP_CfMO_CustomProperties	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3.4.1	3.4.1	CustomProperty1	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.4.2	3.4.2	CustomProperty2	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		ws_prop_type=Integer
3.4.3	3.4.3	CustomPropertyN	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		ws_prop_type=Boolean
3.4.4	3.4.4	ObjectEventId	String							
3.5	3.5	ObjectEventId	String							
4	4	ObjectEventId	String							
5	5			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		

그림 19. 요청 처리를 위한 HTTP 프로토콜 구성 MO 속성

HTTP 프로토콜 구성 MO 속성은 다음에 설명되어 있습니다.

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

요청 처리를 위한 사용자 정의 등록 정보: HTTP 프로토콜 구성 MO에 선택적으로 사용자 정의 등록 정보를 지정할 수 있습니다. 이 작업은 UserDefinedProperties 속성

을 포함함으로써 이루어집니다. 이 속성은 등록 정보 값이 있는 하나 이상의 하위 등록 정보를 지닌 Business Object와 대응됩니다. 이 Business Object의 모든 속성은 다음과 같이 메시지 머리글의 변수 부분에서 읽혀질(또는 동기 응답의 경우에는 쓰여질) 단일 등록 정보를 정의해야 합니다.

- 속성의 유형은 프로토콜 등록 정보 유형과 상관없이 항상 String이어야 합니다. 속성의 응용프로그램 특정 정보에는 속성이 맵핑되는 프로토콜 메시지 등록 정보의 이름 및 형식을 정의하는 두 개의 이름-값 쌍이 포함되어 있습니다.

표 22에서는 이러한 속성에 대한 응용프로그램 특정 정보를 요약합니다.

표 22. 사용자 정의 프로토콜 등록 정보 속성에 대한 응용프로그램 특정 정보: 이름=값 쌍 내용

이름	값	설명
ws_prop_name (대소문자를 구분하지 않으며, 지정하지 않으면 속성 이름이 등록 정보 이름으로 사용됨)	임의의 유효한 프로토콜 등록 정보 이름	프로토콜 등록 정보의 이름입니다. 일부 벤더는 확장 기능을 제공하기 위해 특정 등록 정보를 준비합니다. 일반적으로 이러한 벤더 특정 기능에 대한 액세스가 필요하지 않은 경우에는 JMS로 시작하는 사용자 정의 등록 정보(JMS 프로토콜용)를 정의하지 마십시오.
ws_prop_type (대소문자를 구분하며 JMS에 대해 선택적이며, 지정하지 않으면 String으로 가정함. String 유형만 의미가 있으므로 HTTP(S)에 대해서는 관계가 없음)	String, Integer, Boolean, Float, Double, Long, Short	프로토콜 등록 정보의 유형입니다. JMS 프로토콜의 경우, JMS API는 JMS Message에 등록 정보 값을 설정하기 위한 다수의 메소드를 제공합니다. setIntProperty, setLongProperty, setStringProperty 등. 여기에 지정된 JMS 등록 정보의 유형은 메시지에서 등록 정보 값의 설정에 사용되는 메소드를 표시합니다.

주어진 사용자 정의 등록 정보 ASI(ws_prop_name 또는 ws_prop_type)가 올바르지 않으며 이 머리글을 처리하기 위한 논리적인 방법이 없는 경우(예: HTTP 처리에 대해 등록 정보 유형을 무시함), 커넥터는 경고를 로깅하며 이 등록 정보를 무시합니다. ws_prop_name 또는 ws_prop_type에 대한 필수 확인을 수행한 후에 사용자 정의 등록 정보의 값이 설정되거나 검색되지 않으면 커넥터는 오류를 로깅하며 이벤트가 실패합니다.

UserDefinedProperties 속성이 지정되어 있으며 자체 Business Object가 인스턴스화되어 있는 경우, 커넥터는 이 하위 Business Object의 각 속성을 처리하며 이에 따라 메시지 등록 정보 값을 설정합니다.

동기 요청 처리의 경우, 웹 서비스/URL에서 응답 메시지를 수신하는 시점에 UserDefinedProperties 속성이 지정되어 있으면 커넥터는 UserDefinedProperties Business

Object의 인스턴스를 작성하며 메시지에서 등록 정보 값을 추출하려고 시도한 후에 이를 새 Business Object에 저장합니다. 최소한 하나의 등록 정보 값이 정상적으로 검색되면 커넥터는 프로토콜 구성 MO에 수정된 UserDefinedProperties Business Object를 설정합니다.

메시지 변환 맵: MTM(Message Transformation Map) 기능은 요청 처리 HTTP(S) 프로토콜 핸들러 전용으로 지원됩니다. MessageTransformationMap은 Business Object를 지시하는 프로토콜 구성 MO의 선택적 속성입니다. Business Object에는 규칙에 지정된 MIME 유형 및 charset과 함께 메시지 변환을 위한 규칙이 들어 있습니다. (대소문자가 구별되는) 속성 이름 MessageTransformationMap을 찾고 이 속성이 Business Object 유형(그림 19 참조)인 경우, 커넥터는 해당 오브젝트의 규칙을 사용하여 메시지를 변환합니다.

그림 19에 표시된 대로 MTM 속성에는 하나의 카디널리티 N 하위 Business Object 속성(이름은 TransformationRule임)이 있어야 합니다. 메시지에 대해 TransformationRule을 찾으려고 하는 경우, SOAP/HTTP(s) 프로토콜 핸들러는 먼저 모든 TransformationRules에 지정된 ContentType에 의해 메시지가 정확히 일치하도록 시도합니다. 실패하는 경우, 커넥터는 메시지의 다중 유형에 적용되는 규칙을 찾으려고 시도합니다. 프로토콜 핸들러 처리에 대한 자세한 정보는 82 페이지의 『SOAP/HTTP-HTTPS 프로토콜 핸들러 처리』의 내용을 참조하십시오.

TransformationRule Business Object의 각 인스턴스에는 표 23에 표시된 대로 지정된 속성이 있어야 합니다.

표 23. HTTP 프로토콜 구성 MO의 MessageTransformationMaps에 대한 TransformationRule 속성

속성 이름	필수	유형	기본값	설명
TransformationRule	아니오	Business Object, 카디널리티 N		메시지 변환에 대해 하나의 규칙을 유지하는 속성입니다. MessageTransformationMap 속성 아래에 이 속성의 영(0) 또는 그 이상의 인스턴스가 있을 수 있습니다.
+ContentType	예	String	*/*	이 등록 정보의 값은 이 변환 규칙이 적용되는 메시지의 HTTP ContentType을 지정합니다. 이 속성에 대한 기본값 */*을 사용하여 커넥터는 임의의 ContentType에 이 규칙을 적용할 수 있습니다. 프로토콜 핸들러 처리에 대한 자세한 정보는 82 페이지의 『SOAP/HTTP-HTTPS 프로토콜 핸들러 처리』의 내용을 참조하십시오. 참고로 프로토콜 핸들러가 다른 규칙과 동일한 ContentType을 지닌 둘 이상의 규칙을 찾으면 프로토콜 핸들러는 경고를 로깅하고 모든 중복 규칙을 무시하지만 고유 규칙을 사용합니다.

표 23. HTTP 프로토콜 구성 MO의 MessageTransformationMaps에 대한 TransformationRule 속성 (계속)

속성 이름	필수	유형	기본값	설명
+MimeType	아니오			이 Business Object에 지정된 ContentType의 메시지를 처리하는 중에 Data Handler를 호출할 때 사용할 MIME 유형입니다.
+Charset	아니오			이 Business Object에 지정된 ContentType의 요청을 변환할 때 사용할 charset입니다.

요청 처리를 위한 HTTP 신임 전파: 신임 전파의 목적을 위해, 커넥터는 HTTP 프로토콜 구성 MO에서 Authorization_UserId 및 Authorization_Password 속성을 지원합니다. HTTP 기본 인증 설계의 일부로서 이 신임의 전파만이 지원됩니다.

요청 처리 중 신임 전파를 원하는 경우, Authorization_UserId 및 Authorization_Password 속성을 WSDL ODA에서 생성한 프로토콜 구성 MO에 수동으로 추가해야 합니다. Business Object 및 Meta Object 정의를 생성한 후에 Business Object Designer에서 이를 수행합니다(WSDL ODA에 대한 자세한 정보는 157 페이지의 제 6 장 『요청 처리용 협업 사용 기능』을 참조하십시오).

협업은 프로토콜 구성 MO에서 Authorization_UserId 및 Authorization_Password 속성 값을 설정합니다. 이 속성이 널(null)이 아니고 비어 있지 않은 경우, 커넥터는 대상 웹 서비스로 송신하는 요청에 대한 권한 머리글을 작성합니다. SOAP HTTP/HTTPS 프로토콜 핸들러는 권한 부여 머리글 작성 시 *HTTP Authentication: Basic and Digest Access Authentication(RFC 2617)*을 따릅니다.

주: 다이제스트 인증 설계는 지원되지 않으며 Rfc2617에 정의된 HTTP 인증에 대한 선택적 시도-응답(challenge-response) 메커니즘도 지원되지 않습니다. HTTP(s) 프로토콜 핸들러가 신임을 필요로 하는 서버를 호출하는 경우, 커넥터는 서버에서의 시도 응답을 대기하지 않습니다. 대신에 계속해서 신임을 전송합니다.

비동기 요청 처리 TLO

그림 20에서는 비동기 요청 처리용 Business Object 계층 구조를 보여줍니다. 요청 오브젝트만 필수이며 이 오브젝트에는 SOAP Data Handler용 SOAP 구성 MO와 두 가지 프로토콜 구성 MO(SOAP/JMS 및 SOAP/HTTP/HTTPS 프로토콜 핸들러용)가 있습니다. 다음 섹션에 설명되어 있습니다.

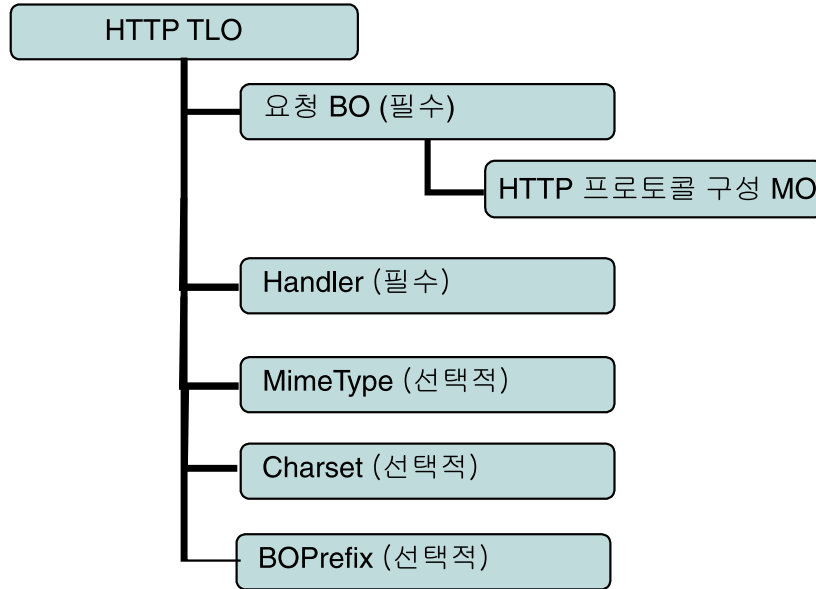


그림 20. 비동기 요청 처리용 Business Object 계층 구조

TLO에는 속성 레벨 ASI가 있는 속성뿐만 아니라 오브젝트 레벨 ASI를 포함합니다. 두 종류의 ASI 모두가 아래에 설명되어 있습니다. 머리글 컨테이너 및 머리글 하위 Business Object에 대한 정보는 38 페이지의 『머리글 컨테이너 Business Object』를 참조하십시오.

비동기 이벤트 처리 TLO에 대한 오브젝트 레벨 ASI

그림 21에서는 비동기 요청 처리를 위한 샘플 TLO인 CLIENT_ASYNC_Order_TLO를 보여줍니다.

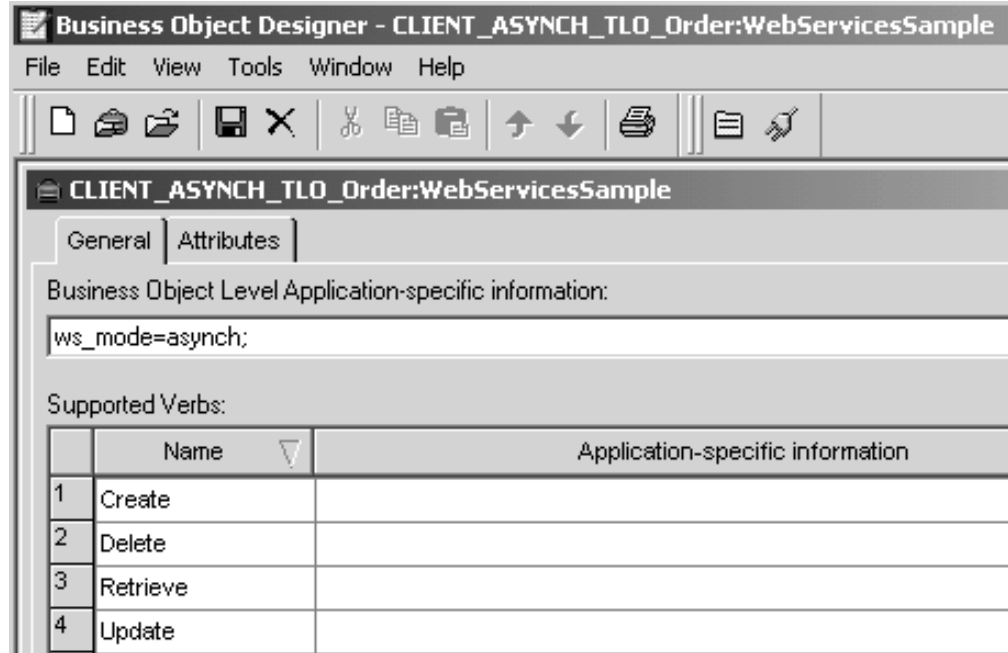


그림 21. 비동기 요청 처리용 최상위 레벨 Business Object

아래 표 24는 비동기 요청 처리 TLO에 대한 오브젝트 레벨 ASI에 대해 설명합니다.

표 24. 비동기 요청 처리 TLO 오브젝트 ASI

오브젝트 레벨 ASI	설명
ws_mode=asynch	요청 처리 중에 커넥터가 이 ASI 등록 정보를 사용하여 협업을 동기식으로(synch) 또는 비동기식으로(asynch) 호출할지 여부를 판별합니다. 비동기 요청 처리의 경우, 이 ASI는 asynch로 설정되어야 합니다. 기본값은 asynch입니다.

비동기 요청 처리 TLO에 대한 속성 레벨 ASI

그림 22에서는 샘플 요청 처리 TLO인 CLIENT_ASYNC_TLO_Order의 속성을 보여줍니다.

	Pos	Name	Type	Key	Required	Card	Maximum Length	Default	App Spec Info
1	1	Handler	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		255		soap/http
2	2	MimeType	String	<input type="checkbox"/>	<input type="checkbox"/>		255		xml/soap
3	3	BOPrefix	String	<input type="checkbox"/>	<input type="checkbox"/>		255		
4	4	Request	CLIENT_ASYNC_TLO_Order	<input type="checkbox"/>	<input type="checkbox"/>	1			ws_botype=request
5	5	ObjectEventId	String						

그림 22. 비동기 요청 처리용 TLO 속성

표 25는 비동기 요청 처리 TLO의 요청 속성에 대한 속성 레벨 ASI를 요약합니다.

표 25. 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 처리에만 사용됩니다. (이벤트 처리의 경우, 프로토콜 리스너는 SOAPDHMimeType 커넥터 특정 구성 등록 정보를 사용합니다.) 기본값은 xml/soap입니다.
BOPrefix	없음	String 유형의 이 속성은 차후 개발될 예정이므로 필요하지 않습니다.
Handler	없음	이 속성은 웹 서비스 요청을 처리하는 데 사용할 프로토콜 핸들러를 지정하며 요청 처리용으로만 사용됩니다. 다음 값 중 하나를 사용합니다. <ul style="list-style-type: none"> soap/jms 커넥터는 SOAP/JMS 프로토콜 핸들러를 사용하여 요청을 처리합니다. soap/http 커넥터는 SOAP/HTTP-HTTPS 프로토콜 핸들러를 사용하여 이 웹 서비스 요청을 처리합니다. 기본값은 soap/http입니다.
Request	ws_botype=request	이 속성은 웹 서비스 요청 Business Object에 해당합니다. 커넥터는 이 속성 ASI를 사용하여 TLO 속성이 SOAP 요청 BO 유형인지 판별합니다. 속성 이름이 아닌 이 ASI가 속성 유형을 판별합니다. 둘 이상의 요청 속성이 있는 경우, 커넥터는 첫 번째 속성의 ASI를 사용합니다.

비동기 요청 처리를 위한 요청 Business Object

요청 Business Object는 TLO의 하위이며 비동기 요청 처리를 위해 필수입니다. 비동기 요청 처리용 요청 Business Object에 대한 오브젝트 레벨 ASI가 표 26에 설명되어 있습니다.

표 26. 비동기 요청 처리: 요청 Business Object에 대한 오브젝트 레벨 ASI

오브젝트 레벨 ASI	설명
<code>cw_mo_soap=SOAPCfgMO</code>	이 ASI 값은 SOAP 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 요청 Business Object에 대한 Data Handler 변환을 정의하는 SOAP 구성 MO입니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』를 참조하십시오.
<code>cw_mo_jms=SOAPJMSCfgMO</code>	이 ASI 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이는 JMS 프로토콜 핸들러의 목적지 웹 서비스를 지정하는 프로토콜 구성 MO입니다. 자세한 정보는 51 페이지의 『요청 처리용 요청 Business Object의 JMS 프로토콜 구성 MO』를 참조하십시오.
<code>cw_mo_http=SOAPHTTFCfgMO</code>	이 ASI 값은 프로토콜 구성 MO에 해당하는 속성의 이름과 일치해야 합니다. 이것은 SOAP/HTTP-HTTPS 프로토콜 핸들러에 대한 목적지를 지정하는 별도의 프로토콜 구성 MO입니다. 이 ASI는 SOAP/HTTP-HTTPS 프로토콜 핸들러가 사용됩니다. TLO 요청 속성에는 요청 처리용 JMS 및 HTTP 프로토콜 구성 MO가 모두 있어야 합니다. 자세한 정보는 51 페이지의 『요청 처리용 HTTP 프로토콜 구성 MO』를 참조하십시오.
<code>SOAPAction=SOAPActionURI</code>	커넥터는 이 ASI를 사용하여 요청 메시지에서 SOAPAction 머리글을 설정할지 여부를 판별합니다. 대상 웹 서비스에 SOAPAction 머리글이 필요한 경우에만 이 ASI를 지정하십시오. 이 ASI는 요청 처리에 사용되지만 이벤트 공고에는 사용되지 않습니다.

그림 14에 표시된 샘플에서, 요청 속성에는 SOAP 구성 MO와 머리글 컨테이너 (OrderHeader)는 물론 내용 관련 속성(OrderLineItems)이 있습니다. SOAP 구성 MO, 프로토콜 구성 MO, SOAP 머리글 컨테이너 및 머리글 하위 Business Object의 요구 사항과 특성은 동기 요청 처리에서와 같이 비동기 요청 처리에 대해서도 동일합니다. 자세한 정보는 44 페이지의 『동기 요청 처리 TLO』의 주제를 참조하십시오.

	Pos	Name	Type	Key	Required	Card	Maximum	App Spec Info
1	1	Request	SERVICE_ASYNC_Order	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1		ws_botype=request
1.1	1.1	OrderId	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		255	
1.2	1.2	OrderDate	Date	<input type="checkbox"/>	<input type="checkbox"/>			
1.3	1.3	CustomerId	String	<input type="checkbox"/>	<input type="checkbox"/>		255	
1.4	1.4	OrderLineItems	SERVICE_ASYNC_Order_LineItem	<input type="checkbox"/>	<input type="checkbox"/>	N		type_name=Order_LineItem
1.5	1.5	OrderHeader	SERVICE_ASYNC_Order_Header	<input type="checkbox"/>	<input type="checkbox"/>	1		soap_location=SOAPHeader
1.6	1.6	SOAPCfgMO	SERVICE_ASYNC_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.7	1.7	SOAPJMScfgMO	SERVICE_ASYNC_SOAP_JMS_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.8	1.8	ObjectEventId	String					
2	2	ObjectEventId	String					
3	3			<input type="checkbox"/>	<input type="checkbox"/>		255	

그림 23. 비동기 이벤트 처리용 요청 속성

비동기 요청 처리용 구성 MO

SOAP 구성 MO(SOAPCfgMO)의 속성은 이벤트 처리 SOAP 구성 MO의 속성과 동일합니다. 자세한 정보는 32 페이지의 『SOAP 구성 MO』를 참조하십시오. 121 페이지의 『SOAP 구성 Meta Object: 모든 SOAP Business Object의 하위 오브젝트』도 참조하십시오.

JMS 프로토콜 구성 MO는 JMS 웹 서비스 사용 시 요청 Business Object에서 필수입니다. 자세한 정보는 51 페이지의 『요청 처리용 요청 Business Object의 JMS 프로토콜 구성 MO』를 참조하십시오.

요청 처리 중에, SOAP/HTTP-HTTPS 프로토콜 핸들러가 HTTP 프로토콜 구성 MO를 사용하여 대상 웹 서비스의 목적지를 판별합니다. 이 프로토콜 구성 MO는 요청 Business Object에 필수입니다. 자세한 정보는 51 페이지의 『요청 처리용 HTTP 프로토콜 구성 MO』를 참조하십시오.

Business Object 개발

Business Object Designer를 사용하여 Business Object를 작성하고 Connector Configurator를 사용하여 해당 Business Object를 지원하도록 커넥터를 구성합니다. Business Object Designer 도구에 대한 자세한 정보는 *Business Object 개발 안내서* 및 159 페이지의 제 7 장 『협업을 웹 서비스로 표시』를 참조하십시오. Connector Configurator에 대한 자세한 정보는 203 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

제 4 장 웹 서비스 커넥터

- 64 페이지의 『커넥터 처리』
- 66 페이지의 『SOAP/HTTP(S) 웹 서비스』
- 67 페이지의 『SOAP/JMS 웹 서비스』
- 68 페이지의 『이벤트 처리』
- 80 페이지의 『요청 처리』
- 91 페이지의 『SSL』
- 89 페이지의 『커넥터 및 JMS』
- 94 페이지의 『커넥터 구성』
- 115 페이지의 『시작 시의 커넥터』
- 116 페이지의 『로깅』
- 116 페이지의 『추적』

이 장은 웹 서비스 커넥터 및 구성 방법에 대해 설명합니다.

모든 WebSphere Business Integration 커넥터는 통합 브로커와 함께 작동합니다. 웹 서비스 커넥터는 IBM WebSphere InterChange Server 통합 브로커와 함께 작동하는데, 이 통합 브로커는 *Technical Introduction to IBM WebSphere InterChange Server*에 설명되어 있습니다.

커넥터는 어댑터의 런타임 구성요소입니다. 커넥터는 응용프로그램 특정 구성요소와 커넥터 프레임워크로 구성됩니다. 응용프로그램 특정 구성요소에는 특정 응용프로그램에 알맞은 코드가 들어 있습니다. 해당 코드가 모든 커넥터에 공통인 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소의 매개체로서 작용합니다. 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이에 다음 서비스를 제공합니다.

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

이 문서에는 응용프로그램 특정 구성요소와 커넥터 프레임워크에 대한 정보가 있습니다. 문서에서는 이 구성요소 모두를 커넥터라고 부릅니다.

커넥터에 대한 통합 브로커의 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 기능을 사용하여 커넥터 프레임워크는 다음을 수행할 수 있습니다.

- 협업을 웹 서비스로 표시한 후 웹 서비스 클라이언트에서 호출 처리
- 웹 서비스를 호출하는 협업에 의해 요청 처리

SOAP Data Handler에 대한 자세한 정보는 119 페이지의 제 5 장 『SOAP Data Handler』를 참조하십시오.

주: 커넥터는 SOAP/HTTP 및 SOAP/JMS 바인딩만을 지원합니다.

이벤트 처리 개요

커넥터 이벤트 처리(또는 이벤트 공고)는 웹 서비스 클라이언트의 요청을 처리하는 데 사용됩니다. 이 이벤트 처리 성능은 이 장의 후반부에서 자세히 설명되는 다음 구성요소를 포함하여 프로토콜 리스너 프레임워크를 포함합니다.

- SOAP/HTTP 프로토콜 리스너
- SOAP/HTTPS 프로토콜 리스너
- SOAP/JMS 프로토콜 리스너

커넥터는 리스너를 사용하여 협업을 웹 서비스로 표시하고 웹 서비스에서 표시된 협업으로의 호출에 대한 전송을 청구합니다.

SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너는 협업을 SOAP/HTTP 웹 서비스로 표시합니다. SOAP/JMS 프로토콜 리스너는 협업을 SOAP/JMS 웹 서비스로 표시합니다.

웹 서비스 클라이언트로부터 요청이 도착할 때, 리스너는 SOAP 요청 메시지를 Business Object로 변환하고 협업을 호출합니다. 동기 요청인 경우, 커넥터는 요청 비즈니스와 동일한 유형의 응답 Business Object를 수신합니다. 리스너는 요청 Business Object를 SOAP 응답 메시지로 변환합니다. 그런 다음, 리스너는 SOAP 응답 메시지를 웹 서비스 클라이언트로 전송합니다. 이 커넥터에서는 이벤트 순서 지정이 요구사항이 아닙니다. 커넥터는 임의의 순서로 이벤트를 전달할 수 있습니다.

웹 서비스 커넥터는 SOAP Data Handler를 이용하여 수신 SOAP 요청 메시지를 Business Object로 변환합니다. Data Handler가 수신 중인 SOAP 요청 메시지를 해석할 Business Object를 판별하도록 돕기 위해, 커넥터는 Data Handler에 지원되는 Business Object에 관한 메타 데이터 정보를 제공합니다. 커넥터는 먼저 지원되는 Business Object로부터의 변환을 위한 잠재적인 후보인 모든 Business Object의 목록을 만듭니다. 이 목록은 TLO와 비TLO 모두로 구성될 수 있습니다. 지원되는 TLO Business Object는 오브젝트 레벨 ASI ws_eventtlo=true로 되어 있는 것입니다.

TLO가 사용되는 경우, 프로토콜 리스너가 다음과 같이 TLO의 오브젝트 레벨 ASI를 읽습니다.

- ws_collab= 이것은 호출할 협업을 판별합니다.
- ws_mode= 이것은 동기식(synch) 또는 비동기식(asynch)으로 협업을 호출하는 방법을 판별합니다.

비TLO가 사용되는 경우, 프로토콜 리스너는 WSDL 구성 마법사가 생성하는 WSCollaborations 구성 등록 정보 값에서 협업 및 처리 모드를 읽습니다.

커넥터는 비교하고 SOAP 요청의 BodyName 및 BodyNamespace가 잠재적인 Business Object의 이름에 일치하도록 시도합니다. TLO의 경우, 이 BodyName/BodyNamespace 쌍은 SOAP 요청 Business Object의 SOAP 구성 MO 등록 정보를 사용하여 발견됩니다. 비TLO의 경우, BodyName/BodyNamespace 쌍은 WSCollaborations 커넥터 구성 등록 정보를 사용하여 발견됩니다(커넥터는 WSCollaborations 등록 정보에 항목이 있는, TLO 이외의 것만 고려합니다). Data Handler가 BodyName/BodyNamespace 쌍을 사용하여 SOAP 요청 대 Business Object 변환에 사용할 Business Object를 판별합니다.

커넥터는 SOAP Data Handler에서 리턴한 요청 Business Object를 검사합니다. 이 Business Object에 ws_tloname ASI가 있으면, 커넥터가 이 TLO에 요청 Business Object를 설정합니다. 이 TLO는 협업을 호출하는 데 사용됩니다. 그러나, 이 ASI가 설정되지 않으면 커넥터는 SOAP Data Handler에서 리턴한 요청 Business Object를 사용하여 협업을 호출합니다.

동기 협업 실행의 경우, 커넥터는 SOAP Data Handler를 이용하여 클라이언트로 다시 전송할 SOAP 응답 또는 결합 메시지를 작성합니다. 이 경우, 커넥터는 단순히 SOAP Business Object(TLO의 하위) 또는 비TLO를 Data Handler에 전달합니다. SOAP Data Handler는 전달되는 Business Object를 기초로 SOAP 메시지를 리턴합니다.

요청 처리 개요

협업 대신, 커넥터는 SOAP/HTTP(S) 및 SOAP/JMS에서 웹 서비스를 호출할 수 있습니다. 이 요청 처리 기능은 WSDL ODA(Object Discovery Agent) 및 프로토콜 핸들러 프레임워크에서 지원됩니다. WSDL ODA는 대상 웹 서비스에 대한 정보를 포함하는 SOAP Business Object를 생성하기 위해 사용하는 설계 도구입니다. 자세한 정보는 157 페이지의 제 6 장 『요청 처리용 협업 사용 가능』을 참조하십시오. 프로토콜 핸들러 프레임워크는 다음 구성요소로 구성되는 구성 가능한 런타임 모듈로서, 이 장의 뒤에서 자세히 설명됩니다.

- SOAP/HTTP-HTTPS 프로토콜 핸들러
- SOAP/JMS 프로토콜 핸들러

TLO에 항상(WSDL ODA를 통해) 설정되는 협업 요청 Business Object를 수신할 때, 프로토콜 핸들러 프레임워크가 적당한 프로토콜 핸들러를 로드합니다. 프로토콜 핸들러는 웹 서비스 호출 및 (선택적으로) 응답 보호에 필요한 전송 레벨 세부사항을 관리하며 세 가지의 주요 작업을 수행합니다. 협업 요청 Business Object를 SOAP 요청 메시지로 변환하고 요청 메시지로 엔드포인트 웹 서비스를 호출하며 요청/응답(동기) 모드에 있는 경우, SOAP 응답 메시지를 Business Object로 변환하고 해당 오브젝트를 협업으로 리턴합니다. 커넥터는 SOAP/HTTP-HTTPS 프로토콜 핸들러를 사용하여 SOAP/HTTP(S) 웹 서비스를 호출하고 SOAP/JMS 프로토콜 핸들러를 사용하여 SOAP/JMS 웹 서비스를 호출합니다.

웹 서비스 커넥터는 항상 TLO를 사용하는 협업에서 호출됩니다. 커넥터는 TLO에서 SOAP 요청 Business Object를 판별하고, 이 Business Object로 SOAP Data Handler를 호출합니다. Data Handler는 커넥터가 웹 서비스에 전송하는 요청 메시지를 리턴합니다.

동기 웹 서비스 실행의 경우, 커넥터는 SOAP Data Handler를 이용하여 SOAP 응답 및 결합 메시지를 SOAP 응답 및 결합 Business Object로 변환합니다. 커넥터를 Data Handler가 이러한 SOAP 응답/결합 메시지를 Business Object로 변환하기 위해 해석할 Business Object를 판별하는 데 도움이 되도록 Data Handler에 특정 메타 정보를 제공합니다. 특별히, 커넥터는 호출하는 TLO의 하위인 모든 응답 및 결합 Business Object의 목록을 작성합니다. 응답 Business Object는 하나만 있어야 하며 선택적으로 여러 결합 Business Object가 있을 수 있습니다. 또한 단 하나의 default Business Object가 있을 수도 있습니다. 커넥터가 일치하도록 시도한 후, SOAP BodyName 및 BodyNamespace가 모든 응답 Business Object의 목록에 표시되는 Business Object 이름으로 맵핑시킵니다. SOAP 응답 Business Object의 경우, 이 쌍은 SOAP 응답 Business Object의 SOAP 구성 MO 등록 정보를 사용하여 발견됩니다. SOAP 결합 Business Object의 경우, 이 쌍은 세부사항 요소의 첫 번째 하위에 대한 elem_name 및 elem_ns 속성 레벨 ASI 등록 정보를 사용하여 발견됩니다. default Business Object의 경우, 커넥터는 단순히 Data Handler에 default Business Object의 이름을 알립니다. 이 변환에서 다른 결합 Business Object가 해석되지 않는 경우, 최후의 수단으로 Data Handler가 default Business Object를 해석해야 합니다.

SOAP/HTTP(S) 웹 서비스

웹 서비스는 HTTP 전송 프로토콜을 지원합니다. HTTP는 HTTP 클라이언트가 연결을 열고 HTTP 서버에 요청 메시지를 전송하는 클라이언트-서버 모델을 구현합니다. 클라이언트 요청 메시지는 웹 서비스를 호출하는 것입니다. HTTP 서버는 호출을 포함하는 메시지를 디스패치하고 연결을 닫습니다.

웹 서비스로 표시된 협업에 대한 클라이언트 요청을 처리할 때 커넥터의 SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너는 HTTP 클라이언트-서버 및 요청/응답 모델을 사

용합니다. 그러나, SOAP/HTTP 리스너는 HTTP 서버(프록시, 중간 또는 그 반대)로서 기능하기 위한 것이 아닙니다. 오히려 SOAP/HTTP 리스너는 엔터프라이즈 내에서 및 방화벽 뒤에서 사용하는 엔드포인트로서 기능합니다. 따라서, 클라이언트 요청을 리스너에 라우트하기 위해 방화벽에 별도의 웹 서버 또는 게이트웨이가 전개되어야 합니다. 자세한 정보는 1 페이지의 제 1 장 『커넥터의 개요』를 참조하십시오.

SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너는 협업을 SOAP/HTTP(S) 웹 서비스로 표시합니다. 커넥터가 SOAP/HTTP-HTTPS 프로토콜 핸들러를 사용하여 SOAP/HTTP(S) 웹 서비스를 호출합니다.

동기 SOAP/HTTP(S) 웹 서비스

커넥터 처리의 관점에서, 동기 HTTP 웹 서비스는 요청/응답 경로를 따르는 웹 서비스입니다. SOAP/HTTP 또는 SOAP/HTTPS 프로토콜 리스너가 HTTP 요청 메시지를 성공적으로 처리하는 경우, 본문은 웹 서비스 응답과 HTTP 상태 코드 200 OK를 포함합니다. 결함이 리턴되는 경우, 본문은 결함 메시지와 상태 코드 500을 포함합니다.

비동기 SOAP/HTTP(S) 웹 서비스

커넥터 처리의 관점에서, 비동기 HTTP 웹 서비스는 요청 전용 경로를 따르는 웹 서비스입니다. SOAP/HTTP 또는 SOAP/HTTPS 프로토콜 리스너가 요청 전용 웹 서비스 조작을 성공적으로 수신하고 처리하는 경우, HTTP 상태 코드 202 Accepted가 생성됩니다. 200 OK의 HTTP 상태 코드를 생성하도록 커넥터를 구성할 수도 있습니다. 자세한 정보는 표 40의 HTTPAsyncResponseCode 등록 정보를 참조하십시오. 결함이 발생하면 HTTP 상태 코드 500이 생성됩니다. 잘못된 본문이 리턴될 수 있지만, 응답이 없습니다.

SOAP/JMS 웹 서비스

JMS는 엔터프라이즈가 메시징, 데이터 지속성 및 Java 기반 응용프로그램에 대한 액세스를 위해 웹 서비스 솔루션과 함께 결합할 수 있는 전송 레벨 API입니다. SOAP/JMS 웹 서비스는 JMS 대기열 기반 전송을 구현하는 웹 서비스입니다.

웹 서비스 솔루션이 대기열 또는 주제에 대한 JMS 목적지를 구현할 수 있습니다. 커넥터의 SOAP/JMS 프로토콜 리스너는 대기열 목적지만을 지원합니다. 주제는 지원되지 않습니다. JMS 텍스트 메시지만이 지원됩니다.

이벤트 처리 중에, SOAP/JMS 웹 서비스 클라이언트가 요청 메시지를 JMS 메시지로 래핑하고 이를 JMS 목적지가 커넥터인 대기열에 공개합니다. JMS 목적지는 웹 서비스 요청을 포함하는 JMS 메시지를 검색하고 JMS 메시지에서 SOAP 요청 메시지를 추출합니다. 그런 다음, SOAP 요청 메시지를 처리합니다.

동기 SOAP/JMS 웹 서비스

동기 커넥터 처리(요청/응답)의 경우, 응답 메시지가 JMS 메시지(요청 메시지의 JMS 메시지 같은)로 래핑됩니다. 그런 다음, 웹 서비스 응답을 포함하는 JMS 메시지가 수신 요청으로부터 JMSReplyTo 대기열로 전송됩니다. 응답 메시지의 JMS 머리글은 다음과 같이 JMS 요청 메시지에 있는 머리글의 값으로 설정됩니다.

- 응답 메시지의 JMSCorrelationID는 JMS 요청 메시지의 JMSMessageID 값으로 설정되어야 합니다.
- 응답 메시지의 JMS DeliveryMode는 요청의 JMSDeliveryMode로 설정됩니다.
- 응답 메시지의 JMSPriority는 요청의 JMSPriority로 설정됩니다.
- 요청 메시지의 JMSExpiration은 요청의 JMSExpiration으로 설정됩니다.

이 처리는 74 페이지의 『SOAP/JMS 프로토콜 리스너 처리』에서 자세히 설명됩니다.

비동기 SOAP/JMS 웹 서비스

커넥터 처리의 관점에서, 비동기 SOAP/JMS 웹 서비스는 요청 전용 경로를 따르는 웹 서비스입니다. SOAP/JMS 프로토콜 리스너가 성공적으로 요청 전용 웹 서비스 메시지를 수신하고 처리하는 경우, 응답을 포함하는 JMS 메시지가 클라이언트로 리턴되지 않습니다. ReplyToQueue가 구성되고 JMS 메시지의 수령 시 결합이 발생하면, 결합 메시지가 웹 서비스 클라이언트로 리턴됩니다. 그리고 ErrorQueue가 SOAP/JMS 리스너에 지정되면, 결합 메시지가 거기에서 아카이브됩니다.

이벤트 처리

이벤트 처리 성능 구현의 첫 번째 단계는 비즈니스 프로세스(협업)을 웹 서비스로 표시하는 것입니다. 그런 다음, 이 웹 서비스를 예를 들어 UDDI 레지스트리에 공개하고 협업을 호출하는 웹 서비스 클라이언트에 응답하도록 커넥터를 구성합니다.

이벤트 처리 중에, 커넥터는 프로토콜 리스너와 SOAP Data Handler를 사용하여 웹 서비스 클라이언트의 SOAP 요청 메시지를 웹 서비스로 표시된 협업이 조작할 수 있는 Business Object로 변환합니다. 프로토콜 리스너가 이벤트 처리에서 중요한 역할을 합니다.

프로토콜 리스너

웹 서비스 요청은 HTTP, HTTPS 및 JMS를 포함하여 다양한 전송으로 이루어집니다. 웹 서비스 프로토콜 리스너는 전송 채널에서 이러한 요청의 도달 여부를 모니터링합니다. 다음과 같은 세 가지의 프로토콜 리스너와 해당 채널이 있습니다.

- SOAP/HTTP 프로토콜 리스너
- SOAP/HTTPS 프로토콜 리스너
- SOAP/JMS 프로토콜 리스너

각 리스너는 전송을 청취하는 스레드로 구성됩니다. 클라이언트로부터 SOAP 요청 메시지를 수신하면 리스너는 프로토콜 리스너 프레임워크에 이벤트를 등록합니다.

프로토콜 리스너 프레임워크는 프로토콜 리스너를 관리하여 자원을 사용 가능할 때 요청을 스케줄할 수 있습니다. 커넥터 특정 등록 정보에 값을 설정할 때 리스너 및 프로토콜 리스너 프레임워크의 항목을 구성합니다. 프로토콜 리스너 프레임워크 등록 정보 중에 다음을 구성할 수 있습니다.

- **WorkerThreadCount** 프로토콜 리스너 프레임워크에서 사용할
- **RequestPoolSize** 프로토콜 리스너 프레임워크에 등록할 최대 요청 수. 이 최대 요청 수보다 많은 요청을 수신하면 새 요청을 더 이상 등록하지 않습니다.

위의 두 가지 커넥터 특정 등록 정보는 무한 웹 서비스 이벤트를 사용하여 프로토콜 리스너가 커넥터 작동을 방해하지 않도록 하는 방식으로 메모리 할당을 제어합니다. 할당 알고리즘은 다음과 같습니다. 언제든지 커넥터는 $WorkerThreadCount + RequestPoolSize$ 와 동일한 총 이벤트 수를 수신할 수 있습니다. 요청의 $WorkerThreadCount$ 수를 병렬로 처리할 수 있습니다.

추가 프로토콜 리스너를 프로토콜 리스너 프레임워크에 플러그인할 수 있습니다. 자세한 정보는 114 페이지의 『복수 프로토콜 리스너 작성』 및 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너 처리

SOAP/HTTP(S) 프로토콜 리스너는 웹 서비스 클라이언트의 HTTP(S) 요청을 지속적으로 청취하는 스레드로 구성됩니다. 리스너 스레드는 호스트 및 포트 커넥터 특정 구성(리스너) 등록 정보에 지정되는 호스트 및 포트에 바인드합니다. 다른 구성 등록 정보(RequestWaitTimeout)는 커넥터가 종료했는지 여부를 확인하기 전에 리스너가 요청을 기다리는 간격을 정의합니다.

그림 24는 동기 운영에 대한 SOAP/HTTP 프로토콜 리스너 처리를 보여줍니다.

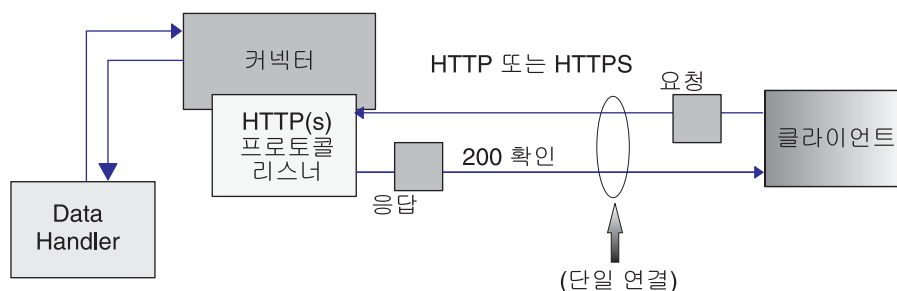


그림 24. SOAP/HTTP 프로토콜 리스너: 동기 이벤트 처리

그림 25는 비동기 운영에 대한 SOAP/HTTP 프로토콜 리스너 처리를 보여줍니다.

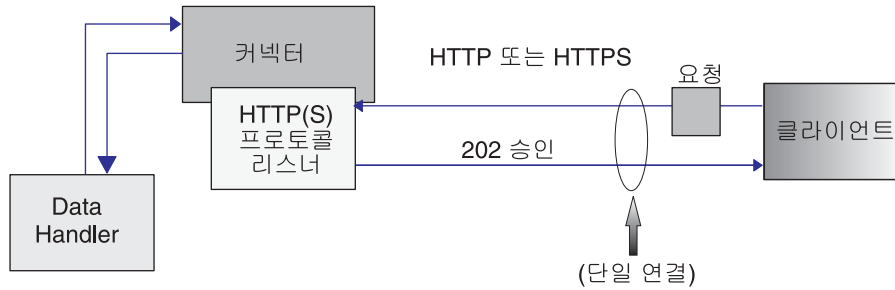


그림 25. SOAP/HTTP 프로토콜 리스너: 비동기 이벤트 처리

웹 서비스 클라이언트는 SOAP/HTTP 또는 SOAP/HTTPS 요청을 시작할 때 SOAP 요청 메시지를 SOAP/HTTP 또는 SOAP/HTTPS 리스너의 URL로 보냅니다. 클라이언트는 프로토콜 리스너 URL을 호출하기 위해 HTTP POST 메소드를 사용해야 합니다.

HTTP(S) 요청이 도착할 때, 리스너는 해당 요청을 프로토콜 리스너 프레임워크에 등록하며, 이 프레임워크는 자원을 사용 가능할 때 이벤트를 처리하도록 스케줄합니다. 그런 다음, 리스너가 요청에서 프로토콜 머리글과 페이로드를 추출합니다.

표 27에는 인바운드 메시지의 Charset, MimeType, ContentType 및 Content-Type 머리글을 판별하기 위해 리스너가 사용하는 규칙 우선순위가 요약되어 있습니다.

표 27. 인바운드 메시지에 대한 SOAP/HTTP(s) 프로토콜 리스너 처리 규칙

우선 순위	Charset	MimeType	ContentType	Content-Type 머리글
1	수신 중인 HTTP 메시지 Content-Type 머리글 값에 있는 Charset 매개변수 값	이 리스너에 대한 URLsConfiguration 커넥터 등록 정보 값	Content-Type 머리글 값으로부터 수신하는 HTTP 메시지 유형/부속 유형 값	수신 중인 HTTP 메시지 Content-Type 머리글
2	이 리스너에 대한 URLsConfiguration 등록 정보 값	SOAPDHMimeType 커넥터 등록 정보 값		
3	요청 메시지 ContentType의 유형이 부속 유형을 가진 text이면(예: text/xml, text/plain 등), 기본값은 ISO-8859-1이 됩니다. 그렇지 않으면 Charset이 사용되지 않습니다.	기본값은 ContentType		

표 27에 표시된 대로:

- 프로토콜 리스너는 다음 규칙에 따라 인바운드 메시지의 Charset을 판별합니다.

1. 리스너는 HTTP 메시지 Content-Type 머리글 값의 charset 매개변수로부터 Charset을 추출하려고 시도합니다.
 2. Content-Type 머리글로부터 Charset 값을 가져올 수 없으면 프로토콜 리스너는 이 리스너의 URLsConfiguration 등록 정보 값을 읽으려고 시도합니다.
 3. 이전 단계에 설명된 방법을 사용하여 Charset 값을 가져올 수 없을 때 메시지 ContentType 유형이 부속 유형을 가진 text이면(예: text/xml, text/plain 등), 리스너는 기본 Charset 값인 ISO-8859-1을 사용합니다. 그렇지 않으면 Charset 값이 사용되지 않습니다.
- 리스너는 다음의 규칙에 따라 응답 메시지에 대한 MimeType을 판별합니다.
 1. 수신 요청 메시지가 사용하는 URL의 TransformationRules를 구성한 경우 요청 ContentType이 TransformationRule의 ContentType과 일치하면, 리스너는 TransformationRule을 사용하여 요청 메시지를 SOAP 요청 business object로 변환하는 데 필요한 MimeType을 추출합니다. 리스너는 요청된 URL의 URLsConfiguration 등록 정보에 있는 ContentType 값(예: text/xml)을 기본으로 정확히 일치하는 TransformationRule을 찾으려고 시도합니다.
 2. 이러한 시도가 실패하면 리스너는 요청 URL(예: */*)에 속한 둘 이상의 ContentType에 적용되는 TransformationRule을 찾으려고 시도합니다.
 3. MimeType과 일치하는 TransformationRule이 없으면 리스너는 SOAPDHMimeType 커넥터 구성 등록 정보를 MimeType 값으로 사용합니다.
 4. 이전의 모든 단계에서 MimeType을 판별하는 데 실패하면, ContentType의 값을 MimeType으로 사용하여 SOAP Data Handler를 호출한 다음 요청 메시지를 SOAP 요청 Business Object로 변환합니다.
 - 리스너는 수신 중인 HTTP 메시지 Content-Type 머리글로부터 유형/부속 유형을 추출하여 ContentType을 판별합니다.
 - 리스너는 수신 중인 HTTP 메시지 Content-Type 머리글의 Content-Type 머리글로부터 Content-Type 머리글을 판별합니다.

협업이 비동기로 호출되는 경우, 리스너는 요청 Business Object를 통합 브로커로 전달하고 HTTP 상태 코드 202 Accepted와 함께 웹 서비스 클라이언트에 응답합니다. 그러면 리스너 처리가 종결됩니다.

동기 호출인 경우, 리스너는 동기식으로 협업을 호출합니다. 협업은 SOAP 응답 Business Object에 대해 응답합니다.

표 28에는 응답 메시지의 Charset, MimeType, ContentType 및 Content-Type 머리글을 판별할 때 리스너가 사용하는 규칙의 우선순위가 요약되어 있습니다.

표 28. 아웃바운드 동기 응답 메시지에 대한 SOAP/HTTP(s) 프로토콜 리스너 처리 규칙

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 메시지 MimeType, 요청 및 응답 ContentType가 일치하는 경우에만.	요청 메시지 ContentType	ContentType 및 Charset을 사용하여 머리글 구성
3	요청 메시지 Charset, 요청 및 응답 ContentType이 일치하는 경우에만.	SOAPDHMimeType 커넥터 등록 정보 값		
4	ContentType이 text/*이면 기본값은 ISO-8859-1이 됩니다. 그렇지 않으면 Charset이 사용되지 않습니다.	ContentType 값을 MimeType으로 사용		

표 28에 표시된 대로:

- 리스너는 다음의 규칙에 따라 응답 메시지에 대한 Charset을 판별합니다.
 1. Charset이 응답 Business Object 프로토콜 구성 MO에 지정되어 있으면 해당 값이 사용됩니다.
 2. Charset 값이 응답 Business Object 프로토콜 구성 MO 머리글에 지정되어 있지 않으며 응답 및 요청 Business Object가 TLO의 하위이면 Charset이 TLO에 지정되어 있는지 여부를 리스너가 확인합니다.
 3. TLO에 Charset가 지정되어 있지 않거나 응답 Business Object가 TLO가 아닐 경우, 응답이 요청과 동일한 ContentType을 갖고 있으면 요청의 Charset가 응답에 사용됩니다.
 4. 이전 단계에서 응답 Charset 값을 판별하는 데 실패할 경우 메시지 ContentType의 유형 부분이 임의의 부속 유형을 가진 text이면(예: text/xml, text/plain 등), 리스너는 기본 Charset 값인 ISO-8859-1을 사용합니다. 그렇지 않으면, Charset 값이 사용되지 않습니다.
- 리스너는 다음의 규칙에 따라 응답 메시지에 대한 MimeType을 판별합니다.
 1. TLO의 MimeType 속성
 2. TLO MimeType 속성이 누락되었을 때 요청 및 응답의 ContentType이 일치하면, 리스너는 응답 메시지에 대한 요청 MimeType을 사용합니다.
 3. 이전 단계가 실패하면 리스너는 SOAPDHMimeType 커넥터 등록 정보의 값을 사용합니다.
 4. 그렇지 않으면 리스너는 ContentType 값을 MimeType으로 사용합니다.
- 리스너는 다음의 규칙에 따라 응답 메시지에 대한 ContentType을 판별합니다.
 1. Content-Type 머리글이 응답 Business Object 프로토콜 구성 MO에 지정되어 있으면, Content-Type 머리글의 유형/부속 유형 부분이 ContentType으로 사용됩니다.

2. Content-Type 머리글이 응답 Business Object 프로토콜 구성 MO에 지정되어 있지 않으면, 리스너는 판별된 ContentType 및 Charset(응답 메시지에 대한 Charset이 판별된 경우)을 사용하여 Content-Type 머리글을 작성합니다.

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

1. 리스너는 특수 속성(예: ObjectEventId)을 무시하기 위해 HTTP 프로토콜 구성 MO의 각 항목을 검토합니다.
2. 비어있지 않은 각 머리글은 전송 메시지에 기록되어 추가 처리(예: Content-Type 머리글)가 발생할 수 있습니다.
3. 이 방법을 사용할 경우 리스너는 표준이 아닌 머리글을 메시지에 설정할 수 있으나 메시지가 논리적으로 또는 의미상 올바른지는 점검하지 않는다는 점을 참고하십시오.
4. HTTP 프로토콜 구성 MO UserDefinedProperties 속성에 하나 이상의 사용자 정의 등록 정보가 있으면 리스너는 이를 엔티티 머리글 섹션(마지막 머리글 섹션)에 추가합니다. 사용자 정의 등록 정보에 대한 자세한 정보는 36 페이지의 『이벤트 처리를 위한 사용자 정의 등록 정보』의 내용을 참조하십시오.

주: Connection, Trailer, Transfer-Encoding, Content-Encoding, Content-Length, Content-MD5, Content-Range 머리글 중 하나를 HTTP 프로토콜 구성 MO에 지정하면 올바르지 않은 HTTP 메시지가 발생할 수 있습니다.

그런 다음, 리스너는 SOAP Data Handler를 호출하여 협업이 리턴하는 응답 Business Object를 SOAP 응답 메시지로 변환합니다.

리스너는 응답 메시지를 웹 서비스 클라이언트에 전달하고 200 OK HTTP 상태 코드를 포함합니다. 협업이 SOAP 결함 Business Object를 리턴하는 경우, 결함 메시지로 변환됩니다. 이 결함 메시지가 500 Internal Server Error HTTP 코드와 함께 웹 서비스 클라이언트에 전달됩니다.

그런 다음, 리스너가 연결을 닫고 이벤트를 처리한 스레드가 사용 가능하게 됩니다.

지원되지 않는 SOAP/HTTP 프로토콜 리스너 처리 기능

SOAP/HTTP 프로토콜 리스너는 다음을 지원하지 않습니다.

- 캐싱: 프로토콜 리스너는 HTTP 스펙(RFC2616)에 정의된 캐싱 기능을 전혀 수행하지 않습니다.
- 프록시: 프로토콜 리스너는 HTTP 스펙(RFC2616)에 정의된 프록시 기능을 전혀 수행하지 않습니다.

- 지속적 연결: 프로토콜 리스너는 HTTP 스펙(RFC2616)에 정의된 지속적 연결을 지원하지 않습니다. 대신에 프로토콜 리스너는 각 HTTP 연결의 범위가 단일 클라이언트 요청이라고 가정합니다. 그리고 서비스 요청이 완료되면 연결을 닫습니다. 프로토콜 리스너는 서비스 호출에서 연결을 다시 사용하려고 시도하지 않습니다.
- 재지정: 프로토콜 리스너는 재지정을 지원하지 않습니다.
- 대형 파일 전송: 프로토콜 리스너를 대형 파일 전송에 사용할 수 없습니다. 아니면 대신에 참조에 의해 대형 파일을 전달하는 것을 고려할 수도 있습니다.
- 상태 관리: 프로토콜 리스너는 RFC2965에 기술된 HTTP 상태 관리 메커니즘을 지원하지 않습니다.
- 쿠키: 프로토콜 리스너는 쿠키를 지원하지 않습니다.

보안 소켓을 사용한 SOAP/HTTPS 리스너 처리

SOAP/HTTPS 프로토콜 리스너 처리는 HTTPS가 보안 소켓을 사용하는 점을 제외하면 SOAP/HTTP 프로토콜 리스너 처리 섹션에 설명된 것과 동일합니다. 자세한 정보는 91 페이지의 『SSL』를 참조하십시오

SOAP/JMS 프로토콜 리스너 처리

SOAP/JMS 프로토콜 리스너는 웹 서비스 클라이언트의 요청에 대한 JMS 목적지인 InputQueue에서 지속적으로 청취하는 스레드로 구성됩니다. RequestWaitTimeout 커넥터 구성 등록 정보가 커넥터가 종료했는지 여부를 확인하기 전에 리스너가 요청에 대기할 기간을 정의합니다.

그림 26은 동기 운영에 대한 SOAP/JMS 프로토콜 리스너 처리를 보여줍니다. 이 그림은 JMS 제공자 정보를 표시하지 않습니다.

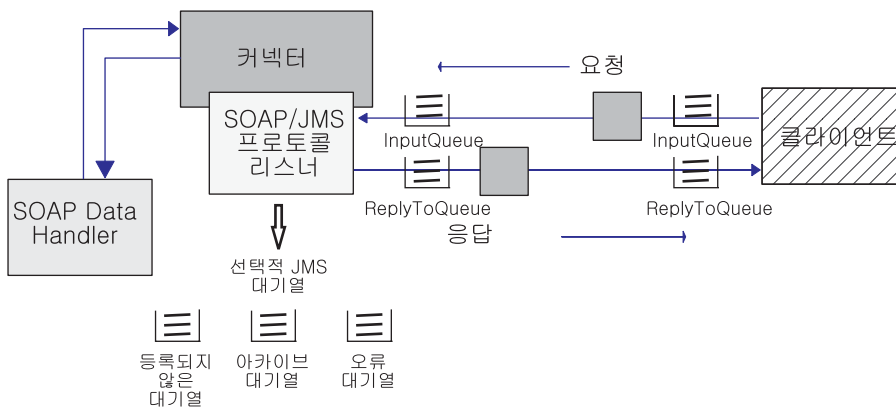


그림 26. SOAP/JMS 프로토콜 리스너: 동기 이벤트 처리

그림 27에 비동기 조작에 대한 SOAP/JMS 프로토콜 리스너 처리가 나와 있습니다.

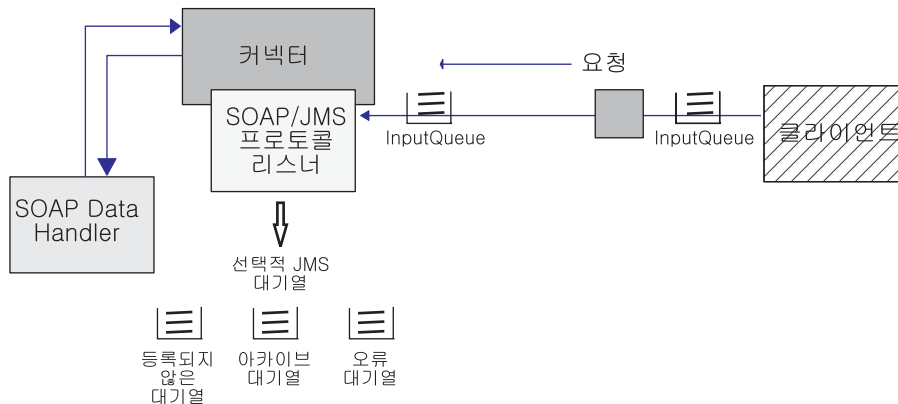


그림 27. SOAP/JMS 프로토콜 리스너: 비동기 이벤트 처리

주: LookupQueueUsingJNDI 구성 등록 정보가 true로 설정되는 경우, SOAP/JMS 프로토콜 리스너는 해당 JNDI를 사용하여 대기열을 찾습니다. JNDI 등록 정보는 커넥터 등록 정보에 지정됩니다. 자세한 정보는 89 페이지의 『커넥터 및 JMS』 및 95 페이지의 『커넥터 특정 구성 등록 정보』의 JNDI 관련 등록 정보를 참조하십시오.

웹 서비스 클라이언트는 SOAP/JMS 요청을 시작할 때 SOAP 요청 메시지를 SOAP/JMS 리스너가 청취 중인 InputQueue에 전송합니다. InputQueue로부터 SOAP 요청 메시지를 수신하면 SOAP/JMS 프로토콜 리스너는 프로토콜 리스너 프레임워크에 요청을 등록합니다. 프로토콜 리스너 프레임워크는 자원이 사용 가능할 때 요청을 스케줄링합니다.

주: 커넥터 구성 등록 정보 InDoubtEvents가 Reprocess로 설정되어 있는 경우, 프로토콜 리스너 프레임워크는 InputQueue의 메시지를 스케줄링하기 전에 InProgressQueue의 JMS 메시지를 스케줄링합니다.

그런 다음 리스너는 이 메시지--본문은 물론 필수 JMS 머리글(JMSCorrelationID, JMSTimeStamp, JMSType)--를 InProgressQueue에 디스패치합니다. 그런 다음, 프로토콜 리스너 프레임워크가 이벤트를 등록합니다.

이를 수행한 후 리스너는 InProgressQueue에서 JMS 메시지를 읽고 메시지 본문과 다음 머리글을 추출합니다.

- JMSDestination
- JMSRedelivered
- JMSCorrelationID
- JMSTimeStamp
- JMSType

- JMSExpiration
- JMSDeliveryMode
- JMSReplyTo
- JMSTimeStamp
- JMSType

JMSType은 TextMessage 또는 BytesMessage 형식일 수 있습니다. TextMessage 형식의 경우, 리스너는 웹 서비스 요청 메시지가 문자열로 추출된 문자열 API를 통해 Data Handler를 호출합니다. BytesMessage의 경우, 리스너는 웹 서비스 요청 메시지가 바이트 배열로 추출된 바이트 Data Handler API를 통해 Data Handler를 호출합니다.

SOAPDHMimeType 커넥터 구성 등록 정보를 사용하여 리스너는 SOAP Data Handler를 호출하여 요청 메시지를 SOAP 요청 Business Object로 변환합니다. 변환 중에 오류가 발생하고 JMSReplyTo JMS 머리글이 지정되는 경우, 리스너는 faultcode를 Client로 설정하고 faultstring을 Cannot Parse로 설정하는 SOAP 결함 메시지로 응답합니다. 결함 메시지는 다른 세부사항을 제공하지 않습니다.

리스너는 Data Handler가 리턴하는 SOAP 요청 Business Object의 오브젝트 레벨 cw_mo_jms ASI를 사용하여 프로토콜 구성 MO를 판별합니다. 이벤트 처리에서 ASI 및 프로토콜 구성 MO는 모두 선택적입니다. 프로토콜 구성 MO를 발견하는 경우, 리스너는 이를 이전에 추출된 JMS 메시지 머리글로 채웁니다. 표 29에서는 프로토콜 구성 MO에 있는 속성과 JMS 메시지 머리글 사이의 맵핑을 보여줍니다.

표 29. JMS 머리글-프로토콜 구성 MO 속성 맵핑

프로토콜 구성 MO 속성	JMS 머리글 이름	설명
CorrelationID	JMSCorrelationID	요청 메시지의 JMSCorrelationID 머리글
MessageId	JMSMessageId	요청 메시지의 JMSMessageID 머리글
Priority	JMSPriority	요청 메시지의 JMSPriority 머리글
Expiration	JMSExpiration	요청 메시지의 JMSExpiration 머리글
DeliveryMode	JMSDeliveryMode	요청 메시지의 JMSDeliveryMode 머리글
ReplyTo	JMSReplyTo	요청 메시지의 JMSReplyTo 머리글. JMS API는 이 머리글을 JMSDestination으로 리턴하지만, SOAP/JMS 프로토콜 리스너는 대기열 이름을 리턴합니다.
Timestamp	JMSTimestamp	요청 메시지의 JMSTimestamp 머리글
Redelivered	JMSRedelivered	요청 메시지의 JMSRedelivered 머리글
유형	JMSType	요청 메시지의 JMSType 머리글
Destination	JMSDestination	요청 메시지의 JMSDestination 머리글

SOAP/JMS 프로토콜 구성 MO UserDefinedProperties 속성에 하나 이상의 사용자 정의 등록 정보가 있으면 리스너는 메시지에서 해당 값을 추출하여 UserDefinedProperties

Business Object를 채웁니다. 사용자 정의 등록 정보에 대한 자세한 정보는 36 페이지의 『이벤트 처리를 위한 사용자 정의 등록 정보』의 내용을 참조하십시오.

TLO(비TLO SOAP 요청 Business Object의 경우)가 통합 브로커에 의해 등록되지 않은 경우, 리스너가 오류를 로그에 기록합니다. JMSReplyTo 머리글이 요청 메시지에 지정되는 경우, 리스너는 SOAP 결합 메시지를 작성하고 그것을 JMSReplyTo 대기열에 넣습니다. faultcode는 Client로 설정되고 faultString은 Not subscribed to this message로 설정됩니다. 결합 메시지에 다른 세부사항은 제공되지 않습니다. 이를 수행하도록 구성된 경우, 리스너는 JMS 머리글을 포함한 원본 JMS 요청 메시지를 UnsubscribedQueue에 아카이브합니다.

협업이 비동기로 호출되는 경우, 리스너는 요청 Business Object를 통합 브로커로 전달합니다. 그런 다음, 리스너는 InProgressQueue에서 해당 메시지를 제거합니다. 이를 수행하도록 구성된 경우, 리스너는 JMS 머리글을 포함한 원본 JMS 요청 메시지를 ArchiveQueue에 아카이브합니다.

비동기 처리 중에 오류가 발생하고 JMSReplyTo가 지정되는 경우, 리스너는 결합 메시지로 응답합니다. 결합 메시지의 faultcode는 Server로 설정되고, faultstring은 Internal Error로 설정됩니다. 이를 수행하도록 구성된 경우, 리스너는 JMS 머리글을 포함한 원본 JMS 요청 메시지를 ErrorQueue에 아카이브합니다.

동기 호출인 경우, 리스너는 동기식으로 협업을 호출합니다. 협업은 SOAP 응답 Business Object에 대해 응답합니다. 그런 다음, 리스너는 SOAP Data Handler를 호출하여 협업이 리턴하는 응답 Business Object를 SOAP/JMS 응답 메시지로 변환합니다. 리스너는 응답 메시지를 ReplyTo 대기열(이는 원래 요청 메시지의 JMSReplyTo 머리글에서 제공됨)로 전달합니다. 그런 후, 리스너는 Data Handler가 리턴하는 응답 메시지를 TextMessage 설정하고, 표 30에 표시된 머리글을 설정합니다.

표 30. 응답 메시지에서 SOAP/JMS 프로토콜 리스너가 설정하는 머리글 값

JMS 머리글 이름	값
JMSCorrelationId	요청 메시지의 JMSMessageId
JMSDeliveryMode	요청 메시지의 JMSDeliveryMode
JMSPriority	요청 메시지의 JMSPriority
JMSExpiration	요청 메시지의 JMSExpiration
JMSRedelivered	요청 메시지의 JMSRedelivered
JMSReplyTo	요청 메시지의 JMSReplyTo
JMSTimestamp	요청 메시지의 JMSTimestamp
JMSType	요청 메시지의 JMSType

리스너는 JMS 사용자 정의 등록 정보가 응답 또는 결합 Business Object의 JMS 프로토콜 구성 MO UserDefinedProperties 속성에 있을 경우 이를 응답 메시지에 설정합니다.

이를 수행하도록 구성된 경우, 리스너는 머리글을 포함한 원래 JMS 메시지(웹 서비스 클라이언트로부터의 요청)를 InProgressQueue에서 ArchiveQueue로 이동시킵니다.

오류가 발생하고 JMSReplyTo가 지정되는 경우, 리스너는 결합 메시지로 응답하고 별도로 구성하는 경우 원래 JMS 요청 메시지를 ErrorQueue에 아카이브합니다.

이벤트 지속성 및 전달

이벤트 지속성은 프로토콜마다 다릅니다.

- **SOAP/HTTP** 프로토콜 리스너 지속성이 없으므로 전달이 보장되지 않습니다.
- **SOAP/HTTPS** 프로토콜 리스너 지속성이 없으므로 전달이 보장되지 않습니다.
- **SOAP/JMS** 프로토콜 리스너 JMS 대기열 이벤트 지속성이 있으며 최소한 1회 이상 전달이 보장됩니다. JMS 대기열에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

이벤트 순서

커넥터는 이벤트를 모든 순서로 전달할 수 있습니다.

이벤트 트리거

이벤트 트리거 메커니즘은 프로토콜 리스너가 구성된 방법에 따라 다릅니다.

- **SOAP/HTTP** 프로토콜 리스너 HTTP 연결 요청에 대해 ServerSocket에서 청취가 발생합니다.
- **SOAP/HTTPS** 프로토콜 리스너 HTTPS 연결 요청에 대해 보안 ServerSocket 계층에서 청취가 발생합니다.
- **SOAP/JMS** 프로토콜 리스너 웹 서비스 요청을 전달하는 수신 JMS 메시지에 대해 입력 대기열에서 청취가 발생합니다. JMS 대기열에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

주: 커넥터는 작성, 갱신, 검색 또는 삭제를 구별하지 않습니다. 이러한 이벤트는 모두 동일한 접근 방식을 따릅니다.

이벤트 검출

이벤트 검출은 각각의 프로토콜 리스너에 의해 수행됩니다. 이벤트 검출 메커니즘은 전송과 각 리스너의 커넥터 특정 등록 정보를 구성하는 방법에 따라 완전히 다릅니다. 이러한 등록 정보에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

이벤트 상태

이벤트 상태는 프로토콜 리스너에서 관리하며 전송과 리스너 구성 방법에 따라 다릅니다.

- **SOAP/HTTP** 프로토콜 리스너 HTTP에는 원래 비지속성 및 동기성이 있습니다. 이에 따라 이벤트 상태는 유지보수되지 않습니다.
- **SOAP/HTTPS** 프로토콜 리스너 HTTP에는 원래 비지속성 및 동기성이 있습니다. 이에 따라 이벤트 상태는 유지보수되지 않습니다.
- **SOAP/JMS** 프로토콜 리스너 JMS는 지속적 전송입니다. 이벤트 상태는 대기열을 사용하여 유지보수됩니다. JMS 대기열에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

이벤트 검색

이벤트 검색은 프로토콜 리스너에서 관리하며 전송과 리스너 구성 방법에 따라 다릅니다.

- **SOAP/HTTP** 프로토콜 리스너 소켓에서 HTTP 요청을 추출하여 이벤트를 검색합니다.
- **SOAP/HTTPS** 프로토콜 리스너 소켓에서 HTTP 요청을 추출하여 이벤트를 검색합니다.
- **SOAP/JMS** 프로토콜 리스너 JMS API를 사용하여 이벤트를 검색합니다. JMS 프로토콜 리스너는 JMS 입력 대기열에서 이벤트를 검색하여 이를 진행 중인 대기열로 이동합니다. JMS 대기열에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

이벤트 아카이브

이벤트 아카이브는 프로토콜 리스너에서 관리하며 전송과 리스너 구성 방법에 따라 다릅니다.

- **SOAP/HTTP** 프로토콜 리스너 전송 수단의 비지속성 및 동기성으로 인해 아카이브는 수행되지 않습니다.
- **SOAP/HTTPS** 프로토콜 리스너 전송의 비지속성 및 동기성으로 인해 아카이브는 수행되지 않습니다.
- **SOAP/JMS** 프로토콜 리스너 비등록 이벤트, 성공한 이벤트 및 실패한 이벤트용 대기열을 포함하여 JMS 대기열로 이벤트를 아카이브하도록 커넥터를 구성할 수 있습니다. JMS 대기열에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

이벤트 복구

이벤트 복구는 프로토콜 리스너에서 관리하며 전송과 리스너 구성 방법에 따라 다릅니다.

- **SOAP/HTTP** 프로토콜 리스너 전송의 비지속성으로 인해 이벤트 복구가 수행되지 않습니다.

- **SOAP/HTTPS** 프로토콜 리스너 전송의 비지속성으로 인해 이벤트 복구가 수행되지 않습니다.
- **SOAP/JMS** 프로토콜 리스너 JMS는 지속적 전송입니다. 이벤트가 처리되는 중에 커넥터가 종료하는 경우, 이벤트는 InProgressQueue에서 계속 사용 가능합니다. 이러한 이벤트를 시작 시에 처리하여 이벤트 복구가 사용 가능하도록 커넥터를 구성할 수 있습니다. InDoubtEvents 커넥터 구성 등록 정보가 이벤트 복구 메커니즘을 판별합니다.

주: SOAP/JMS 리스너는 통합 브로커로의 최소한 1회 이상 전달을 보장합니다. 리스너는 단 한 번의 전달이나 이벤트 순서지정을 보장할 수 없습니다. 즉, 리스너가 수신한 이벤트가 임의의 순서로 통합 브로커에 전달될 수 있습니다.

시작 시에 JMS 프로토콜 리스너가 먼저 InProgressQueue에서 이벤트를 검색하려 시도합니다. 다음에 발생하는 일은 사용자가 InDoubtEvents 구성 등록 정보에 지정하는 값에 의해 결정됩니다. 복구 시나리오가 표에 표시되어 있습니다. JMS 대기열에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

표 31. 응답 메시지에서 SOAP/JMS 프로토콜 리스너가 설정하는 머리글 값

InDoubtEvents 값	이벤트 복구 처리
FailOnStartup	InProgressQueue에서 이벤트를 발견하면, 리스너는 심각한 오류를 로그하고 즉시 종료합니다.
Reprocess	InProgressQueue에서 이벤트를 발견하면, 리스너는 먼저 해당 이벤트를 처리합니다. 리스너는 InProgressQueue에서 발견하는 메시지 수를 추적할 수 있습니다.
Ignore	InProgressQueue의 이벤트가 무시됩니다. 리스너는 InProgressQueue에서 발견되는 이벤트 및 리스너에 의한 이벤트 무시를 추적할 수 있습니다.
LogError	InProgressQueue에서 이벤트를 발견하면, 리스너는 오류를 로그하고 처리를 계속합니다.

요청 처리

커넥터의 요청 처리 성능을 사용하여 협업이 웹 서비스를 호출할 수 있게 합니다. 개발 타스크는 WSDL ODA를 사용한 웹 서비스 최상위 레벨 오브젝트(TLO) 생성 및 이를 전개하도록 협업 구성을 포함합니다. 자세한 정보는 157 페이지의 제 6 장 『요청 처리용 협업 사용 가능』을 참조하십시오. 또한 커넥터와 그의 요청 처리 구성요소, 즉 프로토콜 핸들러 프레임워크와 프로토콜 핸들러를 구성해야 합니다.

런타임 시에, 커넥터는 Business Object의 양식으로 협업으로부터 요청을 수신합니다. Business Object(SOAP 요청, 및 선택적으로 SOAP 응답 및 SOAP 결합 Business Object)는 WSDL ODA가 생성하는 TLO에 포함되고 웹 서비스를 사용하도록 구성된 협업에 의해 발행됩니다. TLO 및 그의 하위 Business Object에는 처리 모드(동기 또는 비동기), Data Handler MIME 유형, 사용할 프로토콜 핸들러 및 대상 웹 서비스의 주소를 지정하는 속성 및 ASI가 들어있습니다. 프로토콜 핸들러가 이 정보를 사용하여

SOAP Data Handler의 인스턴스를 호출하고, 요청 Business Object를 SOAP 요청 메시지로 변환하고 대상 웹 서비스를 호출합니다. 모드가 동기인 경우, 프로토콜 핸들러는 다시 Data Handler를 호출하여 응답 메시지를 SOAP 응답 Business Object로 변환하고 이를 협업으로 리턴합니다.

SOAP 요청 메시지에 대한 응답에서, 커넥터는 원격 거래업체로부터 다음 중 하나를 수신할 수 있습니다.

- 데이터를 포함하는 SOAP 응답 메시지
- 결함 정보를 포함하는 SOAP 응답 메시지

프로토콜 핸들러가 요청 처리에서 중요한 역할을 합니다.

프로토콜 핸들러

협업은 HTTP, HTTPS 또는 JMS 전송을 통해 웹 서비스를 호출할 수 있습니다. 이들 커넥터에는 두 개의 프로토콜 핸들러와 대응하는 채널이 있습니다.

- SOAP/HTTP 및 SOAP/HTTPS 웹 서비스를 호출하기 위한 SOAP/HTTP-HTTPS 프로토콜 핸들러
- SOAP/JMS 웹 서비스를 호출하기 위한 SOAP/JMS 프로토콜 핸들러

프로토콜 핸들러 프레임워크는 시작 시 프로토콜 핸들러 로딩을 관리합니다. 커넥터가 요청 Business Object를 수신할 때, 요청 스레드(각 협업 요청이 고유한 스레드로 제공됨에 유의)가 프로토콜 핸들러 프레임워크를 호출하여 요청을 처리합니다.

프로토콜 핸들러 프레임워크는 TLO 핸들러 속성 ASI를 읽어서 사용할 프로토콜 핸들러를 판별합니다. 일련의 규칙을 적용하여(82 페이지의 『SOAP/HTTP-HTTPS 프로토콜 핸들러 처리』 및 86 페이지의 『SOAP/JMS 프로토콜 핸들러 처리』 참조), 프로토콜 핸들러는 요청 Business Object를 SOAP 요청 메시지로 변환하기 위한 Data Handler를 호출합니다. 프로토콜 핸들러는 요청 메시지를 전송(HTTP(S) 또는 JMS) 메시지로 패키징합니다. 요청 Business Object에서 SOAPAction ASI를 발견하는 경우, 프로토콜 핸들러는 이것을 요청 메시지 머리글에 추가합니다.

그런 다음, 프로토콜 핸들러는 요청 Business Object 프로토콜 구성 MO의 목적지 속성을 읽어서 대상 주소를 판별합니다. 그런 다음, 프로토콜 핸들러가 요청 메시지와 함께 대상 웹 서비스를 호출합니다.

프로토콜 핸들러는 ws_mode TLO ASI를 읽어서 처리 모드가 동기 또는 비동기인지를 판별합니다. 이 ASI가 asynch로 설정되는 경우, 프로토콜 핸들러 처리가 완료됩니다. 그렇지 않으면, 프로토콜 핸들러는 응답 메시지를 기다립니다. 응답 메시지가 도착하는 경우, 프로토콜 핸들러가 프로토콜 머리글과 페이로드를 추출합니다. 그런 다음, Data Handler(MimeType TLO 속성으로 표시됨)를 호출하여 메시지를 응답 또는 결함 Business Object로 변환합니다. 프로토콜 핸들러는 다시 프로토콜 구성 MO를 사용하

여 Business Object에 프로토콜 머리글을 설정합니다. 그런 다음, 프로토콜 핸들러가 협업에 응답 또는 결합 Business Object를 리턴합니다.

커넥터 구성에 따라 커넥터에 플러그인된 프로토콜 핸들러가 하나 이상 있을 수 있습니다. 커넥터 특정 등록 정보를 통해 프로토콜 핸들러를 구성할 수 있습니다.

SOAP/HTTP-HTTPS 프로토콜 핸들러 처리

SOAP/HTTP(S) 프로토콜 핸들러는 이 섹션에서 언급하는 예외를 사용하여 81 페이지의 『프로토콜 핸들러』에 설명한 대로 수행합니다. 그림 28은 동기 운영에 대한 SOAP/HTTP-HTTPS 프로토콜 핸들러를 보여줍니다.

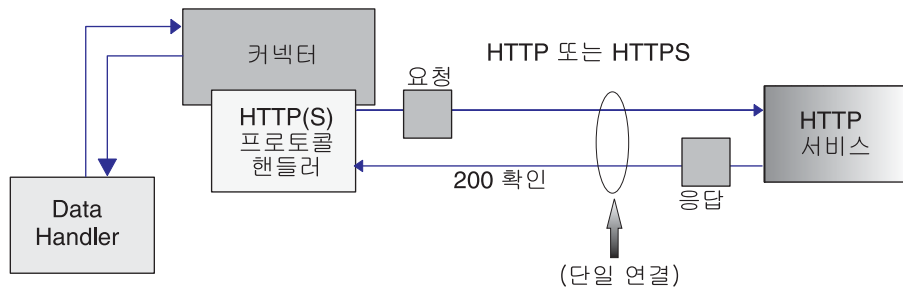


그림 28. SOAP/HTTP-HTTPS 프로토콜 핸들러: 동기 요청 처리

그림 29는 비동기 요청 프로세스에 대한 SOAP/HTTP-HTTPS 프로토콜 핸들러를 보여줍니다.

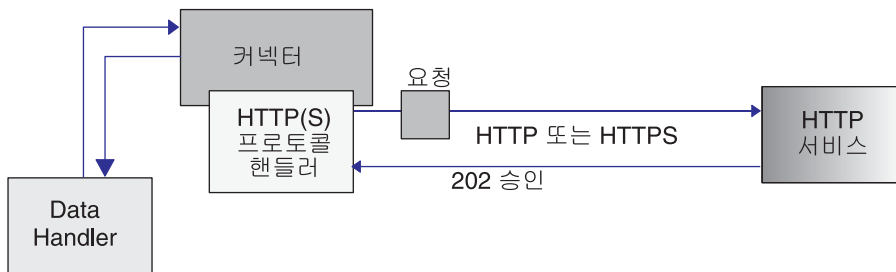


그림 29. SOAP/HTTP-HTTPS 프로토콜 핸들러: 비동기 요청 처리

주: 이 섹션은 SOAP/HTTP 프로토콜 처리만을 설명합니다.

SOAP/HTTP-HTTPS 프로토콜 핸들러는 프로토콜 구성 MO를 판별하기 위해 SOAP 요청 Business Object의 오브젝트 레벨 ASI(cw_mo_http)를 사용합니다.

SOAP/HTTP-HTTPS 프로토콜 핸들러는 HTTP 프로토콜 구성 MO의 목적지 속성을 읽어서 대상 웹 서비스의 URL을 판별합니다. URL이 누락되거나 불안정한 경우, 프로토

콜 핸들러는 서비스 호출에 실패합니다. HTTP 프로토콜 구성 MO 및 해당 속성에 대한 추가 정보는 51 페이지의 『요청 처리용 HTTP 프로토콜 구성 MO』의 내용을 참조하십시오.

SOAP/HTTP-HTTPS 프로토콜 핸들러는 SOAP Data Handler가 리턴하는 SOAP 요청 메시지를 사용하여 웹 서비스를 호출합니다. HTTP 프록시 커넥터 구성 등록 정보가 지정되는 경우, SOAP/HTTP(S) 프로토콜 핸들러가 그에 따라서 작동합니다. 응답이 리턴되는 경우, SOAP/HTTP(S) 프로토콜 핸들러가 응답을 읽습니다.

표 32에는 전송되는 요청 메시지에 대한 Charset, MimeType, ContentType 및 ContentType 머리글을 판별하기 위해 SOAP/HTTP-HTTPS 프로토콜 핸들러에 의해 사용되는 규칙의 우선 순위가 요약되어 있습니다.

표 32. 아웃바운드 메시지에 대한 SOAP/HTTP-HTTPS 프로토콜 핸들러 처리 규칙

우선 순위	Charset	MimeType	ContentType	ContentType 머리글
1	프로토콜 구성 MO의 Content-Type 머리글	TLO 속성의 MimeType 등록 정보	프로토콜 구성 MO의 Content-Type 머리글	프로토콜 구성 MO의 Content-Type 머리글
2	TLO 속성의 Charset 등록 정보	기본값은 ContentType	기본값은 text/xml	ContentType 및 Charset을 사용한 Content-Type 머리글 작성
3	ContentType이 text/*이면 기본값은 ISO-8859-1이 됩니다. 그렇지 않으면 Charset이 사용되지 않습니다.			

표 32에 표시된 대로:

- SOAP/HTTP-HTTPS 프로토콜 핸들러는 다음의 규칙에 따라 응답 메시지에 대한 Charset을 판별합니다.
 1. 요청 Business Object 프로토콜 구성 MO 머리글에 지정되어 있으면 Charset 값이 사용됩니다.
 2. 이전 단계에서 Charset이 판별되지 않으면 프로토콜 핸들러가 TLO 속성에서 Charset 추출을 시도합니다.
 3. 이전 단계에서 설명한 조작이 실패하면 테이블을 사용하여 Charset을 판별합니다.

표 33. 기본 요청 처리 Charset

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

4. 이전 단계에서 Charset가 판별된 경우에는 Charset이 Data Handler에서 설정됩니다.
 5. Data Handler는 요청을 쓰기 위해 필요한 데이터 구조에 따라 스트림 또는 바이트 배열 API로 호출됩니다.
- SOAP/HTTP-HTTPS 프로토콜 핸들러는 다음 규칙에 따라 요청에 대한 MimeType을 판별합니다.
 1. TLO MimeType 속성.
 2. TLO MimeType 속성이 누락되어 있으면 프로토콜 핸들러가 ContentType을 사용하여 MimeType을 판별합니다.
 - SOAP/HTTP-HTTPS 프로토콜 핸들러는 다음 규칙에 따라 요청 메시지에 대한 ContentType을 판별합니다.
 1. Content-Type 머리글이 요청 Business Object 프로토콜 구성 MO에 지정되어 있으면 머리글의 유형/부속 유형이 ContentType으로 사용됩니다.
 2. 그렇지 않으면 핸들러가 기본 ContentType: text/xml을 사용합니다.
 - SOAP/HTTP-HTTPS 프로토콜 핸들러는 다음 규칙에 따라 요청 메시지에 대한 Content-Type 머리글을 판별합니다.
 1. Content-Type 머리글이 요청 Business Object 프로토콜 구성 MO에 지정되어 있으면 해당 값이 전송되는 메시지에 설정됩니다.
 2. Content-Type 머리글이 요청 Business Object 프로토콜 구성 MO에 지정되어 있지 않으면 리스너는 ContentType 및 Charset 매개변수를 사용하여 Content-Type 머리글을 작성합니다(요청 메시지에 대한 Charset이 판별된 경우).

표 34에는 응답 메시지에 대한 Charset, MimeType, ContentType 및 Content-Type 머리글을 판별할 때 핸들러에 의해 사용되는 규칙의 우선 순위가 요약되어 있습니다.

표 34. 인바운드 동기 응답 메시지에 대한 SOAP/HTTP(s) 프로토콜 핸들러 처리 규칙

우선 순위	Charset	MimeType	ContentType	ContentType 머리글
1	수신 중인 HTTP 메시지 Content-Type 머리글 값에 있는 Charset 매개변수 값	요청 Business Object의 프로토콜 구성 MO에서 메시지 TransformationMap 하위 Business Object	Content-Type 머리글 값으로부터 수신하는 HTTP 메시지 유형/부속 유형 값	수신 중인 HTTP 메시지 Content-Type 머리글
2	요청 Business Object의 프로토콜 구성 MO에서 메시지 TransformationMap 하위 Business Object	요청 메시지 MimeType, 요청 및 응답 ContentType가 일치하는 경우에만.		
3	요청 메시지 Charset, 요청 및 응답 ContentType이 일치하는 경우에만.	TLO의 MimeType 등록 정보		
4	TLO의 Charset 등록 정보.	기본값은 ContentType		

표 34. 인바운드 동기 응답 메시지에 대한 SOAP/HTTP(s) 프로토콜 핸들러 처리 규칙 (계속)

5	Content-Type이 text/*이면 기본값은 ISO-8859-1입니다. 그렇지 않으면 Charset이 사용되지 않습니다.			
---	------------------------------------------------------------------------	--	--	--

표 34에 표시된 대로:

- 프로토콜 핸들러는 다음의 규칙에 따라 동기 응답 메시지의 Charset을 판별합니다.
 1. Charset 매개변수가 수신 중인 응답 메시지의 Content-Type 머리글에 설정되어 있으면 프로토콜 핸들러는 Charset 값을 사용하여 Data Handler에서 설정합니다.
 2. Charset 값이 응답 메시지 머리글에 없으면 프로토콜 핸들러는 협업 정의된 Charset을 TLO 요청 프로토콜 구성 MO MessageTransformationMap에서 읽으려고 시도합니다.
 3. Charset 값이 TLO에 지정되지 않았거나 TLO가 없을 경우 응답에 요청과 동일한 ContentType이 있으면 요청의 Charset이 응답에 사용됩니다.
 4. 이전 단계에서 Charset 값을 산출하지 못하면 프로토콜 핸들러는 TLO Charset 속성을 읽으려고 시도합니다.
 5. 이전 단계에 설명된 방법을 사용하여 Charset 값을 가져올 수 없을 때 메시지 ContentType 유형이 부속 유형을 가진 text이면(예: text/xml, text/plain 등), 기본값은 ISO-8859-1이 됩니다. 그렇지 않으면, charset 값이 사용되지 않습니다.
- 프로토콜 핸들러는 다음의 규칙에 따라 동기 응답 메시지의 MimeType을 판별합니다.
 1. 프로토콜 핸들러는 먼저 TLO 요청 프로토콜 구성 MO의 MessageTransformationMap에서 MimeType의 추출을 시도합니다. 특히 프로토콜 핸들러는 MTM에서 정확히 일치하는 ContentType을 찾아 MessageTransformationRule을 추출한 다음 거기에 들어 있는 MimeType 등록 정보 값을 사용하려고 시도합니다. 그렇지 않으면, 프로토콜 핸들러는 둘 이상의 ContentType에 적용되는 MessageTransformationRule을 찾습니다(ContentType이 */*임).
 2. MimeType을 MessageTransformationMap을 사용하여 판별하지 못한 경우, 프로토콜 핸들러는 요청 및 응답 ContentType가 일치하는 경우에만 응답에 경우에 대해 요청 MimeType을 사용합니다.
 3. 이전 단계에서 MimeType을 추출할 수 없는 경우, 프로토콜 핸들러는 TLO의 MimeType 속성 또는 기본 MimeType을 사용합니다(프로토콜 핸들러에 사용 가능한 경우).

4. 모든 이전 단계가 실패하면 프로토콜 핸들러는 `ContentType`을 사용하여 `MimeType`을 설정합니다.

- 핸들러는 수신 중인 HTTP 메시지 `Content-Type` 머리글로부터 유형/부속 유형을 추출하여 `ContentType`을 판별합니다.

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

1. 핸들러는 특수 속성(예: `ObjectEventId`)을 무시하기 위해 HTTP 프로토콜 구성 MO의 각 항목을 검토합니다.
2. 비어있지 않은 각 머리글은 전송 메시지에 기록되어 추가 처리(예: `Content-Type` 머리글)가 발생할 수 있습니다.
3. 이 방법을 사용할 경우 핸들러는 표준이 아닌 머리글을 메시지에 설정할 수 있으나 메시지가 논리적으로 또는 의미상 올바르다고 보장할 수 없다는 점을 참고하십시오.
4. HTTP 프로토콜 구성 MO `UserDefinedProperties` 속성에 하나 이상의 사용자 정의 등록 정보가 있으면 핸들러는 이를 엔티티 머리글 섹션(마지막 머리글 섹션)에 추가합니다. 사용자 정의 등록 정보에 대한 자세한 정보는 52 페이지의 『요청 처리를 위한 사용자 정의 등록 정보』의 내용을 참조하십시오.

주: `Connection`, `Trailer`, `Transfer-Encoding`, `Content-Encoding`, `Content-Length`, `Content-MD5`, `Content-Range` 머리글 중 하나를 HTTP 프로토콜 구성 MO에 지정하면 올바르지 않은 HTTP 메시지가 발생할 수 있습니다.

SOAP/JMS 프로토콜 핸들러 처리

SOAP/JMS 프로토콜 핸들러는 이 섹션에서 언급된 예외를 제외하고는 81 페이지의 『프로토콜 핸들러』에 설명한 대로 수행합니다.

주: `LookupQueueUsingJNDI` 구성 등록 정보가 `true`로 설정되는 경우, SOAP/JMS 프로토콜 핸들러는 해당 JNDI를 사용하여 목적지 대기열을 찾습니다. JNDI 등록 정보는 커넥터 등록 정보에 지정됩니다. 자세한 정보는 89 페이지의 『커넥터 및 JMS』 및 95 페이지의 『커넥터 특정 구성 등록 정보』의 JNDI 관련 등록 정보를 참조하십시오.

SOAP/JMS 프로토콜 핸들러는 SOAP Data Handler에 의해 리턴된 웹 서비스 요청 메시지의 본문을 사용하며 표 35에 표시된 대로 설정된 JMS 머리글을 사용하여 JMS 전송 메시지를 작성합니다.

표 35. 요청 메시지에 SOAP/JMS 프로토콜 핸들러가 설정하는 머리글 값

JMS 머리글 이름	SOAP/JMS 프로토콜 구성 MO에 설정되지 않은 경우 기본값
JMSPriority	4
JMSExpiration	0
JMSDeliveryMode	PERSISTENT
JMSReply	
JMSCorrelationId	
JMSRedelivered	
JMSTimestamp	
JMSType	

대상 웹 서비스가 비동기로 호출되는 경우, JMSReplyTo 머리글이 설정되지 않습니다. 그렇지 않으면(동기 처리의 경우), SOAP/JMS 프로토콜 핸들러가 JMSReplyTo 머리글을 설정합니다. SOAP/JMS 프로토콜 핸들러는 ReplyToQueue 구성 등록 정보를 사용하여 JMSDestination(대상 웹 서비스로부터 응답 또는 결함의 리턴 목적지)을 얻고 이를 JMS 전송 메시지의 JMSReplyTo 머리글에 설정합니다.

그림 30은 동기 요청 운영에 대한 SOAP/JMS 프로토콜 핸들러 처리를 보여줍니다.

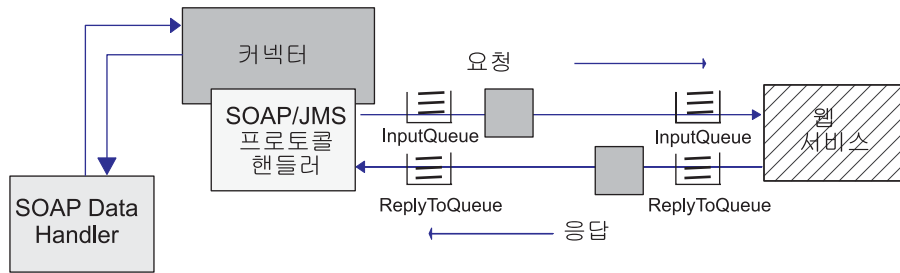


그림 30. SOAP/JMS 프로토콜 핸들러: 동기 요청 처리

그림 31은 비동기 요청 운영에 대한 SOAP/JMS 프로토콜 핸들러 처리를 보여줍니다.

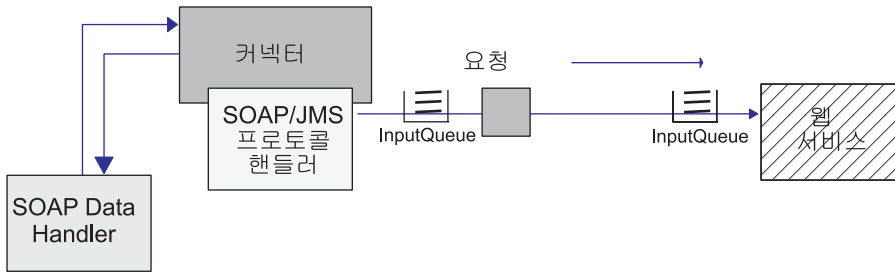


그림 31. SOAP/JMS 프로토콜 핸들러: 비동기 요청 처리

SOAP/JMS 프로토콜 핸들러는 프로토콜 구성 MO를 판별하기 위해 SOAP 요청 Business Object의 오브젝트 레벨 ASI(cw_mo_jms)를 사용합니다. 프로토콜 구성 MO의 목적지 속성이 대상 웹 서비스의 대기열 이름을 제공합니다. JNDI가 사용 가능한 경우, SOAP/JMS 프로토콜 핸들러는 JNDI 오브젝트를 찾아서 SOAP 요청 메시지에 대한 JMSDestination을 확보합니다. 그렇지 않으면, SOAP 프로토콜 구성 MO의 목적지 속성을 사용합니다.

응답이 ResponseWaitTimeout 등록 정보에 지정된 간격 내에 도착하지 않는 경우, SOAP/JMS 프로토콜 핸들러가 협업 요청에 실패합니다. SOAP 응답(또는 결함) 메시지가 도착할 때, SOAP/JMS 프로토콜 핸들러는 SOAP Data Handler에 의한 변환을 위해 JMS 머리글과 페이로드를 추출합니다. 그런 다음, SOAP/JMS 프로토콜 핸들러가 TLO에 SOAP 응답(또는 결함) Business Object를 설정하고, 응답(또는 결함) Business Object의 프로토콜 구성 MO를 사용하여 JMS 머리글을 맵핑합니다. 표 36에서는 이 맵핑을 보여줍니다.

표 36. 동기 요청 처리 중 응답에 대한 프로토콜 구성 MO--JMS 머리글 속성 맵핑

프로토콜 구성 MO 속성	JMS 머리글 이름	설명
Destination	JMSDestination	응답 메시지의 JMSDestination 머리글.
MessageId	JMSMessageId	응답 메시지의 JMSMessageId 머리글
Priority	JMSPriority	응답 메시지의 JMSPriority 머리글
Expiration	JMSExpiration	응답 메시지의 JMSExpiration 머리글
DeliveryMode	JMSDeliveryMode	응답 메시지의 JMSDeliveryMode 머리글
ReplyTo	JMSReplyTo	응답 메시지의 JMSReplyTo 머리글. JMS API는 이 머리글을 JMSDestination으로 리턴하지만, SOAP/JMS 프로토콜 리스너는 대기열 이름을 리턴합니다.
CorrelationId	JMSCorrelationId	응답 메시지의 JMSCorrelationId 머리글
Redelivered	JMSRedelivered	응답 메시지의 JMSRedelivered 머리글
TimeStamp	JMSTimeStamp	응답 메시지의 JMSTimeStamp 머리글
유형	JMSType	응답 메시지의 JMSType 머리글

그런 다음, SOAP/JMS 프로토콜 핸들러가 협업에 TLO를 리턴합니다.

커넥터 및 JMS

주: 이 섹션은 사용자가 JMS 및 JNDI, 특히 JMS의 작동 방법에 익숙하다고 가정합니다. 자세한 정보는 JMS 및 JNDI 소스 문서를 참조하십시오.

커넥터는 협업을 SOAP/JMS 웹 서비스로서 표시할 수 있으며, 또한 협업이 SOAP/JMS 웹 서비스를 호출할 수 있게 합니다. 웹 서비스 커넥터에서 SOAP/JMS 사용을 위한 요구사항은 다음과 같습니다.

1. JMS 서비스 제공자를 설치 및 구성했습니다.
2. JNDI를 설치 및 구성했습니다.
3. JMS 제공자가 JMS API 버전 1.0.2를 지원합니다.
4. 모든 필수 jar 파일이 커넥터의 클래스 경로에 있습니다(모든 필수 jar 파일을 판별하려면 JMS 제공자 문서를 참조하십시오).
5. 모든 필수 라이브러리가 커넥터의 경로에 있습니다(모든 필수 라이브러리를 판별하려면 JMS 제공자 문서를 참조하십시오).

JNDI

SOAP/JMS의 경우, 커넥터는 JNDI 문맥을 사용하는 연결 팩토리를 찾기 위해 JNDI를 사용합니다. 초기화 중에 커넥터는 JNDI에 연결하기 위해 JNDI 커넥터 특정 등록 정보를 읽습니다. 이 등록 정보를 구성하지 않으면 SOAP/JMS를 사용할 수 없습니다. 다음 JNDI 커넥터 특정 등록 정보를 지정할 수 있습니다.

- JNDIProviderURL
- InitialContextFactory
- JNDIConnectionFactoryName
- CTX_ObjectFactories
- CTX_ObjectFactories
- CTX_StateFactories
- CTX_URLPackagePrefixes
- CTX_DNS_URL
- CTX_Authoritative
- CTX_Batchsize
- CTX_Referral
- CTX_SecurityProtocol
- CTX_SecurityAuthentication
- CTX_SecurityPrincipal
- CTX_SecurityCredentials

- CTX_Language
- LookupQueuesUsingJNDI

이들 등록 정보 지정에 대한 지침에 대해서는 JNDI 문서를 참조하십시오. 커넥터와 함께 SOAP/JMS를 사용하려면, 다음 JNDI 커넥터 특정 등록 정보가 필요합니다.

- **JNDIProviderURL** 이 등록 정보를 JNDI 서비스 제공자의 URL로 설정하십시오. 이 등록 정보의 값에 대해서는 JNDI 제공자 문서를 참조하십시오.
- **InitialContextFactory** 이 등록 정보를 JNDI 초기 문맥을 작성할 팩토리 클래스의 완전한 클래스 이름으로 설정하십시오. 이 등록 정보의 값에 대해서는 JNDI 제공자 문서를 참조하십시오. 이 클래스 및 종속성이 커넥터의 클래스 경로에 있는지 확인하십시오.
- **JNDIConnectionFactoryName** 이 등록 정보를 찾아볼(JNDI 문맥 사용) 연결 팩토리의 JNDI 이름으로 설정하십시오. JNDI를 사용하여 이 이름을 찾을 수 있는지 확인하십시오.

LookupQueuesUsingJNDI를 true로 설정하는 경우, 커넥터가 사용하는 모든 대기열을 JNDI를 사용하여 찾을 수 있는지 확인하십시오.

협업을 SOAP/JMS 웹 서비스로 표시

협업을 SOAP/JMS 웹 서비스로서 표시하려면, SOAP/JMS 프로토콜 리스너를 사용해야 합니다. SOAP/JMS 프로토콜 리스너를 사용하려면 사용자가 JNDI 커넥터 등록 정보를 지정해야 합니다.

JMS 제공자 구성은 SOAP/JMS 프로토콜 리스너의 요구사항을 반영해야 합니다. JMS 서비스 제공자가 SOAP/JMS 프로토콜 리스너에 필요한 모든 대기열을 정의하는지 확인하십시오. 반드시 JMS 제공자 문서를 확인하십시오. 대기열 정의 타스크는 제공자별로 다릅니다. SOAP/JMS 프로토콜 리스너에 대해 6개의 대기열을 정의해야 합니다. SOAP/JMS 리스너 구성 등록 정보에 대기열 이름을 설정해야 하며, JNDI ” LookupQueuesUsingJNDI를 true로 설정한 경우, SOAP/JMS 리스너 구성 등록 정보에 대기열의 JNDI 이름도 지정해야 합니다.

다음 SOAP/JMS 리스너 구성 등록 정보의 값으로 대기열 이름을 지정해야 합니다.

- InputQueue
- InProgressQueue
- ArchiveQueue
- UnsubscribedQueue
- ErrorQueue
- ReplyToQueue

InputQueue 및 InProgressQueue는 필수 등록 정보입니다. 이들 대기열을 올바르게 구성했는지 확인하십시오.

ArchiveQueue, UnsubscribedQueue 및 ErrorQueue는 선택적 등록 정보입니다. 이들 대기열은 웹 서비스 요청을 아카이브하는 데 사용됩니다. 이들 등록 정보 중 하나를 사용할 계획인 경우, 해당 JMS 대기열을 올바르게 구성했는지 확인하십시오. JMS 제공자와 함께 이들 대기열을 정의할 때, 이들 대기열의 용량을 주의해서 지정해야 합니다.

SOAP/JMS 웹 서비스를 호출하는 협업

협업이 SOAP/JMS 웹 서비스를 호출할 수 있게 하려면, SOAP/JMS 프로토콜 핸들러를 사용합니다. SOAP/JMS 프로토콜 핸들러는 사용자가 JNDI 커넥터 등록 정보를 지정해야 합니다. 웹 서비스 제공자와 협의하여 JMS 및 JNDI 요구사항을 판별하십시오.

SOAP/JMS 웹 서비스를 호출하려면, 커넥터는 SOAP/JMS 프로토콜 구성 MO의 목적지 속성 값이 대상 웹 서비스의 입력 대기열로 설정되어야 합니다. JNDI "LookupQueuesUsingJNDI"를 true로 설정한 경우, 입력 대기열의 JNDI 이름을 지정해야 합니다.

요청/응답 웹 서비스를 호출할 경우, 웹 서비스 제공자와 협의하여 ReplyTo 대기열에 대한 요구사항을 판별해야 합니다. ReplyTo 대기열이 정의되는지 확인하십시오. 또한 SOAP/JMS 프로토콜 핸들러의 ReplyToQueue 구성 등록 정보에 ReplyTo 대기열의 이름을 지정했는지 확인하십시오. JNDI "LookupQueuesUsingJNDI"가 true로 설정되는 경우, ReplyToQueue 구성 등록 정보의 값이 이 대기열의 JNDI 이름을 제공해야 합니다.

프로토콜 리스너와는 달리 프로토콜 핸들러는 웹 서비스 커넥터에 플러그할 수 없음을 주의하는 것이 중요합니다. 결국, 커넥터는 커넥터가 호출하는 모든 요청/응답 웹 서비스에 대해 동일한 ReplyTo 대기열을 사용합니다.

SSL

이 섹션은 커넥터가 SSL 성능을 구현하는 방법에 대해 설명합니다. 배경 정보에 대해서는 SSL 문서를 참조하십시오. 이 섹션은 SSL 기술에 익숙하다고 가정합니다.

JSSE

커넥터는 협업을 SOAP/HTTPS 웹 서비스로 표시할 수 있고 협업이 SOAP/HTTPS 웹 서비스를 호출할 수 있게 합니다. 커넥터는 JSSE를 사용하여 HTTPS 및 SSL에 대한 지원을 제공합니다. IBM JSSE는 커넥터와 함께 제공됩니다. 이 성능을 사용 가능하게 하려면, 커넥터와 함께 설치되는 파일 중에 있는 `java.security` 파일에 다음 항목이 있어야 합니다.

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

java.security는 커넥터 설치의 \$ProductDir\lib\security 디렉토리에 위치합니다. 커넥터는 JavaProtocolHandlerPackages 커넥터 등록 정보의 값을 사용하여 시스템 등록 정보 java.protocol.handler.pkgs를 설정합니다. 커넥터와 함께 제공되는 IBM JSSE의 경우, 이 등록 정보의 값이 com.ibm.net.ssl.internal.www.protocol로 설정되어야 합니다.

JavaProtocolHandlerPackages 구성 등록 정보는 기본적으로 이 값으로 설정됩니다.

초기화 중에, 커넥터는 JSSE가 지원하는 모든 익명 암호 모음을 사용 불가능하게 합니다.

키스토어 및 트러스트스토어

커넥터와 함께 SSL을 사용하려면, 키스토어 및 트러스트스토어를 설정해야 합니다. 키스토어, 인증을 설정하거나 키 생성을 위한 도구가 제공되지 않습니다. 타사 소프트웨어를 사용하여 이들 작업을 완료해야 합니다.

SSL 등록 정보

다음 SSL 커넥터 특정 등록 정보를 지정할 수 있습니다.

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

이들 등록 정보는 커넥터 인스턴스에 적용됩니다. 동일한 SSL 등록 정보 값 세트가 커넥터에 플러그인되는 모든 SOAP/HTTPS 프로토콜 리스너 및 각 커넥터 인스턴스에 대한 SOAP/HTTP-HTTPS 프로토콜 핸들러에 의해 사용됩니다. HTTPS/SSL 설정에 대한 자세한 정보는 243 페이지의 부록 E 『HTTPS/SSL 구성』을 참조하십시오.

협업을 SOAP/HTTPS 웹 서비스로 표시

협업을 SOAP/HTTPS 웹 서비스로서 표시할 때, SOAP/HTTPS 프로토콜 리스너를 사용합니다. SOAP/HTTPS 프로토콜 리스너를 사용하려면 SSL 커넥터 특정 등록 정보를 지정해야 합니다. 이들 등록 정보에 지정하는 값은 SSL 요구사항을 반영해야 합니다.

- **SSLVersion** 사용하려는 SSLVersion이 JSSE에서 지원되는지 확인하십시오.
- **KeyStore** SOAP/HTTPS 프로토콜 리스너가 SSL 통신에서 서버로서 작용하기 때문에, 키스토어를 지정해야 합니다. 리스너는 SSL " KeyStore 구성 등록 정보에 지정되는 키스토어를 사용합니다. 이 등록 정보 값은 키스토어 파일의 전체 경로여야

합니다. 키스토어가 커넥터에 대한 키 쌍(개인용 및 공용 키)을 포함하는지 확인하십시오. 개인용 키의 별명이 SSL " KeyStoreAlias 등록 정보로서 지정되어야 합니다. 키스토어에 액세스하기 위해 필요한 암호를 SSL " KeyStorePassword 등록 정보로 지정해야 합니다. 또한 키스토어 및 개인용 키(키스토어에 있는)에 액세스하기 위해 필요한 암호가 동일한지 확인하십시오. 마지막으로, 커넥터의 디지털 인증을 웹 서비스 클라이언트에 분배하여 해당 클라이언트가 커넥터를 인증할 수 있어야 합니다.

- **TrustStore SOAP/HTTPS** 프로토콜 리스너가 웹 서비스 클라이언트를 인증하기 원하는 경우, 클라이언트 인증을 활성화해야 합니다. SSL " UseClientAuth 등록 정보를 true로 설정하여 이 작업을 수행합니다. 또한 다음을 지정해야 합니다.
 - SSL " TrustStore 구성 등록 정보의 값으로서 트러스트스토어의 위치
 - SSL " TrustStorePassword 등록 정보의 값으로서 트러스트스토어에 액세스하기 위해 필요한 암호

트러스트스토어가 웹 서비스 클라이언트의 디지털 인증을 포함하는지 확인하십시오. 웹 서비스 클라이언트가 사용하는 디지털 인증은 자체 서명되거나 CA에서 발행됩니다. 트러스트스토어가 CA의 루트 인증을 신뢰하는 경우, JSSE는 해당 CA에서 발행되는 모든 디지털 인증을 인증합니다.

HTTPS/SSL 설정에 대한 자세한 정보는 243 페이지의 부록 E 『HTTPS/SSL 구성』을 참조하십시오.

SOAP/HTTPS 웹 서비스를 호출하는 협업

협업이 SOAP/HTTPS 웹 서비스를 호출할 수 있게 하려면, SOAP/HTTP-HTTPS 프로토콜 핸들러를 사용합니다. SOAP/HTTP-HTTPS 프로토콜 핸들러와 함께 SSL을 사용하려면 SSL 커넥터 특정 등록 정보를 지정해야 합니다. 이들 등록 정보에 지정하는 값이 웹 서비스 제공자의 HTTPS/SSL 요구사항을 반영해야 합니다.

- **SSLVersion** 사용하려는 SSLVersion이 웹 서비스 제공자와 JSSE에서 지원되는지 확인하십시오.
- **TrustStore SOAP/HTTP-HTTPS** 프로토콜 핸들러가 SSL 통신에서 클라이언트로서 작용하기 때문에 트러스트스토어를 설정해야 합니다. 핸들러는 SSL -> Truststore 구성 등록 정보에 지정된 Truststore를 사용합니다. 이 등록 정보의 값은 트러스트스토어 파일에 대한 전체 경로여야 합니다. SSL -> TrustStorePassword 등록 정보의 Truststore에 액세스하기 위해 필요한 암호를 지정해야 합니다. 트러스트스토어가 웹 서비스 제공자의 디지털 인증을 포함하는지 확인하십시오. 웹 서비스 제공자가 사용하는 디지털 인증은 자체 서명되거나 CA에서 발행될 수 있습니다. 트러스트스토어가 CA의 루트 인증을 신뢰하는 경우, JSSE는 해당 CA에서 발행되는 모든 디지털 인증을 인증합니다.
- **KeyStore** 웹 서비스 제공자가 클라이언트 인증이 필요한 경우, 키스토어를 설정해야 합니다. SOAP/HTTP-HTTPS 프로토콜 핸들러는 SSL " KeyStore 구성 등록 정보에 지정되는 키스토어를 사용합니다. 이 값은 키스토어 파일의 전체 경로여야 합

니다. 키스토어가 커넥터에 대해 구성된 키 쌍(개인용 키 및 공용 키)을 포함하는지 확인하십시오. 개인용 키의 별명이 SSL ” KeyStoreAlias 등록 정보에 지정되어야 합니다. 키스토어에 액세스하기 위해 필요한 암호가 SSL ” KeyStorePassword 등록 정보에 지정되어야 합니다. 마지막으로, 키스토어 및 개인용 키(키스토어에 있는)에 액세스하기 위해 필요한 암호가 동일한지 확인하십시오. 인증을 위해 커넥터의 디지털 인증을 웹 서비스 제공자에 분배해야 합니다.

HTTPS/SSL 설정에 대한 자세한 정보는 243 페이지의 부록 E 『HTTPS/SSL 구성』을 참조하십시오.

커넥터 구성

설치 프로그램을 사용하여 시스템에 커넥터 파일을 설치한 후 표준 및 응용프로그램 특정 커넥터 구성 등록 정보를 설정해야 합니다.

구성 설정 등록 정보

커넥터에는 표준 구성 등록 정보 및 커넥터 특정 구성 등록 정보의 두 유형의 구성 등록 정보가 있습니다. 커넥터를 실행하기 전에 SM(System Manager)을 사용하여 이들 등록 정보의 값을 설정해야 합니다.

표준 구성 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이들 등록 정보의 문서에 대해서는 185 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오. 아래 표에는 부록의 구성 등록 정보와 관련하여 이 커넥터에 특정한 정보를 제공합니다.

등록 정보	설명
CharacterEncoding	이 커넥터는 이 등록 정보를 사용하지 않습니다.
Locale	이 커넥터는 국제화되지 않았으므로, 이 등록 정보의 값을 변경할 수 없습니다. 현재 지원되는 로케일을 판별하려면 커넥터에 대한 릴리스 정보를 참조하십시오.

이 커넥터는 통합 브로커로서 InterChange Server(ICS)만을 지원하기 때문에 통합 브로커와 관련된 유일한 구성 등록 정보는 ICS에 대한 사항입니다.

최소한 다음 표준 커넥터 구성 등록 정보를 설정해야 합니다.

- AgentTraceLevel
- ApplicationName
- ControllerTraceLevel
- DeliveryTransport

커넥터 특정 구성 등록 정보

커넥터 특정 구성 등록 정보는 런타임 중에 커넥터 에이전트에 필요한 정보를 제공합니다. 또한 커넥터 특정 등록 정보는 에이전트를 다시 코드화하고 재빌드할 필요없이 커넥터 에이전트 내에서 정적 정보 또는 논리를 변경하는 방법을 제공합니다.

표 37은 커넥터 특정 구성 등록 정보를 나열합니다. 등록 정보의 설명은 다음의 섹션을 참조하십시오. 일부 등록 정보에는 다른 등록 정보가 포함되어 있습니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

주: 커넥터에 SOAP/JMS 프로토콜 리스너나 SOAP/JMS 프로토콜 핸들러를 사용하지 않으려면, SOAP/JMS에 관련된 커넥터 특정 등록 정보를 삭제하거나 공백으로 두었는지 확인하십시오.

표 37. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수
ConnectorType	임의의 유효한 커넥터 유형	WebService	예
DataHandlerMetaObjectName	Data Handler Meta Object 이름	MO_DataHandler_Default	예
JavaProtocolHandlerPackages	유효한 Java 프로토콜 핸들러 패키지	com.ibm.net.ssl.internal. www.protocol	아니오
ProtocolHandlerFramework	이것은 계층 구조 등록 정보이며 값은 없음	없음	아니오
+ProtocolHandlers	이것은 계층 구조 등록 정보이며 값은 없음		아니오
++SOAPHTTPHTTPSHandler	이는 계층 구조 등록 정보입니다. 서버 등록 정보에 대한 정보는 97 페이지의 『SOAPHTTPHTTPSHandler』를 참조하십시오.		예
++SOAPJMSHandler	이는 계층 구조 등록 정보입니다. 서버 등록 정보에 대한 정보는 98 페이지의 『SOAPJMSHandler』를 참조하십시오.		
ProtocolListenerFramework	이것은 계층 구조 등록 정보이며 값은 없습니다.		아니오
+WorkerThreadCount	사용 가능한 리스너 스레드의 수를 제공 하는 1보다 큰 정수	10	아니오
+RequestPoolSize	자원 풀 크기를 제공하는 WorkerThreadCount보다 큰 정수	20	아니오
+ProtocolListeners	이것은 계층 구조 등록 정보이며 값은 없음		
++Listener1	고유하게 이름 지정된 프로토콜 리스너		예
+++Protocol	soap/http, soap/https, soap/jms		예
+++SOAPDHMimeType	SOAP Data Handler의 유효한 MIME 유형	xml/soap	
+++ListenerSpecific	리스너에 고유하거나 리스너에 필요한 등록 정보 100 페이지의 『ListenerSpecific』을 참조하십시오.		
ProxyServer	이것은 계층 구조 등록 정보이며 값은 없음		아니오
+HttpProxyHost	HTTP 프록시 서버의 호스트 이름		아니오
+HttpProxyPort	HTTP 프록시 서버의 포트 번호	80	아니오
+HttpNonProxyHosts	직접 연결이 필요한 HTTP 호스트		아니오

표 37. 커넥터 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
+HttpsProxyHost	HTTPS 프록시 서버의 호스트 이름		아니오
+HttpsProxyPort	HTTPS 프록시 서버의 포트 번호	443	아니오
+HttpsNonProxyHosts	직접 연결이 필요한 HTTPS 호스트		아니오
+SocksProxyHost	소켓 프록시 서버 이름		아니오
+SocksProxyPort	소켓 프록시 서버 포트		아니오
+HttpProxyUsername	Http 프록시 서버 사용자 이름		아니오
+HttpProxyPassword	Http 프록시 서버 암호		아니오
+HttpsProxyUsername	Https 프록시 서버 사용자 이름		아니오
+HttpsProxyPassword	Https 프록시 서버 암호		아니오
SSL	이것은 계층 구조 등록 정보이며 값은 없음		아니오
+SSLVersion	SSL, SSLv2, SSLv3, TLS, TLSv1	SSL	아니오
+SSLDebug	true, false	false	아니오
+KeyStoreType	유효한 키스토어 유형	JKS	아니오
+KeyStore	KeyStore 파일 경로		아니오
+KeyStorePassword	키스토어의 개인용 키 암호		아니오
+KeyStoreAlias	키스토어에 있는 키 쌍의 별명		아니오
+TrustStore	TrustStore 파일 경로		아니오
+TrustStorePassword	TrustStore 암호		아니오
+UseClientAuth	true false	false	아니오
WSCollaborations	이것은 WSDL 구성 마법사가 작성하는 계층 구조 등록 정보이며 값은 없음. 110 페이지의 『WSCollaborations』를 참조하십시오.		
+Collaboration1	이것은 계층 구조 등록 정보이며 값은 없음		
++CollaborationPort1	협업 포트의 이름		예
+++WebServiceOperation1	이것은 계층 구조 등록 정보이며 값은 없음		예
++++BodyName	웹 서비스 메소드의 이름. 유효한 XML 요소 이름이어야 합니다		예
++++BodyNS	웹 서비스 메소드의 이름 공간. 유효한 XML 이름 공간이어야 합니다		예
++++BOName	조작할 요청 Business Object의 이름		예
++++Mode	synch asynch	asynch	아니오
JNDI	이것은 JMS 관련 계층 구조 등록 정보이며 값은 없음		아니오
+LookupQueuesUsingJNDI	true false	false	아니오
+JNDIProviderURL	유효한 JNDI URL		아니오
+InitialContextFactory	초기 문맥의 팩토리 클래스 이름		아니오
+JNDIConnectionFactoryName	JNDI 문맥을 사용하여 찾을 연결 팩토리의 이름.		아니오
+CTX_ObjectFactories	JNDI 문맥에서 보안 및 오브젝트 찾아보기에 대한 추가 정보를 지정하는 등록 정보		아니오
+CTX_properties			

ConnectorType: 이 등록 정보를 WebService로 설정하면, 협업 포트를 바인드할 때 System Manager가 커넥터를 웹 서비스 커넥터로 표시합니다. 그렇지 않으면 정상 커넥터로 표시됩니다.

기본값 = WebService

DataHandlerMetaObjectName: 이것은 Data Handler가 구성 등록 정보를 설정하는 데 사용하는 Meta Object의 이름입니다.

기본값 = MO_DataHandler_Default

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

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

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

기본값 = 없음

ProtocolHandlers: 이 계층 구조 등록 정보에는 값이 없습니다. 첫 번째 레벨의 하위는 개별 프로토콜 핸들러를 나타냅니다.

기본값 = 없음

SOAPHTTPHTTPSHandler: SOAP/HTTP-HTTPS 프로토콜 핸들러의 이름. 이는 계층 구조 등록 정보입니다. 리스너와는 달리 프로토콜 핸들러는 중복될 수 없으므로 각 프로토콜마다 단 하나의 핸들러만 있을 수 있습니다. 아래에 있는 표 38에서는 SOAP/HTTP-HTTPS 프로토콜 핸들러에 대한 종속 등록 정보를 보여줍니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 38. SOAP/HTTP-HTTPS 프로토콜 핸들러 구성 등록 정보

이름	가능한 값	기본값	필수
++SOAPHTTPHTTPSHandler	이것은 계층 구조 등록 정보이며 값은 없습니다.		예
+++Protocol	핸들러가 구현하는 프로토콜 종류. SOAP/HTTP 및 SOAP/HTTPS의 경우, 값은 soap/http입니다. 주: 이 등록 정보의 값이 지정되지 않으면, 커넥터는 프로토콜 핸들러를 초기화하지 않습니다.		예
+++HTTPReadTimeout	원격 호스트(웹 서비스)에서 읽는 동안의 제한시간 간격(밀리초 단위)을 지정하는 SOAP/HTTP 특정 등록 정보 이 등록 정보를 지정하지 않거나 0으로 설정하면 원격 호스트로부터 읽는 동안 SOAP/HTTP 프로토콜 핸들러가 무한정 블록화됩니다.		아니오

그림 32에서는 Connector Configurator에 표시되는 등록 정보를 보여줍니다.

Standard Properties		Application Config Properties		Supported Business Objects		Trace	
	Property	Value	Update	Encrypt			
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>			
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>			
3	<input type="checkbox"/> ProxyServer		agent restart	<input type="checkbox"/>			
4	<input type="checkbox"/> SSL		agent restart	<input type="checkbox"/>			
5	<input type="checkbox"/> ProtocolListenerFramework		agent restart	<input type="checkbox"/>			
6	<input type="checkbox"/> ProtocolHandlerFramework		agent restart	<input type="checkbox"/>			
7	<input type="checkbox"/> ProtocolHandlers		agent restart	<input type="checkbox"/>			
8	<input type="checkbox"/> SOAPHTTPHTTPSHandler		agent restart	<input type="checkbox"/>			
9	Protocol	soap/http	agent restart	<input type="checkbox"/>			
10	HTTPReadTimeout	0	agent restart	<input type="checkbox"/>			
11	<input type="checkbox"/> SOAPJMSHandler		agent restart	<input type="checkbox"/>			
12	<input type="checkbox"/> JNDI		agent restart	<input type="checkbox"/>			
13	<input type="checkbox"/> WSCollaborations		agent restart	<input type="checkbox"/>			

그림 32. SOAP/HTTP-HTTPS 프로토콜 핸들러 등록 정보

SOAPJMSHandler: SOAP/JMS 프로토콜 핸들러의 이름. 이는 계층 구조 등록 정보입니다. 리스너와는 달리 프로토콜 핸들러는 중복될 수 없으므로 각 프로토콜마다 단 하나의 핸들러만 있을 수 있습니다. 아래에 있는 표 39에서는 SOAP/JMS 프로토콜 핸들러에 대한 종속 등록 정보를 보여줍니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 39. SOAP/JMS 프로토콜 핸들러 구성 등록 정보

이름	가능한 값	기본값	필수
++SOAPJMSHandler	이것은 계층 구조 등록 정보이며 값은 없습니다.		예
+++Protocol	핸들러가 구현하는 프로토콜 종류. SOAP/JMS의 경우, 값은 soap/jms입니다. 주: 이 등록 정보의 값이 지정되지 않으면, 커넥터는 프로토콜 핸들러를 초기화하지 않습니다.		예
+++ResponseWaitTimeout	프로토콜 핸들러가 동기 요청 처리를 위해 ReplyToQueue에서 기다리는 제한시간 간격(밀리초 단위)을 지정하는 JMS 프로토콜 핸들러 특정 등록 정보입니다. 이 간격 중에 응답이 도달하지 않으면 핸들러는 협업 요청에 실패합니다. 이 등록 정보를 지정하지 않거나 0으로 설정하면 프로토콜 핸들러는 ReplyToQueue에서 무한정 대기합니다.	0	아니오
+++ReplyToQueue	이는 ReplyTo 대기열의 이름을 지정하는 JMS 프로토콜 핸들러 특정 등록 정보입니다. 동기 요청 처리의 경우, 핸들러는 JMSReplyTo 필드를 이 JMS 목적으로 설정합니다. LookupQueuesUsingJNDI = true일 경우 SOAP/JMS 프로토콜 핸들러는 JNDI를 사용하여 이 대기열을 찾습니다.	none	예

그림 33에서는 Connector Configurator에 표시되는 등록 정보를 보여줍니다.

Standard Properties		Application Config Properties		Supported Business Objects		Trace/Log Files		D...	
	Property	Value	Update Method	Encrypt	Description				
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>					
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>					
3	ProxyServer		agent restart	<input type="checkbox"/>					
4	SSL		agent restart	<input type="checkbox"/>					
5	ProtocolListenerFramework		agent restart	<input type="checkbox"/>					
6	ProtocolHandlerFramework		agent restart	<input type="checkbox"/>					
7	ProtocolHandlers		agent restart	<input type="checkbox"/>					
8	SOAPHTTPHTTTPSHandler		agent restart	<input type="checkbox"/>					
9	SOAPJMSHandler		agent restart	<input type="checkbox"/>					
10	Protocol	soap/jms	agent restart	<input type="checkbox"/>					
11	ResponseWaitTimeout	0	agent restart	<input type="checkbox"/>					
12	ReplyToQueue		agent restart	<input type="checkbox"/>					
13	JNDI		agent restart	<input type="checkbox"/>					
14	WSCollaborations		agent restart	<input type="checkbox"/>					

그림 33. SOAP/JMS 프로토콜 핸들러 등록 정보

ProtocolListenerFramework: 프로토콜 리스너 프레임워크는 이 등록 정보를 사용하여 프로토콜 리스너를 로드합니다. 이것은 계층 구조 등록 정보이며 값은 없습니다.

WorkerThreadCount: 1보다 큰 정수여야 하는 이 등록 정보는 프로토콜 리스너 프레임워크에 사용할 수 있는 프로토콜 리스너 작업 스레드 수를 설정합니다. 자세한 정보는 68 페이지의 『프로토콜 리스너』. 기본값 = 10의 내용을 참조하십시오.

RequestPoolSize: WorkerThreadCount보다 큰 정수여야 하는 이 등록 정보는 프로토콜 리스너 프레임워크의 자원 풀 크기를 설정합니다. 프레임워크는 최대 WorkerThreadCount + RequestPoolSize 요청을 동시에 처리할 수 있습니다.

기본값 = 20

ProtocolListeners: 이것은 계층 구조 등록 정보이며 값은 없습니다. 이 등록 정보의 각 첫 번째 레벨 하위 요소는 개별 프로토콜 리스너를 나타냅니다.

Listener1: 프로토콜 리스너의 이름. 여러 개의 프로토콜 리스너가 있을 수 있습니다. 이는 계층 구조 등록 정보입니다. 이 등록 정보의 복수 인스턴스를 작성하고 추가로 고유하게 이름 지정된 리스너를 작성할 수 있습니다. 그렇게 할 때, 리스너 특정 등록 정보를 변경할 수 있지만 프로토콜 등록 정보는 변경할 수 없습니다. 복수 리스너의 이름은 고유해야 합니다. 가능한 이름(값이 아님): SOAPHTTPListener1, SOAPHTTPSLListener1, SOAPJMSListener1

Protocol: 이 등록 정보는 이 리스너가 구현하고 있는 프로토콜을 지정합니다. 가능한 값: soap/http, soap/https, soap/jms.

주: 이 등록 정보의 값이 지정되지 않으면, 커넥터는 프로토콜 리스너를 초기화하지 않습니다.

SOAPDHMimeType: 리스너가 수신한 요청에 사용할 SOAP Data Handler MIME 유형

기본값 = xml/soap

ListenerSpecific: 리스너 특정 등록 정보는 지정된 프로토콜 리스너에 고유하거나 반드시 필요합니다. 예를 들어, HTTP 리스너에는 리스너가 요청을 모니터링하는 포트 번호를 나타내는 리스너 특정 등록 정보 포트가 있습니다. 표 40은 HTTP-HTTPS 리스너 특정 등록 정보를 요약합니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 40. SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너 특정 구성 등록 정보

이름	가능한 값	기본값	필수
+++SOAPHTTPListenerI	HTTP 프로토콜 리스너의 고유 이름. <i>ProtocolListenerFramework</i> -> <i>ProtocolListeners</i> 계층 구조 등록 정보의 하위입니다. 여러 개의 리스너가 있을 수 있습니다. 이 등록 정보와 계층 구조를 작성하여 추가 HTTP 리스너를 플러그인할 수 있습니다.		예
++++Protocol	SOAP/HTTP 프로토콜 리스너의 경우 soap/http, SOAP/HTTPS 프로토콜 리스너의 경우 soap/https 주: 이 등록 정보의 값이 지정되지 않으면, 커넥터는 프로토콜 리스너를 초기화하지 않습니다.		예
++++SOAPDHMimeType	xml/soap	xml/soap	아니오
++++BOPrefix	이 등록 정보의 값은 <i>Data Handler</i> 로 전달됩니다.		아니오
++++Host	리스너는 이 등록 정보의 값으로 지정된 IP 주소에서 청취합니다. 호스트가 지정되지 않으면 localhost로 기본값이 설정됩니다. 리스너가 실행 중인 시스템의 호스트 이름(DNS 이름) 또는 IP 주소를 지정할 수 있습니다. 한 시스템에 여러 개의 IP 주소나 여러 개의 이름이 있을 수 있습니다.	localhost	아니오
++++Port	리스너가 요청을 청취하는 포트 지정하지 않으면, 포트의 기본값은 SOAP/HTTP의 경우에는 80, SOAP/HTTPS의 경우에는 443으로 설정됩니다. 커넥터 내에서 리스너를 복제할 경우, 호스트 및 포트 등록 정보의 조합은 고유하며 그렇지 않을 경우 리스너가 요청을 승인하기 위해 포트에 바인드할 수 없습니다.	SOAP/HTTP 리스너의 경우 80, SOAP/HTTPS 리스너의 경우 443	아니오
++++SocketQueueLength	수신 연결 요청용 대기열(소켓 대기열)의 길이. 호스트가 연결을 거부하기 전에 한 번에 저장할 수 있는 수신 연결 수를 지정합니다. 최대 대기열 길이는 운영 체제에 따라 다릅니다.	연 5	아니오
++++RequestWaitTimeout	웹 서비스 요청이 도달하기를 기다리는 동안 호스트 및 포트에서 리스너 스레드를 블록화할 시간 간격(밀리초). 이 간격 이전에 웹 서비스 요청을 수신하면 리스너는 이 요청을 처리합니다. 그렇지 않으면 리스너 스레드가 커넥터 종료 플래그의 설정 여부를 확인합니다. 값이 설정되어 있으면 커넥터를 종료하고 그렇지 않으면 <i>RequestWaitTimeout</i> 간격 동안 계속 블록화됩니다. 이 등록 정보를 0으로 설정할 경우 커넥터는 무한정 블록화됩니다. 값을 지정하지 않으면 60000ms 기본값으로 설정됩니다.	60000(ms)	아니오

표 40. SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
+++++HTTPReadTimeout	클라이언트로부터 웹 서비스 요청을 읽는 동안 리스너를 블 록화할 시간 간격(밀리초). 이 매개변수를 0으로 설정할 경 우 리스너는 전체 요청 메시지를 수신할 때까지 무한정 블 록화됩니다.	0	아니오
+++++HttpAsyncResponseCode	리스너로의 비동기 요청에 대한 HTTP 응답 코드: 200 (확인) 202 (승인)	202 (승인)	아니오
+++++URLsConfiguration	계층 구조 등록 정보이며 값이 없습니다. 이 리스너에서 지 원하는 URL에 대한 하나 이상의 구성 및 선택적으로 MIME 유형 및 charset 값이 포함됩니다. 참고로 이는 ProtocolListenerFramework -> ProtocolListeners -> SOAPHTTPListener1 계층 구조 등록 정보의 하위입니다. 이 등록 정보를 지정하지 않을 경우, 리스너는 기본값을 사 용합니다.	ContextPath: / Enabled: true Data handler MimeType: 요청 의 ContentType과 동일함 Charset: 없음. 자세한 정보 는 69 페이지의 『SOAP/HTTP 및 SOAP/HTTPS 프 로토콜 리스너 처 리』를 참조하십시 오	아니오
+++++URL1	이것은 계층 구조 등록 정보이며 값은 없습니다. 해당 하 위는 이 리스너에서 지원하는 URL의 이름을 제공합니다. 다중 지원 URL이 있을 수 있습니다. 이 등록 정보 및 해 당 계층 구조를 복제함으로써 추가 URL을 플러그인할 수 있습니다.		아니오
+++++ContextPath	리스너에서 받은 HTTP 요청에 대한 URI입니다. 이 값은 URLsConfiguration 등록 정보 아래의 ContextPath 값들 중에서 고유해야 합니다. 그렇지 않으면 커넥터가 오류를 로 강하며 시작에 실패합니다. ContextPath는 대소문자를 구 분합니다. 그러나 여기에 프로토콜, 호스트 이름 및 포트(대 소문자를 구분함)가 포함될 수 있습니다. 프로토콜이 ContextPath에 지정되어 있으면 이는 http가 되어야 합니 다. 호스트가 지정되면 이는 호스트 리스너 등록 정보의 값 과 동일해야 합니다. 포트를 지정할 경우, 포트 리스너 등 록 정보 값과 동일해야 합니다.		아니오
+++++Enabled	이 등록 정보의 값은 상위 URL 계층 구조 등록 정보가 커 넥터에 대해 사용 가능한지 여부를 판별합니다.	True	아니오
+++++TransformationRules	이것은 계층 구조 등록 정보이며 값은 없습니다. 하나 이 상의 변환 규칙을 보유합니다.		
+++++TransformationRule1	이것은 계층 구조 등록 정보이며 값은 없습니다. 변환 규 칩을 보유합니다.		아니오

표 40. SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
+++++++ContentType	이 등록 정보의 값은 특정 핸들링(Data Handler MIME 유형 또는 charset)을 적용해야 하는 수신 중인 요청의 ContentType을 지정합니다. TransformationRuleN 계층 구조 등록 정보에 ContentType이 지정되지 않은 경우, 커넥터는 경고 메시지를 로그한 다음 TransformationRuleN 등록 정보를 무시합니다. 이 등록 정보에 대해 특수 값 */*를 지정하면 프로토콜 리스너가 임의의 ContentType에 이 규칙을 적용할 수 있습니다. 리스너가 ContentType을 공유하는 동일 컨텍스트 경로에 대한 하나 이상의 규칙을 찾으면 리스너는 오류를 로깅하며 초기화에 실패함을 참고하십시오.		아니오
+++++++MimeType	지정된 ContentType 요청을 처리하기 위해 Data Handler를 호출할 때 사용할 MIME 유형입니다.		아니오
+++++++Charset	지정된 ContentType 요청을 Business Object로 변환할 때 사용할 Charset입니다.		아니오

그림 34에서는 Connector Configurator에 표시되는 등록 정보를 보여줍니다.

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 34. SOAP/HTTP 프로토콜 리스너 등록 정보

표 41에는 SOAP/JMS 프로토콜 리스너 특정 등록 정보가 요약되어 있습니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 41. SOAP/JMS 프로토콜 리스너 특정 구성 등록 정보

이름	가능한 값	기본값	필수
+++SOAPJMSListener1	JMS 프로토콜 리스너의 고유 이름. ProtocolListenerFramework -> ProtocolListeners 계층 구조 등록 정보의 하위입니다. 여러 개의 리스너가 있을 수 있습니다. 이 등록 정보와 계층 구조를 작성하여 추가 JMS 리스너를 플러그인할 수 있습니다.		예

표 41. SOAP/JMS 프로토콜 리스너 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
++++Protocol	soap/jms		예
++++SOAPDHMimeType	xml/soap	xml/soap	아니오
++++BOPrefix	이 등록 정보의 값은 SOAPDHMimeType 등록 정보에서 지정한 Data Handler로 전달됩니다.		아니오
++++RequestWaitTimeout	이 등록 정보는 웹 서비스 요청을 기다리는 동안 SOAP/JMS 리스너 스레드가 InputQueue를 블록화하는 시간 간격을 설정합니다. 이 간격 내에서 웹 서비스 요청을 수신하면 리스너는 이 요청을 처리합니다. 이 간격 내에서 요청을 수신하지 않으면, 리스너 스레드는 먼저 커넥터 종료 플래그가 설정되어 있는지 확인합니다. 커넥터 종료 플래그가 설정되어 있으면 커넥터를 종료하고, 그렇지 않으면 RequestWaitTimeout 간격 동안 계속 블록화됩니다. 이 등록 정보를 0으로 설정할 경우, 커넥터는 무한정 블록화됩니다.	60000	아니오
++++SessionPoolSize	주어진 리스너 및 작업자 스레드에 할당될 수 있는 최대 세션 수. 최소 세션 수(및 기본값)는 2입니다. 더 큰 세션 풀 크기의 경우, 커넥터는 더 많은 메모리를 필요로 합니다.		아니오
++++InputQueue	이 등록 정보는 리스너가 웹 서비스의 인바운드 메시지를 풀링하는 입력 대기열의 이름을 제공합니다. LookupQueuesUsingJNDI = true일 경우, 리스너는 JNDI를 사용하여 이 대기열을 찾고 InputQueue 등록 정보의 값은 SOAP/JMS 바인딩의 jms:address 요소에 대한 jndiDestinationName 속성으로 설정됩니다. jms:address 요소는 WSDL 문서의 wsdl:port 섹션에 지정됩니다. WSDL 생성 중에 SOAP/JMS 리스너를 선택할 경우, 시스템 관리자는 이 등록 정보의 값을 사용하여 자동으로 jndiDestinationName 속성을 작성합니다. LookupQueueUsingJNDI = false인 경우, 시스템 관리자는 대신 jmsProviderDestinationName 속성을 작성합니다.		예
++++InProgressQueue	이 등록 정보는 진행 중인 대기열의 이름을 제공합니다. 리스너는 InputQueue에서 InProgressQueue로 인바운드 메시지 사본을 전송합니다. LookupQueuesUsingJNDI = true일 경우 리스너는 JNDI를 사용하여 이 대기열을 찾습니다.		예
++++ArchiveQueue	이 등록 정보는 아카이브 대기열의 이름을 제공합니다. 리스너는 성공적으로 처리한 메시지 사본을 InProgressQueue에서 ArchiveQueue로 전송합니다. LookupQueuesUsingJNDI = true일 경우, 리스너는 JNDI를 사용하여 이 대기열을 찾습니다.		아니오
++++UnsubscribedQueue	이 등록 정보는 비등록 대기열의 이름을 제공합니다. 리스너는 비등록 메시지 사본을 InProgressQueue에서 UnsubscribedQueue로 전송합니다. LookupQueuesUsingJNDI = true일 경우, 리스너는 JNDI를 사용하여 이 대기열을 찾습니다.		아니오
++++ErrorQueue	이 등록 정보는 오류 대기열의 이름을 제공합니다. 리스너는 실패한 메시지 사본을 ErrorQueue로 전송합니다. LookupQueuesUsingJNDI = true일 경우, 리스너는 JNDI를 사용하여 이 대기열을 찾습니다.		아니오

표 41. SOAP/JMS 프로토콜 리스너 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
++++InDoubtEvents	<p>이 등록 정보는 예상치 못한 커넥터 종료로 완전하게 처리되지 않은 메시지를 <i>InProgressQueue</i>에서 처리하는 방법을 지정합니다. 다음 값 중 하나를 사용할 수 있습니다.</p> <ul style="list-style-type: none"> • <i>FailOnStartup</i> 오류를 로그하고 시스템을 즉시 종료합니다. • <i>Reprocess InProgressQueue</i>에서 나머지 메시지를 처리합니다. • <i>Ignore</i> 진행 중인 대기열에 있는 메시지를 무시합니다. • <i>LogError</i> 오류를 로그하지만 시스템을 종료하지 않습니다. 	Ignore	아니오
++++ReplyToQueue	<p>이 등록 정보는 <i>ReplyTo</i> 대기열의 이름을 제공합니다. WSDL 구성 마법사가 이 등록 정보를 읽고 WSDL 문서에 기록합니다. 이 등록 정보를 지정하지 않으면, 유틸리티는 WSDL 문서의 SOAP/JMS 바인딩에서 <i>ReplyTo JMS</i> 머릿글을 작성하지 않습니다. (리스너는 이 등록 정보를 사용하지 않습니다.) JNDI 등록 정보를 지정하고 <i>LookupQueueUsingJNDI = false</i>일 경우에도 WSDL 생성 유틸리티는 WSDL 문서에서 JNDI 특정 속성을 작성합니다. SOAP/JMS 바인딩이 JNDI 없이 <i>ReplyTo</i> 속성을 지정하는 방식을 제공하지 않기 때문에 이러한 JNDI 특정 속성이 필요합니다. <i>InputQueue</i>의 경우 JNDI 찾아보기가 필수는 아니지만 <i>ReplyTo</i> 대기열의 경우에는 JNDI 특정 등록 정보가 필수입니다. WSDL 유틸리티가 JNDI 특정 등록 정보를 찾지 않으면 유틸리티는 SOAP/JMS 바인딩에서 <i>ReplyTo</i> 속성을 작성할 수 없습니다.</p>		
++++ JMSVendorURI	<p>JMS 구현을 고유하게 식별하고 SOAP/JMS 바인딩의 <i>jms:address</i> 요소에 대한 <i>jmsVendorURI</i> 속성에 해당하는 문자열. <i>jms:address</i> 요소는 WSDL 문서의 <i>wSDL:port</i> 섹션에 지정됩니다. 리스너는 이 등록 정보를 사용하지 않습니다.</p>		아니오

그림 35에서는 Connector Configurator에 표시되는 등록 정보를 보여줍니다.

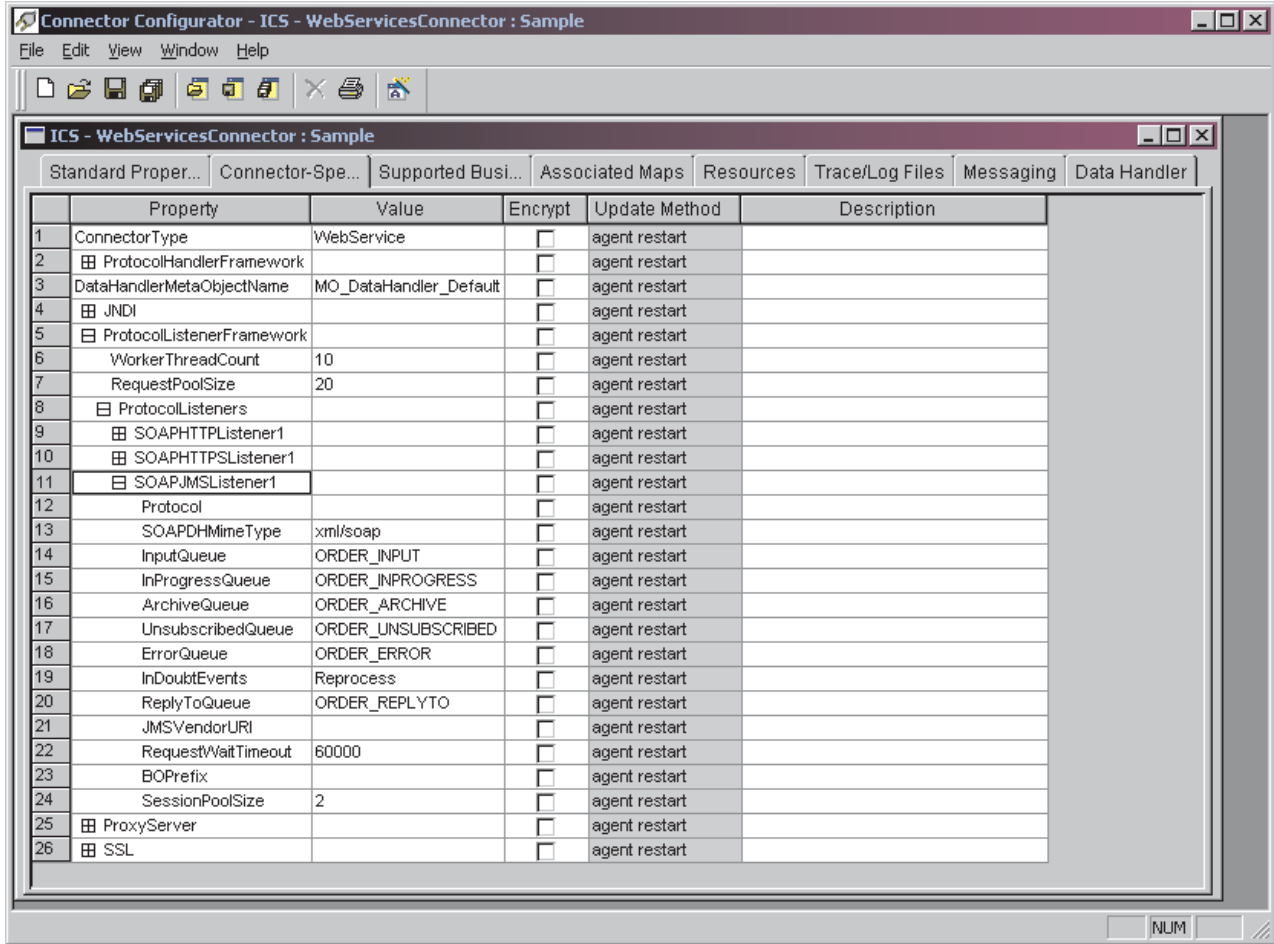


그림 35. SOAP/JMS 프로토콜 리스너 등록 정보

주: 다음 등록 정보에 지정된 대기열 이름이 고유한지 확인하십시오.

- InputQueue
- InProgressQueue
- ArchiveQueue
- UnsubscribedQueue
- ErrorQueue

ProxyServer: 네트워크가 프록시 서버를 사용하는 경우, 이 등록 정보에 값을 구성하십시오. 이것은 계층 구조 등록 정보이며 값은 없습니다. 이 등록 정보에 지정된 값은 SOAP/HTTP/HTTPS 프로토콜 핸들러에서 사용됩니다.

그림 36에서는 Connector Configurator에 표시되며 아래에서 논의되는 ProxyServer 등록 정보를 보여줍니다.

Standard Properties		Connector-Specific Properties	Supported Business Objects	Associated Maps	Resources
	Property	Value	Encrypt	Update Method	Description
1	<input checked="" type="checkbox"/> ProtocolHandlerFramework		<input type="checkbox"/>	agent restart	
2	DataHandlerMetaObjectName	MO_DataHandler_Default	<input type="checkbox"/>	agent restart	
3	<input checked="" 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 checked="" type="checkbox"/> SSL		<input type="checkbox"/>	agent restart	

그림 36. ProxyServer 등록 정보

HttpProxyHost: HTTP 프록시 서버의 호스트 이름. 네트워크에서 HTTP 프로토콜의 프록시 서버를 사용하는 경우, 이 등록 정보를 지정하십시오.

기본값 = 없음

HttpProxyPort: 커넥터가 HTTP 프록시 서버에 연결하는 데 사용하는 포트 번호

기본값 = 80

HttpNonProxyHosts: 이 등록 정보의 값은 프록시 서버를 통하지 않고 직접 연결해야 하는 하나 이상의 호스트(HTTP용)를 제공합니다. 값은 각각 "/"로 구분되는 호스트 목록일 수 있습니다.

기본값 = 없음

HttpsProxyHost: HTTPS 프록시 서버의 호스트 이름

기본값 = 없음

HttpsProxyPort: 커넥터가 HTTPS 프록시 서버에 연결하는 데 사용하는 포트 번호

기본값 = 443

HttpsNonProxyHosts: 이 등록 정보의 값은 프록시 서버를 통하지 않고 직접 연결해야 하는 하나 이상의 호스트(HTTPS용)를 제공합니다. 값은 각각 "|"로 구분되는 호스트 목록일 수 있습니다.

기본값 = 없음

SocksProxyHost: Socks 프록시 서버의 호스트 이름. 네트워크에서 Socks 프록시를 사용하는 경우, 이 등록 정보를 지정하십시오.

주: 기본 JDK가 Socks를 지원해야 합니다.

기본값 = 없음

SocksProxyPort: Socks 프록시 서버에 연결할 포트 번호. 네트워크에서 Socks 프록시를 사용하는 경우, 이 등록 정보를 지정하십시오.

기본값 = 없음

HttpProxyUsername: HTTP 프록시 서버의 사용자 이름. 웹 서비스 요청에 대한 목적지가 HTTP URL이며 ProxyServer -> HttpProxyUsername을 지정하는 경우, SOAP HTTP/HTTPS 프로토콜 핸들러는 프록시에 대해 인증할 때 Proxy-Authorization 머리를 작성합니다. 핸들러는 인증을 위해 CONNECT 메소드를 사용합니다.

프록시-인증 머리는 Base-64 인코딩되어 있으며 구조는 다음과 같습니다.

Proxy-Authorization: 기본

Base64EncodedString

핸들러는 사용자 이름 및 암호 등록 정보 값을 콜론(:)으로 분리하여 연결함으로써 base64 인코딩 문자열을 작성합니다.

기본값 = 없음

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

기본값 = 없음

HttpsProxyUsername: HTTPS 프록시 서버의 사용자 이름. 웹 서비스 요청에 대한 목적지가 HTTPS URL이며 ProxyServer -> HttpsProxyUsername을 지정하는 경우, SOAP HTTP/HTTPS 프로토콜 핸들러는 프록시에 대한 인증을 위해

Proxy-Authorization 머리글을 작성합니다. 핸들러는 HttpsProxyUsername 및 HttpsProxyPassword 구성 등록 정보 값을 콜론(:)으로 분리하여 연결함으로써 Base-64 인코드화된 문자열을 작성합니다.

기본값 = 없음

HttpsProxyPassword: HTTPS 프록시 서버의 암호. 이 값이 사용되는 방식에 대한 자세한 정보는 108 페이지의 『HttpsProxyUsername』을 참조하십시오.

기본값 = 없음

SSL: 커넥터에 SSL을 구성하려면 이 등록 정보에서 값을 지정하십시오. 이것은 계층 구조 등록 정보이며 값은 없습니다.

그림 37에서는 Connector Configurator에 표시되며 아래에서 논의되는 SSL 등록 정보를 보여줍니다.

Standard Properties		Application Config Properties		Supported Business Objects		Trace/Log Files		D:	
	Property	Value	Update Method	Encrypt	Description				
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>					
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>					
3	ProxyServer		agent restart	<input type="checkbox"/>					
4	SSL		agent restart	<input type="checkbox"/>					
5	SSLVersion	SSL	agent restart	<input type="checkbox"/>					
6	SSLDebug	False	agent restart	<input type="checkbox"/>					
7	KeyStoreType	JKS	agent restart	<input type="checkbox"/>					
8	KeyStore		agent restart	<input type="checkbox"/>					
9	KeyStorePassword		agent restart	<input type="checkbox"/>					
10	KeyStoreAlias		agent restart	<input type="checkbox"/>					
11	TrustStore		agent restart	<input type="checkbox"/>					
12	TrustStorePassword		agent restart	<input type="checkbox"/>					
13	UseClientAuth	False	agent restart	<input type="checkbox"/>					
14	ProtocolListenerFramework		agent restart	<input type="checkbox"/>					
15	ProtocolHandlerFramework		agent restart	<input type="checkbox"/>					
16	JNDI		agent restart	<input type="checkbox"/>					
17	WSCollaborations		agent restart	<input type="checkbox"/>					

그림 37. SSL 등록 정보

SSLVersion: 커넥터에서 사용할 SSL 버전. 자세한 정보는 지원되는 SSL 버전에 대한 IBM JSSE 문서를 참조하십시오.

기본값 = SSL

SSLDebug: 이 등록 정보의 값을 true로 설정할 경우, 커넥터는 javax.net.debug 시스템 등록 정보의 값을 true로 설정합니다. IBM JSSE는 이 등록 정보를 사용하여 추적 기능을 켭니다. 자세한 정보는 IBM JSSE 문서를 참조하십시오.

기본값 = false

KeyStoreType: 이 등록 정보의 값은 KeyStore 및 TrustStore 유형을 제공합니다. 자세한 정보는 유효한 KeyStore 유형에 대한 IBM JSSE 문서를 참조하십시오.

기본값 = JKS

KeyStore: 이 등록 정보는 KeyStore 파일의 전체 경로를 제공합니다. KeyStore 및/또는 KeyStoreAlias 등록 정보를 지정하지 않으면 KeyStorePassword, KeyStoreAlias, TrustStore 및 TrustStorePassword 등록 정보는 무시됩니다. 이 등록 정보에 지정된 경로를 사용하여 KeyStore를 로드할 수 없는 경우 커넥터는 시작되지 않습니다. 경로는 키스토어 파일의 전체 경로여야 합니다.

기본값 = 없음

KeyStorePassword: 이 등록 정보는 KeyStore에서 개인용 키의 암호를 제공합니다.

기본값 = 없음

KeyStoreAlias: 이 등록 정보는 KeyStore에서 키 쌍의 별명을 제공합니다. SOAP/HTTPS 리스너는 KeyStore에서 이 개인용 키를 사용합니다. 또한, SOAP/HTTP-HTTPS 프로토콜 핸들러가 클라이언트 인증이 필요한 웹 서비스를 호출할 때 KeyStore의 이 별명을 사용합니다. 등록 정보는 유효한 JSSE 별명으로 설정되어야 합니다.

기본값 = 없음

TrustStore: 이 등록 정보는 TrustStore의 전체 경로를 제공합니다. TrustStore는 커넥터가 신뢰하는 인증을 저장하는 데 사용됩니다. TrustStore는 KeyStore와 같은 유형이어야 합니다. TrustStore 파일의 전체 경로를 지정해야 합니다.

기본값 = 없음

TrustStorePassword: 이 등록 정보는 TrustStore의 암호를 제공합니다.

기본값 = 없음

UseClientAuth: 이 등록 정보는 SSL 클라이언트 인증의 사용 여부를 지정합니다. true로 설정되면, SOAP/HTTPS 리스너는 클라이언트 인증을 사용합니다.

기본값 = false

WSCollaborations: 이 등록 정보는 협업 오브젝트를 웹 서비스로서 표시할 때 자동으로 작성되며 BITLO에 사용됩니다. 이것은 계층 구조 등록 정보이며 값은 없습니다. 이 등록 정보의 각 첫 번째 레벨 하위 요소는 웹 서비스로 표시된 협업을 표시합니

다. 이들 등록 정보를 자동으로 작성하는 데 사용되는 도구에 대한 정보는 159 페이지의 제 7 장 『협업을 웹 서비스로 표시』를 참조하십시오.

주: 시스템 관리자에서 협업 또는 해당 포트를 삭제할 경우, 커넥터는 협업을 표시하는 등록 정보를 자동으로 삭제하지 않습니다. 이러한 등록 정보는 Connector Configurator를 사용하여 삭제해야 합니다.

그림 38에서는 Connector Configurator에 표시되며 아래에서 논의되는 WSCollaborations 등록 정보를 보여줍니다.

Standard Properties		Application Config Properties		Supported Business Objects		Trace/Log Files		D
	Property	Value	Update	Encrypt	Description			
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>				
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>				
3	<input type="checkbox"/> ProxyServer		agent restart	<input type="checkbox"/>				
4	<input type="checkbox"/> SSL		agent restart	<input type="checkbox"/>				
5	<input type="checkbox"/> ProtocolListenerFramework		agent restart	<input type="checkbox"/>				
6	<input type="checkbox"/> ProtocolHandlerFramework		agent restart	<input type="checkbox"/>				
7	<input type="checkbox"/> JNDI		agent restart	<input type="checkbox"/>				
8	<input type="checkbox"/> WSCollaborations		agent restart	<input type="checkbox"/>				
9	<input type="checkbox"/> Collaboration1		agent restart	<input type="checkbox"/>				
10	<input type="checkbox"/> CollaborationPort1		agent restart	<input type="checkbox"/>				
11	<input type="checkbox"/> WebServiceOperation1		agent restart	<input type="checkbox"/>				
12	BodyName		agent restart	<input type="checkbox"/>				
13	BodyNS		agent restart	<input type="checkbox"/>				
14	BOName		agent restart	<input type="checkbox"/>				
15	BOVerb		agent restart	<input type="checkbox"/>				
16	Synchronous		agent restart	<input type="checkbox"/>				

그림 38. WSCollaborations 등록 정보

Collaboration1: 이 등록 정보는 해당 커넥터를 통해 웹 서비스로 표시되는 협업 오브젝트의 이름입니다. 이것은 계층 구조 등록 정보이며 값은 없습니다. 이러한 등록 정보는 여러 개가 있을 수 있으며, 웹 서비스로 표시되는 각 협업 오브젝트에 하나씩 존재합니다. 이 등록 정보의 각 첫 번째 레벨 하위 요소는 해당 협업의 포트를 표시합니다.

CollaborationPort1: 이 등록 정보는 협업 포트에 이름을 지정합니다. 이것은 계층 구조 등록 정보이며 값은 없습니다. 이러한 등록 정보는 여러 개가 있을 수 있으며, 커넥터로 바인드되는 이 협업의 각 포트에 하나씩 존재합니다. 이 등록 정보의 각 첫 번째 레벨 하위 요소는 웹 서비스 협업을 나타냅니다.

WebServiceOperation1: 이 등록 정보는 협업 오브젝트에 해당하는 웹 서비스 작업을 나타냅니다. 이것은 계층 구조 등록 정보이며 값은 없습니다. 이러한 등록 정보는 하나 이상 있을 수 있으며, WSDL 문서 생성 시 사용자가 정의한 각 웹 서비스 작업에 하나씩 존재합니다.

BodyName: 이 등록 정보는 웹 서비스 메소드의 이름을 제공하며 유효한 XML 요소 이름이어야 합니다.

기본값 = 없음

BodyNS: 이 등록 정보는 웹 서비스 메소드의 이름 공간을 제공하며 유효한 XML 이름 공간이어야 합니다.

기본값 = 없음

BOName: 이 등록 정보는 해당 조作的 요청 Business Object 이름을 제공합니다.

기본값 = 없음

Mode: 이 등록 정보는 조작에 대한 처리 모드를 지정합니다. synch로 설정할 경우, 커넥터는 동기식으로 협업을 호출합니다. 그렇지 않으면, 기본적으로 요청 전용 조작으로서 협업을 비동기식으로 호출합니다.

기본값 = asynch

JNDI: 커넥터는 JNDI에 연결할 때 SOAP/JMS 프로토콜 핸들러와 JMS 프로토콜 리스너에서 사용하는 하나의 JNDI(Java Naming and Directory Interface) 제공자 등록 정보 세트를 유지보수합니다. 이것은 계층 구조 등록 정보이며 값은 없습니다. 커넥터는 JNDI를 사용하여 JMS 연결 팩토리 오브젝트를 찾습니다. WSDL 구성 마법사가 SOAP/JMS 바인딩을 생성할 때 이 등록 정보를 사용합니다.

그림 39에서는 Connector Configurator에 표시되며 아래에서 논의되는 JNDI 등록 정보를 보여줍니다.

Standard Properties		Application Config Properties		Supported Business Objects		Trace/Log Files		D:	
	Property	Value	Update	Encrypt	Description				
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>					
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>					
3	ProxyServer		agent restart	<input type="checkbox"/>					
4	SSL		agent restart	<input type="checkbox"/>					
5	ProtocolListenerFramework		agent restart	<input type="checkbox"/>					
6	ProtocolHandlerFramework		agent restart	<input type="checkbox"/>					
7	JNDI		agent restart	<input type="checkbox"/>					
8	LookupQueuesUsingJNDI	False	agent restart	<input type="checkbox"/>					
9	InitialContextFactory		agent restart	<input type="checkbox"/>					
10	JNDIConnectionFactoryName		agent restart	<input type="checkbox"/>					
11	CTX_ObjectFactories		agent restart	<input type="checkbox"/>					
12	CTX_StateFactories		agent restart	<input type="checkbox"/>					
13	CTX_URLPackagePrefixes		agent restart	<input type="checkbox"/>					
14	CTX_DNS_URL		agent restart	<input type="checkbox"/>					
15	CTX_Authoritative		agent restart	<input type="checkbox"/>					
16	CTX_Batchsize		agent restart	<input type="checkbox"/>					
17	CTX_Referral		agent restart	<input type="checkbox"/>					
18	CTX_SecurityProtocol		agent restart	<input type="checkbox"/>					
19	CTX_SecurityAuthentication		agent restart	<input type="checkbox"/>					
20	CTX_SecurityPrincipal		agent restart	<input type="checkbox"/>					
21	CTX_SecurityCredentials		agent restart	<input type="checkbox"/>					
22	CTX_Language		agent restart	<input type="checkbox"/>					
23	WSCollaborations		agent restart	<input type="checkbox"/>					

그림 39. JNDI 등록 정보

LookupQueuesUsingJNDI: 이 등록 정보의 값을 true로 설정할 경우, 커넥터의 SOAP/JMS 리스너 및 SOAP/JMS 프로토콜 핸들러는 JNDI를 사용하여 대기열을 찾습니다.

기본값 = false

JNDIProviderURL: 이 등록 정보는 JNDI 서비스 제공자의 URL을 제공합니다. 이는 SOAP/JMS 바인딩의 jms:address 요소에 대한 jndiProviderURL 속성에 해당합니다. jms:address 요소는 wsdl:port 섹션에 지정됩니다. 이는 기본 JNDI 제공자로 사용되며 유효한 JNDI URL이어야 합니다. 자세한 정보는 JNDI 스펙을 참조하십시오.

기본값 = 없음

InitialContextFactory: 이 등록 정보는 초기 문맥을 작성하는 팩토리 클래스의 완전한 클래스 이름(예: com.ibm.NamingFactory)을 제공합니다. 이는 SOAP/JMS 바인딩의 jms:address 요소에 대한 initialContextFactory 속성에 해당합니다. jms:address 요소는 wsdl:port 섹션에 지정됩니다.

기본값 = 없음

JNDIConnectionFactoryName: 이 등록 정보는 JNDI 문맥을 사용하여 찾을 연결 팩토리의 이름을 제공합니다. 이는 SOAP/JMS 바인딩의 jms:address 요소에 대한 jndiConnectionFactoryName 속성에 해당합니다. jms:address 요소는 wsdl:port 섹션에 지정됩니다.

기본값 = 없음

CTX_ObjectFactories: JNDI 문맥에서 보안 및 오브젝트 찾아보기에 대한 추가 정보를 지정하는 등록 정보. 표 42는 이러한 등록 정보가 요약되어 있습니다. + 문자는 등록 정보 계층 구조에서 항목의 위치를 표시합니다.

표 42. JNDI(Java Naming and Directory Interface) 제공자 등록 정보

등록 정보 이름	설명
+CTX_StateFactories	JNDI 문맥에서 보안 및 오브젝트 찾아보기에 대한 추가 정보를 지정하는 등록 정보. 자세한 정보는 J2EE 문서를 참조하십시오. 이 등록 정보는 JMS용 어댑터에서 사용하는 등록 정보를 반영합니다.
+CTX_URLPackagePrefixes	
+CTX_DNS_URL	
+CTX_Authoritative	
+CTX_Batchsize	
+CTX_Referral	
+CTX_SecurityProtocol	
+CTX_SecurityAuthentication	
+CTX_SecurityPrincipal	
+CTX_SecurityCredentials	
+CTX_Language	

복수 프로토콜 리스너 작성

프로토콜 리스너의 여러 인스턴스를 작성할 수 있습니다. 프로토콜 리스너는 ProtocolListenerFramework -> ProtocolListeners 커넥터 등록 정보의 하위 등록 정보로 구성됩니다. 각 하위(ProtocolListenerFramework -> ProtocolListeners의)는 커넥터마다 별도로 프로토콜 리스너를 식별합니다. 따라서, ProtocolListeners 등록 정보 아래에서 새로운 하위 등록 정보를 구성하여 프로토콜 리스너를 더 작성할 수 있습니다. 새로 작성한 리스너 등록 정보의 모든 하위 등록 정보를 지정했는지 확인하십시오. 각 리스너에 고유한 이름을 지정해야 합니다. 그러나 리스너 프로토콜 등록 정보(soap/http, soap/https 또는 soap/jms)는 변경하지 않으며, 리스너의 여러 인스턴스에서 동일하게 남아 있습니다.

주: 프로토콜 등록 정보가 스위치 역할을 하기 때문에 매우 중요합니다. 리스너나 핸들러를 사용하지 않으려면, 이 등록 정보를 비워두십시오.

SOAP/HTTP 또는 SOAP/HTTPS 리스너의 여러 인스턴스를 작성하고 있으면, 인스턴스마다 다른 포트나 호스트 등록 정보를 지정했는지 확인하십시오. 여러 SOAP/JMS 리스너를 지정하고 있으면, 인스턴스마다 반드시 다른 대기열 세트를 사용하십시오.

핸들러의 복수 인스턴스는 작성할 수 없습니다. 각 프로토콜에 대해 단 하나의 핸들러만이 존재할 수 있습니다.

시작 시의 커넥터

커넥터 시작 시 `init()` 메소드는 시스템 관리자의 Connector Configurator를 사용하여 설정한 구성 등록 정보를 읽습니다. 제대로 작동하게 하려면 커넥터 폴링을 사용 불가능으로 하지 마십시오(커넥터 폴링은 기본적으로 사용 가능합니다). 다음 섹션에서는 이와 관련된 발생사항에 대해 설명합니다.

프록시 설정

ProxyServer 커넥터 특정 등록 정보를 지정할 경우, 커넥터는 프록시 시스템 등록 정보를 설정합니다. 프록시 서버는 요청 처리용으로만 SOAP/HTTP-HTTPS 프로토콜 핸들러와 함께 사용됩니다. 커넥터는 또한 설정하는 시스템 등록 정보 각각을 추적합니다. ProxyServer 등록 정보에 대한 자세한 정보는 95 페이지의 『커넥터 특정 구성 등록 정보』를 참조하십시오.

JNDI 초기화

커넥터 특정 등록 정보 JNDI는 커넥터에서 사용할 JNDI를 지정합니다. 커넥터는 JNDI를 사용하여 JMS 연결 팩토리 오브젝트를 찾습니다. JNDI "LookupQueuesUsingJNDI"를 true로 설정하면 커넥터는 JNDI를 사용하여 JMS 대기열 오브젝트를 찾습니다.

SOAP/JMS(SOAP/JMS 프로토콜 리스너 및 SOAP/JMS 프로토콜 핸들러)를 사용하지 않으려면 JNDI 등록 정보를 지정할 필요가 없습니다. JNDI 등록 정보를 지정했지만 커넥터가 JNDI를 초기화할 수 없는 경우 커넥터를 종료합니다. 다음 커넥터 특정 JNDI 등록 정보를 모두 지정하지 않으면 커넥터는 JNDI를 초기화하지 않습니다.

- JNDIProviderURL
- InitialContextFactory
- JNDIConnectionFactoryName

주: JNDI 구현은 커넥터와 함께 제공되지 않습니다.

프로토콜 리스너 프레임워크 초기화

시작 시 커넥터는 프로토콜 리스너 프레임워크를 인스턴스화하고 이를 초기화합니다. 이 프레임워크는 커넥터 특정 등록 정보 ProtocolListenerFramework를 읽고, 커넥터는 WorkerThreads 및 RequestPoolSize 커넥터 등록 정보의 값을 읽습니다.

ProtocolListenerFramework 등록 정보가 지정되지 않거나 누락된 경우, 커넥터는 웹 서비스 클라이언트로부터 요청을 수신할 수 없으며 경고를 로그합니다.

그 다음 커넥터는 ProtocolListenerFramework -> ProtocolListeners 등록 정보를 읽습니다. ProtocolListeners 등록 정보의 첫 번째 레벨 등록 정보는 모두 프로토콜 리스너를 나타냅니다. 프로토콜 리스너 프레임워크는 리스너 각각을 로드하여 초기화하고 추적하려고 합니다. 영속적 이벤트 기능이 있는 경우, 리스너는 이벤트 복구를 시도합니다.

프로토콜 핸들러 프레임워크 초기화

커넥터는 커넥터 특정 등록 정보 ProtocolHandlerFramework를 읽고 프로토콜 핸들러 프레임워크를 인스턴스화하고 초기화합니다. 이 등록 정보가 누락되거나 적절하게 설정되지 않으면 커넥터는 요청 처리를 수행할 수 없고 경고를 로그합니다. 그 다음 커넥터는 프로토콜 핸들러에 해당하는 모든 ProtocolHandlerFramework " ProtocolHandlers 등록 정보를 읽은 다음 이를 로드, 초기화프로토콜 핸들러는 커넥터 초기화 중에 로드되며 협업이 서비스 요청을 수행할 경우에는 인스턴스화되지 않습니다. 프로토콜 핸들러에는 다중 스레드가 발생하지 않습니다.

로깅

커넥터는 다음 경우에 경고를 로그합니다.

- ProtocolListenerFramework 등록 정보가 지정되어 있지 않습니다. 커넥터는 이벤트 공고를 수행할 수 없음을 경고합니다(커넥터는 웹 서비스로 표시된 협업을 호출할 수 없습니다).
- ProtocolHandlerFramework 등록 정보가 지정되어 있지 않습니다. 커넥터는 (협업) 요청 처리를 수행할 수 없음을 경고합니다.

추적

추적은 커넥터 작동을 엄중하게 따르기 위해 켤 수 있는 선택적 디버깅 기능입니다. 기본적으로, 추적 메시지는 STDOUT에 기록됩니다. 추적 메시지 구성에 대한 자세한 정보는 커넥터 구성 등록 정보를 참조하십시오. 추적을 사용 가능하게 하고 설정하는 방법에 대한 자세한 정보는 *Connector Development Guide for Java*를 참조하십시오.

커넥터 추적 레벨은 다음과 같습니다.

- | | |
|------|--------------------------------------------------------------------------------------------------------------------|
| 레벨 0 | 이 레벨은 커넥터 버전을 식별하는 추적 메시지에 사용됩니다. |
| 레벨 1 | pollForEvents 메소드를 호출할 때마다 추적합니다. ICS 전달을 위해 리스너가 작성한 TLO 이름을 추적합니다. TLO에서 요청 Business Object 이름과 해당 속성 이름을 추적합니다. |
| 레벨 2 | gotAppEvent() 또는 executeCollaboration()에서 InterChange |

Server로 Business Object를 게시할 때마다 로그하는 추적 메시지에 이 레벨을 사용합니다. 또한 요청을 처리 중인 프로토콜 핸들러를 추적합니다.

레벨 3 처리 중인 Business Object의 ASI를 추적합니다. 처리 중인 Business Object의 속성을 추적합니다. 이벤트 공고 중에 SOAP 요청 Business Object의 TLO를 추적합니다. Data Handler에서 리턴한 Business Object를 추적합니다.

레벨 4 다음과 연관된 전송 머리글을 추적합니다.

- 전송에서 프로토콜 리스너가 검색한 SOAP 요청 메시지
- 프로토콜 리스너가 클라이언트로 전송한 응답 메시지

스레드 산출 항목, 처리된 모든 ASI, 중요한 함수의 모든 입력 및 종료 항목을 추적합니다.

레벨 5 다음을 추적합니다.

- 중요한 각 메소드의 입력 및 종료 항목
- 모든 구성 특정 등록 정보
- 각 프로토콜 리스너의 로딩
- 전송에서 프로토콜 리스너가 검색한 요청 메시지
- 전송에서 프로토콜 리스너가 클라이언트로 전송한 응답 메시지
- 각 프로토콜 핸들러의 로딩
- SOAP Data Handler에서 리턴된 메시지
- 협업으로 전송된 TLO의 Business Object 덤프
- Data Handler에서 리턴된 Business Object 덤프

제 5 장 SOAP Data Handler

- 『SOAP Data Handler 구성』
- 127 페이지의 『SOAP Data Handler 처리』
- 154 페이지의 『SOAP 양식 및 지침 사용』
- 155 페이지의 『XML 제한사항』

Data Handler는 1차 역할은 Business Object를 SOAP 메시지로 변환(반대의 경우 포함)하는 데이터 변환 모듈입니다. SOAP Data Handler는 다음 기능을 수행합니다.

- 요청 처리
 - SOAP 요청 Business Object를 SOAP 요청 메시지로
 - SOAP 응답 메시지를 SOAP 응답 Business Object로
 - SOAP 결합 메시지를 SOAP 결합 Business Object로
- 이벤트 처리
 - SOAP 요청 메시지를 SOAP 요청 Business Object로
 - SOAP 응답 Business Object를 SOAP 응답 메시지로
 - SOAP 결합 Business Object를 SOAP 결합 메시지로

이 장은 SOAP Data Handler를 구성하는 방법, SOAP Data Handler가 메시지 및 오브젝트를 처리하는 방법과 Data Handler를 사용자 정의하는 방법에 대해 설명합니다.

SOAP Data Handler 구성

SOAP Data Handler는 Connector for Web Services의 핵심 구성요소입니다. 커넥터는 Business Object를 웹 서비스 준수 SOAP 메시지로 변환하기 위해 Data Handler를 호출합니다.

협업이 웹 서비스로 표시될 때 커넥터는 SOAP Data Handler도 호출합니다. 그러면 Data Handler가 원격 Trading Partner(또는 내부 클라이언트)로부터 전송된 SOAP 메시지를 Business Object로 변환합니다. 커넥터는 해당 Business Object를 웹 서비스용으로 구성된 협업에 전달합니다.

Data Handler Meta Object의 정보가 이들 변환에서 결정적인 역할을 수행합니다. 제품 파일을 설치한 후 시작하기 전에 이 정보를 구성합니다. 사용자 정의 이름 핸들러를 추가하지 않으면 기본 SOAP Data Handler 구성을 사용하여 시간을 절약할 수 있습니다. 그러나 Data Handler 변환마다 특정 Meta Object 정보를 구성해야 합니다. Data Handler Meta Object는 다음 섹션에서 설명합니다.

Meta Object 요구사항

Meta Object는 구성 정보가 있는 Business Object입니다. 커넥터는 런타임 시 Meta Object를 사용하여 Data Handler를 구성하고 해당 인스턴스를 작성합니다. SOAP Data Handler는 또한 Meta Object를 사용하여 SOAP 메시지의 본문을 찾고, 본문과 대응되는 Verb와 Business Object를 판별하며, SOAP 메시지에서 Business Object를 인코드하고, 이 장에 설명된 여러 가지 다른 작업을 수행합니다. 이 섹션은 이들 Meta Object의 요구사항에 대해 설명합니다.

Meta Object 계층 구조 및 용어

그림 28에서는 Adapter for Web Services 제품에 대한 Meta Object 구조를 보여줍니다. Meta Object의 이름은 그림에서 굵은체로 표시되며 아래에 설명되어 있습니다.

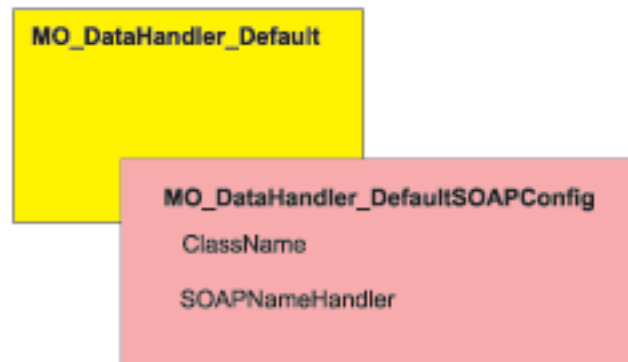


그림 40. Meta Object 구조

다음 용어가 Meta Object에 대해 논의할 때 이 책 전체에서 사용됩니다.

- **MO_DataHandler_Default** - 인스턴스화할 Data Handler를 판별하기 위해 커넥터 에이전트가 사용하는 Data Handler Meta Object. 이것은 커넥터의 `DataHandlerMetaObjectName` 등록 정보에 지정됩니다.
- **MO_DataHandler_DefaultSOAPConfig** - 특별히 SOAP Data Handler를 위한 하위 Data Handler Meta Object
- SOAP 구성 Meta Object(SOAP 구성 MO) - 각 SOAP Business Object의 하위 오브젝트로 지정되고 Business Object에서 SOAP 메시지(또는 그 반대)로의 단일 변환에 대한 구성 정보를 포함하는 Meta Object입니다.

MO_DataHandler_Default

MO_DataHandler_Default는 커넥터에서 호출되는 모든 Data Handler에 대한 최상위 레벨 Meta Object입니다. 이들 Meta Object에 포함된 MIME 유형은 사용할 Data Handler를 판별합니다. 커넥터 에이전트는 이 Meta Object를 사용하여 SOAP Data

Handler의 인스턴스를 작성합니다. 따라서, MO_DataHandler_Default 오브젝트에는 MO_DataHandler_DefaultSOAPConfig 유형의 xml_soap 속성이 있어야 합니다.

MO_DataHandler_Default 오브젝트를 설치한 후 구성할 수 있습니다. 유형 MO_DataHandler_DefaultSOAPConfig의 xml_soap를 추가해야 합니다.

MO_DataHandler_DefaultSOAPConfig

캐릭터 에이전트는 이 Meta Object를 사용하여 런타임 시 SOAP Data Handler를 작성 및 구성합니다. MO_DataHandler_DefaultSOAPConfig에는 다음을 지정하는 유형 string의 두 속성이 있습니다.

- SOAP Data Handler의 클래스 이름
- SOAP 이름 핸들러

이들 속성이 표 43에 표시됩니다

이 장에서 나중에 설명하는 사용자 정의 이름 핸들러를 구현하려는 경우가 아니면, MO_DataHandler_DefaultSOAPConfig를 제공되고 설치된 대로 사용할 수 있습니다. 구성할 필요가 없습니다.

표 43. MO_DataHandler_DefaultSOAPConfig에 대한 Meta Object 속성

이름	유형	기본값	설명
ClassName	문자열	com.ibm.adapters .dataHandlers.xml.soap	createHandler 메소드에 전달된 MIME 유형을 기반으로 클래스 이름을 찾기 위해 Data Handler 기본 클래스가 사용하는 표준 속성.
SOAPName Handler	문자열		사용할 SOAP 이름 핸들러의 이름

SOAP 구성 Meta Object: 모든 SOAP Business Object의 하위 오브젝트

SOAP 구성 MO는 하나의 Data Handler 변환(SOAP 메시지에서 Business Object로, 또는 Business Object에서 SOAP 메시지로의 변환)을 정의합니다. SOAP 구성 MO는 SOAP Business Object의 하위 오브젝트입니다. 이들 하위 SOAP 구성 MO는 기본 Business Object 해석에 중요합니다. 기본 Business Object 해석을 사용할 때, 모든 하위 SOAP 구성 MO에는 요청, 응답 또는 결합 오브젝트에 대한 것인지에 상관없이 BodyName 및 BodyNS의 기본값에 대해 고유한 항목이 있어야 합니다. 표 44에서는 SOAP 구성 MO의 이들 속성 및 기타 속성을 보여줍니다.

표 44. SOAP Config MO의 속성

이름	필수	설명
BodyNS	예	SOAP 본문에 사용할 이름 공간
BodyName	예	SOAP 메시지의 본문 이름. SOAP 결합의 경우 기본값을 soap:fault로 설정하십시오.
BOVerb	예	SOAP 구성 MO를 포함하는 Business Object의 Verb
TypeInfo	아니오	유형 정보(xsi:type)를 SOAP 요소에 기록하고 읽을지 여부를 지시하는 true 또는 false 속성. 기본값 = false
TypeCheck	아니오	TypeInfo를 true로 설정한 경우에만 이 등록 정보를 읽습니다. 가능한 값은 none 및 strict입니다. none일 경우, Business Object로 SOAP 메시지를 읽어들이는 때 유형의 유효성 검증을 건너뛵니다. strict일 경우, Data Handler는 Business Object의 응용프로그램 특정 정보에 대해 모든 SOAP 유형 이름과 이름 공간의 유효성을 엄격하게 검증합니다. 기본값 = none
Style	아니오	이 등록 정보는 SOAP 메시지 양식을 지시하며 BodyName 및 BodyNS와 같은 다른 속성에 대한 암시가 들어 있습니다. 이 속성의 가능한 값은 rpc 및 document입니다. 기본값 = rpc
사용	아니오	이 등록 정보는 SOAP 메시지의 사용을 지시하며 Business Object에서 SOAP 본문을 구성하는 방식에 영향을 줍니다. 가능한 값은 literal 및 encoded입니다. 기본값은 literal입니다.

그림 41에서는 SOAP Business Object와 SOAP Config MO의 관계를 보여줍니다.

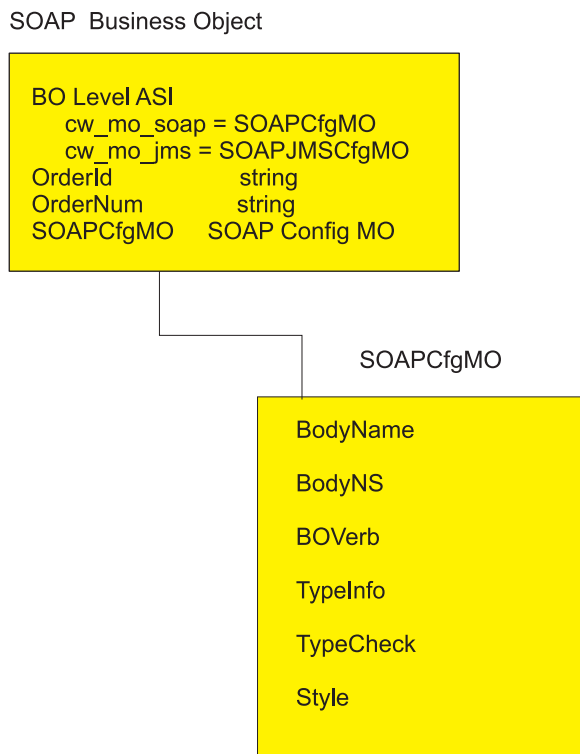


그림 41. SOAP 구성 Meta Object

그림 41에서는 SOAP 응답 Business Object 및 해당 하위 Business Object를 보여줍니다. 하위 Business Object인 SOAPCfgMO는 Business Object 응답에서 SOAP 응

답 메시지로의 변환을 위해 SOAP Data Handler 작동을 지정하는 SOAP 구성 MO입니다. 하위 SOAP 구성 MO를 표시하는 속성은 cw_mo_soap로 시작하는 이름-값 쌍을 사용해야 합니다.

일반적으로 cw_mo_로 시작하는 Business Object 레벨의 응용프로그램 특정 정보를 읽을 때 Data Handler는 이름-값 쌍에 지정한 하위 오브젝트에 변환 Meta Object 정보가 있음을 인식하므로 변환 중인 메시지 본문에 이 하위 오브젝트를 내용으로 포함하지 않습니다. 예에서 이름-값 쌍인 cw_mo_jms 및 cw_mo_soap로 표시된 하위 오브젝트는 Meta Object로 인식되므로 SOAP 응답 메시지에 기록되지 않습니다. 또한 SOAP Data Handler는 cw_mo_soap를 제외하고 cw_mo_로 시작하는 모든 Business Object 레벨의 응용프로그램 특정 정보를 무시합니다. 따라서, SOAP Data Handler는 cw_mo_tpi 같은 응용프로그램 특정 정보를 무시합니다. 그러나 SOAP Data Handler는 cw_mo_soap에 지정한 SOAP 구성 MO를 읽고 이를 사용하여 Business Object에서 SOAP 메시지로의 SOAP 응답 변환을 실행합니다.

모든 SOAP Business Object에는 하위 SOAP 구성 MO가 있어야 하며 이 하위 오브젝트는 Business Object 레벨에서 응용프로그램 특정 정보로 지정되어야 합니다. 이 중 대부분은 자동화됩니다. WSDL ODA를 사용하여 SOAP 메시지용 Business Object를 생성할 때 SOAP 구성 MO는 자동으로 생성됩니다.

SOAP 메시지의 Style 및 Use 영향

SOAP 구성 MO 선택적 등록 정보, Style 및 Use는 SOAP 메시지가 작성된 방식에 영향을 미칩니다. Style의 가능한 값은 rpc 및 document이며, Use의 가능한 값은 literal 및 encoded입니다. 아래 섹션에서는 Style 및 Use 조합이 SOAP 메시지 작성에 영향을 미치는 방식을 설명합니다.

rpc/literal: Style 등록 정보를 rpc로 설정하고 Use 등록 정보를 literal로 설정하면, SOAP 메시지의 본문 이름과 본문 이름 공간은 각각 SOAP ConfigMO의 BodyName 및 BodyNS 등록 정보에서 읽혀집니다.

다음은 본문 이름 및 본문 이름 공간이 각각 getOrderStatus 및 OrderStatusNS로 해석되는 rpc/literal 양식 메시지의 예입니다.

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getOrderStatus xmlns:ns1="http://www.ibm.com/">
      <Part1>
        <ns2:Elem1 xmlns:ns2="http://www.ibm.com/element1">
          <Child1>1</Child1>
          <Child2>2</Child2>
        </ns2:Elem1>
        <ns3:Elem1 xmlns:ns3="http://www.ibm.com/element1">
          <Child1>3</Child1>
          <Child2>4</Child2>
        </ns3:Elem1>
      </Part1>
    </ns1:getOrderStatus>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        <Elem2>10</Elem2>
    </Part1>
</ns1:getOrderStatus>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

그림 42에서는 이 rpc/literal 메시지에 해당하는 Business Object를 보여줍니다.

Name	Type	Key	Card	Default	App Spec Info
☐ Part1	SOAP_Part1Type	<input checked="" type="checkbox"/>	1		
☐ Elem1	SOAP_MaxType	<input checked="" type="checkbox"/>	N		maxoccurs=5;elem_ns=http://www.ibm.com/elem1
Child1	String	<input checked="" type="checkbox"/>			
Child2	String	<input type="checkbox"/>			
ObjectEventId	String				
Elem2	String	<input type="checkbox"/>			
ObjectEventId	String				
☐ SOAPConfigMO	SOAP_Req_Cfg_MO	<input type="checkbox"/>	1		
BodyName	String	<input checked="" type="checkbox"/>		getOrderStatus	
BodyNS	String	<input type="checkbox"/>		http://www.ibm.com	
BOVerb	String	<input type="checkbox"/>		Retrieve	
TypeInfo	String	<input type="checkbox"/>		false	
TypeCheck	String	<input type="checkbox"/>		none	
Style	String	<input type="checkbox"/>		rpc	
Use	String	<input type="checkbox"/>		literal	

그림 42. rpc/literal SOAP 구성 MO

주: 해당 SOAP 메시지가 작성되도록 이 등록 정보 및 Business Object 속성을 적절하게 구성해야 합니다.

rpc/encoded: Style 등록 정보가 rpc로 서정되고 Use가 encoded로 설정되면, SOAP 메시지의 본문 이름과 본문 이름 공간은 각각 Child ConfigMO의 BodyName 및 BodyNS 등록 정보에서 읽혀집니다. 또한 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" 속성이 본문 태그에 추가됩니다.

다음은 본문 이름 및 본문 이름 공간이 각각 getOrderStatus 및 OrderStatusNS로 해석되는 rpc/encoded 메시지의 예입니다.

```

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

```



```

<SOAP-ENV:Body SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getOrderStatus xmlns:ns1="http://www.ibm.com/">
    <Part1 xsi:type="ns1:SOAP_Part1Type">
      <ns2:Elem1 SOAP-ENC:arrayType="ns2:SOAP_MaxType[2]"
        xsi:type="SOAP-ENC:Array" xmlns:ns2="http://www.ibm.com/Elem1">
        <item>
          <Child1 xsi:type="xsd:string">1</Child1>
          <Child2 xsi:type="xsd:string">2</Child2>
        </item>
        <item>
          <Child1 xsi:type="xsd:string">3</Child1>
          <Child2 xsi:type="xsd:string">4</Child2>
        </item>
      </ns2:Elem1>
      <Elem2 xsi:type="xsd:string">10</Elem2>
    </Part1>
  </ns1:getOrderStatus>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

그림 43에서는 이 rpc/encoded 메시지에 해당하는 Business Object를 보여줍니다.

Name	Type	Key	Card	Default	App Spec Info
☐ Part1	SOAP_Part1Type	<input checked="" type="checkbox"/>	1		
☐ Elem1	SOAP_MaxType	<input checked="" type="checkbox"/>	N		elem_ns=http://www.ibm.com/Elem1
Child1	String	<input checked="" type="checkbox"/>			
Child2	String	<input type="checkbox"/>			
ObjectEventId	String				
Elem2	String	<input type="checkbox"/>			
ObjectEventId	String				
☐ SOAPConfigMO	SOAP_Req_Cfg_MO	<input type="checkbox"/>	1		
BodyName	String	<input checked="" type="checkbox"/>		getOrderStatus	
BodyNS	String	<input type="checkbox"/>		http://www.ibm.com	
BOVerb	String	<input type="checkbox"/>		Retrieve	
TypeInfo	String	<input type="checkbox"/>		true	
TypeCheck	String	<input type="checkbox"/>		none	
Style	String	<input type="checkbox"/>		rpc	
Use	String	<input type="checkbox"/>		encoded	

그림 43. rpc/encoded SOAP 구성 MO

document/literal: Style 등록 정보를 document로 설정하고 Use 등록 정보를 literal로 설정하면, 둘러싸는 모든 본문 이름 태그가 존재하지 않습니다. 다음은 위의 BO에 기초하는 문서 양식 SOAP 메시지의 예입니다.

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:Elem1 xmlns:ns1="http://www.ibm.com/elem1">
      <Child1>1</Child1>
      <Child2>2</Child2>
    </ns1:Elem1>
    <ns2:Elem1 xmlns:ns2="http://www.ibm.com/elem1">
      <Child1>3</Child1>
      <Child2>4</Child2>
    </ns2:Elem1>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

그림 44에서는 이 document/literal 메시지에 해당하는 Business Object를 보여줍니다.

Name	Type	Key	Card	Default	App Spec Info
☐ Elem1	SOAP_Elem1	<input checked="" type="checkbox"/>	1		maxoccurs=3;elem_ns=http://www.ibm.com/elem1
Child1	String	<input checked="" type="checkbox"/>			
Child2	String	<input type="checkbox"/>			
ObjectEventId	String				
☐ SOAPConfigMO	SOAP_Req_Cfg_MO	<input type="checkbox"/>	1		
BodyName	String	<input checked="" type="checkbox"/>		getOrderStatus	
BodyNS	String	<input type="checkbox"/>		http://www.ibm.com	
BOVerb	String	<input type="checkbox"/>		Retrieve	
TypeInfo	String	<input type="checkbox"/>		false	
TypeCheck	String	<input type="checkbox"/>		none	
Style	String	<input type="checkbox"/>		document	
Use	String	<input type="checkbox"/>		literal	

그림 44. document/literal SOAP 구성 MO

XML 코드 단편의 encodingStyle 속성이 설정되지 않았음을 참고하십시오.

document/encoded: 이 Style/Use 조합은 지원되지 않습니다. Style이 document로 설정되고 Use가 encoded로 설정된 SOAP ConfigMO가 발생한 경우, Data Handler가 실패합니다.

SOAP Data Handler 처리

SOAP Data Handler는 다음 방법으로 SOAP 메시지와 Business Object 사이의 변환을 수행합니다.

- **SOAP 메시지를 Business Object로 처리**
 - 웹 서비스 클라이언트가 웹 서비스로 표시된 협업을 호출할 때 이벤트 처리 시 해당 단계에서 요청 메시지를 SOAP 요청 Business Object로 데이터 처리가 발생합니다.
 - 웹 서비스가 호출한 협업으로 SOAP 응답 메시지를 리턴할 때 요청 처리 중에 응답 메시지를 SOAP 응답 Business Object로 데이터 처리가 발생합니다. 또는 이 단계에서 결합 메시지를 SOAP Business Object로 데이터 처리가 발생할 수 있습니다.

이러한 처리에 대한 자세한 설명은 이 섹션의 뒤에 있는 『SOAP 본문 메시지 대 Business Object 처리』를 참조하십시오.

- **Business object를 SOAP 메시지로 처리**
 - 웹 서비스로 표시된 협업에서 응답 Business Object를 리턴할 때 이벤트 처리 중에 Business Object를 SOAP 응답 메시지로 데이터 처리가 발생합니다. 또는 이 단계에서 결합 Business Object를 SOAP 결합 메시지로 데이터 처리가 발생할 수 있습니다.
 - 협업을 Business Object를 SOAP 요청 메시지로 변환하도록 커넥터를 서비스 호출을 수행할 때 요청 처리 단계에서 Business Object를 SOAP 요청 메시지로 데이터 처리가 발생합니다.

이러한 처리에 대한 자세한 설명은 이 섹션의 뒤에 있는 130 페이지의 『Business Object 대 SOAP 메시지 본문 처리』를 참조하십시오.

SOAP 본문 메시지 대 Business Object 처리

이 섹션은 SOAP 본문 메시지를 Business Object로 변환의 단계별 설명을 제공합니다.

1. SOAP Data Handler가 SOAP 메시지를 수신합니다.
2. Apache SOAP API를 사용하여 Data Handler가 SOAP 메시지를 구문 분석합니다.
3. Data Handler가 SOAP 메시지의 구성요소인 봉투, 머리글 및 본문을 추출합니다.
4. 머리글 처리에 대한 자세한 정보는 129 페이지의 『SOAP 머리글 메시지 대 Business Object 처리』를 참조하십시오.
5. 본문 처리 - Data Handler는 SOAP 본문의 첫 번째 요소를 읽어 이 요소가 결합을 전달하는지 또는 데이터를 전달하는지 여부를 판별합니다. 본문 내용이 결합이 아닌 경우, Data Handler는 다음을 수행합니다.

- a. Business Object를 변환에 사용할지 여부를 판별하도록 Business Object 분석을 수행합니다. 사용자 정의 이름 핸들러를 구성한 경우, 아래에 설명된 기본 Business Object 분석은 적용되지 않을 수 있습니다. 플러그 가능한 이름 핸들러 지정에 대한 자세한 정보는 153 페이지의 『플러그 가능 이름 핸들러 지정』을 참조하십시오.
 - b. Data Handler는 또한 변환에 사용할 SOAP 구성 MO(Data Handler가 작성 중인 SOAP Business Object의 하위 오브젝트)도 분석합니다. SOAP 구성 MO의 인스턴스가 없으면 Data Handler는 인스턴스를 작성하고 기본값을 읽습니다. ConfigMO 속성 값에서 Data Handler는 Business Object Verb를 읽습니다. Data Handler는 SOAP Business Object를 인스턴스화하고 해당 Verb를 설정합니다. 이는 Data Handler가 SOAP 메시지를 작성하려고 하는 Business Object입니다.
 - c. Data Handler는 SOAP 메시지를 한 번에 한 요소씩 계속 구문 분석합니다. rpc의 경우, Data Handler는 첫 번째 요소가 상위일 것으로 예상합니다.
 - d. Data Handler는 Business Object의 속성(또는 응용프로그램 특정 정보, 자세한 정보는 135 페이지의 『Business Object를 SOAP 메시지로 변환의 ASI』 참조)의 이름이 하위 요소와 이름과 동일할 것으로 예상합니다. 속성이 Business Object에 없으면 Data Handler는 예외를 발생합니다. 하위 요소는 단순 유형이거나 복합 유형일 수 있습니다. 복합 요소는 하위 요소가 있는 요소입니다.
 - e. 단순 요소 - 하위 요소가 단순 요소인 경우, 기본적으로 Data Handler는 단순 요소의 이름과 동일한 이름(또는 ASI)이 있는 Business Object 속성을 예상합니다. Data Handler는 단순 요소의 값을 읽고 Business Object에 이 값을 설정합니다.
 - f. 복합 요소 - 하위 요소가 복합 유형인 경우, Data Handler는 Business Object가 하위 Business Object와 이름(또는 ASI) 및 유형이 같은 속성이 있는 것으로 예상합니다. 이 속성은 복합 SOAP 요소 또는 SOAP 배열이 있을지 여부에 따라서 단일 카디널리티 또는 복수 카디널리티일 수 있습니다. 그 다음 Data Handler는 하위 Business Object(기본적으로, 속성의 유형은 하위 Business Object의 이름을 제공함)를 인스턴스화하고 복합 요소의 모든 하위 요소를 읽어 하위 Business Object에 값을 설정합니다. Data Handler는 이 속성의 카디널리티를 확인한 후 하위 Business Object를 상위 Business Object 속성으로 설정합니다. 속성이 카디널리티 n인 경우, Data Handler는 이 Business Object를 컨테이너에 추가합니다. 복합 요소에는 단순 또는 복합 하위 요소가 있을 수 있습니다. 이들 요소 또한 같은 방식으로 처리됩니다. 단순 요소일 경우에는 Data Handler는 하위 BO에서 값을 설정하고, 복합 요소일 경우에는 Data Handler는 하위 Business Object를 인스턴스화합니다.
6. 결함 처리 - Data Handler는 SOAP 본문의 첫 번째 요소의 이름을 읽고 이 요소가 결함인지 여부를 판별합니다. 첫 번째 요소의 이름이 Fault인 경우, Data Handler는 이것이 결함 메시지라고 결론짓습니다. 결함 Business Object 분석은 결함 메시

지가 변환되어야 하는 Business Object를 판별하기 위해 발생합니다. 그러면 Data Handler는 본문 처리와 같은 처리 과정을 수행합니다. Data Handler는 하위 Business Object에서 지정한 Business Object에 다음 속성이 있는 것으로 예상합니다.

- a. faultcode: 필수. String 속성
 - b. faultstring: 필수. String 속성
 - c. faultactor: 필수가 아님. String 속성
 - d. detail: 필수가 아님. 하위 BO
7. 어떠한 이유로 결함 처리에 실패할 경우, 발생한 예외에는 SOAP 결함 메시지에 있는 faultcode, faultstring 및 faultactor 요소의 텍스트가 포함됩니다.

주: 결함 메시지의 SOAP 스펙에 따라 faultcode, faultstring 및 faultactor는 단순 요소인 반면 detail은 복합 요소(하위 요소가 있는 요소)입니다. 또한 faultcode, faultstring, faultactor 및 detail은 SOAP 봉투 이름 공간에 속하는 반면 detail 하위 요소는 사용자 정의 이름 공간에 속할 수 있습니다.

SOAP 머리글 메시지 대 Business Object 처리

이 섹션은 Data Handler가 SOAP 메시지의 머리글을 Business Object로 변환하는 방법에 대해 설명합니다.

1. SOAP Data Handler가 SOAP 메시지의 본문을 처리합니다. 본문 처리에서 SOAP Business Object가 작성됩니다.
2. SOAP 메시지에 SOAP 머리글 요소가 있으면 SOAP Data Handler는 본문 처리에서 얻은 Business Object에서 SOAP 머리글 속성을 예상합니다. SOAPHeader 속성은 Business Object의 하위 속성이며, 그의 응용프로그램 특정 정보로서 soap_location=SOAPHeader를 포함합니다. 이러한 속성이 없는 경우, SOAP Data Handler는 오류를 발생시킵니다. SOAPHeader 속성은 SOAP 머리글 컨테이너 Business Object 유형이어야 합니다. SOAP Data Handler가 1단계에서 얻은 SOAP Business Object에서 이 속성의 인스턴스를 작성합니다.
3. SOAP-Env:Header 요소의 각 바로 아래 하위에 대한 자세한 정보는 다음과 같습니다.
 - a. Data Handler는 SOAP 머리글 컨테이너 Business Object에서 하위 속성을 예상합니다. 이 속성의 이름은 머리글 요소의 이름과 동일해야 하며 SOAP 머리글 하위 Business Object를 준수해야 합니다. Data Handler가 이러한 속성을 찾을 수 없는 경우, 오류를 발생시킵니다. 따라서 이 요소의 이름 공간이 이 속성의 elem_ns 응용프로그램 특정 정보에 지정된 이름 공간과 동일해야 합니다. 동일하지 않으면, Data Handler는 오류를 발생시킵니다.
 - b. Data Handler는 SOAP 머리글 하위 Business Object의 인스턴스를 작성하고 그것을 2단계에서 작성된 SOAP 머리글 컨테이너 Business Object의 인스턴스에 배치합니다.

- c. 이 머리글 요소에 actor 속성이 있는 경우, Data Handler는 위에서 작성된 하위 Business Object에 actor 속성이 있다고 예상합니다. actor 속성을 발견하지 못하는 경우, Dats Handler는 오류를 발생시킵니다.

주: actor 속성을 추가하려는 경우, 139 페이지의 『SOAP 속성 지정』을 참조하십시오.

- d. 이 머리글 요소에 mustUnderstand 속성이 없는 경우, Dats Handler는 mustUnderstand 속성이 위에서 작성된 하위 Business Object에 있다고 예상합니다. mustUnderstand 속성을 발견하지 못하는 경우, Dats Handler는 오류를 발생시킵니다.

주: mustUnderstand 속성을 추가하려는 경우, 139 페이지의 『SOAP 속성 지정』을 참조하십시오.

- e. 이 머리글 요소의 각 하위 요소에 대해 Dats Handler는 하위 Business Object에서 동일한 이름이 있는 속성을 예상합니다. 이들 요소는 SOAP-Env:Body 요소의 하위 요소와 동일한 방법으로 처리됩니다.

Business Object 대 SOAP 메시지 본문 처리

다음은 Business Object를 SOAP 본문 메시지로 변환의 단계별 설명입니다. 응용프로그램 특정 정보와 관련된 특수한 경우에는 135 페이지의 『Business Object를 SOAP 메시지로 변환의 ASI』를 참조하십시오.

1. SOAP Data Handler는 변환 중인 SOAP Business Object에 해당하는 SOAP ConfigMO를 찾습니다.
2. Dats Handler가 SOAP 메시지의 봉투 및 머리글을 작성합니다.
3. Data Handler는 SOAP ConfigMO를 분석합니다. SOAP ConfigMO의 인스턴스가 없으면 Data Handler는 인스턴스를 작성하고 기본값에서 읽습니다. 기본적으로, Data Handler는 SOAP ConfigMO의 BodyName 속성 값을 읽어서 결함 Business Object를 처리 중인지 판별합니다. 이 값이 soap:fault로 설정되면 Business Object는 SOAP 결함 Business Object로 간주됩니다. 결함 Business Object가 아닌 경우 Data Handler는 아래의 본문 작성에 설명된 처리를 수행하고, 그렇지 않으면 결함 작성에 설명된 처리를 수행합니다.
4. 본문 작성 - 다음 단계는 Business Object로부터 SOAP 메시지의 본문을 작성하기 위해 Data Handler가 수행하는 처리를 자세히 설명합니다.
 - Data Handler는 SOAP ConfigMO 속성에서 BodyName 및 BodyNS를 가져온 후에 SOAP 메시지의 본문의 첫 번째 (상위) 요소를 작성합니다. 첫 번째 요소의 이름은 기본적으로 BodyName의 값입니다. 이 문서에서는 이 요소를 본문 요소라고 합니다. 본문 요소의 이름 공간은 기본적으로 BodyNS에서 판별된 값입니다. SOAP ConfigMO의 양식 속성이 document로 설정되는 경우, 이 단계 (첫 번째 본문 요소 작성)는 건너뛸니다.

- 그런 다음, Dats Handler가 Business Object의 속성을 읽고 유형에 의해 이를 처리합니다. 속성의 각 유형에 대한 처리가 아래에 설명되어 있습니다.
 - 단순 속성 - 속성이 단순 유형이면 Data Handler는 본문 요소에서 속성과 같은 이름의 하위 요소를 작성합니다(특수 응용프로그램 특정 정보에 달리 지정되지 않은 경우). Dats Handler는 이 요소의 값을 Business Object에 있는 속성 값으로 설정합니다.
 - 카디널리티 1 하위 Business Object 속성
속성이 단일 카디널리티 하위 Business Object인 경우 Dats Handler는 본문 요소의 하위 요소를 작성합니다. 이 요소를 하위 Business Object 요소라고도 합니다. 작성되는 하위 요소의 이름은 속성과 동일합니다(특수 ASI 등록 정보에 달리 지정되지 않은 경우). Data Handler는 하위 Business Object의 속성을 고려하여 수신 Business Object의 속성을 처리하는 것과 동일한 방식으로 속성의 하위 요소를 작성합니다. 그러나 하위 요소는 본문 요소가 아니라 하위 Business Object 요소의 하위로 작성됩니다.
 - 카디널리티 n 하위 Business Object 속성 - 속성이 카디널리티 n 하위 Business Object인 경우, Data Handler는 SOAP 배열을 작성합니다. 각 속성은 단일 카디널리티 하위 Business Object가 처리되는 것과 동일한 방식으로 처리됩니다.
- 5. 결합 작성 - 다음 섹션에서는 Data Handler가 결합 메시지를 작성하는 프로세스에 대해합니다.
 - Data Handler는 Business Object에서 다음의 속성을 예상합니다.
 - faultcode: 필수, String 속성
 - faultstring: 필수, String 속성
 - faultactor: 필수가 아님. String 속성
 - detail: 필수가 아님. 하위 BO 속성.
 임의의 필수 속성이 누락되면 Data Handler 오류가 발생합니다.
 - Data Handler는 faultcode에 대한 요소를 작성합니다. Business Object의 faultcode 속성에서 지정된 값을 설정합니다.
 - Data Handler는 faultstring에 대한 요소를 작성합니다. Business Object의 faultstring 속성에서 지정된 값을 설정합니다.
 - Data Handler는 faultactor를 작성합니다. Business Object의 faultactor 속성에서 지정된 값을 설정합니다.
 - 세부사항 속성이 Business Object에 있으면 속성은 하위 Business Object 유형이어야 합니다. 그렇지 않으면, Data Handler 오류가 발생합니다. 위의 본문 작성 섹션에서 강조된 것처럼 각각의 세부 Business Object의 속성을 처리합니다.
- 6. CxIgnore 처리 - Data Handler가 속성 값이 CxIgnore로 설정된 것을 확인하는 경우, Data Handler는 이 속성에 대한 요소를 작성하지 않습니다.

7. **CxBlank** 처리 - Data Handler가 속성 값이 CxBlank로 설정된 것을 확인하는 경우, Data Handler는 이 속성에 대한 요소를 작성하지만 값은 설정하지 않습니다.

Business Object 대 SOAP 메시지 머리글 처리

이 섹션은 SOAP 머리글 속성만 처리에 대해 설명합니다. 다른 모든 속성은 130 페이지의 『Business Object 대 SOAP 메시지 본문 처리』에서 설명한 것과 같이 처리됩니다.

1. Data Handler는 Business Object에서 SOAPHeader 속성을 얻습니다. 이 속성에는 그 응용프로그램 특정 정보로서 soap_location=SOAPHeader가 있습니다. SOAP Data Handler는 이 속성이 널(null)이 아닌 경우에만 SOAP-Env:Header 요소를 작성합니다. Business Object가 둘 이상의 SOAPHeader 속성을 포함하는 경우, 첫 번째 속성이 처리되고 나머지는 본문의 파트로서 취급됩니다.
2. Data Handler는 SOAPHeader 속성이 SOAP 머리글 컨테이너 Business Object를 표시하는 단일 카디널리티 하위라고 예상합니다. Data Handler는 SOAP 머리글 하위 Business Object의 유형이 있는 SOAP 머리글 컨테이너 Business Object의 하위 속성을 처리합니다.
3. SOAP 머리글 컨테이너 Business Object의 각 속성에 대해 Data Handler가 다음을 수행합니다.
 - a. 카디널리티를 검사합니다. 이 속성이 1 또는 n 카디널리티 하위 오브젝트가 아닌 경우 무시됩니다.
 - b. 값을 검사합니다. 이 속성의 값이 NULL인 경우 속성은 무시됩니다.
 - c. 속성이 1 또는 n 카디널리티 하위 오브젝트인 경우, Data Handler는 단계 1에서 작성된 SOAP-Env:Header 요소의 바로 아래 하위인 머리글 요소를 작성합니다. 이 머리글 요소의 이름은 속성의 이름과 동일합니다. 이 요소의 이름 공간은 이 속성의 elem_ns 응용프로그램 특정 정보에 의해 제공됩니다.
 - d. 속성이 SOAP 머리글 하위 Business Object인 경우, 이 Business Object의 모든 속성이 처리됩니다. 이 속성에는 actor 및 mustUnderstand 속성이 있을 수 있습니다.

주: mustUnderstand 또는 actor 속성을 추가하려는 경우, 139 페이지의 『SOAP 속성 지정』을 참조하십시오.
 - e. SOAP 머리글 하위 Business Object에 널(null)이 아닌 actor 속성이 아닌 경우, Data Handler는 단계 c에서 작성된 머리글 요소에 actor 속성을 작성합니다.
 - f. SOAP 머리글 하위 Business Object에 널이 아닌 mustUnderstand 속성이 있는 경우, Data Handler는 c단계에서 작성된 머리글 요소에 mustUnderstand 속성을 작성합니다.

- g. SOAP 머리글 하위 Business Object의 널이 아닌 다른 모든 속성은 이 머리글 요소의 하위 요소가 됩니다. 해당 요소는 SOAP-Env:Body 요소의 하위 요소와 동일한 방식으로 작성됩니다.

머리글 결합 처리

SOAP 스펙은 머리글과 관련된 오류가 머리글에 리턴되어야 한다고 명시합니다. 이들 머리글은 SOAP 결합 메시지로 리턴됩니다. 메시지 머리글이 요청 및 응답 Business Object의 SOAPHeader 속성에 지정되는 것처럼, 결합 머리글은 결합 Business Object의 SOAPHeader 속성에 지정됩니다.

요청 또는 응답 Business Object의 가능한 각 머리글이 오류를 유발할 수 있습니다. 이러한 오류는 결합 메시지의 머리글에 보고됩니다.

WSDL 문서에는 사용자가 결합 머리글을 지정할 수 있는 SOAP 바인딩 머리글 요소가 있습니다. 자세한 정보는 1장에 나열된 SOAP 및 WSDL 스펙을 참조하십시오.

headerfault의 응용프로그램 특정 정보를 사용하여 사용자의 각 머리글에 대한 머리글 결합을 지정할 수 있습니다. SOAP 머리글 컨테이너 Business Object의 각 속성에 대해 headerfault 응용프로그램 특정 정보를 지정할 수 있습니다. 결합 Business Object에 대한 SOAP 머리글 컨테이너 Business Object에 있는 속성 목록은 다음과 같습니다.

headerfault=attr1, attr2, attr3...

WSDL 구성 마법사가 요청 또는 응답 오브젝트의 SOAP 머리글 하위 Business Object에서 headerfault 응용프로그램 특정 정보를 찾는 경우, 유틸리티는 WSDL에 이들 머리글에 대해 생성되는 headerfault 요소를 작성합니다. WSDL을 사용하여 사용자의 각 요청(입력) 및 응답(출력) 머리글에 여러 머리글 결합을 지정할 수 있습니다. 그러므로 이 응용프로그램 특정 정보의 값은 쉼표로 구분된 속성 목록입니다.

응용프로그램 특정 정보 기능 사용

오브젝트 및 속성 레벨의 응용프로그램 특정 정보(ASI)를 지정하여 SOAP Data Handler 기능을 확장 및 향상시킬 수 있습니다. 표 45에서는 아래 섹션에서 설명된 속성을 보여줍니다. 표의 모든 항목은 별도로 언급되지 않는 한 속성 레벨 ASI입니다.

표 45. SOAP 오브젝트 ASI 요약

ASI	가능한 값	설명
soap_location	SOAPHeader	머리글 속성과 같은 이 Business Object 속성을 지정합니다.
headerfault	String	결합 BO에서 해당하는 SOAP 머리글의 BO 속성 이름을 식별합니다.

표 45. SOAP 오브젝트 ASI 요약 (계속)

ASI	가능한 값	설명
elem_name	String	BO 속성에 해당하는 SOAP 요소의 이름을 지정합니다.
elem_ns	String	BO 속성에 해당하는 SOAP 요소의 이름 공간을 지정합니다.
type_name	String	BO 속성에 해당하는 SOAP 요소의 유형을 지정합니다.
type_ns	String	BO 속성에 해당하는 요소의 유형 이름 공간을 지정합니다.
xsdtype	true	이전 xsd 버전(예: 1999, 2000 등)을 xsd의 최신 버전(예를 들어, 2001)으로 대체하면서, BO 속성에 해당하는 요소의 이름 공간으로 xsd를 지정합니다.
attr_name	String	BO 속성에 해당하는 SOAP 속성의 이름을 지정합니다.
attr_ns	String	BO 속성에 해당하는 SOAP 속성의 이름 공간을 지정합니다.
arrayof	String	요소의 배열을 작성하는 데 사용해야 하는 n 카디널리티 하위 Business Object 속성의 이름을 지정합니다.
dh_mimetype	String	복합 유형의 속성을 변환하는 데 사용할 Data Handler의 mimeType을 지정합니다.
cw_mo_*	String	이 Business Object 레벨 ASI는 SOAP Data Handler를 통해 내용이 아닌 메타 데이터로 해석되는 하위 구성 MO의 이름을 지정합니다. cw_mo_soap만 메타 데이터로 처리되는 하위 구성 MO를 지정합니다. 기타 모든 cw_mo_*는 다른 구성요소를 표시하므로 SOAP Data Handler 처리에서 제외됩니다. 다른 모든 cw_mo*는 무시됩니다.
cw_mo_soap	String	이 Business Object 레벨 ASI는 Business Object 변환 시 사용해야 하는 하위 구성 MO 속성의 이름을 지정합니다.
cw_mo_jms	String	이 Business Object 레벨 ASI는 사용할 JMS 프로토콜 구성 MO의 이름을 지정합니다.
cw_mo_http	String	이 Business Object 레벨 ASI는 사용할 HTTP 프로토콜 구성 MO의 이름을 지정합니다.

표 45. SOAP 오브젝트 ASI 요약 (계속)

ASI	가능한 값	설명
wrapper	true	Business Object에 있는 래퍼 오브젝트의 속성 이름을 지정합니다. 래퍼 오브젝트는 특정 스키마 표시기에 사용되며, 일련화되지 않아야 합니다.
maxoccurs	Integer	Business Object 속성의 최대 발생 가능성을 지정합니다. maxoccurs 값에 따라 Business Object에는 래퍼가 있을 수도 있고 없을 수도 있습니다.
minoccurs	Integer	Business Object 속성의 최소 발생 가능성을 지정합니다. minoccurs 값에 따라 오브젝트에는 래퍼가 있을 수도 있고 없을 수도 있습니다.
all	String	스키마에서 모든 표시기를 표시하는 하위 속성을 지정합니다.
choice	String	스키마에서 choice 표시기를 표시하는 하위 속성을 지정합니다.

Business Object를 SOAP 메시지로 변환의 ASI

SOAP Data Handler는 Business Object의 ASI를 사용하여 SOAP 메시지 구성 방법을 판별합니다. 달리 명시되지 않는 한 다음 섹션에 설명된 모든 ASI는 속성 레벨 ASI를 가리키며 모든 문자열 기반 비교는 대소문자에 관계없이 수행됩니다.

elem_name 및 elem_ns 처리

이 섹션에 설명된 예에서는 속성 이름이 OrderId이고, SOAP 요소 이름 공간의 접두부가 ns0인 것으로 가정합니다.

1. elem_name과 elem_ns를 모두 지정하지 않는 경우, elem_name은 속성 이름으로 기본값이 설정되고 elem_ns는 요소 상위의 이름 공간으로 기본값이 설정됩니다. ASI는 지정되지 않습니다.

```
<OrderId>1</OrderId>
```

2. elem_name은 지정하고 elem_ns는 지정하지 않은 경우, elem_name은 ASI elem_name 값으로 설정되고 elem_ns는 SOAP 본문의 이름 공간으로 기본값이 설정됩니다. ASI는 다음과 같습니다.

```
elem_name=CustOrderId
<CustOrderId>2</CustOrderId>
```

3. elem_ns는 지정하고 elem_name은 지정하지 않은 경우, elem_name은 속성 이름으로 기본값이 설정되고 elem_ns는 ASI elem_ns 값으로 설정됩니다. xmlns 속성은 해당 요소 범위에서 요소 이름 공간을 찾지 못한 경우에만 명시적으로 작성됩니다. 요소 이름 공간을 찾으면 이미 정의된 이름 공간 접두부가 사용됩니다. 그렇지

않으면(요소 이름 공간을 찾지 못한 경우), elem_ns의 고유 접두부가 생성됩니다. 다음 예를 고려해 보십시오. 이 예에서는 접두부가 이미 범위에 정의된 것으로 가정합니다. (ns1은 이 요소의 범위에 이미 정의된 이름 공간에 해당하는 접두부를 표시합니다.) ASI는 다음과 같습니다.

```
elem_ns= http://www.w3.org/2001/XMLSchema
<ns1:OrderId>3</ns1:OrderId>
```

다음 예에서는 접두부가 없는 것으로 가정합니다(ns2는 고유한 접두부를 표시합니다). ASI는 다음과 같습니다.

```
elem_ns=CustOrderIdNamespace
<ns2:OrderId xmlns:ns2="CustOrderIdNamespace">3</ns2:OrderId>
```

4. elem_name 및 elem_ns가 모두 지정된 경우, elem_name 및 elem_ns는 ASI 값으로 설정됩니다. 이미 정의된 이름 공간과 관련하여 위의 3에서 수행한 것과 동일한 확인이 적용됩니다. 3과 같이 이름 공간을 아직 정의하지 않은 경우, elem_ns의 고유 접두부가 생성됩니다. ASI는 다음과 같습니다.

```
elem_name=CustOrderId;elem_ns=CustOrderIdNamespace
<ns2:CustOrderId xmlns:ns2="CustOrderIdNamespace">1</ns2:OrderId>
```

단순 속성에 대한 type_name 및 type_ns 처리

이 섹션에 있는 예의 경우 속성 이름은 OrderId, SOAP 요소 이름 공간 접두부는 ns0, 속성 유형은 String입니다.

주: type_name 및 type_ns 처리는 구성 MO 속성 TypeInfo가 true일 경우에만 발생합니다.

1. type_name 또는 type_ns가 지정되지 않은 경우, type_name의 기본값은 단순 유형이며 type_ns의 기본값은 xml 스키마 정의 이름 공간(xsd)입니다. ASI는 지정되지 않습니다.

```
<OrderId xsi:type="xsd:string">1</OrderId>
```

2. type_name은 지정하고 type_ns는 지정하지 않은 경우 type_name은 ASI type_name 값으로 설정되고 type_ns는 요소 이름 공간으로 기본값이 설정됩니다. ASI는 다음과 같습니다.

```
type_name=CustString
<OrderId xsi:type="ns0:CustString">2</OrderId>
```

3. type_ns는 지정하고 type_name은 지정하지 않은 경우 type_ns는 단순 유형 이름으로 기본값이 설정되고 type_name은 ASI type_ns 값으로 설정됩니다. 접두부는 elem_ns 작성과 유사한 방식으로 처리됩니다. 요소 범위에 이름 공간이 없는 경우, 유형 이름 공간의 고유 접두부가 생성됩니다. ASI는 다음과 같습니다.

```
type_ns=CustStringNamespace
<OrderId xmlns:ns2="CustStringNamespace" xsi:type="ns2:String">3</OrderId>
```

4. `type_name`과 `type_ns`를 모두 지정한 경우, 지정된 ASI 값으로 설정됩니다. 유형 이름 공간의 고유 접두부가 생성됩니다. ASI는 다음과 같습니다.

```
type_name=CustString;type_ns=CustStringNamespace
<OrderId xmlns:ns2="CustStringNamespace" xsi:type="
"ns2:CustString">1</OrderId>
```

단일 카디널리티 속성에 대한 `type_name` 및 `type_ns` 처리

이 섹션에 있는 예의 경우 속성 이름은 `OrderStaus`, SOAP 요소 이름 공간 접두부는 `ns0`, 속성 유형은 `OrderStatus`입니다.

주: `type_name` 및 `type_ns` 처리는 구성 MO 속성 `TypeInfo`가 `true`일 경우에만 발생합니다.

1. `type_name`과 `type_ns`를 모두 지정하지 않은 경우, `type_name`은 Business Object 이름으로 기본값이 설정되고 유형 namespace는 요소의 이름 공간으로 기본값이 설정됩니다. ASI는 지정되지 않습니다.

```
<OrderStatus xsi:type="ns0:OrderStatus">1</OrderStatus>
```

2. `type_name`이 지정되고 `type_ns`가 지정되어 있지 않은 경우, `type_name`은 지정된 ASI 값으로 설정되며 `type_ns`의 기본값은 요소의 이름 공간이 됩니다. ASI는 다음과 같습니다.

```
type_name=CustOrderStatus
<OrderStatus xsi:type="ns0:CustOrderStatus">1</OrderStatus>
```

3. `type_ns`가 지정되고 `type_name`이 지정되지 않은 경우, `type_name`의 기본값은 Business Object 이름이 되며 `type_ns`는 지정된 `type_ns` 값으로 설정됩니다. 유형 이름 공간의 고유 접두부가 생성됩니다. ASI는 다음과 같습니다.

```
type_ns=CustTypeNS
<OrderStatus xsi:type="ns2:SOAP_OrderStatusLine
" xmlns:ns2="CustTypeNS">1</OrderStatus>
```

4. `type_name`과 `type_ns`를 모두 지정한 경우, 지정된 ASI 값으로 설정됩니다. 유형 이름 공간의 고유 접두부가 생성됩니다. ASI는 다음과 같습니다.

```
type_name=CustOrderStatus;type_ns=CustTypeNS
<OrderStatus
xsi:type="ns2:CustOrderStatus" xmlns:ns2="CustTypeNS">1</OrderStatus>
```

복수 카디널리티 속성에 대한 `type_name` 및 `type_ns` 처리

이 섹션에 설명된 모든 예제에서는 속성 이름이 `MultiLines`이고, SOAP 요소 이름 공간의 접두부가 `ns0`인 것으로 가정합니다. 속성 유형은 `OrderStatus`인 것으로 가정합니다.

주: `type_name` 및 `type_ns` 처리는 구성 MO 속성 `TypeInfo`가 `true`일 경우에만 발생합니다.

1. type_name과 type_ns를 모두 지정하지 않은 경우, type_name은 Business Object 이름으로 기본값이 설정되고 type_ns는 요소의 이름 공간으로 기본값이 설정됩니다. ASI는 다음과 같습니다.

```
<MultiLines SOAP-ENC:arrayType="ns0:OrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="SOAP-ENC:Array">
```

2. type_name은 지정하고 type_ns는 지정하지 않은 경우, type_name은 지정된 ASI type_name 값으로 설정되고 type_ns는 요소 이름 공간으로 기본값이 설정됩니다. ASI는 다음과 같습니다.

```
type_name=CustOrderStatus
<MultiLines SOAP-ENC:arrayType="ns0:CustOrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="SOAP-ENC:Array">
```

3. type_ns가 지정되고 type_name이 지정되지 않은 경우, type_name의 기본값은 Business Object 이름이 되며 type_ns는 지정된 ASI type_ns 값으로 설정됩니다. 유형 이름 공간의 고유 접두부가 생성됩니다. ASI는 다음과 같습니다.

```
type_ns=CustTypeNS
<MultiLines SOAP-ENC:arrayType="ns2:OrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="CustTypeNS" xsi:type="SOAP-ENC:Array">
```

4. type_name과 type_ns를 모두 지정한 경우, 지정된 ASI 값으로 설정됩니다. 유형 이름 공간의 고유 접두부가 생성됩니다. ASI는 다음과 같습니다.

```
type_name=CustOrderStatus;type_ns=CustTypeNS
<MultiLines SOAP-ENC:arrayType="ns2:CustOrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="CustTypeNS" xsi:type="SOAP-ENC:Array">
```

주: 각 배열 요소의 상위 요소를 나타내는 항목 요소의 유형 및 이름 공간은 arrayType 과 같습니다.

단순, 단일 및 복수 카디널리티 유형의 xsdtype

단순, 단일 및 복수 카디널리티 유형의 경우, 유형 이름의 xsdtype ASI 속성을 true 로 설정하여 SOAP 메시지의 현재 XSD를 따르십시오. type_name 및 type_ns 등록 정보를 설정할 경우에만 xsdtype 등록 정보를 읽습니다. type_name 및 type_ns가 제공된 경우, SOAP Data Handler는 먼저 SOAP API를 사용하여 쌍을 Java 유형에 맵핑하려고 합니다. 그런 다음, SOAP 메시지의 현재 XSD를 사용하여 Java 유형을 다시 SOAP 요소 유형으로 변환합니다. 예를 들어, 현재 XSD가

```
http://www.w3.org/2001/XMLSchema
```

이면 ASI는 다음과 같습니다.

```
type_name=timeInstant;type_ns=http://www.w3.org/1999/XMLSchema;xsdtype=true
```

SOAP 메시지 유형 이름은 다음과 같이 작성됩니다.

```
<OrderDate xsi:type="xsd:dateTime">
```

이는 dateTime이 1999 XSD의 timeInstant에 해당되는 2001 XSD이기 때문입니다.

xsdtype 및 단순 유형 배열

복수 카디널리티 오브젝트의 경우, 다음과 같은 단순 유형 배열을 작성할 수 있습니다.

```
<MultiLines SOAP-ENC:arrayType="xsd:string[4]"  
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"  
xsi:type="SOAP-ENC:Array">
```

이를 수행하려면 type_name 등록 정보를 원하는 단순 유형(예: string)으로 설정하고 type_ns 등록 정보는 적절한 XSD 스펙으로 설정하십시오. 그런 다음, 유형이 현재 XSD 유형으로 변환되도록 xsdtype 등록 정보를 true로 설정하십시오. 마지막으로 arrayOf 등록 정보는 단순 유형 값을 보유해야 하는 컨테이너의 속성 이름으로 설정해야 합니다. 다음은 문자열 배열과 관련하여 ASI가 찾을 항목과 관련된 예입니다.

```
arrayof=size;type_name=string;type_ns=http://www.w3.org/2001/XMLSchema;xsdtype=true
```

결합 처리 시 ASI 효과

faultcode, faultfactor, faultstring 및 detail 요소는 다음 규칙을 따릅니다.

1. 이러한 속성에 있는 임의의 elem_name, elem_ns, type_name 및 type_ns ASI는 무시됩니다.
2. 세부사항 요소의 모든 하위는 본문 처리에서 설명한 대로 정확히 기록됩니다.

머리글 처리 시 ASI 효과

머리글 하위 오브젝트 레벨 및 그 아래에서 모든 ASI 등록 정보(표 45 참조)를 사용할 수 있습니다.

SOAP 속성 지정

단순 유형에 대한 attr_name 처리

이것은 지원되지 않습니다. Business Object 속성은 작성할 SOAP 속성의 경우에는 복합(1 또는 n 카디널리티) 유형이어야 합니다.

단일 및 복수 카디널리티 유형에 대한 attr_name 처리

Business Object 속성을 SOAP 요소 대신 SOAP 속성으로 변환하는 ASI를 지정할 수 있습니다. Data Handler는 복합 단일 및 n-card 유형에 SOAP 속성 추가만을 지원합니다. 다음 샘플을 고려하십시오.

```
<CustInfo City="4" State="5" Street="2" Zip="6">  
<Name xsi:type="xsd:string">1</Name>  
<Street2 xsi:type="xsd:string">3</Street2>  
</CustInfo>
```

이 Business Object 정의 구조(그림 45에서 각 속성의 오른쪽에 속성 레벨 ASI가 지정됨)의 경우, Data Handler는 다음 처리 단계를 따릅니다.

Name	Type	App Spec Info
☐ CustInfo	CustomerInfo	
Name	String	
Street1	String	attr_name=Street
Street2	String	
City	String	attr_name=City
State	String	attr_name=State
Zip	String	attr_name=Zip

그림 45. attr_name Business Object

1. 복합 속성을 살펴볼 때, Data Handler는 먼저 이 복합 Business Object 속성에 대한 대응하는 태그를 생성합니다. 이 예에서는 CustInfo가 복합 Business Object 속성을 나타냅니다.
2. Data Handler는 복합 Business Object의 하위를 반복합니다. 단순 유형 속성만이 속성 작성을 위해 고려됩니다. 단순 유형이 attr_name이라는 ASI 등록 정보가 있는 경우, Data Handler는 이 단순 유형을 SOAP 요소에 속성으로 기록합니다. 이 예에서는 요소(CustInfo)에 Street, City, State 및 Zip의 네 개의 속성이 있습니다.
3. Business Object의 나머지 속성은 표준 BODY 처리를 사용하여 기록됩니다. 이것은 모든 관련 ASI도 attr_name ASI가 없는 Business Object 속성에 대해 평가할 것임을 의미합니다.

복수 카디널리티 유형 처리를 위한 논리는 단일 카디널리티 유형 처리의 경우와 동일합니다. 특히, 각 <item> 태그는 복수 카디널리티 오브젝트의 각 Business Object 인스턴스에 대응하며 ASI를 사용하여 처리됩니다. 예를 들어, 대응하는 ASI가 있는 다음 복수 카디널리티 Business Object 정의 구조의 경우를 가정하십시오.

Name	Type	Card	App Spec Info
☐ CustInfo	CustomerInfo	N	
Name	String		
Street1	String		attr_name=Street
Street2	String		
City	String		attr_name=City
State	String		attr_name=State
Zip	String		attr_name=Zip

그림 46. attr_name 복수 카디널리티 Business Object

Data Handler로 전송된 이벤트에 이 복수 카디널리티 오브젝트의 두 인스턴스가 있으면, 작성되는 SOAP 메시지는 다음과 유사합니다.

```
<CustInfo>
  <item City="Armonk" Street="Main Street">
    <Name>IBM</Name>
    <Street2>None</Street2>
  </item>
  <item City="Burlingame" State="Ca" Street="577 Airport Blvd" Zip="94010">
    <Name>Burlingame Labs</Name>
    <Street2>Suite 600</Street2>
  </item>
</CustInfo>
```

item 태그가 복합 요소 유형으로 취급됨을 주목하십시오. BO 정의에 있는 모든 속성이 대응하는 item 태그의 SOAP 속성이 됩니다.

arrayof 처리

arrayof ASI 등록 정보를 사용하여 배열 요소 자체(하위 태그가 아닌)에 SOAP 속성을 지정합니다. 단일 카디널리티 복합 유형의 ASI에 arrayOf 등록 정보를 추가합니다. arrayOf 등록 정보의 값은 SOAP 메시지가 표시할 복수 카디널리티 하위의 이름이어야 합니다. SOAP 메시지를 작성할 때, 복수 카디널리티 하위는 배열의 하위 태그를 나타냅니다.

arrayof 요소가 있고 대응하는 ASI가 표시된 Business Object 정의가 그림 47에 표시되어 있습니다.

Name	Type	Card	App Spec Info
Customer	Customer	1	arrayof=CustInfo
CustID	String		attr_name=ID
CustInfo	CustomerInfo	N	
Name	String		
Street1	String		attr_name=Street
Street2	String		
City	String		attr_name=City
State	String		attr_name=State
Zip	String		attr_name=Zip

그림 47. arrayOf Business Object

Data Handler로 전송한 이벤트에 카디널리티 n 오브젝트의 두 인스턴스가 있으면 작성된 SOAP 메시지는 다음과 같습니다.

```
<Customer ID="12">
  <CustInfo City="4" State="5" Street="2" Zip="6">
    <Name>1</Name>
    <Street2>3</Street2>
  </CustInfo>
  <CustInfo City="10" State="11" Street="8" Zip="12">
```



```

    <Name>7</Name>
    <Street2>9</Street2>
  </CustInfo>
</Customer>

```

arrayof 등록 정보를 item 이외의 이름이 있는 배열 항목을 작성하는 데 사용할 수 있음을 주목하십시오. 이 예에서, item 태그가 CustInfo 태그로 변경됩니다. 이 요소 이름은 arrayof ASI 등록 정보가 지시하는 Business Object 속성의 이름에서 파생됩니다.

attr_name 및 attr_ns 처리

작성된 SOAP 속성에 해당하는 이름 공간을 제공해야 할 수 있습니다. 단순 유형에 attr_ns ASI 등록 정보를 지정하여 이 작업을 수행합니다. 같은 속성의 ASI에 attr_name 이 있을 경우에만 Data Handler는 attr_ns 등록 정보를 처리합니다. attr_name 및 attr_ns 는 다음의 규칙을 따릅니다.

1. attr_name과 attr_ns가 모두 존재하지 않는 경우, Business Object 속성은 SOAP 요소로 변환됩니다.
2. attr_name만 설정된 경우, SOAP 속성의 이름 공간은 요소 이름 공간으로 기본값이 설정됩니다.

```
<CustInfo Street="577 Airport"></CustomerInfo>
```

3. attr_ns만을 설정할 때, 등록 정보는 무시되고 Business Object 속성이 SOAP 요소로 변환됩니다.
4. attr_name과 attr_ns 모두 존재할 경우, SOAP 속성은 다음과 같이 작성됩니다.

```
<CustInfo ns2:Street="577 Airport" xmlns:ns2=
  "AttrNS"></CustomerInfo>
```

dh_mimetype: Data Handler 호출

SOAP Data Handler는 다른 Data Handler를 호출하여 Data Handler가 존재하는 형식으로 Business Object를 작성할 수 있습니다. SOAP 하위 Business Object를 SOAP 문자열로 전송할 때 SOAP 메시지에 인코딩된 텍스트를 추가하여 이를 수행합니다.

RNIF 문서는 SOAP 요소의 값을 인코딩할 수 있는 형식 중 하나입니다. 이 기능을 사용하려면 SOAP 하위 Business Object 레벨에서 RNIF BO를 추가하십시오. RNIF Business Object를 문자열로 변환할 때 다른 Data Handler를 호출하도록 SOAP Data Handler에 신호를 보내려면 dh_mimetype 등록 정보를 속성의 ASI에 추가하십시오. dh_mimetype ASI 등록 정보의 값은 MO_DataHandler_Default Meta Object에 지정한 유효한 mimeType이어야 합니다. mimeType은 Business Object를 처리할 때 호출하는 Data Handler를 판별하는 데 사용됩니다.

그림 48에서는 CustomerInfo가 복합 하위이고 RNET_Pip3A2PriceAndAvailabilityQuery가 RNIF Business Object인 SOAP 하위 Business Object를 보여줍니다.

Name	Type	App Spec Info
[-] CustomerInfo	CustomerInfo	
Name	String	
CustID	String	
[+] RNIFMsg	RNET_Pip3A2PriceAndAvailabilityQuery	elem_name=RNIFexample;dh_mimetype=application/x_rosettanet_agent;type_name=base64Binary;type_ns=http://www.w3.org/2001/XMLSchema;xsdtype=true

그림 48. dh_mimetype이 있는 RNIF Business Object

이 Business Object에서 작성된 SOAP 메시지는 다음과 같습니다.

```
<CustomerInfo>
<Name>IBM Corporation</Name>
<CustID>95626</CustID>
<RNIFexample
xsi:type="xsd:base64Binary">1AWERYER238W98EYR9238728374871892787ASRJK23423
JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASRJK234
34JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASRJK2
4234JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASR
234234JKAWERJ234AWERIJHI423488R4HASFWR234
</RNIFexample>
</CustomerInfo>
```

RNIF 예 요소에는 그의 요소 값으로 base64 2진 인코딩된 RNIF 인코딩 문자열이 있습니다. 또한 elem_name, elem_ns, type_name, type_ns 및 xsdtype ASI 등록 정보는 Business Object 속성에 관련된 상태로 유지됩니다. 이 예에서 지정된 elem_name은 메시지 작성 시 SOAP 요소의 이름을 표시합니다.

주: 호출한 Data Handler에서 리턴된 요소 값이 인코딩된 텍스트일 경우, type_name 등록 정보는 base64Binary로 설정되어야 하고 type_ns는 xsd 이름 공간과 일치해야 하며 xsdtype은 true로 설정되어야 합니다.

xsd:base64Binary: xsd:base64Binary로 분석하도록 type_name 및 type_ns를 설정할 경우, SOAP Data Handler는 해당 요소의 값을 설정하기 전에 Business Object로부터 값을 인코딩합니다. Data Handler는 Apache API를 사용하여 base64Binary 순차기에 대해 레지스트리를 조회하고 호출한 Data Handler에서 리턴된 문자열을 순차화한 후 요소의 값을 설정합니다.

스키마 complexType 표시기

다음 섹션은 Business Object의 스키마 complexType 표시기 효과에 대해 설명합니다. 표시기는 다음과 같습니다.

- maxOccurs

- minOccurs
- all
- sequence
- choice

단순 유형에 대한 maxOccurs 및 minOccurs 표시기: maxOccurs 표시기는 요소가 복합 유형 내에 발생할 수 있는 최대 횟수를 지정합니다. minOccurs 표시기는 요소가 complexType 내에 발생해야 하는 최소 횟수를 지정합니다.

다음 스키마를 고려해 보십시오.

```
<xs:element name="Address" type="Address">
<xs:complexType name="Address">
  <xs:sequence>
    <xs:element name="AddressLine" type="xsd:string" maxOccurs="10"/>
    <xs:element name="SuiteNumber" type="xsd:string" minOccurs="3"
      maxOccurs="unbounded"/>
    <xs:element name="City" type="xsd:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

위의 예는 AddressLine 요소가 Address 요소에서 최대 10회 발생할 수 있는 반면 SuiteNumber 요소는 최소한 3회 이상 발생해야 함을 나타냅니다. 이 스키마에 해당하는 Business Object에는 다음의 ASI가 있는 각 maxOccurs/minOccurs 표시기에 적합한 N 카디널리티 랩퍼 오브젝트가 있어야 합니다.

```
maxOccurs=N;wrapper=true
```

또는

```
minOccurs=3;wrapper=true;
```

wrapper=true ASI는 오브젝트가 랩퍼이므로 SOAP 메시지에 명시적으로 기록되지 않음을 나타냅니다. 대신 이 랩퍼 오브젝트에는 단순 유형의 하위 하나가 있어야 합니다. 런타임 시 SOAP를 Business Object로 변환의 경우, Data Handler는 랩퍼의 N 하위 오브젝트를 읽고 각 오브젝트마다 해당하는 요소를 작성합니다. Business Object를 SOAP 메시지로 변환 수행 시 Data Handler는 발생하는 모든 요소마다 N 카디널리티 랩퍼에서 하위 오브젝트를 작성합니다.

해당하는 SOAP Business Object는 그림 49에 표시된 것과 유사합니다.

Pos	Name	Type	Key	Card	App Spec Info
1	☐ Address	Address	<input checked="" type="checkbox"/>	1	
1.1	☐ AddressLine	AddressLine_wrap	<input checked="" type="checkbox"/>	N	maxoccurs=10;wrapper=true
1.1.1	AddressLine	String	<input checked="" type="checkbox"/>		
1.1.2	ObjectEventId	String			
1.2	☐ SuiteNumber	SuiteNumber_wrap	<input checked="" type="checkbox"/>	N	minoccurs=3;wrapper=true
1.2.1	SuiteNumber	String	<input checked="" type="checkbox"/>		
1.2.2	ObjectEventId	String			
1.3	City	String	<input type="checkbox"/>		
1.4	ObjectEventId	String			
2	ObjectEventId	String			

그림 49. SOAP Business Object에 있는 단순 유형 ASI의 minOccurs 및 maxOccurs

그림 49에 표시된 Business Object에 해당하는 SOAP 메시지는 다음과 같습니다.

```
<Address xsi:type="ns0:Address">
  <AddressLine xsi:type="xsd:string">Line1</AddressLine>
  <AddressLine xsi:type="xsd:string">Line2</AddressLine>
  <SuiteNumber xsi:type="xsd:string">600</SuiteNumber>
  <SuiteNumber xsi:type="xsd:string">650</SuiteNumber>
  <SuiteNumber xsi:type="xsd:string">700</SuiteNumber>
  <City xsi:type="xsd:string">San Francisco</City>
</Address>
```

주: SOAP Data Handler는 요소의 최대 또는 최소 발생 횟수의 유효성을 검증하지 않고 동일한 방식으로 maxOccurs와 minOccurs 표시기를 처리합니다. Data Handler는 단순히 컨테이너 구조를 제공하여 maxOccurs 및 minOccurs 표시기가 있는 특정 요소의 복수 인스턴스를 보유합니다. 이는 단순 및 복합 유형에 적용됩니다.

복합 유형에 대한 maxOccurs 및 minOccurs 표시기: <maxOccurs> 표시기는 복합 유형에서 요소가 발생할 수 있는 최대 횟수를 지정합니다. <minOccurs> 표시기는 complexType에서 요소가 발생해야 하는 최소 횟수를 지정합니다. 다음 스키마에서 maxOccurs 표시기를 고려해 보십시오.

```
<xs:element name="Address" type="Address">
<xs:complexType name="Address">
  <xs:sequence>
    <xs:element name="AddressInfo" type="AddressInfo" maxOccurs="3"/>
    <xs:element name="City" type="xsd:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:complexType name="AddressInfo">
  <xs:sequence>
    <xs:element name="StreetLine" type="xsd:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

위의 예는 AddressInfo 복합 유형 요소가 Address 요소에서 최대 세 번 발생할 수 있음을 표시합니다. 이 스키마에 해당하는 Business Object는 complexType AddressInfo 자체가 N 카디널리티가 될 수 있으므로 래퍼 오브젝트가 없습니다. 다음 ASI는 N 카디널리티 속성인 maxOccurs=3에 위치합니다.

그림 50에서는 해당 SOAP Business Object를 보여줍니다.

Pos	Name	Type	Key	Card	App Spec Info
1	☐ Address	Address	<input checked="" type="checkbox"/>	1	
1.1	☐ AddressInfo	AddressInfo	<input checked="" type="checkbox"/>	N	maxOccurs=3
1.1.1	StreetLine	String	<input checked="" type="checkbox"/>		
1.1.2	ObjectEventId	String			
1.2	City	String	<input type="checkbox"/>		
1.3	ObjectEventId	String			
2	ObjectEventId	String			

그림 50. SOAP Business Object에 있는 복합 유형 ASI의 minOccurs 및 maxOccurs

그림 50에 표시된 Business Object에 해당하는 SOAP 메시지는 다음과 같습니다.

```
<Address xsi:type="ns0:Address">
  <AddressInfo xsi:type="ns0:AddressInfo">
    <StreetLine xsi:type="xsd:string">100 Market St.</ StreetLine>
    <StreetLine xsi:type="xsd:string">Apt 15</ StreetLine>
  </AddressInfo>
  <City xsi:type="xsd:string">San Francisco</City>
</Address>
```

all 표시기: all 표시기는 기본적으로 complexType의 하위 요소가 모든 순서로 나타날 수 있고 각 하위 요소는 0 또는 1회 발생해야 한다는 것을 표시합니다. 다음 스키마를 고려하십시오.

```
<complexType name="Item">
  <all>
    <element name="quantity" type="xsd:int"/>
    <element name="product" type="xsd:string"/>
  </all>
</complexType>
```

위의 예는 요소 quantity 및 product가 SOAP 메시지에서 모든 순서로 발생할 수 있음을 나타냅니다. quantity 요소가 첫 번째로 발생한 후 product 요소가 두 번째로 발생하거나 또는 그 반대로 발생할 수 있습니다.

그림 51에서는 이 스키마 단편에 해당하는 Business Object를 보여줍니다.

Pos	Name	Type	Card	App Spec Info
1	☐ Item	Item	1	all=Item_wrapper
1.1	☐ Item_wrapper	Item_wrapper	N	wrapper=true
1.1.1	quantity	String		
1.1.2	product	String		
1.1.3	ObjectEventId	String		
1.2	ObjectEventId	String		
2	ObjectEventId	String		

그림 51. SOAP Business Object에 있는 all 표시기 ASI

해당 SOAP 메시지 단편은 다음과 같습니다.

```
<Item xsi:type="ns0:Item">
  <quantity xsi:type="xsd:string">12</quantity>
  <product xsi:type="xsd:string">2</product>
</Item>
```

sequence 표시기: sequence 표시기는 하위 요소가 complexType에 지정된 순서로 나타나야 함을 지정합니다.

```
<complexType name="Item">
  <sequence>
    <element name="quantity" type="int"/>
    <element name="product" type="string"/>
  </sequence>
</complexType>
```

SOAP Data Handler에는 이 표시기의 특수 ASI 또는 랩퍼 오브젝트가 필요하지 않습니다. 기본적으로, Data Handler는 Business Object에 지정된 순서대로 SOAP 요소를 읽고 씁니다.

choice 표시기: choice 표시기는 complexType의 요소 중 하나만 SOAP 메시지에 나타날 수 있음을 지정합니다. 다음 스키마를 고려하십시오.

```
<complexType name="Item">
  <choice>
    <element name="quantity" type="int"/>
    <element name="product" type="string"/>
  </choice>
</complexType>
```

SOAP Data Handler에는 이 표시기의 특수 ASI 또는 랩퍼 오브젝트가 필요하지 않습니다. Business Object를 SOAP 메시지로 변환할 때 Data Handler는 SOAP 메시지에 나타나는 요소 중 사용자 선택사항에 따릅니다. SOAP 메시지를 Business Object로 변환할 경우, Data Handler는 기존 요소를 읽고 해당하는 속성을 채웁니다.

SOAP을 Business Object로 변환의 ASI

SOAP Data Handler는 Business Object의 ASI를 사용하여 수신 SOAP 메시지를 읽고 유효성을 검증합니다. SOAP Data Handler는 ASI 유효성 검증에 다음 규칙을 적용합니다.

- 머리글 및 본문 처리는 동일합니다.
- SOAP ConfigMO 등록 정보인 TypeCheck는 strict로 설정하고 TypeInfo는 Data Handler가 아래 섹션에 설명된 유효성 검증을 수행하도록 true로 설정해야 합니다.
- type_name 및 type_ns 유효성 검증은 유형 유효성 검증이 일반적으로 두 개의 등록 정보 모두의 영향을 받으므로 동시에 수행됩니다.

주: 달리 명시되지 않는 한 다음 섹션에 설명된 모든 ASI는 속성 레벨 ASI입니다.

elem_name 유효성 검증

다음 규칙은 단순, 카디널리티 1 및 카디널리티 n 속성의 유효성 검증에 적용됩니다.

1. SOAP 메시지를 구문 분석하는 동안 요소가 발생하면 요소 이름을 elem_name 값과 일치시켜 보면서 Data Handler는 우선 Business Object 레벨에서 모든 ASI를 검색합니다.
2. 일치하는 항목을 찾을 수 없는 경우, Data Handler는 해당 Business Object 레벨에서 각각의 속성 이름에 대해 요소의 이름과 일치하도록 시도합니다.
3. 두 가지 검색이 모두 실패하는 경우에는 Data Handler가 실패합니다.

elem_ns 유효성 검증

다음 경우는 단순, 카디널리티 1 및 카디널리티 n 속성의 유효성 검증에 적용됩니다.

1. 이 요소에 대한 SOAP 메시지에서 elem_ns ASI 또는 xmlns가 없는 경우에는 요소가 적절하게 유효성이 검증됩니다.
2. elem_ns ASI가 없으며 SOAP 메시지의 대응되는 요소에 xmlns가 지정되어 있지 않으면, Data Handler는 elem_ns의 기본값을 범위에 있던 Business Object에서 읽은 마지막 elem_ns로 설정합니다. Data Handler는 이 값을 SOAP 메시지의 xmlns 값과 비교합니다. 일치 항목이 없으면 유효성 검증은 실패합니다.
3. elem_ns ASI가 없으며 SOAP 메시지의 대응되는 요소에 xmlns가 지정되어 있지 않은 경우, Data Handler는 ASI에 지정되어 있는 elem_ns가 SOAP 메시지의 현재 범위에 있는 이름 공간 중 하나와 일치하는지 확인합니다. 일치 항목이 없으면 유효성 검증은 실패합니다.

type_name 및 type_ns 유효성 검증

아래 섹션은 type_name 및 type_ns의 유효성 검증을 설명합니다.

단순 속성: 다음 규칙은 xsi:type이 true일 때 type_name 및 type_ns의 유효성 검증에 적용됩니다.

- **type_name**과 **type_ns**를 모두 지정할 경우 - Data Handler는 type_name 및 type_ns 쌍을 사용하여 해당하는 Java 클래스 오브젝트를 작성합니다. 수신 SOAP 메시지 typename 및 typenamespace를 사용하여 다른 Java 클래스 오브젝트를 조회합니다. 두 개의 Java 클래스 오브젝트가 일치하면 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.
- **type_name**과 **type_ns**를 모두 지정하지 않은 경우 - Data Handler는 단순 Business Object 속성을 Java 클래스 오브젝트에 맵핑합니다. 수신 SOAP 메시지 typename 및 typenamespace를 사용하여 다른 Java 클래스 오브젝트를 조회합니다. 두 개의 Java 클래스 오브젝트가 일치하면 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.
- **type_name**만 지정됨 - 단순 유형 유효성 검증이 실패합니다. xsiType이 true일 경우, type_name과 type_ns를 둘다 지정하거나 둘다 지정하지 않아야 합니다.
- **type_ns**만 지정됨 - 단순 유형 유효성 검증이 실패합니다. xsiType가 true일 경우, type_name 값과 type_ns를 둘다 지정하거나 둘다 지정하지 않아야 합니다.

다음 규칙은 xsiType이 false일 때 type_name 및 type_ns의 유효성 검증에 적용됩니다.

- **type_name**과 **type_ns**를 모두 지정한 경우 - Data Handler는 SOAP 메시지 typename과 typenamespace 쌍과 ASI에 지정된 type_name 및 type_ns 값 간에 직접 비교를 수행합니다. 이들 쌍이 정확히 일치할 경우 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.
- **type_name**과 **type_ns**를 모두 지정하지 않은 경우 - Data Handler는 단순 Business Object 속성을 Java 클래스 오브젝트에 맵핑합니다. 수신 SOAP 메시지 typename 및 typenamespace를 사용하여 다른 Java 클래스 오브젝트를 조회합니다. 두 개의 Java 클래스 오브젝트가 일치하면 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.
- **type_name**만 지정한 경우 - type_ns 값은 Business Object ASI에서 발견된 요소 이름 공간으로 기본값이 설정됩니다. Data Handler는 ASI에 지정된 기본 type_ns 및 type_name을 사용하여 이들 값과 SOAP 메시지 typename과 typenamespace 간에 직접 비교를 수행합니다. 이들 쌍이 정확히 일치할 경우 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.
- **type_ns**만 지정한 경우 - type_name 값은 Business Object 속성 유형으로 기본값이 설정됩니다. Data Handler는 ASI에 지정된 기본 type_ns 및 type_name을 사용하여 이들 값과 SOAP 메시지 typename과 typenamespace 간에 직접 비교를 수행합니다. 이들 쌍이 정확히 일치할 경우 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.

복합 속성(카디널리티 1 및 n): 다음 규칙은 xsiType이 true일 때 type_name 및 type_ns의 유효성 검증에 적용됩니다.

- 두 개의 **type_name** 및 **type_ns**가 지정됨 - xsdType이 무시됩니다. Data Handler는 xsdType이 false인 것처럼 처리합니다.
- **type_name**과 **type_ns**를 모두 지정하지 않은 경우 - xsdType은 무시됩니다. Data Handler는 xsdType이 false인 것처럼 처리합니다.
- **type_name**만 지정됨 - xsdType이 무시됩니다. Data Handler는 xsdType이 false인 것처럼 처리합니다.
- **type_ns**만 지정됨 - xsdType이 무시됩니다. Data Handler는 xsdType이 false인 것처럼 처리합니다.

다음 규칙은 xsdType이 false일 때 type_name 및 type_ns의 유효성 검증에 적용됩니다.

- **type_name**과 **type_ns**를 모두 지정한 경우 - Data Handler는 SOAP 메시지 typename과 typenamespace 쌍과 ASI에 지정된 type_name 및 type_ns 값 간에 직접 비교를 수행합니다. 이들 쌍이 정확히 일치할 경우 유효성이 검증됩니다. 그렇지 않으면, 유효성 검증이 실패합니다.
- **type_name**과 **type_ns**를 모두 지정하지 않을 경우 - type_name은 Business Object 속성 유형으로 기본값이 설정됩니다. type_ns 값은 Business Object ASI에 있는 요소 이름 공간으로 기본값이 설정됩니다. Data Handler는 이러한 기본 작동을 사용하여 이들 값과 SOAP 메시지 typename과 typenamespace 쌍 간에 직접 비교를 수행합니다. 이들 쌍이 정확히 일치할 경우 유효성이 검증됩니다. 그렇지 않으면, 유효성 검증이 실패합니다.
- **type_name**만 지정한 경우 - type_ns 값은 Business Object ASI에서 발견된 요소 이름 공간으로 기본값이 설정됩니다. Data Handler는 ASI에 지정된 기본 type_ns 및 type_name을 사용하여 이들 값과 SOAP 메시지 typename과 typenamespace 간에 직접 비교를 수행합니다. 이들 쌍이 정확히 일치할 경우 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.
- **type_ns**만 지정한 경우 - type_name 값은 Business Object 속성 유형으로 기본값이 설정됩니다. Data Handler는 ASI에 지정된 기본 type_ns 및 type_name을 사용하여 이들 값과 SOAP 메시지 typename과 typenamespace 간에 직접 비교를 수행합니다. 이들 쌍이 정확히 일치할 경우 유효성이 검증됩니다. 그렇지 않으면, 유효성이 검증됩니다.

attr_name 및 attr_ns 유효성 검증

SOAP 메시지를 Business Object로 읽어들이는 동안, SOAP 속성에 대한 각 SOAP 요소를 검색합니다. 속성이 발견되면, 이들 속성이 대응하는 BO의 attr_name 등록 정보 값과 비교됩니다. 예를 들어, 이 SOAP 메시지를 고려하십시오.

```
<CustInfo City="4" State="5" Street="2" Zip="6">
  <Name xsi:type="xsd:string">1</Name>
  <Street2 xsi:type="xsd:string">3</Street2>
</CustInfo>
```


이제 그림 52에 표시된 Business Object 정의 구조(각 속성의 오른쪽에 속성 레벨 ASI가 지정됨)를 고려하십시오.

Name	Type	App Spec Info
☐ CustInfo	CustomerInfo	
Name	String	
Street1	String	attr_name=Street
Street2	String	
City	String	attr_name=City
State	String	attr_name=State
Zip	String	attr_name=Zip

그림 52. attr_name 및 attr_ns 유효성 검증

Data Handler는 다음의 처리 단계를 수행합니다.

1. 요소 이름 CustInfo를 읽습니다.
2. 이 요소 이름에 해당하는 Business Object 속성을 분석합니다.
3. SOAP 요소의 속성을 읽고 하위 속성의 ASI와 이를 대조합니다. 이러한 경우, SOAP 메시지 Street는 Business Object 속성 Street1과 대조하고 City는 Business Object 속성 City와 대조합니다.
4. CustInfo에 대한 하위 요소는 나머지 본문과 동일한 방식으로 읽고 처리됩니다.

주: attr_ns의 유효성은 검증되지 않습니다.

Data Handler가 제공된 요소에 대해 SOAP 속성을 순환합니다. 발견되는 각 속성의 경우, Data Handler가 대응하는 속성이 있는지 Business Object를 검색합니다. 발견되는 경우, 해당 Business Object 속성이 SOAP 속성의 값으로 채워집니다. 대응하는 Business Object 속성이 없으면, Data Handler는 다음 SOAP 속성을 계속 처리합니다.

SOAP Data Handler 내에서 Data Handler 호출

SOAP Data Handler는 다른 Data Handler를 사용하여 SOAP 메시지의 인코드된 요소 값을 Business Object로 읽어들이 수 있습니다. 예를 들어, RNIF 문서는 SOAP 요소 값이 인코드되는 형식 중 하나일 수 있습니다. 이 기능을 사용하기 위해, SOAP 하위 Business Object의 임의의 레벨에 RNIF Business Object를 추가할 수 있습니다. 이 RNIF 인코드된 String을 RNIF Business Object로 변환할 때 다른 Data Handler가 사용되도록 SOAP Data Handler에 신호를 보내려면, 속성의 ASI에 dh_mimetype 등록 정보를 추가해야 합니다. dh_mimetype ASI의 값은 MO_DataHandler_Default Business Object에 지정되는 유효한 mimeType이어야 합니다. 이 mimeType이 문자열에 사용할 Data Handler를 판별하는 데 사용됩니다. 예를 들어, RNIFExample이 RNIF 인코드된 String을 포함하는 SOAP 요소인 다음 SOAP 메시지를 고려합니다.

```

<CustInfo>
<Name>IBM Corporation</Name>
<CustID>95626</CustID>
<RNIFexample xsi:type="xsd:base64Binary">
1AWERYER238W98EYR9238728374871892787ASRJK234234JKAWER
J234AWERIJHI423488R4HASF1AWERYER238W98EYR923872837487
1892787ASRJK234234JKAWERJ234AWERIJHI423488R4HASF1AWER
YER238W98EYR9238728374871892787ASRJK234234JKAWERJ234A
WERIJHI423488R4HASF1AWERYER238W98EYR92387283748718927
87ASRJK234234JKAWERJ234AWERIJHI423488R4HASFWR234
</RNIFexample>
</CustomerInfo>

```

SOAP Business Object는 그림 53에 표시된 것과 유사합니다.

Name	Type	App Spec Info
☐ CustInfo	CustomerInfo	
Name	String	
Street1	String	attr_name=Street
Street2	String	
City	String	attr_name=City
State	String	attr_name=State
Zip	String	attr_name=Zip

그림 53. RNIFExample Business Object

RNIFexample 요소에 RNIF 인코딩된 String이 요소 값으로서 포함되어 있습니다. 또한, elem_name, elem_ns, type_name, type_ns 및 xsdtype ASI 등록 정보는 여전히 Business Object 속성과 관련된 상태에 있습니다.

주: 호출된 Data Handler가 리턴하는 요소 값이 인코딩된 텍스트인 경우, type_name 등록 정보가 base64Binary로 설정되어야 하고, type_ns가 xsd 이름 공간에 대응해야 하며, xsdtype이 true로 설정되어야 합니다.

기본 Business Object 분석

SOAP을 Business Object로 변환의 경우, SOAP Data Handler와 웹 서비스 커넥터는 Business Object 이름을 분석하기 위해 정보를 교환하는 특수 계약을 준수합니다. 커넥터는 BodyName 및 BodyNamespace 쌍으로 매핑되는 Business Object 이름의 목록을 SOAP Data Handler에 제공합니다. 또한 TLO에 설정된 default Business Object가 있는 경우, 이 정보가 Data Handler에 전달됩니다. 이 정보가 제공될 때, SOAP Data Handler는 다음 단계를 사용하여 처리합니다.

1. Data Handler가 SOAP 메시지를 수신합니다.
2. Data Handler가 이것이 SOAP 요청, 응답 또는 결함 메시지인지를 판별합니다.
 - a. SOAP 요청 또는 응답 메시지인 경우, Data Handler는 SOAP-ENV:Body 요소의 첫 번째 하위 요소에서 BodyName 및 BodyNamespace를 읽습니다.

- b. SOAP 결합 메시지인 경우, Data Handler는 결합 메시지의 세부사항 요소의 첫 번째 하위 요소에서 BodyName 및 BodyNamespace를 읽습니다. 결합 메시지에 세부사항 요소가 없는 경우, Data Handler는 이 변환을 위해 defaultfault Business Object를 사용합니다.
3. defaultfault Business Object가 아직 선택되지 않은 경우, Data Handler는 2단계에서 발견된 BodyName 및 BodyNamespace를 커넥터가 제공하는 목록에서 발견되는 쌍과 일치시키고자 합니다. 일치가 이루어지는 경우, Business Object 분석에 성공합니다. 일치가 이루어지지 않는 경우, Data Handler는 의미있는 오류 메시지를 표시하고 실패합니다.

플러그 가능 이름 핸들러 지정

기본 Business Object 분석을 사용하여, 플러그 가능 이름 핸들러를 지정하여 SOAP 메시지를 Business Object로 변환에 사용할 Business Object를 판별할 수 있습니다. MO_DataHandler_DefaultSOAPConfig를 변경하여 이 작업을 수행합니다.

MO_DataHandler_DefaultSOAPConfig에는 다음을 지정하는 유형 string의 두 속성이 있습니다.

- **ClassName** - SOAP Data Handler 기본 클래스의 클래스 이름 플러그 가능 이름 핸들러 지정 시 이 속성 값은 변경하지 않습니다.
- **SOAPNameHandler** - SOAPNameHandler 속성은 호출하는 이름 핸들러를 지시합니다. 이 등록 정보의 값은 클래스 이름이어야 합니다. SOAPNameHandler 클래스는 다음 서명이 있는 추상 클래스입니다.

```
public static SOAPNameHandler createNameHandler(Object moProps)
public abstract String getBOName(Envelope msgEnv, SOAPProperty prop)
```

SOAP Data Handler에 작성된 모든 이름 핸들러는 getBOName 메소드를 구현해야 합니다. createNameHandler 메소드에 전달된 오브젝트는 이 이름 핸들러의 Meta Object 등록 정보를 설정합니다. getBOName 메소드에 전달되는 SOAPProperty는 Data Handler에 대한 커넥터 호출을 통해 전달된 SOAPProperty를 설정합니다. 다음은 SOAP Data Handler가 메소드를 호출하는 방법입니다.

```
SOAPNameHandler nh = SOAPNameHandler.createNameHandler(moProps);
String boName = nh.getBOName(msgEnv, prop);
```

SOAPNameHandler 속성에 값이 있으면 SOAP Data Handler는 지정된 이름 핸들러를 호출합니다. 값이 없거나 지정된 이름 핸들러가 Business Object 이름을 가져오지 못하면 기본 Business Object 분석을 수행할 수 있도록 기본적으로 SOAP Data Handler를 호출합니다. 기본 비즈니스 분석은 커넥터가 본문 이름, 본문 이름 공간 및 Business Object 이름과 같은 정보가 있는 커넥터 지원 Business Object 컨테이너 목록을 Data

Handler에 전달하는 프로세스에 대해합니다. Data Handler는 이 정보를 사용하여 SOAP 메시지에서 Business Object를 작성합니다. 이름 핸들러가 실패하는 경우, 기본 Business Object 분석이 발생합니다.

제한사항

아래 섹션은 Data Handler 제한사항에 대해 설명합니다.

SOAP 양식 및 지침 사용

SOAP 메시지는 웹 서비스에서 정의된 양식 및 사용법으로 작성됩니다. SOAP Data Handler는 표 46에 표시된 지원 레벨을 제공합니다.

표 46. 양식 및 지침 사용

양식	사용	정의된 파트 사용	Data Handler 지원
document	literal	element	full
document	literal	type	limited(아래 참조)
document	encoded	element	none
document	encoded	type	limited(아래 참조)
rpc	literal	element	none
rpc	literal	type	full
rpc	encoded	element	none
rpc	encoded	type	full

파트 및 파트 요소 순서

SOAP Data Handler가 SOAP 메시지를 Business Object로 변환하고 SOAP 메시지가 document/literal/type 또는 document/encoded/type 형식을 따를 경우 메시지 파트는 WSDL에 설명된 순서를 따라야 합니다. 예를 들어, 다음의 WSDL을 고려해 보십시오.

```
<operation name="GetQuote"
  style="document" ...>
<input>
  <soap:body parts="Part1 Part2 Part3 Part4" use="literal">
</input>
</operation>

<definitions
  xmlns:stns="(SchemaTNS)"
  xmlns:wtns="(WsdITNS)"
  targetNamespace="(WsdITNS)">

<schema targetNamespace="(SchemaTNS)"
  elementFormDefault="qualified">
<element name="SimpleElement" type="xsd:int"/>
<element name="CompositElement" type="stns:CompositType"/>
<complexType name="CompositType">
<all>
  <element name='elem_a' type="xsd:int"/>
```

```

        <element name='elem_b' type="xsd:string"/>
    </all>
</complexType>
</schema>

<message...>
<part name='Part1' type="stns:CompositeType"/>
<part name='Part2' type="xsd:int"/>
<part name='Part3' element="stns:SimpleElement"/>
<part name='Part4' element="stns:CompositeElement"/>
</message>
ö
</definitions>

```

SOAP 메시지는 파트에 정의된 순서를 따라야 합니다. 아래의 SOAP 예에서 Part1 요소가 Part2, Part3, Part4 요소에 선행합니다. 적절한 BO 분석을 위해 이 순서는 유지 보수되어야 합니다.

```

<soapenv:body... xmlns:mns=" (MessageNS) "
  xmlns:stns=" (SchemaTNS) ">
  <stns:elem_a>123</stns:elem_a>
  <stns:elem_b>hello</stns:elem_b>
  <soapenc:int>123</soapenc:int>123</soapenc:int>123</soapenc:int>
  <stns:SimpleElement>123</stns:SimpleElement>
  <stns:CompositeElement>
    <stns:elem_a>123</stns:elem_a>
    <stns:elem_b>hello</stns:elem_b>
  </stns:CompositeElement>
</soapenv:body>

```

SOAP 메시지가 document/literal/type 또는 document/encoded/type 형식을 따를 경우 파트 요소도 이 순서를 따라야 합니다. 위 예의 Part1에서 elem_a 태그는 elem_b 태그보다 선행해야 합니다. 이러한 제한은 Data Handler의 Business Object 분석 프로세스를 통해 지시된 사항입니다. 문서 양식에 대한 기본 Business Object 분석에서 첫 번째 요소의 본문 이름과 이름 공간을 사용하므로 이들 요소는 각각의 경우에서 동일한 Business Object가 분석되도록 특정 요청, 응답 또는 결합의 모든 SOAP 메시지에서 동일한 요소여야 합니다.

주: SOAP 메시지가 document/literal/type 또는 document/encoded/type 형식을 따를 경우 요소는 선택적이어서는 안 됩니다.

XML 제한사항

다음의 XML 구조, 기능 및 표기법은 지원되지 않습니다.

- 다차원 배열
- 부분 전송 배열
- 스파스(Sparse) 배열
- 혼합 항목
- maxOccurs가 1보다 큰 순서, 그룹 및 선택 모델 그룹 구성요소

제 6 장 요청 처리용 협업 사용 가능

- 『요청 처리 협업 선택 목록』

이 장에는 요청 처리에 협업을 사용할 수 있게 하기 위해 따라야 하는 단계가 설명되어 있습니다. 협업은 커넥터를 사용하여 웹 서비스를 호출합니다.

요청 처리 협업 선택 목록

Business Object Designer를 사용하여 Business Object를 생성하는 것은 협업 개발 프로세스의 일부입니다. 아래 섹션에 설명된 다음과 같은 작업을 수행하여 웹 서비스를 호출하는 데 협업에서 사용할 수 있는 Business Object를 생성해야 합니다.

1. URL, UDDI 또는 파일 시스템 중 하나로부터 WSDL 문서를 식별하십시오. 이 작업을 위해 타사 도구를 사용합니다. 즉, 웹 서비스 커넥터는 이 작업에 도구를 제공하지 않습니다.
2. Business Object Designer를 열고 WSDL ODA를 실행하십시오. 자세한 정보는 171 페이지의 『WSDL ODA 시작』을 참조하십시오.
3. ODA를 구성하십시오.
4. 선택사항을 확인하십시오.
5. 요청(동기 요청용), 응답 및 결합 Business Object는 물론 SOAP 구성 MO, 프로토콜 구성 MO, 머릿글 컨테이너 및 하위 오브젝트 그리고 각 오브젝트 및 속성에 적합한 응용프로그램 특정 정보를 포함하는 최상위 레벨 Business Object를 생성하십시오. WSDL ODA는 이러한 프로세스를 자동화합니다.

Business Object를 생성한 후 커넥터와 SOAP Data Handler를 사용하여 협업이 웹 서비스를 호출할 수 있도록 다음 작업을 수행해야 합니다. 협업 템플릿 및 오브젝트 작성과 해당 포트 바인딩을 포함하여 협업 개발에 대한 단계는 *IBM WebSphere InterChange Server Collaboration Development Guide*를 참조하십시오. 일반 Business Object와 WSDL ODA에서 생성된 응용프로그램 특정 Business Object 사이의 맵 작성에 대한 자세한 정보는 *IBM WebSphere InterChange Server Map Development Guide*를 참조하십시오.

제 7 장 협업을 웹 서비스로 표시

- 『프로시저 체크리스트』
- 160 페이지의 『Business Object 식별 또는 개발』
- 160 페이지의 『협업 템플릿 선택 또는 개발』
- 160 페이지의 『새 협업 오브젝트의 포트 바인딩』
- 163 페이지의 『WSDL 구성 마법사』
- 165 페이지의 『WSDL 구성 마법사가 TLO 형식으로 된 Business Object 처리』
- 168 페이지의 『처리 요구사항 및 예외』

이 장은 협업을 웹 서비스로 표시하는 설계 프로시저에 대해 설명합니다. 이를 통해 웹 서비스 클라이언트가 협업을 호출할 때 커넥터는 이벤트를 처리할 수 있습니다.

통합된 설계 도구는 협업을 웹 서비스로 표시하는 작업을 단순화합니다. 웹 서비스를 위한 협업과 Business Object를 구성한 후, WSDL 구성 마법사를 사용합니다. 이 마법사는 협업을 웹 서비스로 표시하는 WSDL 문서와 XML 스키마를 작성합니다. WSDL 출력은 협업을 설명할 뿐만 아니라 웹 서비스 클라이언트를 통한 호출의 기초를 형성합니다.

프로시저 체크리스트

협업을 웹 서비스로서 표시하려면 아래 섹션에 설명되는 다음 작업을 수행해야 합니다.

1. 요청으로 사용하고 선택적으로(동기 이벤트 처리용) 응답 및 결합 SOAP 메시지로 사용할 Business Object를 식별하거나 필요에 따라 개발하십시오. 두 가지 방법으로 이 오브젝트를 생성할 수 있습니다. 1) 수동으로 Business Object Designer를 사용합니다. 2) 웹 서비스에 대한 WSDL 인터페이스 파일이 존재할 경우 WSDL ODA를 사용하여 요청 및 기타(응답 및 결합) Business Object를 생성할 수 있습니다. 두 번째 방법을 따를 경우 다음을 수행하십시오.
 - a. 구성 WSDL ODA 구성 등록 정보에 협업의 이름을 지정하십시오. 이 값은 TLO에서 ws_collab ASI를 표시합니다.
 - b. WSDL_URL 또는 UDDI_InquiryAP를 지정하십시오.
WSDL 인터페이스 파일의 I_URL WSDL ODA 구성 등록 정보(이 파일이 네트워크 또는 로컬로 존재할 경우 이 파일에 대한 디렉토리 경로도 지정할 수 있음).

자세한 정보를 보려면 171 페이지의 『WSDL ODA 시작』의 내용을 참조하십시오.

2. Business Object를 사용하려면 협업 템플릿을 개발하거나 기존 협업 템플릿을 선택하십시오.

3. 웹 서비스를 위한 협업 오브젝트와 포트를 작성하십시오.

먼저 협업 오브젝트가 적절하게 Business Object를 채우는지 확인해야 합니다. 협업 오브젝트 작성에 대한 자세한 정보 및 단계별 절차에 대해서는 *Implementation Guide for WebSphere InterChange Server*를 참조하십시오.

주: 협업 오브젝트에는 적당한 변환에 대해 구성된 맵이 있어야 합니다. 맵은 SOAP 요청 메시지에서 수신된 Business Object를 협업이 사용하는 Business Object로 변환합니다. 맵은 또한 협업이 리턴하는 Business Object를 SOAP 응답 메시지에 임베드되는 Business Object로 변환합니다. 맵핑 및 맵핑 절차에 대한 자세한 정보는 *Map Development Guide*를 참조하십시오.

4. WSDL 문서를 작성하려면 WSDL 구성 마법사를 사용하십시오. 유틸리티는 또한 웹 서비스 커넥터를 구성합니다.

주: WSDL 구성 마법사는 구현, 인터페이스 및 하나 이상의 스키마 파일을 작성합니다. 이 문서에서는 이들 출력을 집합적으로 WSDL 문서라고 부릅니다.

5. 필요에 따라 WSDL 문서를 공개하십시오.

주: 커넥터는 WSDL 문서 공개를 위한 도구를 제공하거나 이를 지원하지 않습니다.

Business Object 식별 또는 개발

Business Object Designer를 사용하여 Business Object를 작성하고 Connector Configurator를 사용하여 해당 Business Object를 지원하도록 커넥터를 구성합니다.

Business Object Designer에 대한 자세한 정보는 *Business Object Designer*를 참조하십시오. 웹 서비스 Business Object에 대한 자세한 정보는 25 페이지의 제 3 장 『Business Object 요구사항』을 참조하십시오.

협업 템플릿 선택 또는 개발

선택 또는 개발하는 협업 템플릿에는 웹 서비스로 표시할 하나 이상의 시나리오가 있어야 합니다. 협업 템플릿에 대한 자세한 정보는 *Collaboration Development Guide*를 참조하십시오.

새 협업 오브젝트의 포트 바인딩

Business Object 유형에 적합한 협업 템플릿의 포트를 구성한 후에 협업 오브젝트를 작성하고 포트를 웹 서비스 커넥터의 인스턴스에 바인드해야 합니다.

새 협업 오브젝트를 작성하고 포트를 웹 서비스 커넥터의 인스턴스에 바인드하려면 다음을 수행하십시오.

1. 마우스 오른쪽 단추로 협업 오브젝트 폴더를 누르고 새 협업 오브젝트 작성을 선택 하십시오. 그러면 템플릿 목록을 표시하는 새 협업 작성 창이 표시됩니다(그림 54 참조).

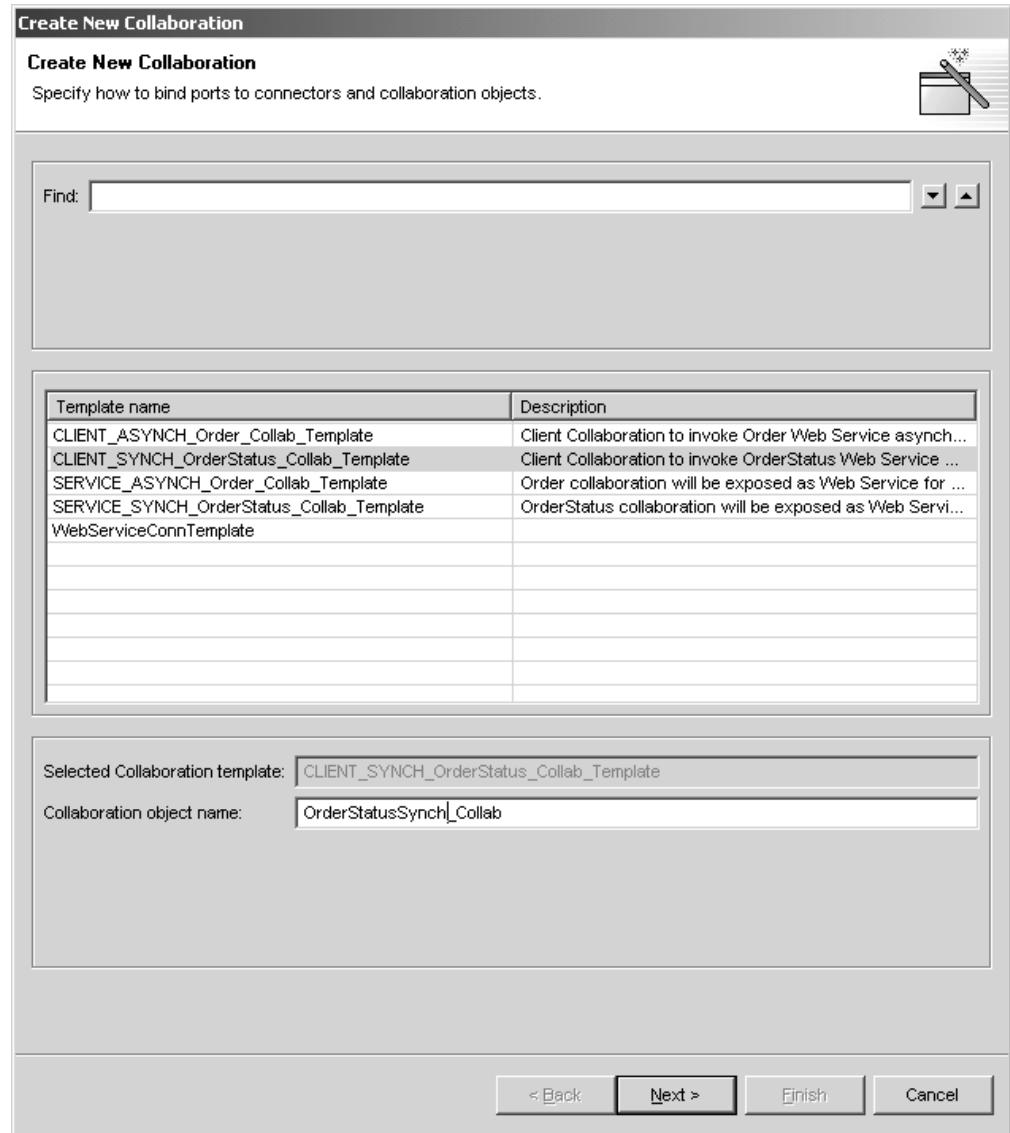


그림 54. 새 협업 작성 창

2. 템플릿 이름에서 협업 템플릿을 선택하고 협업 오브젝트 이름 필드에 협업 오브젝트 이름을 입력하십시오. 그러면 그림 55에 표시된 대로 바인드 포트 창이 표시됩니다.

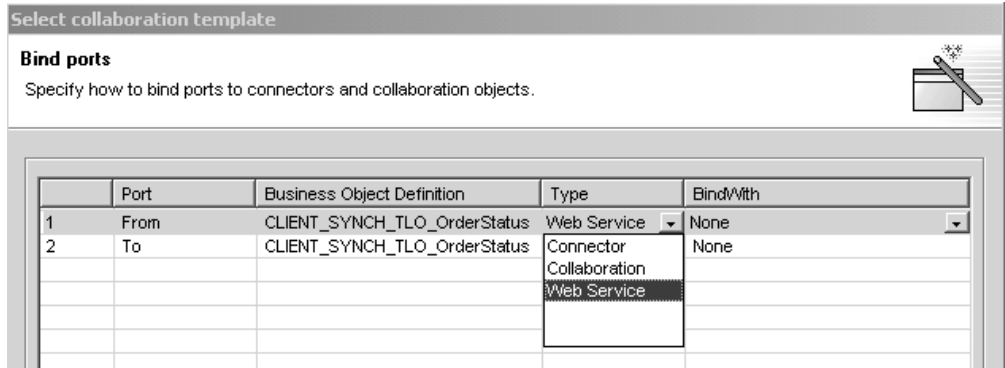


그림 55. 바인드 포트 창

3. 포트를 선택하고, 유형 화살표를 눌러 포트에 대한 풀다운 메뉴를 표시한 후 WebService를 선택하십시오(그림 55에 표시).

웹 서비스 커넥터의 모든 인스턴스에는 ConnectorType 응용프로그램 특정 등록 정보가 있습니다. 기본적으로, 이 등록 정보는 WebService로 설정됩니다. System Manager의 협업 포트 바인드 창은 ConnectorType 등록 정보의 값을 사용하여 웹 서비스 커넥터인 커넥터를 판별합니다.

4. BindWith 화살표를 눌러 커넥터 인스턴스 목록을 표시하십시오. 시스템 관리자는 ConnectorType 등록 정보의 값이 WebService로 설정된 커넥터의 인스턴스를 표시합니다. 웹 서비스 커넥터의 인스턴스를 선택하십시오(그림 56에 예가 나와 있습니다).

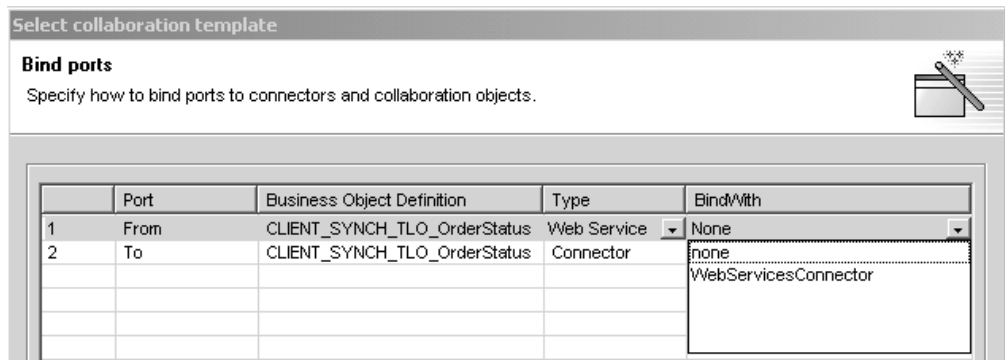


그림 56. 웹 서비스 커넥터의 인스턴스 선택

5. 완료 버튼을 누르십시오.

이제 WSDL 구성 마법사를 실행할 준비가 되었습니다.

WSDL 구성 마법사

협업 오브젝트를 작성하고 해당 트리거링 포트를 웹 서비스 컨넥터의 인스턴스에 바인드하면, WSDL 구성 마법사를 사용할 준비가 되었습니다. 유틸리티는 협업, Business Object 정의 및 컨넥터에 지정된 바인딩, 포트 이름, 조작 및 기타 데이터를 사용하여 WSDL 구현 파일(*.impl.wsdl), WSDL 인터페이스 파일(*.wsdl), xml 스키마 파일(*.xsd)을 생성합니다. 이러한 파일은 웹 서비스로 표시된 협업의 복합 파일이므로 이들 파일을 별도의 파일 또는 하나의 파일로 생성할 것인지 여부를 유틸리티에서 지정할 수 있습니다. 유틸리티는 HTTP, HTTPS 및 JMS 프로토콜을 통해 SOAP를 지원합니다. 프로토콜 리스너 프레임워크의 구성 정보는 컨넥터 특정 등록 정보 ProtocolListenerFramework에서 검색합니다. 이 등록 정보는 또한 리스너 목록을 사용 가능하게 합니다.

마법사 실행

WSDL 구성 마법사를 실행하려면 다음을 수행하십시오.

1. 웹 서비스에 대해 구성된 협업 오브젝트를 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 웹 서비스로 표시를 선택하십시오. 그림 57에 나타난 대로 WSDL 구성 마법사가 표시됩니다.

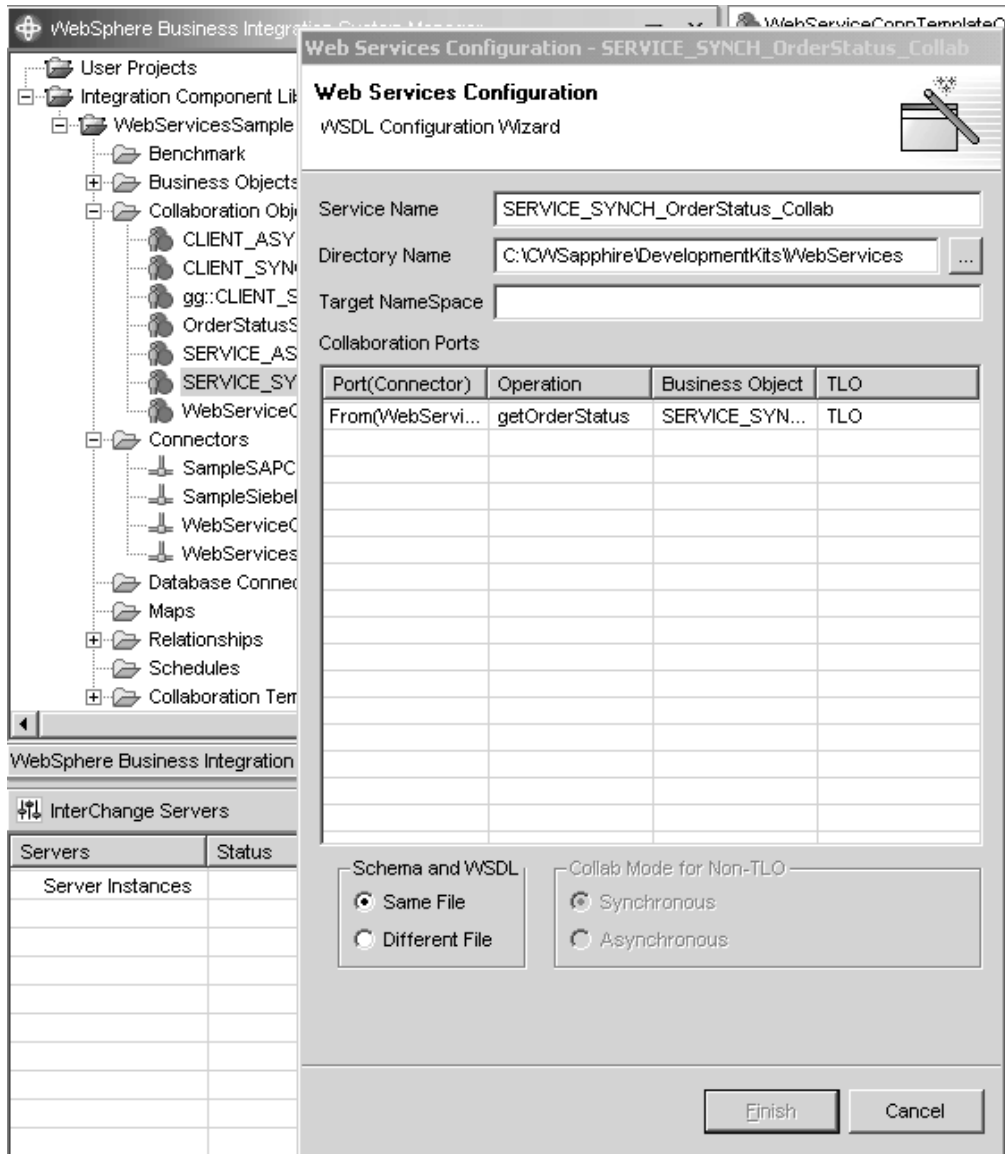


그림 57. WSDL 구성 마법사

그림 57에 표시된 대로, 열은 다음과 같습니다.

- 포트(커넥터) 웹 서비스 커넥터에 바인드된 협업 오브젝트의 트리거링 포트. 마법사는 협업 오브젝트에서 이 정보를 가져옵니다.
- 조작 Business Object가 TLO인 경우, 마법사는 요청 Business Object의 SOAP 구성 Mo BodyName 속성에서 이 정보를 확보합니다. Business Object가 비TLO인 경우, 마법사는 Business Object 이름과 포트 이름을 결합합니다.
- **Business Object** 스키마를 작성하는 데 사용됩니다. 마법사는 이 트리거링 포트에 대한 커넥터의 지원되는 Business Object에서 이 정보를 확보합니다.

2. 필요에 따라 다음을 입력하십시오.

- 서비스 이름 기본적으로, 협업 오브젝트를 설명하는 데 사용한 이름

- 디렉토리 이름 웹 서비스, 협업 템플릿 및 오브젝트용 어댑터가 있는 위치
 - 대상 이름 공간 웹 서비스로 표시 중인 협업의 URL
 - 협업 포트 이 필드의 정보는 협업 오브젝트 구성 절차 중 바인드 포트 창에서 지정한 대로입니다.
 - 비TLO에 대한 협업 모드 TLO를 사용하려는 경우에 이것은 적용되지 않습니다. 그렇지 않으면, 비TLO 오브젝트를 입력으로 사용할 경우 동기 또는 비동기를 지정해야 합니다.
 - 스키마 및 WSDL 출력을 단일 파일 또는 별도의 파일로 할지의 여부를 지정하십시오.
3. 완료 버튼을 누르십시오. 유틸리티가 사용자가 입력한 입력과 스펙을 바탕으로 출력을 생성하는데, 이들은 다음 섹션에 요약되어 있습니다.

WSDL 구성 마법사가 TLO 형식으로 된 Business Object 처리

구성 마법사는 웹 서비스 커넥터에 바인드되는 협업 오브젝트의 각 트리거링 포트에 대한 WSDL 조각을 작성합니다. 조각의 작성은 이 협업의 호출과 연관된 Business Object를 기본으로 합니다.

구성 마법사가 오브젝트 레벨 ASI ws_eventtlo를 읽어서 Business Object가 TLO 형식으로 되어 있음을 판별합니다. ASI 등록 정보가 true로 설정될 경우, Business Object는 TLO입니다. TLO를 사용하여 다음 WSDL 등록 정보를 찾습니다.

- 조각 이름 및 BodyNS 마법사가 TLO 형식으로 된 Business Object를 찾을 때 TLO의 SOAP 요청 Business Object 내의 SOAP 구성 MO의 BodyName 등록 정보를 사용하여 조각 이름을 작성합니다. 마찬가지로, 마법사는 동일한 SOAP 구성 MO에서 BodyNS 등록 정보가 될 메시지 이름 공간을 판별합니다.
- 실행 모드 TLO의 Business Object 레벨 ASI에서 ws_mode 등록 정보를 조사하여 마법사는 모드가 동기 또는 비동기인지를 판별하고 각각 REQUEST_RESPONSE 또는 ONE_WAY WSDL을 작성합니다.

TLO를 기본으로 WSDL 조각을 작성하기 위해 협업은 맵을 사용하거나 사용하지 않는 두 가지 방법으로 구성될 수 있습니다.

맵이 있는 TLO: 협업은 일반적으로 GBO(일반 Business Object) 요청을 승인하도록 구성됩니다. 즉, 협업 템플릿 트리거링 포트가 GBO에 등록합니다. 이 경우에 TLO를 사용하려면, 협업이 웹 서비스 커넥터에 바인드되어야 하며 커넥터가 맵을 통한 GBO를 TLO로 변환을 지원해야 합니다. 그림 58에서는 이 시나리오를 보여줍니다.

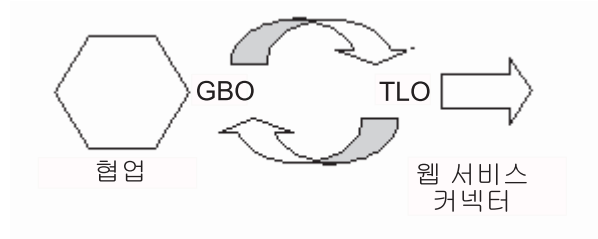


그림 58. 맵이 있는 TLO

협업과 커넥터가 이 방법으로 구성될 때, 마법사는 WSDL 문서에 설명된 조작을 작성하기 위해 TLO Business Object가 사용될 것임을 결정합니다. 이 결정은 커넥터가 지원하는 Business Object 및 연관된 맵을 조사하여 이루어집니다. 구성된 맵이 항상 협업의 GBO를 단 하나의 TLO로 변환하는 것이 웹 서비스 커넥터의 런타임 처리에 중요합니다. 또한, 인바운드 맵의 소스 및 목적지 Business Object가 각각 아웃바운드 맵의 목적지 및 소스 Business Object로 변환하는 것이 중요합니다.

맵이 없는 TLO: 마법사는 또한 맵이 없는 TLO 처리를 지원합니다. 이 경우, 협업 템플릿의 트리거링 포트가 TLO에 직접 등록됩니다. 웹 서비스 커넥터가 TLO를 지원하기 때문에 맵이 필요하지 않습니다. 그림 59에서는 이 시나리오를 보여줍니다.

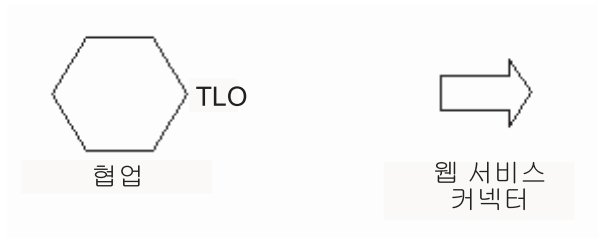


그림 59. 맵이 없는 TLO

협업과 커넥터가 이 방법으로 구성되었을 때, 마법사는 협업에서 발견되는 TLO Business Object를 사용하여 WSDL 문서에 설명된 조작을 작성합니다. 마법사는 이 포트에 대해 구성된 맵이 없음을 판별합니다.

WSDL 구성 마법사가 비TLO 형식으로 된 Business Object 처리

비TLO Business Object에 대한 지원으로 웹 서비스로 표시하기 위해 이미 존재하는 협업과 맵을 사용할 수 있습니다. 이런 이유로 마법사는 또한 TLO 형식이 아닌 Business Object를 사용한 WSDL 조작 작성을 지원합니다.

TLO 프로세스와 유사하게, 마법사는 오브젝트 레벨 ASI ws_eventtlo를 읽어서 Business Object가 비TLO 형식으로 되어 있음을 판별합니다. ASI 등록 정보가 존재하지 않거나 존재하지만 true가 아닌 다른 값으로 설정된 경우, 이 Business Object는 비TLO입니다. 비TLO는 웹 서비스 TLO 구조를 고수하지 않는 모든 Business Object입니다. 비TLO를 사용하여 마법사는 다음 등록 정보를 검색합니다.

- 조작 이름 및 **BodyNS** 마법사가 비TLO 형식으로 된 Business Object를 찾을 때 협업 이름, Business Object 이름 및 포트 이름의 조합을 사용하여 조작 이름을 작성합니다. WSDL 조작에 대한 본문 이름 공간은 WSDL 구성 마법사의 대상 이름 공간 항목을 사용하여 구성됩니다.
- **WSCollaborations** 마법사는 그림 60에 샘플 WSCollaborations 등록 정보가 나와 있습니다.

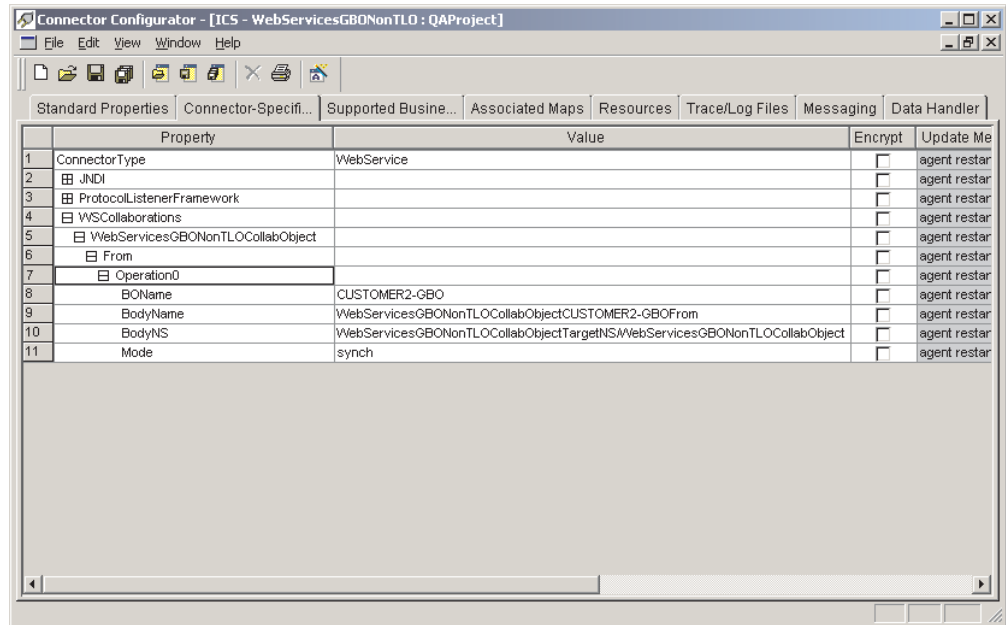


그림 60. WSCollaborations

- **실행 모드** WSDL 조작의 실행 모드는 WSDL 구성 마법사에 있는 비TLO를 기본으로 WSDL 조작을 작성하기 위해 협업은 맵을 사용하거나 사용하지 않는 두 가지 방법으로 구성될 수 있습니다.

맵이 있는 비TLO: 협업은 일반적으로 GBO(일반 Business Object) 요청을 승인하도록 구성됩니다. 동시에, 협업에서 비TLO Business Object로 GBO를 변환하는 사전 존재하는 맵이 있을 수 있습니다. 그림 61에서는 이 시나리오를 보여줍니다.

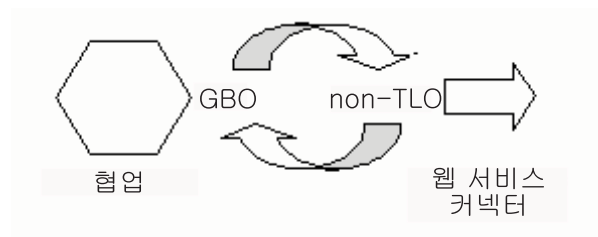


그림 61. 맵이 있는 비TLO

이 경우, 마법사는 비TLO Business Object를 사용하여 WSDL 문서에 기술된 WSDL 조작을 작성합니다. 구성된 맵이 항상 협업의 GBO를 단 하나의 TLO로 변환하는 것이 웹 서비스 커넥터의 런타임 처리에 중요합니다. 또한, 인바운드 맵의 소스 및 목적지 Business Object가 각각 아웃바운드 맵의 목적지 및 소스 Business Object로 정확하게 변환하는 것이 중요합니다.

맵이 없는 비TLO: 아주 특별한 경우에는 GBO 이외의 Business Object로부터의 요청을 승인하도록 협업을 구성할 수 있습니다. 이 경우, 비TLO는 협업에 대한 직접 Business Object이고 맵은 존재하지 않습니다. 그림 62에서는 이 시나리오를 보여줍니다.

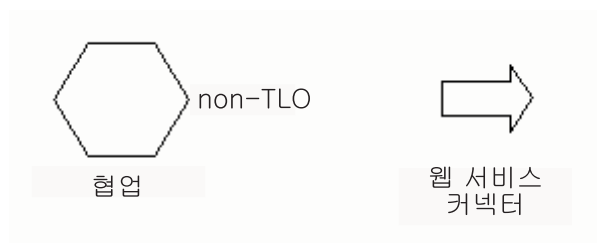


그림 62. 맵이 없는 비TLO

이 경우, 마법사는 이 포트에 대해 맵이 구성되지 않음을 판별하며, 따라서 비TLO Business Object를 사용하여 WSDL 문서에 기술된 WSDL 조작을 작성합니다.

처리 요구사항 및 예외

이 섹션에서는 달리 명시되지 않는 한 모든 유형의 오브젝트(TLO 및 비TLO)에 적용되는 WSDL 구성 마법사의 요구사항에 대해 설명합니다. 웹 서비스 TLO의 Business Object 요구사항에 대한 자세한 정보는 25 페이지의 제 3 장 『Business Object 요구사항』을 참조하십시오.

주: WSDL 도구가 읽는 Business Object 중에, 다음 경우에만 국제화 문자를 가질 수 있습니다.

- elem_name
- elem_ns
- attr_name
- attr_ns
- BodyName
- BodyNS
- type_name
- type_ns

SOAP 구성 MO에서 Use 등록 정보의 지원: WSDL 구성 마법사는 SOAP 구성 MO에서 Use 등록 정보를 지원하지만, SOAP 요청 BO 및 해당 SOAP 응답 BO의 Use 값이 다른 경우, 오류를 발생시킵니다. Use 값을 리터럴로 설정하거나 WSDL 문서를 생성하도록 인코딩할 수 있습니다. Use 등록 정보 및 해당 값에 대한 자세한 정보는 123 페이지의 『SOAP 메시지의 Style 및 Use 영향』을 참조하십시오.

SOAP 구성 MO에서 양식 지원: rpc 양식만이 웹 서비스로 협업을 나타내도록 지원됩니다. SOAP 구성 MO에서 문서로 양식이 지정된 경우, 마법사에서 오류가 발생합니다.

결함 처리: SOAP 결함 Business Object 내의 details 속성에는 단 하나의 하위 속성만 있을 수 있습니다. 그렇지 않으면, 유틸리티는 오류를 생성합니다.

유틸리티는 결함 Business Object를 승인합니다. 여러 개의 결함 Business Object가 발생하면 유틸리티는 첫 번째 또는 기본 결함 Business Object의 머리글 컨테이너를 처리합니다. 처리는 다음과 같습니다.

- 바인딩 섹션 내에서 soap:fault 요소에 이름 공간을 지정하지 않습니다.
- 결함은 항상 문서 양식과 사용 리터럴을 사용하여 지정됩니다.
- 메시지 파트는 요소 속성을 사용하여 지정됩니다.

머리글 결함 처리: 머리글 결함은 WSDL 문서 바인딩 섹션 내에서 soap:header의 하위 요소인 soap:headerfault로 처리됩니다. 머리글 결함은 다음과 같이 머리글 하위 Business Object에 지정된 headerfault ASI를 사용하여 처리됩니다.

- soap:headerfault 요소에 이름 공간을 지정하지 않습니다.
- 머리글 결함은 항상 문서 양식과 사용 리터럴을 사용하여 지정됩니다.
- 메시지 파트가 속성 유형 대신 요소 유형을 사용하여 지정됩니다.

머리글 처리: 여러 개의 머리글 속성은 SOAP 머리글 컨테이너 Business Object 내에 SOAP 머리글 하위 Business Object로 지정됩니다. 머리글 컨테이너 Business Object는 ASI soap_location=SOAPHeader로 식별됩니다. 유틸리티 처리 중에 머리글 컨테이너 Business Object 내의 각 속성에 대한 바인딩 섹션 내에 soap:header 요소가 작성되고 다음의 규칙이 적용됩니다.

- 머리글은 항상 문서 양식과 사용 리터럴을 사용하여 지정됩니다.
- 메시지 파트가 속성 유형 대신 요소 유형을 사용하여 지정됩니다.
- elem_ns를 지정하지 않은 경우 머리글은 본문 이름 공간에 작성됩니다.

주: 머리글 컨테이너 Business Object는 SOAP 요청, 응답 또는 결함 Business Object의 하위가 될 수 있습니다. 이름 공간 속성은 soap:header 요소에 지정되지 않습니다.

elem_ns ASI 처리: 유틸리티는 메시지 파트 레벨에서 elem_ns ASI를 무시합니다. 대신, elem_ns는 두 번째 이하 레벨 속성에서 사용됩니다. elem_ns를 지정하는 경우, 두 번째 레벨 Business Object 속성을 별도의 이름 공간에 정의할 수 있습니다.

JMS 프로토콜 처리: WSDL 문서의 포트 섹션에 있는 SOAP/JMS 바인딩에는 jms:address 요소가 있습니다. 다음은 jms:address 요소의 예입니다 ("?" 접미부가 있는 속성은 선택적입니다).

```
<jms:address destinationStyle = "queue"
jmsVendorURI = "http://ibm.com/ns/mqseries"?
initialContextFactory = "com.ibm.NamingFactory"?
jndiProviderURL = "iiop://something:900/wherever"?
    jndiConnectionFactoryName = "orange"
    jndiDestinationName = "fred"

jmsProviderDestinationName="trash" />
```

LookupQueuesUsingJNDI 커넥터 등록 정보가 true로 설정되는 경우, InputQueue 등록 정보 값은 SOAP/JMS 바인딩의 jms:address 요소의 jndiDestinationName 속성에 해당합니다. jms:address 요소는 wsdl:port 섹션에 지정됩니다.

LookupQueueUsingJNDI가 false로 설정되는 경우, jmsProviderDestinationName 속성은 InputQueue로 설정됩니다. InputQueue는 Listener_JMS hierarchical 등록 정보에서 사용 가능한 커넥터 등록 정보입니다. initialContextFactory, jndiProviderURL 및 jndiConnectionFactoryName 등록 정보는 동기 처리에 대해서만 지정됩니다.

HTTP 프로토콜 처리: WSDL 문서의 샘플 포트 섹션은 다음과 같습니다.

```
<service name="StockQuoteWebService">
<port name="StockQuoteWebServicePort" binding="intf:StockQuoteBinding">
<soap:address location="http://localhost:8080/soap/servlet/messagerouter"/>
</port>
</service>
```

WSDL 구성 마법사는 문맥 경로의 호스트 이름 및 포트의 값을 사용합니다. 문맥 경로가 호스트 이름과 포트 없이 상대 경로만을 포함하는 경우, Listener_HTTP 구성 등록 정보 아래에 있는 호스트 이름 및 포트 등록 정보 값이 soap:address xml 요소에서 위치 속성을 지정하는 데 사용됩니다.

제 8 장 WSDL ODA 사용

- 『WSDL ODA 시작』
- 172 페이지의 『WSDL ODA 실행』
- 173 페이지의 『에이전트 구성』
- 175 페이지의 『WSDL 문서 지정』
- 177 페이지의 『선택사항 확인』
- 178 페이지의 『오브젝트 생성』
- 179 페이지의 『제한사항』

주: WSDL(Web Services Description Language) ODA(Object Discovery Agent)는 요청 처리 및 이벤트 처리 (WSDL 인터페이스 파일이 사용 가능할 경우)에 필요한 Business Object를 생성하는 데 사용됩니다.

협업은 커넥터를 사용하여 웹 서비스를 호출합니다. 아니면 협업을 웹 서비스로 표시할 수 있습니다. 웹 서비스는 WSDL(Web Services Description Language)을 사용하여 설명합니다. 이 장은 WSDL(Web Services Description Language) ODA(Object Discovery Agent)를 사용하여 Business Object를 생성하는 방법에 대해 설명합니다. 커넥터 및 SOAP Data Handler는 협업이 웹 서비스를 호출할 때와 협업을 Business Object로 표시할 때 이 Business Object를 사용합니다.

두 가지 목적으로 WSDL ODA를 사용하여 Business Object를 생성할 수 있습니다.

1. WSDL ODA는 WSDL 구현 파일을 가져와 협업에 필요한 Business Object를 생성하여 외부 웹 서비스를 호출합니다.
2. WSDL ODA는 WSDL 인터페이스 파일을 가져와 웹 서비스로 표시된 협업에 필요한 Business Object를 생성할 수 있습니다.

Business Object Designer 사용 시 WSDL ODA를 실행할 수 있습니다. WSDL ODA는 WSDL 문서를 읽고 커넥터와 SOAP Data Handler에서 필요로 하는 Business Object를 작성합니다. WSDL ODA는 Business Object 개발 작업을 단순화합니다.

주: WSDL ODA는 WSDL에 있는 SOAP/HTTP 및 SOAP/JMS 바인딩을 처리합니다.

WSDL ODA 시작

다음 스크립트 중 하나를 사용하여 WSDL ODA를 시작할 수 있습니다.

- Windows
 - start_WSDLODA.bat

주: 또한 설치 프로그램이 Windows 환경에서 자동으로 작성하는 바로 가기를 사용하여 WSDL ODA를 시작할 수도 있습니다.

- UNIX
 - start_WSDLODA.sh

Business Object Designer를 사용하여 WSDL ODA를 선택, 구성 및 실행합니다. Business Object Designer는 각 스크립트 또는 일괄처리 파일의 AGENTNAME 변수에 지정된 이름으로 각 ODA를 찾습니다.

WSDL ODA 실행

ODA(Object Discovery Agent)는 요청 처리를 위한 Business Object 바인딩 작업을 단순화합니다. Business Object Designer는 사용 가능한 모든 ODA에 그래픽 인터페이스를 제공하고 필요한 에이전트를 찾을 수 있도록 도움을 줍니다. WSDL ODA의 이름은 기본적으로 WSDLODA입니다. WSDL 마법사에 나타나는 이름은 start_WSDLODA.bat 또는 start_WSDLODA.sh 파일에 있는 AGENTNAME 변수의 값에 따라 달라집니다. ODA 및 Business Object 정의, ODA 구성, 시작 및 사용 방법에 대해서는 *IBM WebSphere Business Object Development Guide*를 참조하십시오. 아래의 절차를 수행하는 동안 필요에 따라 문서를 참조하십시오.

Object Discovery Agent를 시작한 후, 다음 단계에 따라서 WSDL ODA를 실행하십시오.

1. Business Object Designer를 여십시오.
2. 파일 메뉴에서, ODA를 사용하여 새로 작성... 하위 메뉴를 선택하십시오. Business Object Designer는 Business Object 마법사의 에이전트 선택 대화 상자를 표시합니다. 그림 63에서는 이 창에 대해 설명합니다.
3. 에이전트 찾기 단추를 눌러 모든 실행 중인 에이전트를 표시하고 WSDL ODA를 선택하십시오.

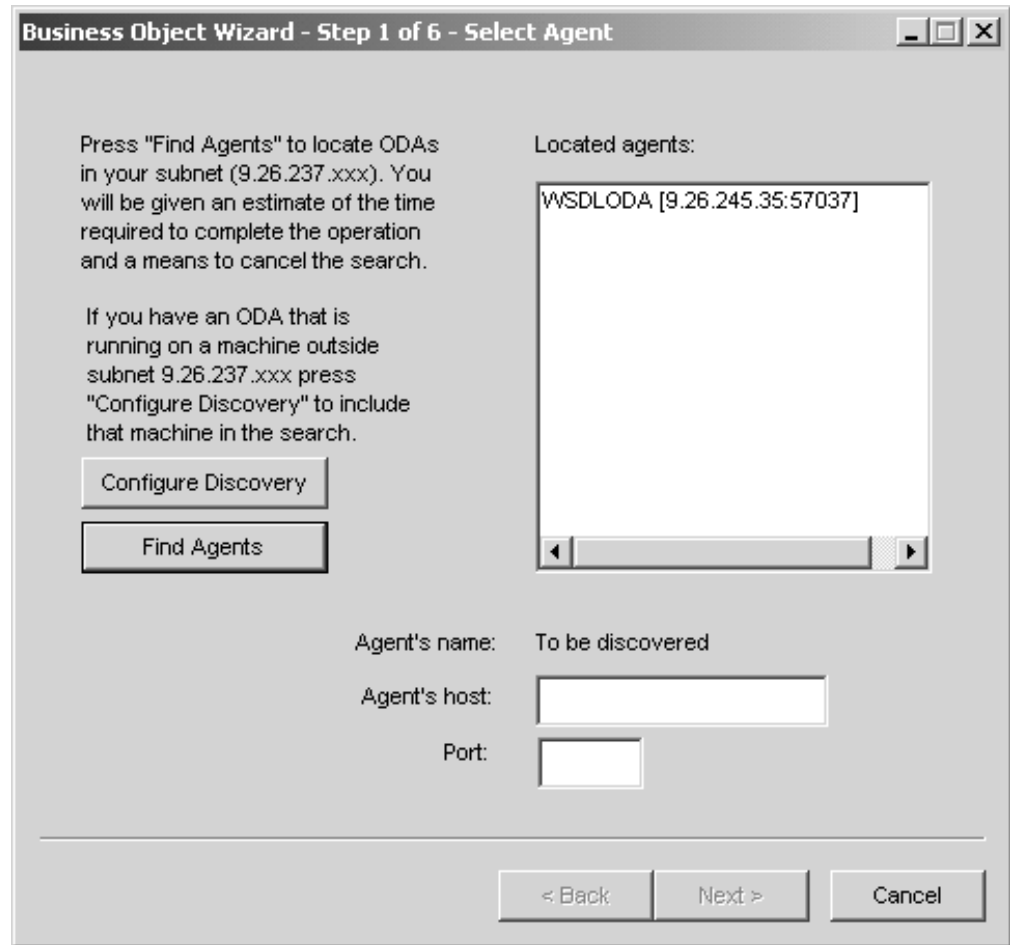


그림 63. 에이전트 선택 창

Business Object Designer가 WSDL ODA를 찾지 못하는 경우, ODA 설정을 확인하십시오.

4. 찾은 에이전트 분할창 목록에서 WSDL ODA를 선택하고 다음을 누르십시오.
그러면 지정해야 하는 구성 등록 정보를 보여주는 에이전트 구성 마법사 창이 표시됩니다.

에이전트 구성

그림 64에서는 WSDL ODA Business Object 마법사의 에이전트 구성 창을 보여줍니다.

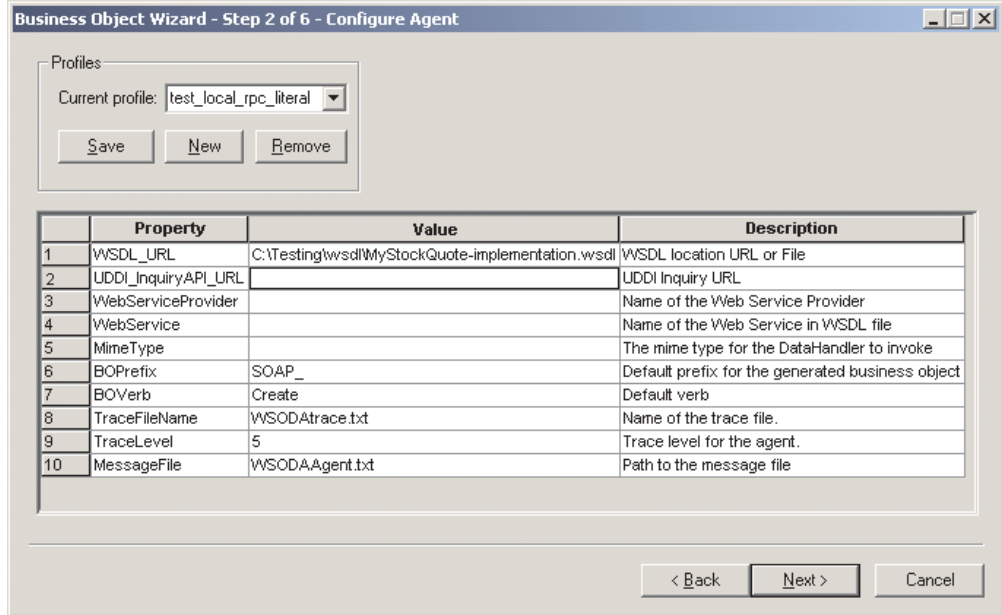


그림 64. 에이전트 구성 창

표 47은 WSDL ODA에 구성해야 하는 등록 정보를 나열합니다.

주: SDL ODA를 처음 사용할 때 각 구성 등록 정보의 값을 지정해야 합니다. 값을 지정한 후 저장 단추를 눌러 프로파일에 등록 정보 값을 저장할 수 있습니다. 다음에 SDL ODA를 사용할 때 “프로파일 선택” 상자에서 저장된 프로파일을 선택할 수 있습니다.

표 47. WSDL ODA 구성 등록 정보

등록 정보	유형	필수	기본값	설명
WSDL_URL	String	예(UDDI를 지정하지 않을 경우)	없음	WSDL 문서의 URL. 이 값은 또한 로컬 WSDL 파일의 절대 경로로 설정될 수 있습니다. URL을 기본 언어로 지정할 수 있습니다.
UDDI_InquiryAPI_URL	String	예(UDDI의 경우)	없음	UDDI 조회 API의 URL
WebServiceProvider	String	예(UDDI의 경우)	없음	대상 웹 서비스 제공자의 이름. 이 이름은 보통 UDDI 레지스트리에 공개된 비즈니스 이름입니다. 이 항목은 대소문자가 구분되며 영문만 사용해야 합니다.
WebService	String	예(UDDI의 경우)		웹 서비스의 이름. 이 항목은 대소문자가 구분되며 영문만 사용해야 합니다.

표 47. WSDL ODA 구성 등록 정보 (계속)

등록 정보	유형	필수	기본값	설명
MimeType	String	아니오	xml/soap	컨텍터가 호출하는 Data Handler의 MIME 유형. Business Object TLO에서 기본값으로 설정되며 영문만 가능합니다.
BOPrefix	String	아니오	SOAP_	이것이 작성되는 각 Business Object의 앞에 추가됩니다. 최대 8문자까지 사용자 구성 가능(영문만)
BOVerb	String	예	Create	요청 Business Object와 선택적으로 응답 및 결합 Business Object의 SOAP 구성 MO에 설정한 Verb
협업	문자열	아니오	없음	이 값은 TLO의 ws_collab ASI를 표시하며 이벤트 처리에 필요한 오브젝트를 생성할 때에는 필수입니다.

다음 섹션은 에이전트 구성 창에서 WSDL 문서를 지정하는 방법에 대해 설명합니다.

WSDL 문서 지정

웹 서비스 Business Object는 WSDL 문서에서 생성됩니다. 이 섹션에서는 ODA의 에이전트 구성 창에서 WSDL 문서 소스를 선택하고 지정하는 방법을 보여줍니다.

WSDL 문서는 로컬 파일 시스템에 상주하거나 웹의 URL 위치 또는 UDDI 레지스트리에 상주할 수 있습니다. WSDL 문서가 상주하는 위치를 지정하면 WSDL ODA가 이 위치를 검색합니다(자세한 WSDL 서비스 설명은 여러 문서로 구성되어 있을 수 있습니다).

URL 위치에서 WSDL 문서 가져오기

위의 그림 64에서처럼

1. 구성 등록 정보 WSDL_URL에 WSDL 문서의 URL을 지정하십시오.

그러면 ODA는 WSDL 문서에서 웹 서비스 목록을 검색하여 가져온 문서의 URL을 분석합니다. WSDL_URL 등록 정보를 통해 URL 구문(예: file:///C:/test/wsdl) 또는 절대 경로(예: C:\test\wsdl)를 사용하여 로컬 파일 시스템의 WSDL 파일 위치를 지정할 수도 있습니다. ODA가 이 문서와 해당되는 종속 문서(가져온 모든 문서)에 액세스할 수 있도록 해야 합니다.

2. 다음을 누르십시오.

ODA는 웹 서비스 제공자 대한 URL을 조회하고 이 URL 위치에서 WSDL에 정의된 서비스 목록을 검색한 후 그림 65에서처럼 목록을 표시합니다.

주: WSDL ODA는 SOAP/JMS 또는 SOAP/HTTP 바인딩만이 있는 포트를 표시하고 다른 유형의 바인딩은 제외합니다.

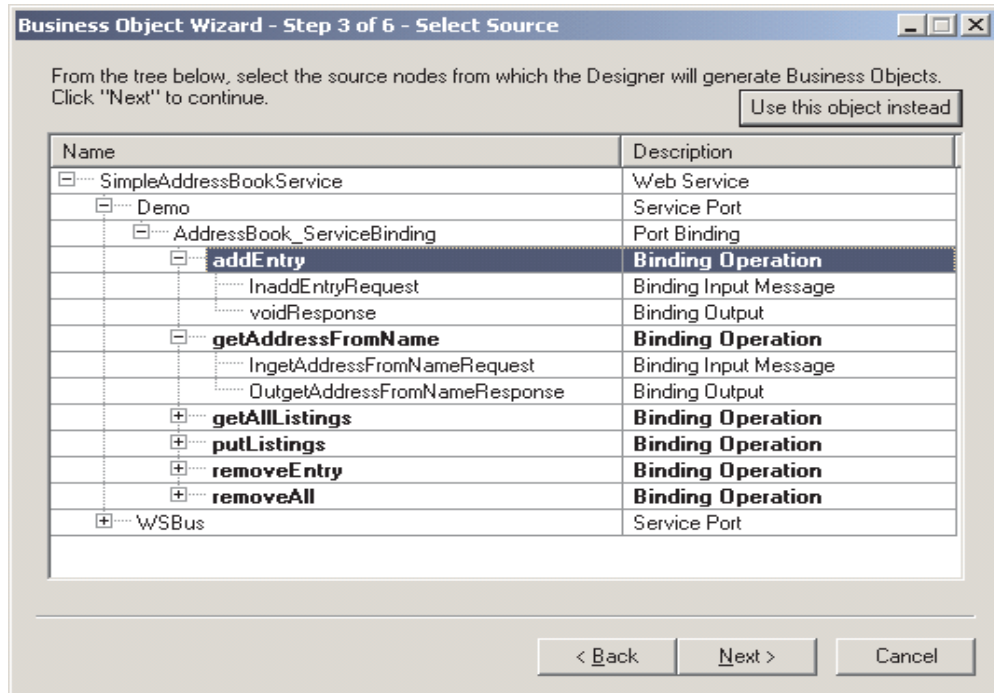


그림 65. 소스 선택 창

3. 포트 목록에서 조작 중 하나만을 선택하십시오(선택 가능한 조작이 강조표시). 서비스 또는 포트 노드는 선택할 수 없으며 표시용으로만 사용됩니다. WSDL 조작 유형에는 ONE_WAY, REQUEST_RESPONSE, SOLICIT_RESPONSE 및 NOTIFICATION이 있습니다. WSDL ODA는 REQUEST_RESPONSE 및 ONE_WAY 조작만 지원하고 표시합니다.
4. 다음을 눌러 177 페이지의 『선택사항 확인』으로 이동하십시오.

UDDI 레지스트리에서 WSDL 문서 가져오기

ODA는 또한 URL 위치 대신 UDDI 레지스트리에서 WSDL 문서를 검색할 수도 있습니다. 이를 위해서는 다음을 수행하십시오.

1. “검색 키”에 대한 에이전트 구성 창에서 다음 등록 정보를 지정하십시오.
 - UDDI_InquiryAPI_URL(예: <https://uddi.ibm.com/ubr/inquiryapi>)
 - WebServiceProvider(예: IBM Corporation)
 - WebService(예: StockQuoteService)
 - WSDL ODA는 UDDI 레지스트리 내에서의 조회를 위해 정확한 이름 일치(findQualifier)를 사용합니다. 매개변수에 대해 유효한 값을 입력하는지 확인하십시오. 일반 UDDI 브라우저를 사용하여 서비스 제공자가 제공하는 서비스를 찾을 수 있습니다.

WSDL ODA는 표 47에 설명된 이 등록 정보를 사용하여 UDDI 레지스트리에 연결합니다.

2. 다음을 누르십시오.

ODA는 웹 서비스 제공자에 대한 UDDI 레지스트리를 조회하고 사용자가 지정한 웹 서비스 매개변수와 일치하는 서비스 목록을 검색합니다. WSDL ODA는 그림 65에서처럼 창에 웹 서비스 제공자가 제공하는 서비스 목록을 표시합니다. UDDI 조회가 둘 이상의 일치물을 리턴할 때 WSDL ODA는 밑줄(_)과 순번이 추가된 목록을 표시합니다(예: StockQuoteService_1, StockQuoteService_2).

주: WSDL ODA는 SOAP/JMS 또는 SOAP/HTTP 바인딩만 있는 포트를 표시합니다.

3. 포트 목록에서 조작 중 하나만 선택하십시오. 서비스 또는 포트 노드는 선택할 수 없으며 표시용으로만 사용됩니다. WSDL 조작 유형에는 ONE_WAY, REQUEST_RESPONSE, SOLICIT_RESPONSE 및 NOTIFICATION이 있습니다. WSDL ODA는 REQUEST_RESPONSE 및 ONE_WAY 조작만 지원하고 표시합니다.

4. 다음을 눌러 『선택사항 확인』으로 이동하십시오.

주: 커넥터는 UDDI 버전 2 API만을 지원합니다. 따라서, UDDI 버전 2를 지원하지 않는 UDDI 레지스트리에서 WSDL을 검색할 수 없습니다.

선택사항 확인

웹 서비스 조작 소스를 선택하면 WSDL ODA Business Object 마법사는 그림 66에서처럼 확인 화면을 표시합니다.

1. 선택사항을 확인하십시오.

2. 다음을 눌러 178 페이지의 『오브젝트 생성』으로 이동하십시오.

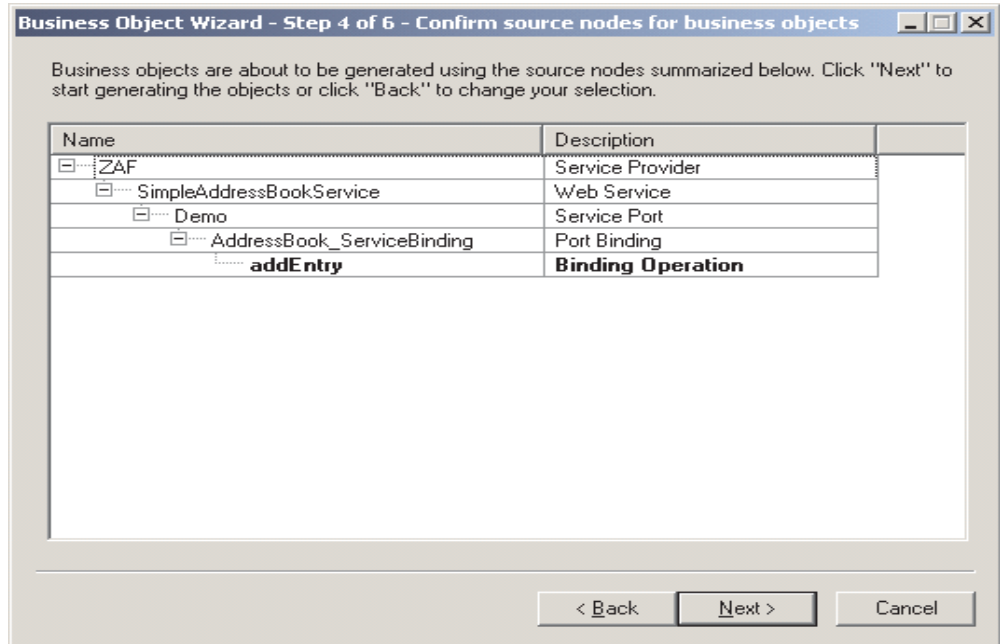


그림 66. 확인 창

오브젝트 생성

WSDL 문서 소스를 확인한 후, WSDL ODA는 호출할 웹 서비스 또는 웹 서비스로서 표시할 협업에 대한 Business Object 및 Meta Object를 생성합니다.

주: WSDL ODA는 최상위 레벨 Business Object에 대한 키 속성을 자동으로 선택할 수 없습니다. 다른 모든 레벨에 있는 Business Object의 경우, WSDL ODA는 첫 번째 속성을 키로 설정합니다. 따라서, Business Object Designer에 WSDL ODA가 생성한 오브젝트를 저장할 때 오류 메시지가 최상위 레벨 오브젝트에 키 속성이 누락되었음을 알려줍니다. 사용자의 비즈니스 데이터 및 Business Object 요구사항을 반영하는 키 속성을 지정한 후 오브젝트를 다시 저장하십시오. 키 속성을 선택할 때 주의하십시오. 키 속성은 이벤트 순서지정에 사용되며 잘못 선택하면 성능 문제가 발생할 수 있습니다.

1. 파일에 Business Object 저장을 선택하거나, 별도의 창에서 Business Object 열기를 선택하십시오. 후자를 선택하면 Business Object Designer가 실행되고 해당 응용프로그램에서 Business Object가 열립니다.
2. 종료 ODA를 선택하고 완료를 누르십시오.

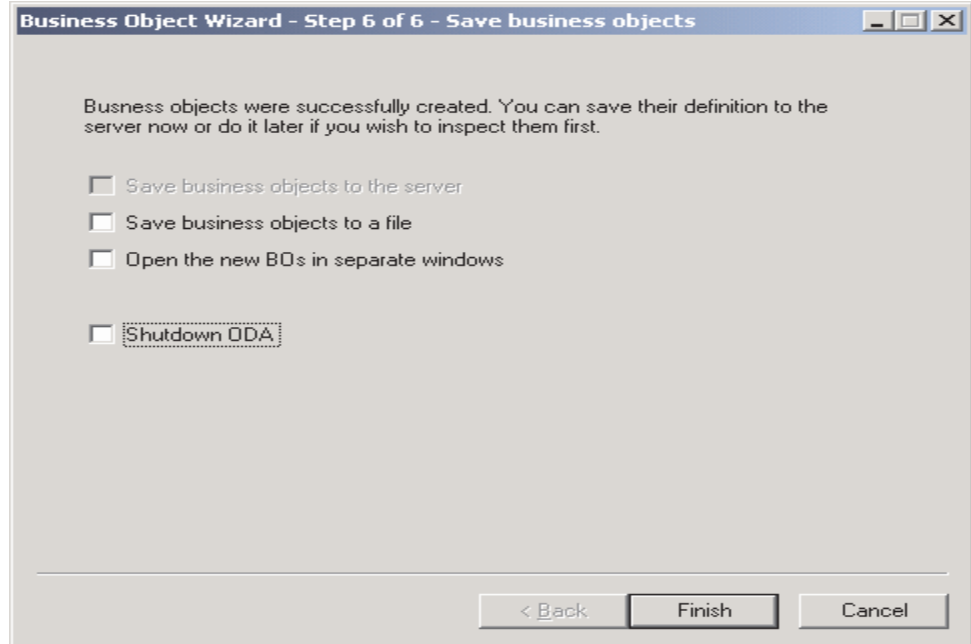


그림 67. 저장 창

요청 처리의 경우, 웹 서비스에 대한 호출에는 요청이 있어야 하며 동기식일 경우, 응답 및 결합 메시지가 있어야 합니다. 이벤트 처리의 경우, 표시된 협업에는 요청이 있어야 하며 동기식일 경우, 응답 및 결합 메시지가 있어야 합니다. WSDL ODA는 모든 레벨의 응용프로그램 특정 정보(ASI), SOAP Data Handler 및 프로토콜 구성 MO를 포함하여 각각에 대해 Business Object를 생성합니다. WSDL 문서의 SOAP 바인딩은 SOAP 메시지의 구조를 판별합니다. Business Object 구조에 대한 자세한 정보는 25 페이지의 제 3 장 『Business Object 요구사항』을 참조하십시오.

제한사항

표 48은 속성 양식, 사용 및 유형과 요소를 사용한 파트 정의의 다양한 조합에 대한 WSDL ODA 지원을 설명합니다.

표 48. WSDL ODA 제한사항

다음을 사용하여 정의된 양식/사용/파트	설명
rpc/인코드/유형	지원됨
rpc/인코드/요소	지원됨
rpc/리터럴/유형	지원됨
rpc/리터럴/요소	지원됨
doc/인코드/유형	지원되지 않음
doc/인코드/요소	지원되지 않음
doc/리터럴/유형	지원됨
doc/리터럴/요소	지원됨

WSDL ODA는 완전히 자체 포함하거나(하나의 파일에) 서비스 요소를 포함하는 구현 파일로 분리되는 WSDL 파일과, 유형, 메시지, portType 및 바인딩을 포함하여 기타 모든 WSDL 요소를 포함하는 인터페이스 파일 그리고 스키마에 해당하는 하나 이상의 파일을 검색할 수 있습니다. WSDL ODA는 예를 들어, 하나의 파일에 메시지와 portType이 있으며 다른 파일에 바인딩이 있는 둘 이상의 인터페이스 파일을 갖는 WSDL 파일을 검색할 수 없습니다.

WSDL 문서의 스키마는 namespace 접두부에 의해 자체 포함되어야 합니다. <types> 요소의 하위인 <schema> 요소에 WSDL 문서의 <definitions>/<types> 요소에 정의된 namespace 접두부를 사용할 수 없습니다. <schema> 요소의 하위 요소에서 사용되는 경우 <schema> 요소에서 namespace 접두부를 다시 정의할 필요가 있습니다. 다음은 자체 포함되지 않은 스키마의 예입니다.

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:NS="NS">
  <types>
    <schema xmlns="http://www.w3.org/1999/XMLSchema">
      <element name="NSElem" type="NS:NSType"/>
    </schema>
  </types>
</definitions>
```

Namespace 접두부 NS는 <definitions> 요소에 정의되며 <schema> 요소의 재정의 없이 사용됩니다. 그러므로 WSDL ODA는 오류를 생성합니다. 이 문제를 해결하려면, 아래 표시된 것처럼 namespace 접두부 NS를 <schema> 요소에서 재정의하십시오.

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:NS="NS">
  <types>
    <schema xmlns="http://www.w3.org/1999/XMLSchema" xmlns:NS="NS">
      <element name="NSElem" type="NS:NSType"/>
    </schema>
  </types>
</definitions>
```

제 9 장 문제점 해결

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

시작 문제점

문제점

알고리즘을 사용할 수 없음/알고리즘 'SSL'을 사용할 수 없음

가능한 솔루션 / 설명

이 오류는 사용자의 JSSE 제공자가 Connector Configurator에 지정된 SSL 버전을 지원하지 않을 때 발생합니다. 해결책: 지원되는 SSL 버전에 대해 JSSE 제공자의 문서를 확인하십시오. IBM JSSE의 경우, *ProductDir/lib/security* 디렉토리에 있는 *java.security* 파일에 다음 항목이 있는지 확인하십시오.

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

여기서 <number>는 보안 제공자 로드를 위한 환경 설정 순서입니다.

키스토어 로드 오류: 키스토어 파일 경로 "<path>"가 잘못 지정됨
:KeyStore가 없음

이 오류는 키스토어 및/또는 트러스트스토어 파일에 대해 잘못된 경로를 지정하는 경우에 발생합니다.

해결책: Connector Configurator의 SSL -> KeyStore 등록 정보에 지정된 키스토어 파일 경로를 확인하십시오. 또한 트러스트스토어를 사용 중인 경우, Connector Configurator에서 SSL -> TrustStore 등록 정보에 지정된 트러스트스토어 파일 경로를 확인하십시오.

KeyManagementError: KeyStore가 훼손됨. KeyManagement 오류

이 오류는 키스토어 및/또는 트러스트스토어가 훼손되거나 손상된 경우에 발생합니다. 이 오류는 또한 암호에 대해 틀린 값을 지정한 경우에도 발생할 수 있습니다.

해결책: 키스토어가 훼손되지 않았는지 확인하십시오. 키스토어를 다시 작성해 보십시오. 또한 SSL -> KeyStorePassword 및 SSL -> TrustStorePassword 커넥터 등록 정보에 유효한 암호를 입력했는지 확인하십시오.

키스토어에서 인증 로드 오류

이 오류는 인증 및/또는 키스토어, 트러스트스토어가 훼손된 경우 발생합니다. 이 오류는 또한 암호에 대해 틀린 값을 지정한 경우에도 발생할 수 있습니다.

해결책: 인증, 키스토어 또는 트러스트스토어가 훼손되었는지 확인하십시오. 또한 SSL -> KeyStorePassword 및 SSL -> TruststorePassword 커넥터 등록 정보에 유효한 암호를 지정했는지 확인하십시오.

서버 소켓 작성 오류, 종료 중: 오류

이 오류는 SOAP/HTTP 또는 SOAP/HTTPS 프로토콜 리스너가 커넥터 등록 정보에 지정된 포트에 바인딩할 수 없는 경우에 발생합니다.

해결책: 모든 SOAP/HTTP 및 SOAP/HTTPS 프로토콜 리스너에 대해 지정된 포트를 확인하십시오. 둘 이상의 리스너에 동일한 포트가 지정되는 경우, 리스너 중 하나만이 시작할 수 있습니다. 또한, 해당 포트에서 실행 중인 다른 서비스가 있는지 확인하십시오. 다른 서비스가 있는 경우, 프로토콜 리스너에 대해 다른 포트를 선택할 수 있습니다.

문제점	가능한 솔루션 / 설명
KeyManagementError:UnrecoverableKeyException, 키를 복구할 수 없음	이 오류는 키스토어 또는 트러스트스토어를 사용할 수 없는데 발생합니다. 해결책: 새 키스토어를 작성하십시오.
SSL 데이터 교환 예외: 알 수 없는 CA	이것은 사용자의 트러스트스토어에 CA 인증이 없는 경우에 발생합니다. 해결책: 자체 서명 인증 뿐만 아니라 CA의 인증이 트러스트스토어에 상주하는지 확인하십시오. 또한, 인증의 DN에 호스트 이름(가능하면 IP 주소)이 있는지 확인하십시오.
로그 파일에서 과도한 JSSE 로깅을 인지합니다.	사용자 콘솔에 모든 기초 JSSE 세부사항이 표시되지 않기를 원하는 경우, Connector Configurator에 있는 SSL -> SSLDebug 등록 정보의 값을 false로 설정하십시오.
프로토콜 리스너를 지정했지만 리스너가 초기화되지 않고 있습니다. 커넥터에 다음 경고 메시지가 표시됩니다.	커넥터가 프로토콜 리스너의 프로토콜 등록 정보에 대한 유효한 값을 추출할 수 없었습니다. 유효한 값은 soap/http, soap/https 또는 soap/jms입니다.
Skipping Protocol Listener Property Set "SOME_LISTENER_NAME" with protocol property "": unable to determine the protocol listener class.}]	해결책: 이것은 오류 조건이 아닙니다. 그러나 커넥터가 이 리스너를 사용하기 원하는 경우, 유효한 프로토콜 등록 정보 값을 지정하십시오.
프로토콜 핸들러를 지정했지만 핸들러가 초기화되지 않습니다. 커넥터에 다음 경고 메시지가 표시됩니다.	커넥터가 핸들러의 프로토콜 등록 정보에 대한 유효한 값을 추출할 수 없습니다. 유효한 값은 soap/http, soap/https 또는 soap/jms입니다.
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: .]	해결책: 이것은 오류 조건이 아닙니다. 그러나 커넥터가 이 핸들러를 사용하기 원하는 경우, 유효한 프로토콜 등록 정보 값을 지정하십시오.
java.lang.NoClassDefFoundError: Javax/jms/JMSEException...	커넥터가 jms.jar을 찾을 수 없습니다. 해결책: jms.jar가 커넥터 클래스 경로에 있는지 확인하십시오.
Fail to lookup, queue: "InProgressQueue" for specified queue name: "<queue name>" queue using JNDI "<queue name>" javax.naming.NameNotFoundException: <queue name>	커넥터와 함께 SOAP/JMS 웹 서비스를 사용할 경우, 대기열을 작성하지 않을 때 이 문제점이 발생합니다. 또한 JNDI -> LookupQueuesUsingJNDI를 true로 설정했고 커넥터가 JNDI를 사용하는 대기열을 찾을 수 없는 경우에도 이 오류가 발생할 수 있습니다.
초기화 중 오류, JNDI 문맥이 초기화되지 않음, 사용자가 JMS 프로토콜을 사용할 수 없음	해결책: 커넥터에 필요한 대기열을 작성하십시오. JNDI -> LookupQueuesUsingJNDI가 true로 설정되는 경우, 커넥터에 필요한 대기열을 JNDI를 사용하여 찾을 수 있는지 확인하십시오. SOAP/JMS 프로토콜 리스너 또는 SOAP/JMS 프로토콜 핸들러를 사용하도록 커넥터를 구성한 경우, JNDI 등록 정보를 지정해야 합니다. 해결책: 필수 JNDI 커넥터 특정 등록 정보를 지정했는지 확인하십시오. JNDI 제공자에 연결하기 위해 필요한 라이브러리 및 jar 파일을 판별하려면 JNDI 제공자 문서를 참조하십시오. 모든 필수 jar 파일이 커넥터의 클래스 경로에 있는지 확인하십시오. 또한, 모든 필수 라이브러리가 커넥터의 경로에 있는지 확인하십시오.

문제점

초기 문맥 가져오기 오류

가능한 솔루션 / 설명

SOAP/JMS 프로토콜 리스너 또는 SOAP/JMS 프로토콜 핸들러를 사용하도록 커넥터를 구성한 경우, JNDI 등록 정보를 지정해야 합니다. 이 오류는 또한 JNDI 등록 정보를 올바르게 지정하지 않은 경우에도 발생할 수 있습니다.

해결책: JNDI 등록 정보를 확인하십시오. JNDI가 제대로 구성되는지 확인하십시오. JNDI 제공자에 연결하기 위해 필요한 라이브러리 및 jar 파일을 판별하려면 JNDI 제공자 문서를 참조하십시오. 모든 필수 jar 파일이 커넥터의 클래스 경로에 있는지 확인하십시오. 또한, 모든 필수 라이브러리가 커넥터의 경로에 있는지 확인하십시오.

런타임 오류

문제점

HTTP 응답 구문 분석 오류: HTTP 응답 머리글을 읽는 중에 스트림 끝에 도달했음

언급된 URL에 오류, 호스트 및 포트 세부사항을 추출할 수 없음, 목적지가 잘못된 <destination URL>임

브로커에 <Verb>가 있는 이벤트 Business Object <BO 이름> 전송 실패. 수신된 실행 상태 "-1" 및 오류 메시지:

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

잠재적 해결책/설명

이 오류는 커넥터가 SOAP/HTTP 웹 서비스를 호출할 때 발생합니다. 대상 웹 서비스가 잘못된 HTTP 응답을 전송했기 때문에 발생합니다.

해결책: 대상 SOAP/HTTP 웹 서비스 엔드포인트 주소가 유효한지 확인하십시오.

이 오류는 커넥터가 SOAP/HTTP 웹 서비스를 호출할 때 발생합니다. SOAP/HTTP 웹 서비스에 대해 잘못된 엔드포인트 주소를 지정했기 때문에 발생합니다.

해결책: 웹 서비스에 대해 유효한 엔드포인트 주소를 지정했는지 확인하십시오.

이 오류는 커넥터가 이벤트를 동기식으로 전송하려는 협업이 존재하지 않거나 Business Object Verb를 승인하지 않기 때문에 통합 브로커가 이벤트를 처리하지 못할 때 발생합니다.

해결책: 이벤트 공고를 위해 웹 서비스 TLO를 사용하려는 경우, TLO의 ws_collab 오브젝트 레벨 ASI를 시험하십시오(TLO의 이름이 오류 메시지에 제공됩니다). ws_collab ASI의 값을 확인하십시오. 이 협업이 존재하고 실행 중인지 확인하십시오. ws_mode BO 레벨 ASI가 synch로 설정되는 경우, ws_collab ASI가 필수입니다. ws_verb 오브젝트 레벨 ASI의 값을 확인하십시오. ws_verb ASI에 지정된 Verb가 ws_collab ASI가 지정하는 협업을 트리거할 수 있는지 확인하십시오. 이벤트 공고를 위해 비TLO를 사용할 경우,

WSCollaborations 커넥터 등록 정보를 적절히 검사하십시오. 이 Business Object가 동기식으로 호출할 협업을 찾으십시오. 이 협업이 존재하고 실행 중인지 확인하십시오.

문제점

SOAP 요청을 요청 Business Object로 변환하지 못했습니다.
Soap 결함:

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

잠재적 해결책/설명

이 오류는 이벤트 공고 중에 커넥터가 통합 브로커로 전송하려 시도 중인 Business Object의 Verb를 커넥터가 판별할 수 없을 때 발생합니다.

해결책: 이벤트 공고를 위해 웹 서비스 TLO를 사용 중인 경우, 이 TLO에 대해 ws_verb 오브젝트 레벨 ASI를 지정했는지 확인하십시오. 이 ASI의 값으로 Verb를 지정하십시오. 이벤트 공고를 위해 비TLO를 사용 중이면, 웹 서비스 클라이언트가 전송하는 SOAP 메시지가 Verb 요소를 포함해야 합니다. SOAP Data Handler는 SOAP 메시지의 Verb 요소 값을 사용하여 Business Object의 Verb를 설정합니다. 웹 서비스 클라이언트가 SOAP 메시지에서 Verb를 전송하지 않는 경우, SOAP Data Handler가 Business Object에 Verb를 설정할 수 없습니다. 이 경우에 커넥터는 Business Object를 통합 브로커로 전달할 수 없습니다. 웹 서비스 클라이언트가 SOAP 메시지에 Verb 요소를 포함하지 않을 수 있다고 의심되는 경우, 이 Business Object에 대해 DefaultVerb Verb 레벨 ASI를 제공할 수 있습니다. 그렇게 하면, 커넥터가 통합 브로커로 전송하기 전에 Business Object에 이 Verb를 설정합니다.

부록 A. 커넥터의 표준 구성 등록 정보

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

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

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

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

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

신규 및 삭제된 등록 정보

다음 표준 등록 정보가 이 릴리스에 추가되었습니다.

새 등록 정보

- XMLNameSpaceFormat

삭제된 등록 정보

- RestartCount
- RHF2MessageDomain

표준 커넥터 등록 정보 구성

어댑터 커넥터에는 두 가지 유형의 구성 등록 정보가 있습니다.

- 표준 구성 등록 정보
- 커넥터 특정 구성 등록 정보

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

Connector Configurator 사용

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

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

등록 정보 값 설정 및 갱신

등록 정보 필드의 기본 길이는 255자입니다.

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

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

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

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

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

표준 등록 정보 요약

표 49은 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 49. 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 49. 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 49. 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 49. 표준 구성 등록 정보 요약 (계속)

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

표준 구성 등록 정보

이 절에서는 표준 커넥터 구성 등록 정보를 나열하고 각각을 정의합니다.

AdminInQueue

관리 메시지를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMININQUEUE입니다.

AdminOutQueue

관리 메시지를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/ADMINOUTQUEUE입니다.

AgentConnections

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

AgentTraceLevel

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

ApplicationName

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

BrokerType

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

CharacterEncoding

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

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

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

ConcurrentEventTriggeredFlows

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

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

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

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

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

ContainerManagedEvents

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

기본값은 값 없음입니다.

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

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

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

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

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

ControllerStoreAndForwardMode

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

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

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

기본값은 true입니다.

ControllerTraceLevel

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

커넥터 제어기의 추적 메시지 레벨입니다. 기본값은 0입니다.

DeliveryQueue

DeliveryTransport가 JMS인 경우에만 적용 가능합니다.

Business Object를 통합 브로커에 전송하기 위해 커넥터에서 사용하는 대기열입니다.

기본값은 CONNECTORNAME/DELIVERYQUEUE입니다.

DeliveryTransport

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

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

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

WebSphere MQ 및 IDL

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

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

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

JMS

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

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

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

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

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

- CWSHaredEnv.sh 스크립트의 LDR_CNTRL 환경 변수.

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

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

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

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

DuplicateEventElimination

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

또한 이 등록 정보는 false로 설정될 수 있습니다.

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

FaultQueue

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

기본값은 CONNECTORNAME/FAULTQUEUE입니다.

JvmMaxHeapSize

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

기본값은 128m입니다.

JvmMaxNativeStackSize

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

기본값은 128k입니다.

JvmMinHeapSize

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

기본값은 1m입니다.

jms.FactoryClassName

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

기본값은 CxCommon.Messaging.jms.IBMMQSeriesFactory입니다.

jms.MessageBrokerName

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

기본값은 crossworlds.queue.manager입니다.

jms.NumConcurrentRequests

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

기본값은 10입니다.

jms.Password

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

기본값은 없습니다.

jms.UserName

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

기본값은 없습니다.

ListenerConcurrency

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

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

Locale

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

로케일 이름의 형식은 다음과 같습니다.

ll_TT.codeset

여기서

ll

2문자 언어 코드(보통 소문자)

TT

2문자 국가 또는 지역 코드(보통 대문자)

codeset

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

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

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

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

LogAtInterchangeEnd

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

MaxEventCapacity

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

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

MessageFileName

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

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

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

MonitorQueue

중복 이벤트를 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 192 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.

- 203 페이지의 『Connector Configurator 개요』
- 204 페이지의 『Connector Configurator 시작』
- 205 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 208 페이지의 『새 구성 파일 작성』
- 211 페이지의 『구성 파일 등록 정보 설정』
- 219 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(205 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, 206 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(210 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 206 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

다음 버튼을 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 - 키벡터 특정 등록 정보 템플릿 대화 상자가 나타납니다.

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

등록 정보가 종속되도록 지정하고 종속 조건을 설정하려면, 다음을 수행하십시오.

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

새 구성 파일 작성

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

브로커를 선택하려면 다음을 수행하십시오.

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

다음 작업도 수행할 수 있습니다.

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

커넥터 특정 템플릿에서 구성 파일 작성

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

1. 파일 > 새로 작성 > 커넥터 구성을 누르십시오.

2. 다음 필드가 있는 새 커넥터 대화 상자가 나타납니다.

- 이름

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

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

- 시스템 연결

ICS, WebSphere Message Brokers 또는 WAS를 누르십시오.

- 커넥터 특정 등록 정보 템플릿

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

사용하려는 템플릿을 선택하고 확인을 누르십시오.

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

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

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

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

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

기존 파일 사용

다음 중 하나 이상의 형식으로 기존 파일을 사용할 수 있습니다.

- 커넥터 정의 파일.

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

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

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

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

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

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

System Manager 프로젝트에서 커넥터 구성을 열려면 다음 단계를 따르십시오.

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

구성 파일 완료

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

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

1. 표준 등록 정보 탭에서 브로커 유형 등록 정보에 대한 값 필드를 선택하십시오. 드롭 다운 메뉴에서 ICS, WMQI 또는 WAS를 선택하십시오.
2. 표준 등록 정보 탭에 선택한 브로커와 연관된 등록 정보가 표시됩니다. 지금 파일을 저장하거나 214 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. 212 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

186 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 Web Services 자습서

- 『자습서 정보』
- 223 페이지의 『시작하기 전에』
- 223 페이지의 『설치 및 구성』
- 230 페이지의 『비동기 시나리오 실행』
- 233 페이지의 『동기 시나리오 실행』

이 부록에는 다음과 같은 단계별 프로시저가 있습니다.

- 요청 및 이벤트 처리 모두에 대한 비동기 및 동기 이벤트 전송을 설명합니다.
- SOAP/HTTPS 샘플의 웹 서비스 커넥터 구성 방법을 설명합니다.
- SOAP/HTTP 샘플의 웹 서비스 커넥터 구성 방법을 설명합니다.
- SOAP/JMS 샘플의 웹 서비스 커넥터 구성 방법을 설명합니다.

자습서 정보

이 자습서는 지원되는 각 프로토콜(SOAP/HTTP, SOAP/HTTPS 및 SOAP/JMS)을 사용하여 Adapter for Web Services의 요청 및 이벤트 처리 국면 모두의 비동기 및 동기 이벤트 전송을 설명하기 위한 것입니다. 각 시나리오에서, 어댑터는 다음과 같이 작동합니다.

- 웹 서비스를 호출하는 협업 웹 서비스 클라이언트
- 웹 서비스로 WebSphere ICS 협업을 나타내는 프록시

자습서는 샘플 시나리오에서 어댑터의 기본 기능을 표시하도록 설계되었습니다.

- 커넥터에서 비동기(요청 전용) 웹 서비스 및 관련 클라이언트를 설명하는 비동기 시나리오. 이 시나리오에는 두 가지 샘플이 있습니다. 구성의 간편함을 위해 동일한 웹 서비스 커넥터가 웹 서비스로써 협업을 나타내기 위해 사용되며 클라이언트로써 웹 서비스를 호출합니다.
 - 웹 서비스로 나타나는 협업 이 예에서, 웹 서비스는 단순히 커넥터에 의해 웹 서비스로 나타나는 WebSphere ICS 내의 협업 SERVICE_ASYNC_Order_Collab입니다. 웹 서비스는 Asynch Order Service로 나타냅니다. 커넥터가 제대로 구성되면, 이 웹 서비스는 웹 서비스 프로토콜(SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS)을 사용하여 호출될 수 있습니다. SERVICE_ASYNC_Order_Collab는 SERVICE_ASYNC_TLO_Order를 얻는 단순한 전달 협업입니다. 이 협업의 트리거링 포트(시작)는 웹 서비스 커넥터에 바인드됩니다. 서비스 포트(끝)는 SampleSiebelConnector로 바인드됩니다.

- 웹 서비스 클라이언트로 호출되는 협업 이 샘플에서, 웹 서비스 클라이언트는 웹 서비스 커넥터를 사용하여 웹 서비스 Asynch Order Service를 호출할 WebSphere ICS 내의 또 다른 협업 CLIENT_ASYNC_Order_Collab입니다. 커넥터가 제대로 구성되면, 이 웹 서비스 클라이언트는 웹 서비스 프로토콜(SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS)에서 웹 서비스를 호출할 수 있습니다. CLIENT_ASYNC_Order_Collab는 CLIENT_ASYNC_TLO_Order를 얻는 단순한 전달 협업입니다. 이 협업의 트리거링 포트(시작)는 SampleSAPConnector에 바인드됩니다. 서비스 포트(끝)는 웹 서비스 커넥터로 바인드됩니다.

비동기 시나리오에서 샘플 모두 두 개의 응용프로그램과 관련됩니다.

- SampleSiebel: 클라이언트의 순서를 작성합니다.
- SampleSAP: 순서를 작성합니다.
- 커넥터에서 동기(요청-응답) 웹 서비스 및 관련 클라이언트를 설명하는 동기 시나리오. 이 시나리오에는 두 가지 샘플이 있습니다. 구성의 간편함을 위해 동일한 웹 서비스 커넥터가 웹 서비스로써 협업을 나타내기 위해 사용되며 클라이언트로써 웹 서비스를 호출합니다.

- 웹 서비스로 나타나는 협업. 이 예에서, 웹 서비스는 단순히 커넥터에 의해 웹 서비스로 나타나는 WebSphere ICS 내의 협업 SERVICE_SYNC_OrderStatus_Collab입니다. 이 샘플에서, 이 웹 서비스는 Synch OrderStatus Service로 나타냅니다. 커넥터가 제대로 구성되면, 웹 서비스는 웹 서비스 프로토콜(SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS)을 사용하여 호출될 수 있습니다. SERVICE_SYNC_OrderStatus_Collab은 SERVICE_SYNC_TLO_OrderStatus를 얻는 단순한 전달 협업입니다. 이 협업의 트리거링 포트(시작)는 웹 서비스 커넥터에 바인드됩니다. 서비스 포트(끝)는 SampleSiebelConnector로 바인드됩니다.

- 웹 서비스 클라이언트로 호출되는 협업 이 샘플에서, 웹 서비스 클라이언트는 웹 서비스 커넥터를 사용하여 Synch OrderStatus Service를 호출할 WebSphere ICS 내의 또 다른 협업 CLIENT_SYNC_OrderStatus_Collab입니다. 커넥터가 제대로 구성되면, 이 웹 서비스 클라이언트는 웹 서비스 프로토콜(SOAP/HTTP, SOAP/HTTPS 또는 SOAP/JMS)에서 웹 서비스를 호출할 수 있습니다. CLIENT_SYNC_OrderStatus_Collab은 CLIENT_SYNC_TLO_OrderStatus를 얻는 단순한 전달 협업입니다. 이 협업의 트리거링 포트(시작)는 SampleSAPConnector에 바인드됩니다. 서비스 포트(끝)는 웹 서비스 커넥터로 바인드됩니다.

동기 시나리오에서 샘플 둘 모두 두 개의 응용프로그램과 관련됩니다.

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

시나리오 둘 모두 두 개의 Test Connector를 사용하여 SampleSiebelConnector 및 SampleSAPConnector를 시뮬레이트하는 것과 관련되어 있습니다.

시작하기 전에

학습을 시작하기 전에 다음을 확인하십시오.

- WebSphere ICS 4.2.x 이상을 설치했으며 이 제품에 대한 경험이 있습니다.
- WebSphere ICS 홈 디렉토리에 WebSphere Business Integration Adapter For Web Service를 설치했습니다.
- 웹 서비스 기술 경험이 있습니다.
- SOAP 기술 경험이 있습니다.

설치 및 구성

다음 섹션에서 *WBI_folder*는 현재 WebSphere ICS 설치를 포함하는 폴더를 나타냅니다. 모든 환경 변수 및 파일 분리자가 Windows NT/2000 형식으로 지정됩니다. AIX 또는 Solaris 플랫폼에서 실행 중인 경우 적절히 변경하십시오. (예를 들어, *WBI_folder\connectors*는 *WBI_folder/connectors*가 됩니다).

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 사용자 서버를 서버 인스턴스로 등록하고 연결하십시오.

샘플 내용 로드

Component Navigator Perspective에서

1. 새 통합 구성요소 라이브러리를 작성하십시오.
2. *WBI_folder\connectors\WebServices\samples\WebSphereICS*에 있는 *WebServicesSample.jar* 파일을 가져오십시오.

협업 템플릿 컴파일

WebSphere Business Integration System Manager 사용:

- *WebServicesSample.jar* repos 파일에서 가져온 협업 템플릿의 모두 컴파일.

커넥터 구성

1. 아직 구성하지 않은 경우, 이 안내서에서 설명한 대로 사용자 시스템에 따라 커넥터를 구성하십시오.

2. WebSphere Business Integration System Manager를 사용하여, Connector Configurator에서 WebServicesConnector를 여십시오.
3. 또한 샘플에서 사용하려는 프로토콜의 WebServicesConnector를 구성해야 합니다.
 - SOAP/HTTP를 사용하려는 경우, 『SOAP/HTTP 프로토콜 시나리오 구성』을 참조하여 SOAP/HTTP의 커넥터를 구성하십시오.
 - SOAP/HTTPS를 사용하려는 경우, 225 페이지의 『SOAP/HTTPS 프로토콜 시나리오 구성』을 참조하여 SOAP/HTTPS의 커넥터를 구성하십시오.
 - SOAP/JMS를 사용하려는 경우, 227 페이지의 『SOAP/JMS 프로토콜 시나리오 구성』을 참조하여 SOAP/JMS의 커넥터를 구성하십시오.

SOAP/HTTP 프로토콜 시나리오 구성

이 섹션에서는 SOAP/HTTP 샘플 시나리오의 커넥터를 구성하는 방법을 표시합니다. 이 문서의 본문에서 설명한 대로, 커넥터는 SOAP/HTTP 프로토콜 리스너 및 SOAP/HTTP-HTTPS 프로토콜 핸들러를 포함합니다. 샘플 시나리오는 SERVICE_ASYNC_Order_Collab 및 SERVICE_SYNC_OrderStatus_Collab 협업을 SOAP/HTTP 웹 서비스로 나타냅니다. 협업을 SOAP/HTTP 웹 서비스로 나타내기 위해, 커넥터는 SOAP/HTTP 프로토콜 리스너를 사용합니다. 샘플 시나리오는 SOAP/HTTP 웹 서비스의 SOAP/HTTP 클라이언트인 CLIENT_ASYNC_Order_Collab 및 CLIENT_SYNC_OrderStatus_Collab 협업에서 제공됩니다. SOAP/HTTP 웹 서비스를 호출하기 위해, 커넥터는 SOAP/HTTPHTTPS Protocol Handler를 사용합니다.

다음 단계와 설명에서, 계층 구조 커넥터 구성 등록 정보가 ” 기호로 표시됩니다. 예를 들어, A” B는 A가 계층 구조 등록 정보이며 B는 A의 하위 등록 정보임을 나타냅니다.

이 샘플의 SOAP/HTTP 프로토콜 리스너를 구성하려면 다음과 같이 하십시오.

1. Connector Configurator에서, WebServicesConnector의 커넥터 특정 등록 정보를 누르십시오.
2. ProtocolListeners 하위 등록 정보를 표시하려면 **ProtocolListenerFramework** 등록 정보를 펼치십시오.
3. **ProtocolListeners** 하위 등록 정보를 펼쳐 **SOAPHTTPListener1** 하위 등록 정보를 표시하십시오.
4. **SOAPHTTPListener1”Host** 및 **SOAPHTTPListener1”Port** 등록 정보의 값을 확인하십시오.호스트에서 실행 중이며 이 TCP/IP 포트에서 청취 중인 다른 프로세스가 없는지 확인하십시오. 선택적으로, 커넥터를 실행할 시스템 이름으로 **SOAHTTPListener1”Host**의 값을 설정하려고 할 수 있습니다.

샘플의 SOAP/HTTP-HTTPS 프로토콜 핸들러를 구성할 필요가 없습니다.

SOAP/HTTPS 프로토콜 시나리오 구성

이 섹션에서는 SOAP/HTTPS 샘플 시나리오의 커넥터를 구성하는 방법을 표시합니다. 커넥터는 SOAP/HTTPS 프로토콜 리스너 및 SOAP/HTTP-HTTPS 프로토콜 핸들러를 포함합니다. 샘플 시나리오는 SERVICE_ASYNC_Order_Collab 및 SERVICE_SYNC_OrderStatus_Collab 협업을 SOAP/HTTPS 웹 서비스로 나타냅니다. 협업을 SOAP/HTTPS 웹 서비스로 나타내기 위해, 커넥터는 SOAP/HTTPS 프로토콜 리스너를 사용합니다. 샘플 시나리오는 SOAP/HTTPS 웹 서비스의 SOAP/HTTPS 클라이언트인 CLIENT_ASYNC_Order_Collab 및 CLIENT_SYNC_OrderStatus_Collab 협업에서 제공됩니다. SOAP/HTTPS 웹 서비스를 호출하기 위해, 커넥터는 SOAP/HTTPHTTPS 프로토콜 핸들러를 사용합니다.

다음 단계와 설명에서, 계층 구조 커넥터 구성 등록 정보가 ” 기호로 표시됩니다. 예를 들어, A” B는 A가 계층 구조 등록 정보이며 B는 A의 하위 등록 정보임을 나타냅니다.

주: 223 페이지의 『시작하기 전에』에서 위에 나열한 사전 설치 항목 이외에, 키 및 인증 관리 소프트웨어를 사용하여 사용자 키스토어 및 트러스트스토어를 작성하고 테스트해야 합니다.

SSL 커넥터 특정 등록 정보 구성: SOAP/HTTPS의 경우, 커넥터는 SSL 커넥터 특정 계층 구조 등록 정보를 구성하도록 요구합니다.

1. Connector Configurator에서, WebServicesConnector의 커넥터 특정 등록 정보 탭을 누르십시오.
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
 - **SSL”KeyStorePassword** KeyStore에 액세스하는
 - **SSL”KeyStoreAlias** KeyStore의 개인용
 - **SSL”TrustStore** 키
 - **SSL”TrustStorePassword** TrustStore에 액세스하는

주: Connector Configurator에서의 변경사항을 잊지 마십시오.

SOAP/HTTPS 프로토콜 리스너 구성:

1. Connector Configurator에서, WebServicesConnector의 커넥터 특정 등록 정보를 누르십시오.
2. **ProtocolListenerFramework** 등록 정보를 펼쳐 **ProtocolListeners** 하위 등록 정보를 표시하십시오.

3. **ProtocolListeners** 하위 등록 정보를 펼쳐 **SOAPHTTPSListener1** 하위 등록 정보를 표시하십시오. **SOAPHTTPSListener1”Host** 및 **SOAPHTTPSListener1”Port** 등록 정보의 값을 확인하십시오. 호스트에서 실행 중이며 이 TCP/IP 포트에서 청취 중인 다른 프로세스가 없는지 확인하십시오. 선택적으로, 커넥터를 실행할 시스템 이름으로 **SOAHTTPSListener1”Host**의 값을 설정하려고 할 수 있습니다.

샘플의 SOAP/HTTP-HTTPS 프로토콜 핸들러를 구성할 필요가 없습니다.

KeyStore 및 TrustStore 설정: 빠르게 KeyStore 및 TrustStore를 설정하여 샘플 시나리오에서 사용할 수 있습니다. 프로덕션 시스템의 경우, 제3자 소프트웨어를 사용하여 인증 및 키 생성은 물론 키스토어를 설정하고 관리해야 합니다. 이 자원을 설정하고 관리하기 위해 Adapter for Web Services의 파트로 제공된 도구가 없습니다.

이 섹션에서는 Java Virtual Machine이 사용자 시스템에 설치되어 있으며 사용자가 JVM(Java Virtual Machine)에 제공된 keytool과 익숙하다는 것을 가정하고 있습니다. 자세한 정보나 keytool에서의 문제점 해결 정보는 JVM과 함께 제공되는 문서를 참조하십시오.

KeyStore를 설정하려면:

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

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

2. keytool이 즉시 암호를 입력하도록 프롬프트합니다. SSL”KeyStorePassword 커넥터 등록 정보의 값에 입력한 암호를 지정하십시오.

참고로 위의 예제에서 명령행에 -keystore c:\security\keystore를 지정한 경우에는 c:\security\keystore를 SSL”KeyStore 등록 정보의 값으로서 입력합니다. 또한 명령행에 -alias wsadapter를 지정한 경우에는 wsadapter를 SSL”KeyStoreAlias 커넥터 등록 정보의 값으로서 입력합니다. 그러면 인증의 세부 사항을 입력하도록 프롬프트합니다. 다음은 각 프롬프트에 입력할 수 있는 내용을 보여주지만, 단지 예입니다. 항상 keytool 문서를 참조하십시오.

```
성과 이름이 무엇입니까?
[알 수 없음]: HostName
조직 단위의 이름이 무엇입니까?
[알 수 없음]: myunit
조직의 이름이 무엇입니까?
[알 수 없음]: myorganization
구/군/시의 이름이 무엇입니까?
[알 수 없음]: mycity
시/도의 이름이 무엇입니까?
[알 수 없음]: mystate
이 단위의 두 자리 국가 코드가 무엇입니까?
[알 수 없음]: mycountryIs <CN=HostName, OU=myunit, O=myorganization,
L=mycity, ST=mystate, C=mycountry> 맞습니까?
[아니오]: 예
```

- 성과 이름은 무엇입니까?의 경우, 커넥터를 실행 중인 시스템의 이름을 입력해야 합니다. 그런 다음, keytool은 다음과 같이 프롬프트합니다.

<wsadapter>에 대한 키 암호를 입력하십시오(키스토어 암호와 같으면 RETURN을 누르십시오).

- 리턴을 눌러 동일한 암호를 사용하십시오. 자체 서명된 인증을 사용하려면, 위에서 작성한 인증을 내보낼 수 있습니다. 그렇게 수행하려면, 명령행에서 다음을 입력하십시오.

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

- 이제 keytool이 키스토어 암호를 입력하도록 프롬프트합니다. 위에서 입력한 암호를 입력하십시오.

TrustStore를 설정하려면:

- 신뢰된 인증을 TrustStore로 가져오려면, 다음 명령을 입력하십시오.

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

- 이제 keytool이 키스토어 암호를 입력하도록 프롬프트합니다. -keystore c:\security\truststore를 입력하면, SSL TrustStore 등록 정보를 c:\security\truststore로 설정했는지 확인하십시오. 또한 SSLTrustStorePassword 등록 정보의 값을 위에서 입력한 암호로 설정하십시오.

SOAP/JMS 프로토콜 시나리오 구성

이 섹션에서는 SOAP/JMS 샘플 시나리오의 커넥터를 구성하는 방법을 표시합니다. 샘플 시나리오는 SERVICE_ASYNC_Order_Collab 및 SERVICE_SYNC_OrderStatus_Collab 협업을 SOAP/JMS 웹 서비스로 나타냅니다. 협업을 SOAP/JMS 웹 서비스로 나타내기 위해, 커넥터는 SOAP/JMS 프로토콜 리스너를 사용합니다. 샘플 시나리오는 SOAP/JMS 웹 서비스의 SOAP/JMS 클라이언트인 CLIENT_ASYNC_Order_Collab 및 CLIENT_SYNC_OrderStatus_Collab 협업에서 제공됩니다. SOAP/JMS 웹 서비스를 호출하기 위해, 커넥터는 SOAP/JMS 프로토콜 핸들러를 사용합니다.

다음 단계와 설명에서, 계층 구조 커넥터 구성 등록 정보가 ” 기호로 표시됩니다. 예를 들어, A” B는 A가 계층 구조 등록 정보이며 B는 A의 하위 등록 정보임을 나타냅니다.

주: 223 페이지의 『시작하기 전에』에서 위에 나열한 사전 설치 항목 이외에도, JMS 서비스 제공자를 설치하고 JNDI를 설치 및 구성해야 합니다.

JNDI 등록 정보 구성: SOAP/JMS의 경우, JNDI 커넥터 구성 등록 정보를 구성해야 합니다.

- Connector Configurator에서, WebServicesConnector의 커넥터 특정 등록 정보를 누르십시오.

2. JNDI 계층 구조 등록 정보를 펼쳐 하위 등록 정보를 표시하십시오. 그런 다음 하위 등록 정보를 확인하거나 변경하여 아래 나열한 값을 일치시키십시오.

- **JNDI”JNDIProviderURL** 이 등록 정보를 JNDI 서비스 제공자의 URL로 설정하십시오. JNDI 제공자 문서를 참조하십시오.
- **JNDI”InitialContextFactory** 이 등록 정보를 JNDI 초기 문맥을 작성할 팩토리 클래스의 완전한 클래스 이름으로 설정하십시오. JNDI 제공자 문서를 참조하십시오.
- **JNDI”JNDIConnectionFactoryName** JNDI 문맥을 사용하여 찾아볼 연결 팩토리의 JNDI 이름으로 이 등록 정보를 설정하십시오. JNDI를 사용하여 이 이름을 찾을 수 있는지 확인하십시오.
- 다음 등록 정보 중 하나가 JNDI 제공자에서 필요한지 여부를 알아보려면 JNDI 문서를 참조하십시오.
 - **JNDI”CTX_ObjectFactories**
 - **JNDI”CTX_ObjectFactories**
 - **JNDI”CTX_StateFactories**
 - **JNDI”CTX_URLPackagePrefixes**
 - **JNDI”CTX_DNS_URL**
 - **JNDI”CTX_Authoritative**
 - **JNDI”CTX_Batchsize**
 - **JNDI”CTX_Referral**
 - **JNDI”CTX_SecurityProtocol**
 - **JNDI”CTX_SecurityAuthentication**
 - **JNDI”CTX_SecurityPrincipal**
 - **JNDI”CTX_SecurityCredentials**
 - **JNDI”CTX_Language**

3. Connector Configurator에 변경사항을 저장하십시오.

JMS 대기열 및 SOAP/JMS 프로토콜 리스너 구성: 시나리오에서는 6개의 대기열을 JMS 서비스 제공자로 정의하도록 요구합니다. 그렇게 수행하기 전에, JMS 제공자 문서를 검사하십시오. 대기열 정의는 제공자 간에 다양합니다.

1. 다음 대기열 정의(또는 JNDI 찾아보기를 통해 사용 가능하게 함)

- ORDER_INPUT
- ORDER_INPROGRESS
- ORDER_ERROR
- ORDER_ARCHIVE
- ORDER_UNSUBSCRIBED

- ORDER_REPLYTO
2. CSM에서 Connector Configurator의 **WebServicesConnector**를 여십시오. 아직 구성하지 않았으면 사용자 시스템의 설치 안내서에서 설명한 대로 커넥터를 구성하십시오.
 3. Connector Configurator에서 응용프로그램 구성 등록 정보를 누르십시오.
 4. ProtocolListeners 하위 등록 정보를 표시하려면 **ProtocolListenerFramework** 등록 정보를 펼치십시오.
 5. SOAPJMSListener1 하위 등록 정보를 표시하려면 **ProtocolListeners** 등록 정보를 펼치십시오.
 6. 아래 나열한 것과 일치하도록 SOAPJMSListner1 하위 등록 정보의 값을 확인 또는 변경하십시오.
 - SOAPJMSListener1”Protocol Set를 soap/jms로
 - SOAPJMSListener1”Protocol Set를 soap/jms로
 - SOAPJMSListener1”InputQueue Set를 ORDER_INPUT로
 - SOAPJMSListener1”InProgressQueue Set를 ORDER_INPROGRESS로
 - SOAPJMSListener1”ArchiveQueue Set를 ORDER_ARCHIVE로
 - SOAPJMSListener1”UnsubscribedQueue Set를 ORDER_UNSUBSCRIBED로
 - SOAPJMSListener1”ErrorQueue Set를 ORDER_ERROR로
 - SOAPJMSListener1”ReplyToQueue Set를 ORDER_REPLYTO로
 7. Connector Configurator에 변경사항을 저장하십시오.

SOAP/JMS 프로토콜 핸들러 구성:

1. System Manager에서 Connector Configurator의 **WebServicesConnector**를 여십시오. 아직 그렇게 수행하지 않은 경우, 사용자 시스템의 설치 안내서에서 설명한 대로 커넥터를 구성하십시오.
2. Connector Configurator에서 커넥터 구성 등록 정보를 누르십시오.
3. ProtocolHandlers 하위 등록 정보를 표시하려면 **ProtocolHandlerFramework** 등록 정보를 펼치십시오.
4. SOAPJMShandler 하위 등록 정보를 표시하려면 **ProtocolHandlers** 하위 등록 정보를 펼치십시오. 아래와 일치하도록 SOAPJMShandler 하위 등록 정보의 값을 확인 또는 변경하십시오.
 - SOAPJMShandler”Protocol Set를 soap/jms로
 - SOAPJMShandler”ReplyToQueue Set를 값 ORDER_REPLYTO로
5. Connector Configurator에 변경사항을 저장하십시오.

사용자 프로젝트 작성

- WebSphere Business Integration System Manager를 사용하여, 새 사용자 프로젝트를 작성하십시오. 223 페이지의 『샘플 내용 로드』에 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 유효함을 보장하려면 ICS를 다시 시동하십시오.
2. 모든 협업 오브젝트, 커넥터 제어기 및 맵이 녹색 상태인지 확인하려면 System Monitor 도구를 사용하십시오.

비동기 시나리오 실행

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

1. CLIENT_ASYNC_TLO_Order.Create 이벤트는 Test Connector의 한 인스턴스에서 실행 중인 응용프로그램 SampleSAP에서 시작합니다.
2. 이벤트는 SampleSAP에서 협업 CLIENT_ASYNC_Order_Collab으로 전송됩니다.
3. 그런 다음 이벤트는 협업에서 웹 서비스 커넥터로 전송됩니다.
4. 웹 서비스 커넥터는 CLIENT_ASYNC_TLO_Order의 하위인 CLIENT_ASYNC_Order 오브젝트를 찾습니다.
5. 요청 Business Object는 SOAP Data Handler를 사용하여 SOAP 메시지로 변환됩니다.
6. 웹 서비스 커넥터는 웹 서비스 Asynch Order Service의 끝점(목적지)으로 SOAP 메시지를 전송합니다. 끝점은 프로토콜 구성 MO(Meta Object)의 목적지 속성으로 제공됩니다. 커넥터에서 사용하는 프로토콜 구성 MO는 CLIENT_ASYNC_TLO_Order의 핸들러 속성 값에 의해 좌우됩니다. soap/http 또는 soap/https로 설정된 경우, CLIENT_ASYNC_Order_SOAP_HTTP_CfgMO의 목적지 속성은 웹 서비스의 URL로 끝점을 제공합니다. 그렇지 않고 핸들러 속성이 soap/jms로 설정되면, CLIENT_ASYNC_Order_SOAP_JMS_CfgMO의 목적지 속성은 목적지 대기열 이름으로 끝점을 제공합니다.
7. Asynch Order Service 웹 서비스는 SOAP 요청을 수신합니다. 이전에 언급한 대로, 웹 서비스 커넥터는 이 웹 서비스의 끝점입니다. 끝점(요청이 전송된 지점)에서 청취 중인 커넥터의 프로토콜 리스너는 SOAP 메시지를 수신합니다.

8. 커넥터는 SOAP 메시지를 SERVICE_ASYNC_Order로 변환한 후 SERVICE_TLO_Order 오브젝트를 작성합니다. SERVICE_ASYNC_Order 오브젝트는 SERVICE_TLO_Order 오브젝트의 하위로 설정됩니다.
9. 웹 서비스 커넥터는 이제 비동기적으로 SERVICE_TLO_Order 오브젝트를 ICS로 게시합니다. 이것은 비동기 웹 서비스 호출을 완료합니다.

이것은 비동기 웹 서비스(요청 전용)이므로, 웹 서비스 클라이언트로 다시 응답이 전송되지 않습니다. SERVICE_ASYNC_Order_Collab이 이 오브젝트를 수신하면, 협업은 Test Connector의 두 번째 인스턴스로서 실행되는 SampleSiebel이라는 응용프로그램으로 Business Object를 전송합니다. 오브젝트가 Test Connector에 표시됩니다. 응답 성공이 SampleSiebel 응용프로그램에서 선택되면, 이벤트는 다시 SERVICE_ASYNC_Order_Collab으로 전송됩니다.

비동기 시나리오를 실행하려면 다음을 수행하십시오.

1. ICS 통합 브로커를 아직 실행 중이지 않은 경우 시작하십시오.
2. 웹 서비스 커넥터를 시작하십시오.
3. Test Connector의 두 개의 인스턴스를 시작하십시오.
4. Test Connector를 사용하여, SampleSAPConnector 및 SampleSiebelConnector의 프로파일을 정의하십시오.
5. 각 Test Connector 메뉴에서 **FILE”CONNECT AGENT**를 선택하여 에이전트 시뮬레이트를 시작하십시오.
6. Test Connector를 사용하여 SampleSAPConnector를 시뮬레이트하는 동안, 메뉴에서 **EDIT”LOAD BO**를 선택하십시오. 다음 파일을 로드하십시오.

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

Test Connector는 CLIENT_ASYNC_TLO_Order가 로드되었음을 표시해야 합니다.

7. 웹 서비스 끝점 주소를 확인하십시오.
 - **SOAP/HTTP** 웹 서비스 SOAP/HTTP를 사용하려는 경우
 - a. SOAP/HTTP의 웹 서비스 커넥터를 구성했는지 확인하십시오. Test Connector에서, CLIENT_ASYNC_TLO_Order Business Object의 핸들러 속성 값이 soap/http로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
 - b. CLIENT_ASYNC_TLO_Order의 요청 속성을 펼치십시오. 이 속성은 CLIENT_ASYNC_Order Business Object 유형입니다.
 - c. CLIENT_ASYNC_Order의 SOAPHTTPCfgMO 속성을 펼치십시오. 이 속성은 CLIENT_ASYNC_Order_SOAP_HTTP_CfgMO 유형입니다.

- d. CLIENT_ASYNC_Order_SOAP_HTTP_CfgMO의 목적지 속성 값이 `http://localhost:8080/wbia/webservices/samples`로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
- **SOAP/HTTPS** 웹 서비스 SOAP/HTTPS를 사용하려는 경우
 - a. SOAP/HTTPS의 웹 서비스 커넥터를 구성했는지 확인하십시오. Test Connector에서, CLIENT_ASYNC_TLO_Order Business Object의 핸들러 속성 값이 `soap/https`로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
 - b. CLIENT_ASYNC_TLO_Order의 요청 속성을 펼치십시오. 이 속성은 CLIENT_ASYNC_Order Business Object 유형입니다.
 - c. CLIENT_ASYNC_Order의 SOAPHTTPCfgMO 속성을 펼치십시오. 이 속성은 CLIENT_ASYNC_Order_SOAP_HTTP_CfgMO 유형입니다.
 - d. CLIENT_ASYNC_Order_SOAP_HTTP_CfgMO의 목적지 속성 값이 `https://localhost:8443/wbia/webservices/samples`로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
- **SOAP/JMS** 웹 서비스 SOAP/JMS를 사용하려는 경우
 - a. SOAP/JMS의 웹 서비스 커넥터를 구성했는지 확인하십시오. Test Connector에서, CLIENT_ASYNC_TLO_Order Business Object의 핸들러 속성 값이 `soap/jms`로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
 - b. CLIENT_ASYNC_TLO_Order의 요청 속성을 펼치십시오. 이 속성은 CLIENT_ASYNC_Order Business Object 유형입니다.
 - c. CLIENT_ASYNC_Order의 SOAPJMScfgMO 속성을 펼치십시오. 이 속성은 CLIENT_ASYNC_Order_SOAP_JMS_CfgMO 유형입니다.
 - d. CLIENT_ASYNC_Order_SOAP_JMS_CfgMO의 목적지 속성 값이 `ORDER_INPUT`으로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
- 8. Test Connector로 SampleSAPConnector를 시뮬레이트하는 동안, 로드된 **Test BO**를 누르십시오. 메뉴에서 **REQUEST”SEND**를 선택하십시오. 이벤트 플로우에 대한 세부사항은 이 섹션에서 이전의 단계별 개요를 참조하십시오.
- 9. Test Connector로 SampleSiebelConnector를 시뮬레이트하는 동안, **REQUEST”ACCEPT REQUEST**를 선택하십시오. SERVICE_ASYNC_TLO_Order.Create로 레이블된 이벤트가 Test Connector의 오른쪽 패널에 표시됩니다.
- 10. Business Object를 두 번 누르십시오. Business Object가 창에서 위로 열립니다.

11. Business Object의 요청 속성을 펼치십시오. 요청 속성은 SERVICE_ASYNC_Order 유형입니다. OrderId, Customarily 및 기타 SERVICE_ASYNC_Order 속성을 검사하여 Order를 수신했는지 확인하십시오. 이것은 비동기 시나리오의 실행을 완료합니다.
12. Business Object를 검사하고 나면, 창을 닫으십시오.
REQUEST”REPLY”SUCCESS를 선택하십시오.

동기 시나리오 실행

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

1. CLIENT_SYNCH_TLO_OrderStatus.Retrieve 이벤트는 Test Connector의 한 인스턴스에서 실행 중인 응용프로그램 SampleSAP에서 시작합니다.
2. 이벤트는 SampleSAP에서 CLIENT_SYNCH_OrderStatus_Collab 협업으로 전송됩니다.
3. 그런 다음 이벤트는 협업에서 웹 서비스 커넥터로 전송됩니다.
4. 웹 서비스 커넥터는 CLIENT_SYNCH_TLO_OrderStatus의 하위인 CLIENT_SYNCH_OrderStatus_Request 오브젝트를 찾습니다.
5. 웹 서비스 커넥터는 SOAP Data Handler를 호출하여 CLIENT_SYNCH_OrderStatus_Request Business Object를 SOAP 메시지로 변환합니다.
6. 웹 서비스 커넥터는 웹 서비스 Synch OrderStatus의 끝점(목적지)으로 SOAP 메시지를 전송합니다. 끝점은 프로토콜 구성 MO의 목적지 속성으로 제공됩니다. 커넥터에서 사용하는 프로토콜 구성 MO는 CLIENT_SYNCH_TLO_OrderStatus의 핸들러 속성 값에 의해 좌우됩니다. soap/http 또는 soap/https로 설정된 경우, CLIENT_SYNCH_OrderStatus_Request_SOAP_HTTP_CfgMO의 목적지 속성은 웹 서비스의 URL로 끝점을 제공합니다. 그렇지 않고 핸들러 속성이 soap/jms로 설정되면, CLIENT_SYNCH_OrderStatus_Request_SOAP_JMS_CfgMO의 목적지 속성은 웹 서비스의 대기열 이름으로 끝점을 제공합니다.
7. 웹 서비스 Synch OrderStatus Service는 SOAP 요청을 수신합니다. 이전에 언급한 대로, 웹 서비스 커넥터는 대상 끝점입니다. 끝점(요청이 전송된 지점)에서 청구 중인 커넥터의 프로토콜 리스너는 SOAP 메시지를 수신합니다.
8. 커넥터는 SOAP 메시지로 SOAP Data Handler를 호출합니다. SOAP 메시지는 SOAP Data Handler에 의해 SERVICE_SYNCH_OrderStatus_Request 오브젝트로 변환됩니다. 그런 다음 웹 서비스는 SERVICE_TLO_OrderStatus 오브젝트를 작성합니다. SERVICE_SYNCH_OrderStatus_Request 오브젝트는 SERVICE_TLO_OrderStatus 오브젝트의 하위로 설정됩니다.

9. 웹 서비스 커넥터는 이제 WebSphere ICS에서 실행하는 SERVICE_SYNCH_OrderStatus_Collab 협업으로 SERVICE_TLO_OrderStatus 오브젝트를 동기적으로 게시합니다. 이것은 동기 실행이므로, 웹 서비스 커넥터는 협업이 실행하고 응답을 리턴할 때까지 웹 서비스 커넥터는 차단됩니다.
10. SERVICE_SYNCH_OrderStatus_Collab은 SERVICE_TLO_OrderStatus 오브젝트를 수신합니다. 그런 다음 협업은 Test Connector의 두 번째 인스턴스로 실행 중인 응용프로그램 SampleSiebel로 Business Object를 전송합니다.
11. SampleSiebel 응용프로그램에서 응답 성공을 선택하면, 이벤트는 다시 SERVICE_SYNCH_OrderStatus_Collab 협업으로 전송됩니다.
12. SERVICE_SYNCH_OrderStatus_Collab은 SERVICE_TLO_OrderStatus 오브젝트를 수신합니다. 그런 다음 협업은 Business Object를 웹 서비스 커넥터로 전송합니다.
13. 웹 서비스 커넥터가 SERVICE_SYNCH_OrderStatus_TLO의 하위인 SERVICE_SYNCH_OrderStatus_Response Business Object(또는 Business Object가 채워진 경우 SERVICE_SYNCH_OrderStatus_Fault)를 찾습니다. 이 Business Object는 SOAP Data Handler에 의해 SOAP 응답 메시지(또는 SOAP 결함 메시지)로 변환됩니다.
14. 웹 서비스 커넥터는 SOAP 응답 메시지(또는 SOAP 결함 메시지)를 웹 서비스 클라이언트로 리턴합니다.
15. 이 경우 커넥터인 웹 서비스 클라이언트는 응답을 수신합니다. 커넥터는 응답 메시지로 SOAP Data Handler를 호출합니다.
16. SOAP Data Handler는 Order Synch Service에서 리턴된 내용에 따라 응답 메시지를 CLIENT_SYNCH_OrderStatus_Response 또는 CLIENT_SYNCH_OrderStatus_Fault Business Object로 변환합니다. 웹 서비스 커넥터는 이 오브젝트를 CLIENT_SYNCH_OrderStatus_TLO의 하위로 설정합니다. CLIENT_SYNCH_OrderStatus_TLO는 CLIENT_SYNCH_OrderStatus_Collab 협업으로 리턴됩니다.
17. 그런 다음 CLIENT_SYNCH_OrderStatus_Collab은 Test Connector의 첫 번째 인스턴스로 실행 중인 SampleSAP 응용프로그램으로 CLIENT_SYNCH_OrderStatus_TLO를 전송합니다. Test Connector는 이 오브젝트를 표시합니다.

동기 시나리오를 실행하려면 다음을 수행하십시오.

1. ICS 통합 브로커를 아직 실행 중이지 않은 경우 시작하십시오.
2. 웹 서비스 커넥터를 시작하십시오.
3. Test Connector의 두 개의 인스턴스를 시작하십시오.
4. Test Connector를 사용하여, SampleSAPConnector 및 SampleSiebelConnector의 프로파일을 정의하십시오.

5. 각 Test Connector 메뉴에서 **FILE”CONNECT AGENT**를 선택하여 에이전트 시물레이트를 시작하십시오.
6. Test Connector를 사용하여 SampleSAPConnector를 시물레이트하는 동안, 메뉴에서 **EDIT”LOAD BO**를 선택하십시오. 다음 파일을 로드하십시오.

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

Test Connector는 CLIENT_SYNCH_TLO_OrderStatus가 로드되었음을 표시합니다.

7. 웹 서비스 끝점 주소를 확인하십시오.
 - **SOAP/HTTP** 웹 서비스 SOAP/HTTP를 사용하려는 경우
 - a. SOAP/HTTP의 웹 서비스 커넥터를 구성했는지 확인하십시오. Test Connector에서, CLIENT_SYNCH_TLO_OrderStatus Business Object의 핸들러 속성 값이 soap/http로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
 - b. CLIENT_SYNCH_TLO_OrderStatus의 요청 속성을 펼치십시오. 이 속성은 CLIENT_SYNCH_OrderStatus Business Object 유형입니다.
 - c. CLIENT_SYNCH_OrderStatus의 SOAPHTTPCfgMO 속성을 펼치십시오. 이 속성은 CLIENT_SYNCH_OrderStatus_SOAP_HTTP_CfgMO 유형입니다.
 - d. CLIENT_SYNCH_OrderStatus_SOAP_HTTP_CfgMO의 목적지 속성 값이 http://localhost:8080/wbia/webservices/samples로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
 - **SOAP/HTTPS** 웹 서비스 SOAP/HTTPS를 사용하려는 경우
 - a. SOAP/HTTPS의 웹 서비스 커넥터를 구성했는지 확인하십시오. Test Connector에서, CLIENT_SYNCH_TLO_OrderStatus Business Object의 핸들러 속성 값이 soap/https로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
 - b. CLIENT_SYNCH_TLO_OrderStatus의 요청 속성을 펼치십시오. 이 속성은 CLIENT_SYNCH_OrderStatus Business Object 유형입니다.
 - c. CLIENT_SYNCH_OrderStatus의 SOAPHTTPCfgMO 속성을 펼치십시오. 이 속성은 CLIENT_SYNCH_OrderStatus_SOAP_HTTP_CfgMO 유형입니다.
 - d. CLIENT_SYNCH_OrderStatus_SOAP_HTTP_CfgMO의 목적지 속성 값이 https://localhost:8443/wbia/webservices/samples로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
 - **SOAP/JMS** 웹 서비스 SOAP/JMS를 사용하려는 경우
 - a. SOAP/JMS의 웹 서비스 커넥터를 구성했는지 확인하십시오. Test Connector에서, CLIENT_SYNCH_TLO_OrderStatus Business Object의 핸들러 속성 값이 soap/jms로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.

- b. CLIENT_SYNCH_TLO_OrderStatus의 요청 속성을 펼치십시오. 이 속성은 CLIENT_SYNCH_OrderStatus Business Object 유형입니다.
 - c. CLIENT_SYNCH_OrderStatus의 SOAPJMSCfgMO 속성을 펼치십시오. 이 속성은 CLIENT_SYNCH_OrderStatus_SOAP_JMS_CfgMO 유형입니다.
 - d. CLIENT_SYNCH_OrderStatus_SOAP_JMS_CfgMO의 목적지 속성 값이 ORDER_INPUT으로 설정되었는지 확인하십시오. 이 값에서는 따옴표가 허용되지 않습니다.
8. Test Connector로 SampleSAPConnector를 시뮬레이트하는 동안, 로드된 **Test BO**를 누르십시오. 메뉴에서 **REQUEST”SEND**를 선택하십시오. 데이터 플로우에 대한 세부사항은 이 섹션에서 이전의 단계별 개요를 참조하십시오.
9. SERVICE_SYNCH_TLO_OrderStatus.Retrieve로 레이블된 이벤트가 SampleSiebelConnector를 시뮬레이트하는 Test Connector 인스턴스의 오른쪽 패널에 표시됩니다. 창에 표시할 Business Object를 두 번 누르십시오.
10. Business Object의 요청 속성을 펼치십시오. 요청 속성은 SERVICE_SYNCH_OrderStatus_Request 유형입니다. 이것이 상태가 필수인 주문인지 확인하려면 SERVICE_ASYNC_Order의 OrderId 속성을 검사하십시오.
- 주문 상태를 아는 경우 다음을 수행하십시오.
 - a. SERVICE_SYNCH_TLO_OrderStatus의 응답 속성을 누르십시오. 응답 속성은 CLIENT_SYNCH_OrderStatus_Response 유형입니다.
 - b. 응답 속성을 마우스 오른쪽 단추로 누르십시오.
 - c. 인스턴스 추가 옵션을 누르십시오. CLIENT_SYNCH_OrderStatus_Response Business Object의 새 인스턴스가 작성됩니다.
 - d. **OrderId, CustomerId**의 값과, 사용자가 이 주문에 대해 아는 기타 모든 세부사항을 입력하십시오. 이 주문의 모든 세부사항을 입력하고 나면, 이 창을 닫으십시오.
 - 주문 상태를 모르는 경우 다음을 수행하십시오.
 - a. SERVICE_SYNCH_TLO_OrderStatus의 결함 속성을 누르십시오. 결함 속성은 CLIENT_SYNCH_OrderStatus_Fault 유형입니다.
 - b. 결함 속성을 마우스 오른쪽 단추로 누르십시오.
 - c. 인스턴스 추가 옵션을 누르십시오. CLIENT_SYNCH_OrderStatus_Fault의 새 인스턴스가 작성됩니다.
 - d. faultcode, faultstring, 그리고 사용자가 SOAP 결함 메시지에 전송하려는 기타 모든 세부사항을 입력하십시오. 이 결함의 모든 값을 입력하고 나면, 이 창을 닫으십시오.

11. **REQUEST**”**REPLY**”**SUCCESS**를 선택하십시오.
SERVICE_SYNCH_TLO_OrderStatus.Retrieve로 레이블된 이벤트가 SampleSAPConnector를 시뮬레이트하는 Test Connector의 오른쪽 패널에 표시됩니다.
12. 창에 표시된 **SERVICE_SYNCH_TLO_OrderStatus.Retrieve** Business Object를 두 번 누르십시오.
 - SampleSiebelConnector에서 주문 상태를 리턴한 경우, 채워진 Business Object의 응답 속성을 보아야 합니다. 주문 상태를 확인하려면 응답 속성을 펼치십시오.
 - SampleSiebelConnector에서 결함을 리턴한 경우, 채워진 Business Object의 결함 속성을 보아야 합니다. 결함을 판별하려면 결함 속성을 펼치십시오.
13. Business Object를 검사하고 나면, 창을 닫으십시오.
REQUEST”**REPLY**”**SUCCESS**를 선택하십시오.
이것은 동기 시나리오의 실행을 완료합니다.

부록 D. 3.0.x로 이주

- 『역호환성』
- 『업그레이드 작업』
- 『웹 서비스 생성 유틸리티』
- 240 페이지의 『SOAP Data Handler』
- 241 페이지의 『SOAP 커넥터』
- 241 페이지의 『웹 서비스 커넥터』

이 부록은 Adapter for Web Service의 3.0.x 릴리스 역호환성과 Web Services Adapter 1.x 및 2.x 릴리스로부터의 이주 방법에 대해 설명합니다.

역호환성

Adapter for Web Services, 버전 3.0.x는 역호환되지 않습니다.

- 새 구성요소(웹 서비스 커넥터, SOAP Data Handler, WSDL ODA)는 이 제품의 이전 버전에서 해제된 구성요소와 함께 또는 개별적으로 사용할 수 없습니다.
- 이 제품의 이전 버전에서 해제된 구성요소(SOAP 커넥터, SOAP Data Handler, Web Services Generation Utility)는 버전 3.0.x와 함께 또는 개별적으로 사용할 수 없습니다.
- 이 제품 솔루션의 이전 버전에서 작성하거나 사용된 진단 지원 파일은 버전 3.0.x에서 사용하지 못할 수도 있습니다.
- 이 제품의 3.0.x 버전은 버전 4.2 이전의 WebSphere ICS 릴리스와 함께 사용할 수 없습니다.

업그레이드 작업

아래 섹션은 이 제품의 1.x 및 2.x 버전으로부터 구성요소를 업그레이드하는 방법에 대해 설명합니다.

웹 서비스 생성 유틸리티

웹 서비스 생성 유틸리티는 더 이상 사용할 수 없습니다. 대신 시스템 관리자 도구를 사용하여 협업을 웹 서비스로 표시합니다. 웹 서비스 생성 유틸리티에 의해 생성된 모든 진단 지원 파일은 이 릴리스에서 사용할 수 없습니다.

- 프록시 클래스 웹 서비스 커넥터는 이벤트 공고를 지원합니다. 따라서, 프록시 클래스는 더 이상 협업을 웹 서비스로 표시하는 데 필요하지 않습니다. 웹 서비스 생성

유틸리티를 사용하여 생성된 프록시 클래스가 System Manager 버전 4.2 도구를 사용하여 웹 서비스로서 드러난 협업을 호출할 수 없습니다.

- **WSDL 문서 웹 서비스 생성 유틸리티**는 ICS 버전 4.2 협업용 WSDL 문서를 생성하는 데 사용할 수 없습니다. 대신 시스템 관리자 도구를 사용하여 WSDL 문서를 생성해야 합니다. 자세한 정보는 163 페이지의 『마법사 실행』을 참조하십시오. 요청 처리용 협업을 사용 가능하게 하는 경우, 웹 서비스 생성 유틸리티를 사용하여 생성한 WSDL 문서는 커넥터 3.0.x 릴리스에서 사용 가능한 WSDL ODA에서 사용하지 못할 수 있습니다.

SOAP Data Handler

SOAP Data Handler는 웹 서비스 커넥터에서만 사용할 수 있습니다. 이 Data Handler는 다른 커넥터 또는 서버 액세스 인터페이스에서 사용할 수 없습니다. 아래의 섹션은 추가 지원 사항에 대해 설명합니다.

Meta Object

이전 릴리스에서 사용한 최상위 레벨 SOAP Data Handler Meta Object는 더 이상 지원되지 않습니다. 대신 3.0.x 릴리스 SOAP Data Handler에서 사용할 새 최상위 레벨 Meta Object를 작성해야 합니다. 이 Meta Object에는 Classname 및 SOAPNameHandler 속성만 있어야 합니다.

새 Meta Object의 경우, SOAP 메시지를 Business Object로 변환 및 Business Object를 SOAP 메시지로 변환에 하위 Meta Object가 필요하지 않습니다. 이에 따라 최상위 레벨 Meta Object에 SOAPToBOConfigMO 및 BOToSOAPConfigMO 속성이 없는지 확인하십시오.

이전에 SOAP 메시지를 Business Object로 변환 및 Business Object를 SOAP 메시지로 변환을 설명한 하위 Meta Object는 이제 SOAP 요청, SOAP 응답 및 SOAP 결합 Business Object의 하위 오브젝트로 추가해야 합니다. 자세한 정보는 119 페이지의 제 5 장 『SOAP Data Handler』 및 25 페이지의 제 3 장 『Business Object 요구사항』을 참조하십시오.

응용프로그램 특정 정보

새 SOAP Data Handler에는 새 응용프로그램 특정 정보(ASI) 기능을 특징으로 합니다. 특정 ASI를 SOAP Business Object에 추가하여(그러나 필수는 아님) 이러한 향상된 기능을 이용할 수 있습니다. 하위 SOAP 구성 MO를 Business Object에 추가하는 것을 제외하고 커넥터의 이전 릴리스에서 작성한 SOAP Business Object를 전개하여 3.0.x 버전에서 사용할 수 있습니다.

커넥터 호환성

새 SOAP Data Handler는 3.0.x 웹 서비스 커넥터와 함께만 사용이 가능합니다. 새 SOAP Data Handler를 이전 릴리스에서의 구성요소와 함께 사용할 수는 없습니다(예 : SOAP Connector 또는 서버 액세스 인터페이스).

3.0.x 웹 서비스 커넥터에서는 이전 SOAP Data Handler를 사용할 수 없습니다.

SOAP 커넥터

SOAP 커넥터는 릴리스 3.0.x를 지원하지 않습니다. 대신 웹 서비스 커넥터를 사용하여 웹 서비스를 호출해야 합니다.

웹 서비스 커넥터

릴리스 3.0.x에서는 협업을 웹 서비스로 표시하고 웹 서비스를 호출하는 데 웹 서비스 커넥터를 사용합니다. 협업을 웹 서비스로 표시하는 데 새 이벤트 공고 기능이 사용됩니다. 새 요청 처리 기능은 이제 웹 서비스를 호출하는 데 사용합니다. 아래 섹션에서는 웹 서비스 커넥터를 사용하기 위해 완료해야 하는 이주 작업을 강조합니다.

주: 아래에 설명된 이주 작업은 세부적이 아닐 수도 있습니다. 또한 순서에 상관 없이 작업을 완료할 수 있습니다.

이벤트 공고

3.0.x 커넥터는 웹 서버에서의 프록시 클래스 작성 및 전개를 위해 요구사항 없이 동기식 또는 비동기식으로 협업을 호출할 수 있습니다. 커넥터에는 커넥터에 이벤트를 알리는 리스너 프레임워크가 있습니다. 리스너 프레임워크는 SOAP/HTTP, SOAP/HTTPS 및 SOAP/JMS 바인딩을 지원합니다. 리스너를 적절하게 구성하면 커넥터는 웹 서비스로 표시된 협업을 위해 웹 서비스 클라이언트를 검출하여 응답할 수 있습니다. 리스너 프레임워크 및 구성 방법에 대한 자세한 정보는 68 페이지의 『프로토콜 리스너』를 참조하십시오.

이벤트 공고용 Business Object: 이벤트 공고 최상위 레벨 오브젝트(TLO)를 작성해야 합니다. TLO는 SOAP 요청 Business Object, 선택적으로(동기 요청 처리용) SOAP 응답 및 SOAP 결합 Business Object의 컨테이너입니다. TLO의 구조 구성요소는 단일 웹 서비스 작동을 예상합니다. SOAP 요청 Business Object는 SOAP 요청 메시지에 해당되고 SOAP 응답 Business Object는 SOAP 응답 메시지에 해당되며 SOAP 결합 Business Object는 SOAP 결합 메시지에 해당됩니다. 자세한 정보는 26 페이지의 『동기 이벤트 처리 TLO』를 참조하십시오.

이벤트 공고 및 SOAP Business Object: 이전 릴리스에서 서버 액세스 인터페이스와 함께 사용한 SOAP Business Object는 위의 240 페이지의 『SOAP Data Handler』에 설명한 수정사항과 함께 3.0.x 릴리스에서 사용할 수 있습니다. 이벤트 공고 TLO에서 하위 오브젝트로 SOAP Business Object를 지정해야 합니다.

요청 처리

이전 릴리스의 SOAP 커넥터와 같이 3.0.x 웹 서비스 커넥터는 웹 서비스를 호출할 수 있습니다. 또한 새 커넥터는 SOAP/JMS 바인딩을 사용한 웹 서비스 호출을 지원합니다. 아래 섹션은 커넥터 요청 처리에서의 추가 변경사항에 대해 설명합니다.

최상위 레벨 오브젝트 요청 처리: 요청 처리 TLO를 작성해야 합니다. TLO는 SOAP 요청 Business Object, 선택적으로(동기 요청 처리용) SOAP 응답 및 SOAP 결합 Business Object의 컨테이너입니다. TLO의 구조 구성요소는 단일 웹 서비스 작동을 예상합니다. SOAP 요청 Business Object는 SOAP 요청 메시지에 해당되고 SOAP 응답 Business Object는 SOAP 응답 메시지에 해당되며 SOAP 결합 Business Object는 SOAP 결합 메시지에 해당됩니다. 이러한 의미에서 3.0.x 요청 처리 TLO는 이전 릴리스의 SOAP 커넥터에 사용된 TLO에 해당됩니다. 요청 처리 TLO에 대한 자세한 정보는 44 페이지의 『동기 요청 처리 TLO』를 참조하십시오.

SOAP Business Object: 이전 릴리스의 SOAP 커넥터에 사용된 SOAP Business Object는 240 페이지의 『SOAP Data Handler』의 설명에 따라 수정사항과 함께 사용할 수 있습니다. 이러한 Business Object는 요청 처리 TLO의 하위 오브젝트로 지정해야 합니다. 이전 릴리스에서 이 Business Object는 SOAP 커넥터에 사용된 TLO의 하위 오브젝트였습니다.

3.0.x 웹 서비스 커넥터에는 SOAP 요청 Business Object에 대한 추가 요구사항이 있습니다. 각각의 SOAP 요청 Business Object에는 프로토콜 구성 MO(Meta Object) 유형의 속성이 있어야 합니다. 커넥터는 프로토콜 구성 MO를 사용하여 요청 메시지의 대상을 판별합니다. 프로토콜 구성 MO 각각에는 대상 웹 서비스의 주소를 제공하는 Destination 속성이 있습니다. 대상 웹 서비스를 호출하는 데 SOAP/HTTP 또는 SOAP/HTTPS를 사용하는 경우, Destination 속성은 이전 릴리스의 SOAP 커넥터에 사용된 TLO의 URL 속성에 해당됩니다. 자세한 정보는 33 페이지의 『프로토콜 구성 MO』를 참조하십시오.

부록 E. HTTPS/SSL 구성

- 『키스토어 설정』
- 244 페이지의 『TrustStore 설정』
- 245 페이지의 『공용 키 인증을 위해 CSR(Certificate Signing Request) 생성』

SSL을 사용할 계획인 경우, 키스토어, 인증 및 키 생성 관리를 위해 타사 소프트웨어를 사용해야 합니다. 웹 서비스 커넥터는 이 작업을 위한 도구와 함께 제공되지 않습니다. 그러나, IBM JRE와 함께 출시된 keytool을 사용하여 자체 서명한 인증을 작성하고 키스토어를 관리할 수 있습니다.

키 및 인증 관리 유틸리티인 keytool을 사용하면 자신의 공용/개인용 키 쌍과 연관된 인증을 관리할 수 있습니다. 자체 인증(자신을 다른 사용자나 서비스에게 인증시키는)이나 디지털 서명을 사용하는 데이터 무결성과 인증 서비스에서 사용하기 위한 것입니다. keytool 유틸리티를 사용하여 통신하고 있는 상대방(peer)의 공용 키(인증 양식으로 된)를 저장할 수도 있습니다.

이 부록에서는 keytool을 사용하여 키스토어를 설정하는 방법에 대해 설명합니다. 이 부록의 예는 설명만을 목적으로 하며, keytool에 대한 문서와 관련 서적을 대신하는 것을 목적으로 하지 않습니다. 항상 키스토어를 설정하는 데 사용하는 도구에 대한 소스 문서를 참조하십시오. keytool에 대한 자세한 정보는 다음을 참조하십시오.

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

키스토어 설정

keytool을 사용하여 KeyStore를 작성하려면, 먼저 KeyStore에 키 쌍을 작성해야 합니다. 예를 들어서, 다음 명령행을 입력하면,

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

keytool이 즉시 암호를 입력하도록 프롬프트합니다. 선택한(keytool 매개변수에서) 암호를 입력할 수도 있으나, keytool에서 SSL " KeyStorePassword 커넥터 등록 정보의 값으로 입력한 암호를 지정해야 합니다. 자세한 정보는 110 페이지의 『KeyStorePassword』를 참조하십시오.

샘플 명령을 실행하면 keystore라는 키스토어가 c:\security\keystore 디렉토리에 작성됩니다. 따라서, c:\security\keystore를 SSL " KeyStore 커넥터 계층 구조 등록 정보의 값으로 입력합니다. 또한, 위의 명령행 예에서 -alias wsadapter를 SSL " KeyStoreAlias 커넥터 계층 구조 등록 정보의 값으로 입력합니다. 그러면, keytool 유틸

리터에서 인증에 대한 세부사항을 넣으라고 프롬프트합니다. 다음은 각각의 프롬프트에 입력할 수 있는 것을 보여줍니다. (keytool 문서 참조하십시오.)

```
성과 이름이 무엇입니까?  
[알 수 없음]: HostName  
조직 단위의 이름이 무엇입니까?  
[알 수 없음]: wbi  
조직의 이름이 무엇입니까?  
[알 수 없음]: IBM  
구/군/시의 이름이 무엇입니까?  
[알 수 없음]: Burlingame  
시/도의 이름이 무엇입니까?  
[알 수 없음]: CA  
이 단위의 두 자리 국가 코드가 무엇입니까?  
[알 수 없음]: US  
<CN=HostName, OU=wbi, O=IBM, L=Burlingame,  
ST=CA, C=US>가 맞습니까?  
[아니오]: 예
```

다음으로, keytool이 암호를 입력하도록 프롬프트합니다.

<wsadapter>에 대한 키 암호를 입력하십시오(키스토어 암호와 같으면 RETURN을 누르십시오).

Return을 눌러서 같은 암호를 사용하십시오. 자체 서명된 인증을 사용하려면, 위에서 작성한 인증을 내보낼 수 있습니다. 이런 경우, 명령행에서 다음을 입력하십시오.

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

이제 keytool이 키스토어 암호를 입력하도록 프롬프트합니다. 위에서 입력한 암호를 입력하십시오.

TrustStore 설정

다음은 위해 TrustStore를 설정하려고 할 수 있습니다. SOAP/HTTPS 프로토콜 리스너가 웹 서비스 클라이언트를 인증하기를 원하는 경우, SSL " UseClientAuth 커넥터 구성 등록 정보를 true로 설정하십시오. 이런 경우, SOAP/HTTPS 프로토콜 리스너에서는 TrustStore가 신뢰된 모든 웹 서비스 클라이언트에 대해 인증을 포함함을 예측합니다. 커넥터에서 JSSE 기본 메커니즘을 사용하여 클라이언트를 신뢰한다는 것을 알아두십시오. SOAP/HTTPS 웹 서비스를 호출하는 경우, SOAP/HTTP-HTTPS 프로토콜 핸들러에서는 TrustStore가 웹 서비스를 신뢰해야 합니다. TrustStore에 신뢰된 모든 웹 서비스의 인증이 있어야 한다는 것을 의미합니다. 커넥터에서 JSSE 기본 메커니즘을 사용하여 클라이언트를 신뢰한다는 것을 알아두십시오. 신뢰된 인증을 TrustStore로 가져오려면, 다음과 같은 명령을 입력하십시오.

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

이제 keytool이 키스토어 암호를 입력하도록 프롬프트합니다. -keystore c:\security\truststore를 입력하면, SSL " TrustStore 계층 구조 등록 정보를

c:\security\truststore로 설정했는지 확인하십시오. 또한, SSL ” TrustStorePassword 계층 구조 등록 정보의 값을 이전에 입력한 암호로 설정해야 합니다.

공용 키 인증을 위해 CSR(Certificate Signing Request) 생성

사용자의 ID를 신뢰하는, 이미 신뢰된 상대방들 사이에 SSL 데이터 교환이 있으면, 자체 서명된 인증이 적당할 수 있습니다. 그러나 인증이 CA(Certifying Authority)에 의해 서명될 때 다른 사람에 의해 신뢰될 가능성이 많습니다.

keytool 유틸리티를 사용하여 CA가 서명한 인증을 얻으려면, 먼저 CSR(Certificate Signing Request)을 생성한 후에 CSR을 CA에게 주어야 합니다. 그런 후, CA가 인증을 서명하고 사용자에게 리턴합니다.

다음 명령을 입력하여 CSR을 생성합니다.

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

명령에서 alias는 개인용 키를 위해 작성한 키스토어 별명입니다. keytool 유틸리티가 CA에게 제공하는 CSR 파일을 생성합니다. 그러면, CA가 사용자에게 서명된 인증을 제공합니다. 이 인증을 키스토어로 가져와야 합니다. 가져오려면, 다음 명령을 입력합니다.

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

가져오면, 키스토어에 있는 자체 서명된 인증이 CA 서명된 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