



Adapter for JText 사용자 안내서

버전 5.4.x



Adapter for JText 사용자 안내서

버전 5.4.x

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 117 페이지의 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

이 개정판은 새 개정판에 별도로 명시하지 않는 한 커넥터 버전 5.3.x 및 모든 후속 릴리스와 수정에 적용됩니다.

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

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

© Copyright International Business Machines Corporation 2000, 2003. All rights reserved.

목차

이 책의 정보	v
이 책의 사용자	v
관련 서적	v
인쇄 규칙	vi
이 릴리스의 새로운 기능	vii
버전 5.4.x	vii
이전 버전	vii
제 1 장 JText Adapter 개요	1
어댑터 구성요소	2
JText Connector가 사용하는 Business Object	4
커넥터 작동 방법	7
커넥터 기능	15
로케일 종속 데이터 처리	16
제 2 장 JText Adapter 설치 및 구성	19
호환성	19
전제조건	20
JText Adapter 설치	20
커넥터 구성	20
다중 커넥터 인스턴스 작성	24
지원되는 Business Object 추가	25
커넥터 시작	26
커넥터 중지	27
제 3 장 JText Connector Meta Object 사용	29
JText Meta Object 이름 지정 규칙	30
JText Meta Object 구조	30
공통 구성 작업	41
제 4 장 JText Connector 문제점 해결	67
오류 메시지 로깅	67
Meta Object 이름 지정에 대한 문제점	67
이벤트 트리거링에 대한 문제점	68
JText 장애 핸들링	68
제 5 장 JText Connector로 이주 또는 업그레이드	75
업그레이드 시나리오	75
버전 4.0.x에서 5.3.x로 업그레이드	75
버전 3.2.0에서 4.0.x로 업그레이드해야 하는 이유	77
버전 4.0.x로 업그레이드	77
Text Connector로부터 업그레이드해야 하는 이유	79
JText Connector로 업그레이드	79

부록 A. 커넥터의 표준 구성 등록 정보	81
신규 및 삭제된 등록 정보	81
표준 커넥터 등록 정보 구성	81
표준 등록 정보 요약	83
표준 구성 등록 정보	86
부록 B. Connector Configurator	99
Connector Configurator 개요	99
Connector Configurator 시작	100
System Manager에서 Configurator 실행	101
커넥터 특정 등록 정보 템플릿 작성	101
새 구성 파일 작성	104
기존 파일 사용	105
구성 파일 완료	106
구성 파일 등록 정보 설정	107
구성 파일 저장	114
구성 파일 변경	114
구성 완료	115
국제화된 환경에서 Connector Configurator 사용	115
주의사항	117
프로그래밍 인터페이스 정보	118
상표 및 서비스표	119

이 책의 정보

IBM^(R) WebSphere^(R) Business Integration Adapter 포트폴리오는 주요 e-business 기술, 엔터프라이즈 응용프로그램, 레거시 및 메인프레임 시스템에 대한 통합 연결성을 제공합니다. 제품에는 비즈니스 프로세스 통합을 위한 구성요소를 사용자 정의, 작성 및 관리할 수 있는 도구와 템플릿이 포함되어 있습니다.

이 책은 IBM WebSphere Business Integration Adapter for JText^(TM)에 대한 설치, 구성, 비즈니스 오브젝트 개발 및 문제점 해결에 대해 설명합니다.

이 책의 사용자

이 문서는 WebSphere 컨설턴트 및 고객을 위한 것입니다. 통합 브로커의 기초, Business Object 개발의 기초 및 Data Handler 개발을 잘 알고 있어야 합니다.

관련 서적

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

다음 사이트에서 관련 문서를 설치할 수 있습니다.

일반적인 어댑터 정보, WebSphere 메시지 브로커(WebSphere MQ Integrator, WebSphere MQ Integrator Broker, WebSphere Business Integration Message Broker)와 함께 어댑터를 사용하는 경우 및 WebSphere Application Server와 함께 어댑터를 사용하는 경우:

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

InterChange Server와 함께 어댑터를 사용하는 경우:

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

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

메시지 브로커(WebSphere MQ Integrator Broker, WebSphere MQ Integrator 및 WebSphere Business Integration Message Broker)에 대한 자세한 정보:

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

WebSphere Application Server에 대한 자세한 정보:

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

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

인쇄 규칙

이 책에서는 다음 규칙을 사용합니다.

courier 글꼴	명령어, 파일 이름, 사용자가 입력하는 정보 또는 시스템이 화면에 인쇄하는 정보 같은 리터럴 값을 표시합니다.
굵은체	처음으로 나타나는 새로운 용어를 표시합니다.
기울임, <i>italic</i>	변수 이름 또는 상호 참조를 표시합니다.
파란색 윤곽선	온라인으로 매뉴얼을 보는 경우에만 볼 수 있는 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 바로 가려면 윤곽선 내부를 누르십시오.
{ }	구문 행에서 사용자가 반드시 하나만을 선택해야 하는 옵션 세트는 중괄호 안에 표시됩니다.
[]	구문 행에서 선택적 매개변수는 대괄호 안에 표시됩니다.
...	구문 행에서 생략 부호(...)는 이전 매개변수의 반복을 표시합니다. 예를 들어, <code>option[,...]</code> 은 쉼표로 구분된 여러 개의 옵션을 입력할 수 있음을 의미합니다.
< >	이름 지정 규칙에서 꺾쇠 괄호는 <code><server_name><connector_name></code> <code>tmp.log</code> 에서와 같이 이름의 개별 요소를 묶어 이를 서로 구별합니다.
/, \	이 문서에서 백슬래시(\)는 디렉토리 경로에 대한 규칙으로 사용됩니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체하십시오. 모든 WebSphere Business Integration System 제품 경로 이름은 제품이 시스템에 설치된 디렉토리에 상대적입니다.
%text% 및 \$text	퍼센트(%) 부호 내의 텍스트는 Windows <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>	제품이 설치된 디렉토리를 표시합니다.

이 릴리스의 새로운 기능

버전 5.4.x

5.4 버전을 시작하면, Microsoft Windows NT에서 Adapter for JText가 더 이상 지원되지 않습니다.

어댑터 설치 정보가 이 안내서에서 옮겨졌습니다. 해당 정보에 대한 새로운 위치는 20 페이지의 『JText Adapter 설치』의 2장을 참조하십시오.

이전 버전

이전 버전의 기능 및 변경사항

버전 5.3.x

JText Connector 버전 5.3.x에 대한 이 문서의 릴리스는 다음 새 기능을 포함합니다.

- 새 FTPTransferType Meta Object 등록 정보가 추가되어 ASCII 및 2진 파일 전송을 허용합니다.
- 2진 데이터에 대한 추가 데이터 구문 분석 메소드, FixedBOSize가 JText Adapter에 추가되었습니다.
- 중복 파일 이름을 피하기 위한 시간 소인을 제공함으로써 호스트 파일 시스템 MVS에 대한 지원이 향상되었습니다.
- 어댑터에 새 DataProcessingMode Meta Object 등록 정보가 추가되어 2진 데이터를 읽고 쓸 수 있습니다.
- JText Adapter에 새 Meta Object 속성, FTPDataStructure가 추가되어 원격 MVS 플랫폼의 파일 또는 레코드 지향 데이터 구조에 대한 지원을 제공합니다.

이제 어댑터가 WebSphere Application Server를 통합 브로커로 사용할 수 있습니다. 자세한 정보는 19 페이지의 『호환성』을 참조하십시오. 이제 어댑터는 다음 플랫폼에서 실행됩니다.

- Solaris 7, 8
- AIX 5.x
- HP UX 11.i

버전 5.2.x

다음은 2003년 3월에 갱신된 사항입니다. “CrossWorlds”라는 이름은 전체 시스템을 설명하거나 이전과 거의 유사한 구성요소 또는 도구의 이름을 수정하기 위해서 더 이상

사용되지 않습니다. 예를 들어, “CrossWorlds System Manager”는 이제 “System Manager”로, “CrossWorlds InterChange Server”는 “WebSphere InterChange Server”로 사용됩니다.

버전 5.1.x

JText Connector 버전 5.1.x에 대한 이 문서 릴리스는 다음과 같은 신규 정보 또는 수정된 정보를 포함합니다.

- 파일 기록 시 커넥터가 스테이징 디렉토리를 사용하게 하는 StagingDir Meta Object 속성의 문서. 이 커넥터는 Business Object를 나타내는 파일을 스테이징 디렉토리에 기록한 후 구성된 출력 디렉토리로 이동시켜, 출력 디렉토리에서 파일을 조작할 수 있는 외부 프로세스가 파일이 완전히 기록되기 전에는 파일을 수신하지 않게 합니다. 자세한 정보는 39 페이지의 『StagingDir』을 참조하십시오.
- 파일 작성 시 EndBODelimiter 속성용으로 지정된 값을 커넥터가 포함하게 하거나, 데이터 스트림의 값을 그대로 둘 수 있게 하는 IncludeEndBODelimiter Meta Object 속성의 문서. 자세한 정보는 38 페이지의 『IncludeEndBODelimiter』를 참조하십시오.
- 로케일에 따라 다른 형식 정보로 커넥터가 파일을 읽을 수 있게 하는 FTPFileListingFormat Meta Object 속성의 문서. 자세한 정보는 36 페이지의 『FTPFileListingFormat』을 참조하십시오.
- 커넥터가 FTP 서버와의 지속적 연결을 유지보수하게 하는 FTPKeepConnectionOpen Meta Object 속성의 문서. 자세한 정보는 37 페이지의 『FTPKeepConnectionOpen』을 참조하십시오.
- 커넥터가 통신하는 FTP 서버가 MVS 플랫폼을 실행 중인 경우 값 MVS에 설정되어야 하는 FTPOSPlatform Meta Object 속성의 문서. 자세한 정보는 37 페이지의 『FTPOSPlatform』을 참조하십시오.
- 이벤트를 폴링 중인 FTP 서버가 사용 불가능한 경우 커넥터가 종료하는지 여부를 지정하게 하는 FTPPollTerminateIfServerDown Meta Object 속성의 문서. 자세한 정보는 37 페이지의 『FTPPollTerminateIfServerDown』을 참조하십시오.
- 사용 불가능한 FTP 서버로 요청 처리 수행 시 커넥터가 종료하는지 여부를 지정하게 하는 FTPRequestTerminateIfServerDown Meta Object 속성의 문서. 자세한 정보는 37 페이지의 『FTPRequestTerminateIfServerDown』을 참조하십시오.
- DataEncoding Meta Object 속성과 관련된 정보의 개정. 속성을 모든 Java 지원 인코딩으로 설정할 수 있습니다. 자세한 정보는 34 페이지의 『DataEncoding』을 참조하십시오.

버전 5.0.x

커넥터가 국제화되어 있습니다. 자세한 정보는 16 페이지의 『로케일 종속 데이터 처리』 및 81 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

버전 4.5.x

WebSphere Business Integration Adapter for JText는 Connector for JText를 포함하고 있습니다. 이 어댑터는 ICS(InterChange Server) 및 WMQI(WebSphere MQ Integrator) 통합 브로커 모두와 함께 작동합니다. 통합 브로커는 이기종 세트의 응용프로그램 통합을 수행하는 응용프로그램으로, 데이터 라우팅을 포함하는 서비스를 제공합니다.

이 어댑터는 다음을 포함합니다.

- JText에 고유한 응용프로그램 구성요소
- \connectors\JText\Samples\ 디렉토리에 들어 있는 샘플 Business Object
- 다음으로 구성되는 IBM WebSphere Adapter Framework.
 - 커넥터 프레임워크
 - 개발 도구(Business Object Designer 및 Connector Configurator 포함)
 - API(ODK, JCDK 및 CDK 포함)

이 매뉴얼은 ICS(InterChange Server) 및 WMQI(WebSphere MQ Integrator)의 두 통합 브로커 모두와 함께 이 어댑터 사용에 대한 정보를 제공합니다.

버전 4.4.x

JText Connector 버전 4.4.x에 대한 이 문서의 릴리스는 다음 변경사항을 포함합니다.

- 원격 FTP 파일 시스템을 사용하기 위한 커넥터 구성에 대한 지원이 추가되었습니다. 세부사항은 53 페이지의 『원격 FTP 파일 시스템 지정』을 참조하십시오.
- 폐기된 DataHandlerFormatter 및 ByNameValue, ByDelimiter 및 BySize 포맷터의 문서는 제거되었습니다. 이들 포맷터 사용에 대한 설명은 커넥터용 안내서의 3.0.0 또는 2.0.3 릴리스에 대한 JText Connector 문서를 참조하십시오.
- 동적 하위 Meta Object의 문서가 개선되었으며 해당 매뉴얼은 JTextWrapper 구성이 사용되지 않음을 명시합니다.
- BODelimiter Meta Object 속성의 문서가 개선되었습니다.
- MO_JText_DHFormatter.txt 파일이 제품 디렉토리의 \repository\Jtext 디렉토리에서 제거되었습니다.

버전 4.3.x

IBM WebSphere Business Integration for JText 제품에는 Adapter for e-Mail이 포함됩니다. 이 어댑터는 InterChange Server 및 WebSphere MQ Integrator 통합 브로커 모두와 작동합니다. 통합 브로커는 이기종 세트의 응용프로그램 통합을 수행하는 응용프로그램으로, 데이터 라우팅을 포함하는 서비스를 제공합니다.

이 어댑터는 다음을 포함합니다.

- 전자 우편 기술에 고유한 응용프로그램 특정 구성요소.
- \connectors\EMail\samples에 위치한 샘플.
- 다음으로 구성되는 IBM WebSphere Adapter Framework.
 - 어댑터 프레임워크
 - 개발 도구(Business Object Designer 및 Connector Configurator 포함)
 - API(ODK, JCDK 및 CDK 포함)

이 매뉴얼은 InterChange Server 및 WebSphere MQ Integrator 두 통합 브로커 모두와 함께 이 어댑터를 사용하는 방법에 대한 정보를 제공합니다.

어댑터가 국제화되어 있습니다. 자세한 정보는 16 페이지의 『로케일 종속 데이터 처리』 및 81 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

버전 4.2.x

JText Connector 버전 4.2.x에 대한 이 문서의 릴리스는 다음의 신규 또는 정정된 정보를 포함합니다.

- 동적 하위 Meta Object에 대한 지원이 추가되었습니다. 커넥터는 이제 동적 파일 스펙을 위해 랩퍼 오브젝트 또는 동적 하위 Meta Object를 사용하도록 구성될 수 있습니다. 세부사항은 5 페이지의 『동적 하위 Meta Object 사용』을 참조하십시오.
- 커넥터가 DataHandlerFormatter를 통해 호출하는 대신 Data Handler를 직접 호출할 수 있도록 하기 위해 EventDataHandler 및 OutputDataHandler Meta Object 속성이 추가되었습니다. 이러한 속성을 사용하여 DataHandlerFormatter의 Meta Object 안에서 간접적으로 지정하는 대신 JText Connector Meta Object에 Data Handler를 직접 지정할 수 있습니다. 세부사항은 34 페이지의 표 5를 참조하십시오.
- 임의의 EndBODelimiter 또는 줄 바꾸기 문자 없이 파일에 Business Object 속성을 출력할 때 EndBODelimiter Meta Object를 구성하기 위해 새로운 사용자 옵션인 None이 추가되었습니다. 세부사항은 50 페이지의 표 7을 참조하십시오.
- 사용자가 기본 UTF7 대신 UTF8 인코딩을 지정할 수 있도록 DataEncoding Meta Object 속성이 추가되었습니다. 세부사항은 34 페이지의 표 5를 참조하십시오.

JText Connector 버전 4.1.x에 대한 문서의 릴리스에는 다음의 새 정보가 들어 있습니다.

- 커넥터의 아카이브 기능이 다시 설계되었습니다.
- 포맷터 Meta Object에 대한 이름 지정 규칙이 변경되었습니다.

중요: Business Object 처리를 위해 Data Handler만을 사용하고 포맷터를 사용하지 않는 것이 좋습니다.

- CwJTFormatter.jar 파일이 더 이상 분배되지 않습니다.
- event.log 파일의 형식과 사용법이 변경되었습니다.

- Data Handler 지정 프로세스에 대한 내용이 명확하게 정리되고 수정되었습니다.
- 이벤트 공고 및 Data Handler 처리에 대한 설명이 명확하게 정리되고 수정되었습니다.
- 사용자 정의 Meta Object 작성에 대한 새 절이 추가되었습니다.
- 68 페이지의 『이벤트 트리거링에 대한 문제점』 및 68 페이지의 『JText 장애 핸들링』 절의 내용이 명확하게 정리되고 수정되었습니다

제 1 장 JText Adapter 개요

이 장에서는 IBM WebSphere Business Integration Adapter for JText에 대해 설명합니다. 어댑터는 통합 브로커가 텍스트 또는 2진 파일을 교환하여 응용프로그램과 통신할 수 있게 합니다. 이 커넥터는 API가 부족한 응용프로그램과의 데이터 통합을 용이하게 합니다.

어댑터는 두 파트, 즉 커넥터 프레임워크 및 응용프로그램 특정 구성요소로 구성됩니다. 해당 코드가 모든 커넥터에 공통인 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이의 중재자로서 작용합니다. 응용프로그램 특정 구성요소에는 특정 응용프로그램에 대해 조정된 코드가 들어 있습니다. 커넥터 프레임워크는 통합 브로커와 응용프로그램 특정 구성요소 사이에 다음 서비스를 제공합니다.

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

이 문서에는 커넥터 프레임워크와 응용프로그램 특정 구성요소에 대한 정보가 들어 있습니다. 문서에서는 이러한 구성요소를 모두 어댑터라고 지칭합니다.

이 장의 주제는 다음과 같습니다.

- 2 페이지의 『어댑터 구성요소』
- 4 페이지의 『JText Connector가 사용하는 Business Object』
- 7 페이지의 『커넥터 작동 방법』
- 15 페이지의 『커넥터 기능』
- 16 페이지의 『로케일 종속 데이터 처리』

커넥터와 통합 브로커의 관계에 대한 자세한 정보는 *System Administration Guide* 또는 *Implementation Guide for WebSphere MQ Integrator Broker*를 참조하십시오.

다음과 같은 경우 JText Adapter를 사용하십시오.

- 응용프로그램에 통합 브로커가 통신할 수 있는 C, C++ 또는 Java 표준 API가 없는 경우
- 사용자 정의 빌드된 응용프로그램에 대한 이벤트 테이블을 확보하는 것이 불가능한 경우
- 문자열 또는 2진 파일은 데이터 교환을 위한 가장 적합한 방법입니다.

이런 경우, 응용프로그램을 보다 큰 시스템에 통합하기 위한 가장 간단한 방법은 JText Connector를 통해 문자열 또는 2진 파일을 교환하는 것입니다.

어댑터 구성요소

JText Adapter의 구성요소는 다음과 같습니다.

- 『응용프로그램 특정 구성요소』
- 『Data Handler』
- 3 페이지의 『Meta Object』

그림 1은 IBM WebSphere ICS(InterChange Server)를 통합 브로커로 사용하는 경우 JText Connector의 구조를 보여줍니다.

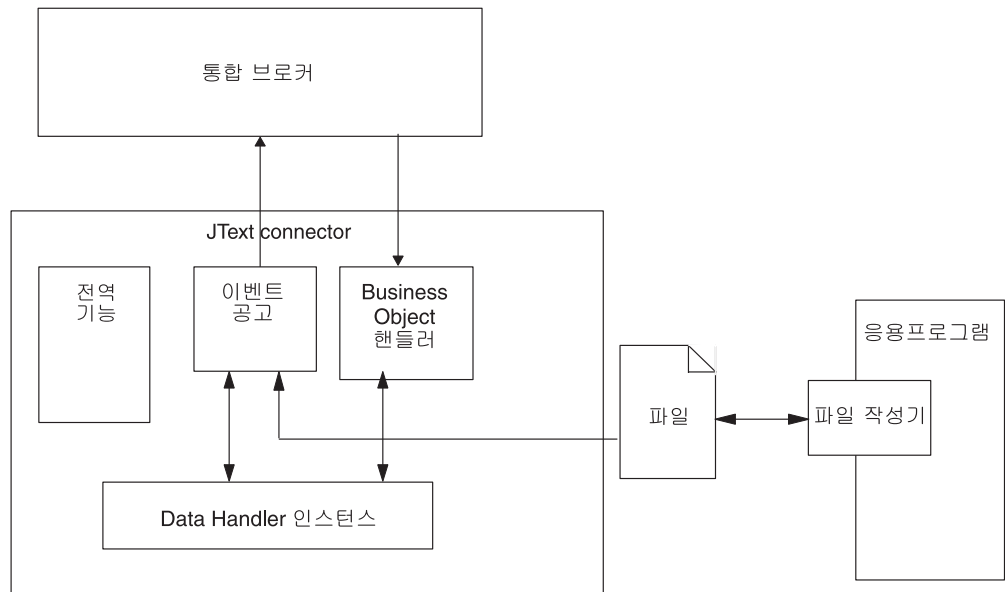


그림 1. JText Connector 구조

응용프로그램 특정 구성요소

JText Adapter의 응용프로그램 특정 구성요소는 파일을 조작하고, 지정된 Data Handler를 호출하여 Business Object와 문자열 또는 바이트 배열 사이에서 데이터를 변환합니다. 또한 통합 브로커와의 통신도 처리합니다.

Data Handler

JText Connector의 목적은 모든 기존 파일 형식과 Business Object 사이의 변환을 제공하는 것입니다. 이를 위해, JText Connector는 커넥터의 Meta Object 구성에 지정되는 Data Handler를 사용합니다.

Data Handler는 파일 읽기 또는 쓰기를 통해 어떠한 방법으로도 파일 시스템과 상호 작용하지 않고 변환을 수행합니다. 텍스트 파일과의 모든 상호작용은 다른 커넥터 구성 요소에 의해 처리됩니다.

데이터 변환을 처리하기 위해 IBM WebSphere Business Integration Adapter Framework가 제공하는 Data Handler나 사용자가 특정 텍스트 형식화 필요성을 처리하기 위해 작성하는 Data Handler를 사용할 수 있습니다. 이 제품은 다음 Data Handler를 제공합니다.

- NameValue -- 이름 지정된 필드를 기본으로 텍스트 데이터를 구문 분석합니다. 이 경우, 텍스트 파일이 Business Object 유형(BusinessObject=B0name), Verb(Verb=verbName) 및 속성 수(AttributeCount=numericValue)를 식별하는 필드를 포함합니다.
- Delimited -- 주로 시스템 읽기의 효율성이 가장 중요한 경우 사용됩니다. Business Object 데이터의 개별 필드를 구분하는 지정된 분리문자를 기본으로 텍스트 데이터를 구문 분석합니다.
- FixedWidth -- 고정 길이 필드를 사용하여 텍스트 데이터를 구문 분석합니다. 필드 길이는 각 Business Object 속성의 MaxLength 등록 정보에 의해 지정됩니다. 이 등록 정보의 값은 Business Object 정의에 저장됩니다.

제품은 NameValue, Delimited 및 FixedWidth Data Handler에 대한 샘플 코드를 제공합니다. 이 코드를 사용하여 Data Handler를 사용자 정의하거나 사용자의 Data Handler를 개발할 수 있습니다. 샘플 코드는 다음 위치에 있습니다.

UNIX:

`$ProductDir/DevelopmentKits/edk/DataHandler/Samples`

Windows:

`%ProductDir%\DevelopmentKits\edk\DataHandler\Samples`

`ProductDir`은 IBM WebSphere Business Integration Adapters 제품이 설치된 디렉토리를 나타냅니다. CROSSWORLDS 환경 변수에는 기본값이 `IBM\WebSphereAdapters`인 `ProductDir` 디렉토리 경로가 포함됩니다.

자세한 정보는 14 페이지의 『Data Handler 처리가 작동하는 방법』을 참조하십시오. 제품이 제공하는 각 Data Handler에 대한 자세한 정보는 *Data Handler* 안내서를 참조하십시오.

Meta Object

사용자가 Connector Configurator에서 설정하는 표준 및 응용프로그램 특정 커넥터 구성 등록 정보 외에, JText Connector에는 사용자가 상이한 Business Object를 서로 다르게 처리하도록 커넥터를 구성할 수 있게 하는 구성 등록 정보 세트가 있습니다. JText Meta Object를 사용하여 이들 등록 정보를 설정합니다. Meta Object는 구성 정보를 포함하는 특수한 종류의 Business Object입니다.

커넥터는 Meta Object 정보를 사용하여, 파일에서 읽는 문자열 또는 바이트 배열을 Business Object로 변환하고 Business Object에서 파일로 문자열 또는 바이트 배열을 형식화하기 위해 사용할 클래스를 판별합니다. JText Meta Object는 이벤트 및 요청 처리 중에 사용할 디렉토리, 파일 확장자, 파일 이름, Business Object 분리문자 및 Data Handler를 지정합니다.

JText Adapter는 Meta Object를 내부적으로 사용합니다. 통합 브로커를 통해 Meta Object를 전송하지 않습니다. 커넥터를 구성하기 위한 Meta Object 사용에 대한 자세한 정보는 29 페이지의 제 3 장 『JText Connector Meta Object 사용』을 참조하십시오.

JText Connector가 사용하는 Business Object

JText Connector를 위한 Business Object는 변환에 대해 지정된 Data Handler에 필요한 형식으로 데이터를 전달해야 합니다. 그러나 JText Connector 응용프로그램 커넥터를 위한 응용프로그램 특정 Business Object에 비해 특별히 설계된 Business Object 세트가 필요하지 않을 수 있습니다.

예를 들어, NameValue Data Handler는 데이터의 각 부분이 데이터를 식별하는 문자열(예: CustomerName=Kumar, Region=NE 및 Department=HR)을 포함할 것을 요구합니다. 모든 일반 Business Object 정의는 해당 이름이 데이터의 각 부분을 식별하는 속성을 포함하기 때문에 JText Connector는 일반 Business Object를 사용할 수 있습니다.

그러나 일반 Business Object는 다수의 상이한 응용프로그램에 필요한 정보의 수퍼세트를 표시하기 때문에 각 일반 Business Object는 대개 하나의 응용프로그램에 필요한 것보다 훨씬 더 많은 정보를 포함하고 있습니다.

그러므로 데이터를 각 응용프로그램에 대한 관리 가능한 크기로 변환하기 위한 좋은 방법은 처리된 데이터의 각 유형에 대해 사용자의 Business Object를 작성하는 것입니다. 해당 Business Object에서 응용프로그램에 필요한 데이터와 Data Handler에 필요한 정보만을 제공하십시오.

예를 들어, FixedWidth Data Handler의 경우 모든 Business Object 속성에 MaxLength 속성 등록 정보에 대해 지정되는 값이 있는지 확인해야 합니다. IBM WebSphere Business Integration Data Handler for XML의 경우, 다른 특정 정보가 필요합니다. 반면 NameValue 및 Delimited Data Handler의 경우 Business Object는 일반 Business Object에 아직 포함되지 않은 어떠한 정보도 포함할 필요가 없습니다. 각 Data Handler에 대한 특정 정보는 *Data Handler* 안내서를 참조하십시오.

Business Object는 데이터 전달 외에, 커넥터가 동적으로 Business Object의 이벤트 파일 이름을 얻거나 출력 파일 이름을 통합 브로커에 리턴할 수 있게 하는 정보를 포

함할 수 있습니다. 이 동적 처리에 맞게 커넥터를 구성하려면, Business Object 레벨의 응용프로그램 특정 정보가 다음 이름-값 쌍 중 하나를 포함해야 합니다.

- `cw_mo_JTextConfig = DynChildMOAttrName`

Business Object가 Data Handler에 의해 사용되는 추가 응용프로그램 특정 정보를 포함하는 경우, 이름-값 쌍은 Business Object에서 첫 번째로 나타나야 하며 세미콜론(;)으로 추가 응용프로그램 특정 정보에서 분리되어야 합니다. 커넥터는 세미콜론까지의 이름-값 쌍을 읽어서 동적 처리 사용 여부를 판별한 후 세미콜론 이후에 나타나는 모든 정보를 Data Handler에 전달합니다.

동적 하위 Meta Object 사용

동적 하위 Meta Object는 파일 이름을 InterChange Server와 교환할 수 있게 합니다. 이 절에서는 다음에 대해 설명합니다.

- 『동적 하위 Meta Object를 사용하는 이유』
- 6 페이지의 『동적 하위 Meta Object를 사용하는 방법』
- 6 페이지의 『동적 하위 Meta Object의 속성』

동적 하위 Meta Object를 사용하는 이유

커넥터가 다음을 수행하게 하려면 동적 하위 Meta Object를 작성하여 사용하십시오.

서비스 호출 요청

- 각 유형의 Business Object(통합 브로커에 의해 하위의 OutFileName 속성에 삽입되는 값을 기본으로 함) 또는 각 개별 Business Object(통합 브로커가 순서를 지정하는 경우)에 대해 동적으로 출력 파일 이름을 생성합니다.

주: 이 커넥터는 하위의 FileWriteMode 속성을 사용하여 하위의 OutFileName 속성에 지정되는 파일을 겹쳐쓸지 또는 해당 파일에 추가할지 여부를 판별합니다.

- 커넥터가 생성하는 각 출력 파일 이름의 이름을 리턴합니다(하위의 OutFileName 속성에 값이 없는 경우). 이 경우, 커넥터는 다음을 수행합니다.
 - 상위 Business Object의 이름에서 이름을 파생시킵니다.
 - 해당 파일에 오브젝트를 기록합니다.
 - 파생된 이름으로 OutFileName Meta Object 속성을 채웁니다.
 - 파생된 이름을 다시 통합 브로커에 전달하며, 통합 브로커는 지정되지 않은 채 동적으로 작성된 출력 파일 이름을 얻습니다.

이벤트 처리

커넥터는 하위의 InFileName 속성을 Business Object를 읽은 파일의 이름으로 채웁니다.

동적 하위 Meta Object를 사용하는 방법

커넥터가 파일 이름을 동적으로 처리하게 하려면 사용자가 다음을 수행해야 합니다.

1. 특정 속성이 있는 동적 하위 Meta Object를 작성하십시오.
2. 데이터 Business Object에서 동적 하위 Meta Object를 표시하는 속성을 추가하십시오.
3. 데이터 Business Object에서 Business Object 레벨의 응용프로그램 특정 정보에 다음을 지정하십시오.

```
cw_mo_JTextConfig = DynChildMOAttrName
```

여기서 *DynChildMOAttrName*은 동적 하위 Business Object를 나타내는 데이터 Business Object에 있는 속성의 이름입니다. 예제에 대한 정보는 그림 2를 참조하십시오.

중요: Data Handler를 사용할 때 *cw_mo_* 접두부가 필수입니다. 접두부가 누락되는 경우, 커넥터는 동적 하위 Meta Object를 데이터 Business Object인 것처럼 지정된 출력 파일에 기록합니다.

4. 동적 하위 Meta Object에서 동적 하위 Meta Object의 속성에 대한 값을 지정하십시오.

동적 하위 Meta Object의 속성

동적 하위 Meta Object는 다음 속성을 포함해야 합니다.

- **FileWriteMode** -- 커넥터가 기존 출력 파일에 추가할지 또는 이를 겹쳐쓸지 여부를 지정하는 값의 문자열 속성. 이 속성의 값은 추가를 의미하는 "a" 또는 겹쳐쓰기를 의미하는 "o"일 수 있습니다. 커넥터는 첫 번째 문자만을 검사하며 값의 대소문자는 고려하지 않습니다.
- **InFileName** -- 이벤트 파일 이름으로 채워지는 문자열 속성(Business Object를 확보하는 파일 및 절대 경로).
- **OutFileName** -- 값에 파일 이름, 절대 경로와 파일 이름 또는 커넥터가 출력 파일에 기록할 때 사용하는 FTP URL이 포함되어 있는 문자열 속성.
 - 이 속성이 파일 이름만을 포함하는 경우, 커넥터는 커넥터가 시작된 디렉토리에 지정된 파일을 기록합니다.
 - 이 속성이 절대 경로와 파일 이름을 포함하는 경우, 커넥터는 지정된 디렉토리에 지정된 파일을 기록합니다.
 - 이 속성이 FTP URL만을 포함하는 경우, 커넥터는 최상위 레벨 JText Meta Object의 EventDir 속성에서 로그인, 암호 및 포트 값을 얻습니다.
 - 이 속성이 로그인, 암호 및 포트 값이 들어 있는 FTP URL을 포함하는 경우, 커넥터는 이 속성에 지정된 값을 사용하여 최상위 레벨 JText Meta Object의 EventDir 속성에 지정된 값을 대체합니다.

자세한 정보는 53 페이지의 『원격 FTP 파일 시스템 지정』을 참조하십시오.

그림 2는 동적 하위 Meta Object를 포함하는 고객 Business Object의 예를 보여줍니다.

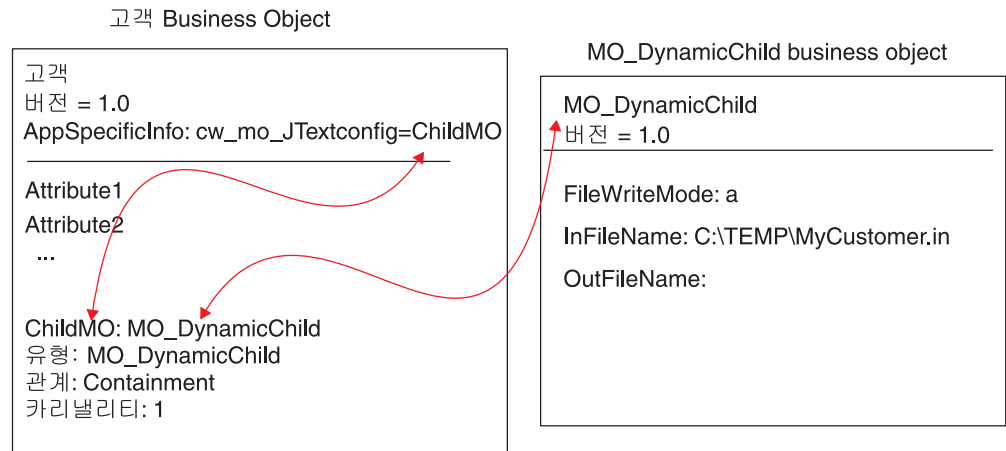


그림 2. 동적 하위 Meta Object의 예

커넥터 작동 방법

JText Connector는 텍스트 또는 2진 파일의 교환을 통해 응용프로그램과 통신합니다. Business Object를 처리할 때 다음 기본 작업을 수행합니다.

- 이벤트 공고
- 요청 처리

이 절에서는 이러한 작업에 대해 설명합니다. 또한 Data Handler 처리가 작동하는 방법과 JText Connector가 Verb를 처리하는 방법도 설명합니다.

이벤트 공고

JText Connector는 다른 커넥터와는 다르게 이벤트를 처리합니다. 타사 응용프로그램에 의존하는 커넥터와는 달리, JText Connector에는 이벤트 테이블이 없습니다. 대신 이벤트 디렉토리를 이벤트 테이블로 취급합니다.

JText Connector가 이벤트를 처리할 때 다음 조작이 발생합니다.

1. 커넥터는 지정된 확장자가 있는 파일에 대해 지정된 디렉토리를 검사하여 이벤트를 폴링합니다. 지정된 디렉토리에 지정된 확장자가 있는 파일의 존재는 이벤트와 동등하게 간주됩니다. 커넥터는 해석하지 않고 이벤트 디렉토리에서 이벤트 파일을 직접 읽습니다. 구문 분석 방법을 사용하여 각 Business Object를 나타내는 서브섹션을 판별합니다. 자세한 정보는 49 페이지의 『특정 Business Object 폴링』을 참조하십시오.
2. 커넥터는 Data Handler의 인스턴스(데이터 Business Object에 대한 JText Meta Object에 지정되는 값을 기본으로 함)를 작성합니다.
3. 커넥터는 Data Handler 인스턴스에서 getB0() 또는 getB0ByteArray()를 호출하고, Business Object를 나타내는 문자열 또는 바이트 배열을 전송합니다. 커넥터는 Business Object를 표시하는 각 요소를 Data Handler로 전달합니다. 파일이 복수 Business Object를 표시하는 경우, 커넥터는 전체 파일이 아닌 한 요소(즉, 단일 Business Object의 문자열 또는 바이트 배열)만을 전송합니다.
4. Data Handler가 문자열 또는 바이트 배열을 Business Object로 변환하고 커넥터에 리턴합니다. Data Handler는 또한 오류를 보고하고 추적을 제공합니다.
5. Data Handler가 기본 Verb 처리를 수행합니다. 커넥터가 논리를 제공하지 않기 때문에 Data Handler를 개발하는 사람은 Verb 설정을 위한 논리를 제공해야 합니다. 샘플 데이터 핸들러가 Verb를 설정하는 방법에 대한 정보는 *Data Handler* 안내서를 참조하십시오.
6. Data Handler에서 Business Object를 작성하지 못하도록 금지하는 오류가 발생하는 경우, 커넥터는 .fail 확장자로 해당 문자열 또는 바이트 배열을 아카이브합니다. Data Handler가 성공하는 경우, 커넥터는 Business Object에 대한 등록을 검사합니다.
 - 커넥터가 Business Object에 등록하지 않은 경우, 커넥터는 확장자가 .unsub인 아카이브 파일에 기록합니다.
 - 커넥터가 Business Object에 등록하는 경우, Business Object를 통합 브로커에 전송합니다.
7. 커넥터가 성공적으로 통합 브로커에 Business Object를 전송하는 경우, 이벤트 파일의 임의의 Business Object가 처리에 실패했는지 여부에 따라서 확장자가 .success 또는 .partial인 파일을 아카이브합니다. 커넥터가 Business Object를 전송하지 못하는 경우 확장자가 .fail인 파일을 아카이브합니다.

구성에 따라서 JText Connector는 이벤트 디렉토리의 모든 파일을 선택하거나 지정된 확장자가 있는 파일만을 선택할 수 있습니다. 자세한 정보는 49 페이지의 『복수 이벤트 파일 또는 복수 이벤트 디렉토리 지정』을 참조하십시오.

JText Connector는 파일의 위치와 상관없이 가장 오래된 것부터 가장 최신 것까지 시간 소인의 순서대로 이벤트 파일을 처리합니다. 즉, JText Connector는 시간 소인의 연순으로 개별 디렉토리에 있는 파일을 처리합니다.

PollQuantity 등록 정보는 커넥터가 제공된 폴에서 통합 브로커에 게시할 수 있는 Business Object의 최대수를 설정합니다. 예를 들어, PollQuantity의 값이 5로 설정되고 커넥터가 폴링 중인 디렉토리에 두 개의 파일이 있다고 가정하십시오. 첫 번째 파일에는 네 개의 Business Object가 있고, 두 번째 파일에는 12개의 Business Object가 있습니다. 첫 번째 폴 호출에서 커넥터는 다음 단계를 수행합니다.

1. 첫 번째 파일에서 네 개의 모든 Business Object를 전송하여 각 Business Object를 처리할 때 아카이브합니다.
2. 두 번째 파일에서 첫 번째 Business Object를 전송합니다.

두 번째 폴 호출 시 커넥터는 두 번째 파일에서 2 - 6번째 Business Object를 전송합니다. 세 번째 폴 호출 시 커넥터는 두 번째 파일에서 7 - 11번째 Business Object를 전송합니다. 네 번째 호출 시 커넥터는 마지막 Business Object를 전송합니다. 커넥터는 각 Business Object를 처리한 후 아카이브합니다. 파일에 있는 임의의 Business Object가 처리에 실패하는 경우, 커넥터는 확장자가 .orig인 전체 파일을 아카이브합니다.

자세한 정보는 다음을 참조하십시오.

- PollQuantity 등록 정보를 사용한 성능 조정에 대해서는 64 페이지의 『JText Connector의 성능 조정』을 참조하십시오.
- 이벤트 디렉토리 및 확장자 지정에 대해서는 42 페이지의 『이벤트 디렉토리 및 확장자 지정』을 참조하십시오.
- 이벤트 처리 지정에 대해서는 41 페이지의 『이벤트 공고 지정』을 참조하십시오.

그림 3은 이벤트 공고 조작을 보여줍니다(그래픽의 숫자는 위에서 약술한 단계와 관련이 없음).

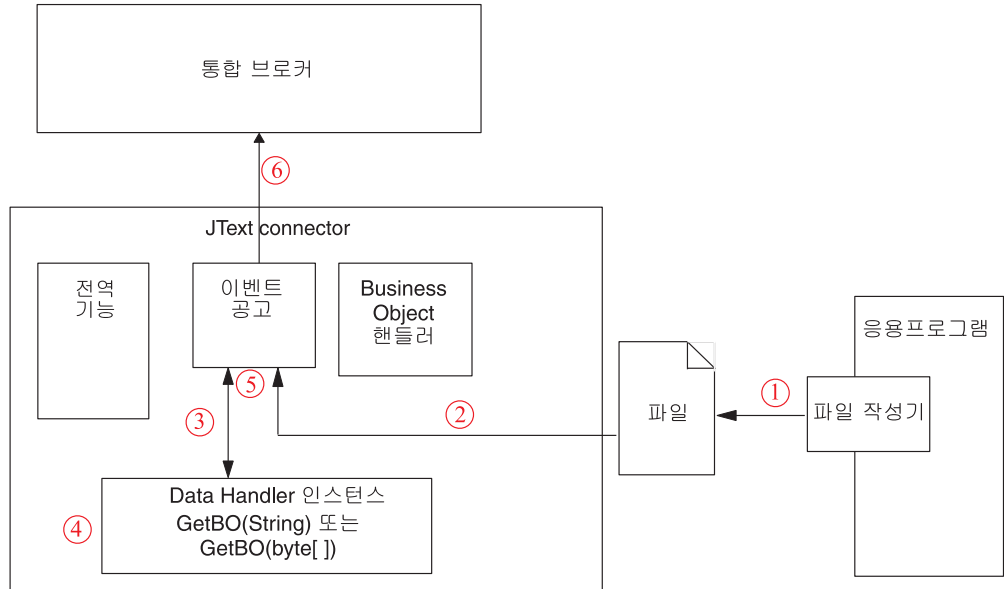


그림 3. 이벤트 광고 조작

이벤트 아카이브

이벤트를 처리한 후 아카이브가 가능하도록 구성되는 경우, JText Connector는 Business Object 문자열 또는 한 Business Object의 2진 배열 표시를 로컬 아카이브 디렉토리의 파일에 기록합니다. 커넥터는 파일을 `밀줄()`, 시간 소인 및 이벤트 상태에 대응하는 파일 확장자로 이름 지정합니다. 제공되는 기본 확장자는 `success`, `partial`, `unsub`, `orig` 및 `fail`입니다. 밀줄 및 시간 소인은 파일 이름 뒤와 파일 확장자 앞에 추가됩니다.

시간 소인은 시스템 시간의 연도, 월, 일, 시, 분, 초 및 밀리초를 포함하는 밀줄로 구분된 목록입니다. 이것은 아카이브된 파일 이름이 고유하고 커넥터에 동일한 이름이 있는 기존 파일을 겹쳐쓰지 않도록 합니다. 아카이브된 파일의 형식은 다음과 같습니다.

`BOName_YYYY_MM_DD_HH_MM_SS_sss.[extension]`

예를 들어, 커넥터는 성공적으로 처리된 `Customer.in` 파일의 이름을 `Customer_2003_11_15_18_24_59_999.success`로 바꿀 수 있습니다.

JText Connector는 형식화 오류가 발생하는 경우 또는 커넥터가 통합 브로커에 Business Object를 전송하지 못하는 경우 Business Object를 `.fail` 파일에 아카이브합니다. 커넥터가 등록하지 않은 경우 JText Connector가 Business Object를 확장자가 `.unsub`인 파일에 아카이브합니다. 이들 아카이브 파일의 시험 및 모든 형식화 오류의 정정 또는 Business Object에 등록된 프로세스의 시작 후, 처리를 위해 이들 파일의 Business Object를 다시 제출합니다.

아카이브에 대한 자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오.

이벤트 및 아카이브 파일의 기본 파일 확장자

JText Connector가 이벤트 및 아카이브 테이블을 사용하지 않기 때문에 파일 확장자를 변경하여 이벤트 상태를 갱신합니다. 표 1에서는 IBM WebSphere Business Integration Adapter for JText가 이벤트 및 아카이브 파일에 대해 전달하는 기본 파일 확장자 값을 보여줍니다.

표 1. 기본 파일 확장자

파일 유형	이벤트 상태/설명	기본 파일 확장자	제공되는 기본 디렉토리
이벤트	새로 작성	in	UNIX:/tmp/JTextConn/Default/Event Windows:C:\temp\JTextConn\Default\Event
아카이브	성공(이벤트 파일의 모든 Business Object가 성공적으로 처리되는 경우, 이 파일은 모든 Business Object를 포함합니다.)	success	UNIX:/tmp/JTextConn/Default/Archive Windows:C:\temp\JTextConn\Default\Archive
아카이브	성공(이벤트 파일의 일부 Business Object가 처리에 실패하는 경우, 이 파일은 성공적으로 처리된 Business Object만을 포함합니다.)	partial	UNIX:/tmp/JTextConn/Default/Archive Windows: C:\temp\JTextConn\Default\Archive
아카이브	등록되지 않음	unsub	UNIX:/tmp/JTextConn/Default/Archive Windows: C:\temp\JTextConn\Default\Archive
아카이브	전체 원래 이벤트 파일(이벤트 파일이 단 하나의 Business Object만을 포함하는 경우에도 임의의 Business Object가 처리에 실패하거나 등록되지 않은 경우에만 작성됩니다.)	orig	UNIX:/tmp/JTextConn/Default/Archive Windows:C:\temp\JTextConn\Default\Archive
아카이브	실패	실패	UNIX:/tmp/JTextConn/Default/Archive Windows:C:\temp\JTextConn\Default\Archive
출력	out	out	UNIX: /tmp/JTextConn/Default/Out Windows: C:\temp\JTextConn\Default\Out

중요: 동시에 동일한 파일에 액세스하고 처리하는 복수 응용프로그램 사이의 액세스 순서가 중요합니다. 제공된 파일에 대해 수행되는 모든 조작을 분석하여 파일 잠금 및 불완전한 데이터의 문제를 회피합니다.

주: 커넥터는 지정된 확장자를 갖는 이벤트 디렉토리의 모든 파일을 입력 파일로 취급합니다. 커넥터가 아카이브된 파일을 이벤트로 취급하는 것을 막으려면 입력 파일 확장자가 아카이브 파일 확장자와 다른지 또는 입력 파일과 아카이브 파일이 서로 다른 디렉토리에 저장되는지 확인하십시오.

사용자 고유 파일 확장자, 이벤트 디렉토리 및 출력 디렉토리 지정에 대한 정보는 34 페이지의 표 5를 참조하십시오.

요청 처리

서비스 호출 요청 처리 시, 커넥터는 Business Object를 출력 문자열 또는 2진 배열로 변환한 후 파일에 기록합니다.

그러나 커넥터는 Business Object를 변환하기 전에 Business Object가 동적 파일 이름 지정에 대해 구성되었는지 즉, Business Object가 동적 하위 Meta Object를 포함하는지 여부를 판별합니다. 이런 경우, 커넥터는 동적으로 출력 파일의 이름을 지정하거나 생성하는 출력 파일의 이름을 리턴합니다.

이 절에서는 다음 경우에 서비스 호출 요청 처리에 대해 설명합니다.

- 『데이터 Business Object가 동적 파일 이름을 지정하지 않음』
- 12 페이지의 『데이터 Business Object가 동적 하위 Meta Object를 포함』

데이터 Business Object가 동적 파일 이름을 지정하지 않음

데이터 Business Object가 동적 파일 이름을 지정하지 않을 때 커넥터는 다음 조작을 수행하여 서비스 호출 요청을 처리합니다.

1. 커넥터가 Business Object 요청을 수신합니다.
2. 커넥터가 Business Object 레벨의 AppSpecificInfo 등록 정보에 다음 내용이 포함되어야 하는지 판별합니다.

```
cw_mo_JTextConfig = DynChildMOAttrName
```

3. 커넥터가 최상위 레벨 JText Meta Object의 구성을 검사하여 호출할 Data Handler를 판별합니다. 기본적으로, 이 Meta Object는

```
MO_DataHandler_DefaultNameValueConfig Data Handler Meta Object를 지정하며, 이는 NameValue Data Handler를 표시합니다.
```

4. 커넥터가 적당한 Data Handler의 인스턴스를 작성하고 해당 인스턴스에 Business Object를 전송합니다.
5. Data Handler가 Business Object를 문자열 또는 바이트 배열로 변환하며, 이는 구성으로 리턴됩니다. Data Handler는 또한 오류를 보고하고 추적을 제공합니다.
6. 커넥터가 파일에 문자열 또는 바이트 배열을 기록합니다.

요청을 처리하기 위한 커넥터 구성에 대한 정보는 45 페이지의 『요청 처리 지정』을 참조하십시오.

데이터 Business Object가 동적 하위 Meta Object를 포함

데이터 Business Object가 동적 하위 Meta Object를 포함할 때 커넥터는 다음 조작을 수행하여 서비스 호출 요청을 처리합니다.

1. 커넥터가 Business Object 요청을 수신합니다.
2. 커넥터가 Business Object 레벨의 AppSpecificInfo 등록 정보가 다음 텍스트를 포함하는지를 판별합니다.

`cw_mo_JTextConfig = DynChildMOAttrName`

주: Business Object의 응용프로그램 특정 정보가 동적 하위 Meta Object를 지정하지 않고 이러한 하위를 포함하지 않는 경우, 커넥터는 12 페이지의 『데이터 Business Object가 동적 파일 이름을 지정하지 않음』에 설명하는 대로 Business Object를 처리합니다.

3. 커넥터가 동적 하위 Meta Object의 OutFileName 속성에서 출력 파일의 이름을 확보합니다.

- 이 속성이 값을 포함하는 경우, 커넥터는 해당 이름의 파일이 이미 존재하는지 검사합니다. 파일이 존재하지 않는 경우, 커넥터는 새 출력 파일을 작성하고 속성 값을 사용하여 파일 이름을 지정합니다. 해당 파일이 이미 존재하는 경우, 커넥터는 기존 파일에 추가하거나 기존 파일을 겹쳐씁니다(하위 Meta Object의 FileMode 값을 기본으로 함).

중요: FileMode 속성 값이 "o"가 아닌 임의의 값으로 시작되는 경우, 커넥터는 추가 모드로 기본 설정됩니다.

- 이 속성이 값을 포함하지 않는 경우(즉, OutFileName=CxIgnore), 커넥터는 이 하위 Meta Object를 포함하는 상위 Business Object의 이름에서 파일 이름을 파생시키고 최상위 레벨 JText Meta Object의 구성을 사용하여 출력 파일의 위치를 판별합니다. 커넥터는 파일에 Business Object를 기록한 후 이 속성에 있는 파일의 이름 및 경로를 리턴합니다.

4. 커넥터가 최상위 레벨 JText Meta Object의 구성을 검사하여 호출할 Data Handler를 판별합니다. 기본적으로, 이 Meta Object는

`MO_DataHandler_DefaultNameValueConfig Data Handler Meta Object`를 지정하며, 이는 NameValue Data Handler를 표시합니다.

5. 커넥터가 적당한 Data Handler의 인스턴스를 작성하고 해당 인스턴스에 Business Object를 전송합니다.

6. Data Handler가 Business Object를 문자열 또는 바이트 배열로 변환하며, 이는 구성으로 리턴됩니다. Data Handler는 또한 오류를 보고하고 추적을 제공합니다.

7. 커넥터가 위의 3단계에서 이름을 파생시킨 파일로 문자열 또는 바이트 배열을 기록합니다.

그림 4는 커넥터가 통합 브로커의 요청을 목적지 응용프로그램에 처리할 때의 JText Connector 구성요소를 보여줍니다.

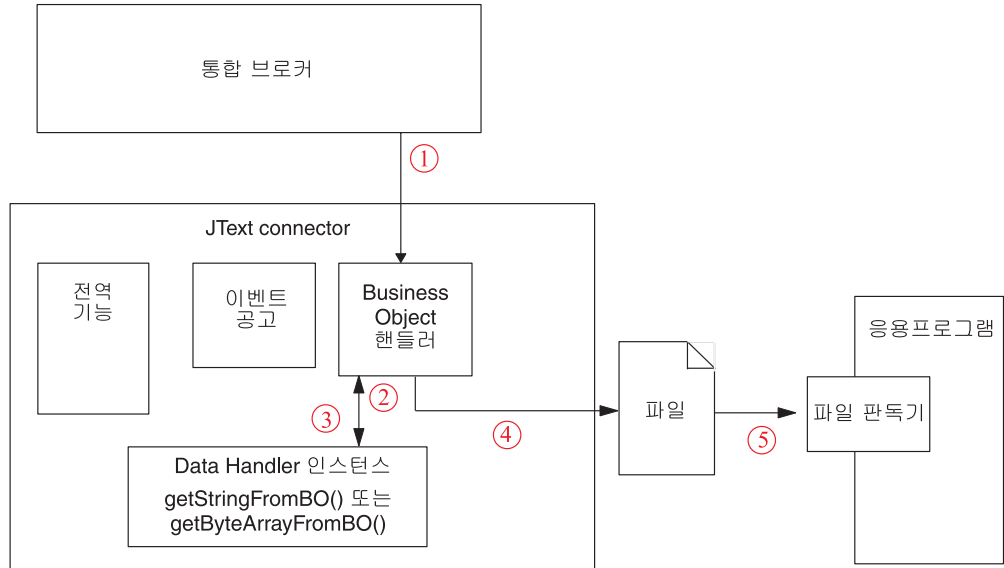


그림 4. Business Object 요청 조작

Data Handler 처리가 작동하는 방법

커넥터는 Data Handler 인스턴스를 사용하여 Business Object와 이벤트 파일에서 읽은 문자열 또는 바이트 배열 사이를 변환합니다. Data Handler는 또한 오류를 보고하고 추적을 제공합니다.

커넥터는 최상위 레벨 JText Meta Object에 있는 EventHandler 및 OutputDataHandler 속성값을 기본으로 Data Handler의 인스턴스를 작성합니다. 이들 속성은 커넥터가 Data Handler의 인스턴스를 작성하는 데 사용하는 Data Handler Meta Object를 식별합니다. Data Handler Meta Object는 전달된 Data Handler 또는 사용자 정의 Data Handler를 표시할 수 있습니다. 자세한 정보는 *Data Handler 안내서*를 참조하십시오.

커넥터는 DataProcessingMode Meta Object의 설정을 기본으로 인터페이스, 문자열 또는 바이트 배열을 결정합니다. 이 Meta Object에 대한 추가 정보는 34 페이지의 표 5를 참조하십시오.

구성 정보를 수신한 후 커넥터는 다음 단계를 수행합니다.

1. Data Handler를 인스턴스화합니다.
2. Data Handler의 setOption() 메소드를 호출하여 Data Handler의 TracingSubSystem 속성을 커넥터 이름으로 설정합니다. Data Handler는 이 값을 사용하여 Data Handler가 작성하는 추적 메시지에 커넥터의 이름을 포함시킵니다.

Data Handler가 작성되고 구성된 후 커넥터는 Data Handler에서 적당한 메소드를 호출하여 Business Object와 데이터 사이의 변환을 수행합니다.

- 이벤트 공고의 경우, 커넥터는 Data Handler에서 `getB0(String)` 또는 `getB0(byte[])` 메소드를 호출합니다. 커넥터는 Data Handler에 Business Object로 변환할 파일의 문자열을 전달합니다. Data Handler가 Business Object를 리턴합니다.
- 요청 처리의 경우, 커넥터는 Data Handler에서 `getStringFromB0()` 또는 `getByteArrayFromB0()` 메소드를 호출합니다. 커넥터는 Data Handler에 문자열 또는 바이트 배열로 변환할 Business Object를 전달합니다. Data Handler는 차례로 직렬화된 Business Object 버전을 문자열 또는 바이트 배열 양식으로 리턴합니다.

`getB0(String)` 또는 `getB0(byte[])` 및 `getStringFromB0()` 또는 `getByteArrayFromB0()` 메소드는 각각 항상 최상위 레벨 상위의 전체 Business Object 계층 구조와 모든 하위 Business Object를 전송 또는 수신합니다.

어느 경우에도 Data Handler는 모든 Meta Object 데이터를 필터링하여 Business Object 특정 데이터만을 전달하는 기능을 담당합니다. 제품이 제공하는 Data Handler는 이 기능을 제공합니다. 사용자 정의 Data Handler를 사용하는 경우에도 이 기능을 제공해야 합니다.

요청에 대한 Business Object Verb 처리

요청을 처리할 때 JText Connector는 한 Verb를 다른 Verb와 동일하게 처리합니다. 커넥터는 Business Object와 연관된 Verb와 상관없이 갱신, 검색 또는 삭제 조작을 수행하지 않고 파일에 기록합니다.

요청을 처리할 때 다음 조건이 해당될 경우 JText Connector는 `CxIgnore`의 값이 있는 모든 속성을 해당 기본값으로 설정합니다.

- Verb는 Create입니다.
- 커넥터의 UseDefaults 등록 정보가 true로 설정되었습니다.
- 속성이 Required로 설정되었습니다.
- 기본값이 Business Object 스펙의 속성에 대해 설정되었습니다.

커넥터 기능

이벤트 공고 및 Business Object 요청 처리와 함께 JText Connector는 다음 기능을 제공합니다.

- 여러 Business Object에 대한 다양한 구성. 예를 들어, 여러 디렉토리 및 파일 확장자 또는 상이한 데이터 형식을 사용하기 위해 상이한 Business Object를 구성할 수 있습니다.

- 파일 확장자, 아카이브 파일 저장을 위한 디렉토리 위치, 형식 유형 및 파일 순서 지정에 대한 구성 기능
- 동적으로 각 Business Object에 대한 출력 파일 이름 판별 또는 생성된 출력 파일의 전체 이름 리턴을 위한 구성 기능
- 장애 복구
- 사용자 정의 Data Handler 기능. 이것은 사용자가 커넥터 코드의 재컴파일 없이 사용자의 Data Handler를 작성할 수 있음을 의미합니다. 사용자가 작성한 새 클래스를 사용하려면 단지 구성 등록 정보를 변경하면 됩니다.
- 로컬 파일 시스템 디렉토리와 마찬가지로 원격 FTP 위치로 데이터를 교환하는 기능

자세한 정보는 19 페이지의 제 2 장 『JText Adapter 설치 및 구성』, 29 페이지의 제 3 장 『JText Connector Meta Object 사용』 및 *Data Handler* 안내서를 참조하십시오.

JText Adapter와 다른 어댑터의 차이점

JText Connector는 소스 응용프로그램에서 다른 어댑터와 같이 목적지 응용프로그램으로의 데이터 전송을 가능하게 하는 반면, 다음 방법에 있어 고유합니다.

- 모든 Business Object를 동일한 방법으로 처리합니다. 즉, 항상 Business Object를 파일에 기록하므로 Create 조작만(수신 Verb와 관계없이) 수행합니다.
- 처리하는 Business Object의 내용을 해석하지 않습니다. 즉, 키 값이 다른 텍스트보다 중요하지 않은 잠재적인 문자열 또는 바이트 배열로서 각 Business Object를 읽습니다.
- 많은 구성을 위해 Meta Object를 사용합니다. 자세한 정보는 29 페이지의 제 3 장 『JText Connector Meta Object 사용』을 참조하십시오.
- 이벤트 테이블이 없습니다. 대신 구성된 이벤트 디렉토리를 이벤트 테이블로 취급합니다.

로케일 종속 데이터 처리

2바이트 문자 세트를 지원하고 지정된 언어로 메시지 텍스트를 전달할 수 있도록 커넥터가 국제화되었습니다. 커넥터가 한 개의 문자 코드 세트를 사용하는 위치에서 다른 코드 세트를 사용하는 위치로 데이터를 전송할 때, 데이터의 의미를 보존하기 위해 문자 변환을 수행합니다. JVM(Java Virtual Machine) 내의 Java 런타임 환경은 Unicode 문자 코드 세트를 나타냅니다. Unicode는 가장 잘 알려진 문자 코드 세트(1바이트 및 다중 바이트)의 문자에 대한 인코딩을 포함합니다. WebSphere Business Integration System에 있는 대부분의 구성요소는 Java로 작성됩니다. 그러므로 데이터가 대부분의 WebSphere Business Integration System 구성요소 사이에서 전송되면, 해당 데이터에는 문자 변환이 필요하지 않습니다. 해당 언어 및 해당 국가 또는 지역에 대한 오류

와 정보 메시지를 기록하려면, 사용자의 환경에 대한 로케일 표준 구성 등록 정보를 구성하십시오. 이들 등록 정보에 대한 자세한 정보는 81 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

제 2 장 JText Adapter 설치 및 구성

이 장에서는 JText Connector를 설치 및 구성하는 방법에 대해 설명합니다.

- 『호환성』
- 20 페이지의 『전제조건』
- 20 페이지의 『JText Adapter 설치』
- 20 페이지의 『커넥터 구성』
- 24 페이지의 『다중 커넥터 인스턴스 작성』
- 25 페이지의 『지원되는 Business Object 추가』
- 26 페이지의 『커넥터 시작』
- 27 페이지의 『커넥터 중지』

호환성

어댑터에 사용되는 어댑터 프레임워크는 어댑터가 통신하고 있는 통합 브로커의 버전과 호환되어야 합니다. 다음 통합 브로커와 함께 어댑터 프레임워크의 다음과 같은 버전에서 Adapter for JText 버전 5.4가 지원됩니다.

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

통합 브로커:

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

예외사항은 릴리스 정보를 참조하십시오.

주: 통합 브로커 설치와 전제조건에 대한 지시사항은 다음 문서를 참조하십시오. WebSphere InterChange Server(ICS)에 대해서는 UNIX 또는 Windows용 시스템 설치 안내서를 참조하십시오.

메시지 브로커(WebSphere MQ Integrator Broker, WebSphere MQ Integrator 및 WebSphere Business Integration Message Broker)에 대해서는 WebSphere

Message Brokers로 어댑터 구현 및 메시지 브로커에 대한 설치 문서를 참조하십시오. 이 중 일부는 웹 사이트 <http://www.ibm.com/software/integration/mqfamily/library/manualsa/>에서 찾을 수 있습니다.

WebSphere Application Server에 대해서는 WebSphere Application Server로 어댑터 구현 및 <http://www.ibm.com/software/webservers/appserv/library.html>의 문서를 참조하십시오.

전제조건

JText Connector를 실행하기 전에 커넥터가 읽고 쓰는 텍스트 파일을 포함하는 아카이브 디렉토리, 출력 및 이벤트에 대한 읽기/쓰기 권한을 작성하십시오. 이 작업은 로컬 서버 및 원격 서버 모두에서 수행해야 합니다.

JText Adapter 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>

커넥터 구성

Connector Configurator에서 커넥터 등록 정보를 구성합니다. 통합 브로커와 상관없이 커넥터가 Business Object마다 서로 다르게 처리할 수 있게 하는 Meta Object를 구성합니다.

주: 병렬 프로세스 등급 자원을 1보다 큰 값으로 설정하여 JText Connector를 실행하는 것은 바람직하지 않습니다. 병렬 프로세스 등급에 대한 자세한 정보는 *System Administration Guide*를 참조하십시오.

커넥터 등록 정보 구성

커넥터는 시작할 때 구성값을 얻습니다. 런타임 세션 중에 하나 이상의 커넥터 등록 정보 값을 변경하기 원할 수 있습니다. 커넥터 등록 정보에 대한 변경사항은 다음일 수 있습니다.

- 동적 -- 이 변경사항은 변경 수행 직후에 적용됩니다.
- 정적 -- 이 변경사항은 커넥터 구성요소 다시 시작 또는 시스템 다시 시작 중 하나를 수행해야 적용됩니다.

등록 정보가 동적인지 또는 정적인지를 판별하려면 통합 브로커에 대한 구성 유틸리티를 참조하십시오.

커넥터는 다음 두 유형의 구성 등록 정보를 갖습니다.

- 표준 구성 등록 정보
- 커넥터 특정 구성 등록 정보

커넥터를 실행하기 전에 이러한 등록 정보 중 일부의 값을 설정해야 합니다.

Business Object 처리 구성

Meta Object를 사용하여 커넥터 작동의 다음 측면을 구성합니다.

- 사용할 Data Handler
- JText 프로세스의 모드(2진 또는 텍스트)
- 이벤트 파일을 폴링할 디렉토리
- 이벤트 파일에 사용할 파일 확장자
- 아카이브하는 경우, 파일 아카이브에 사용할 디렉토리
- 아카이브하는 경우, 처리된 파일을 아카이브할 때 사용할 파일 확장자
- 다른 디렉토리의 다른 오브젝트에서 이벤트를 선택하는지 또는 동일한 이벤트 디렉토리에서 복수 파일을 선택하는지 여부

Meta Object에 대한 정보는 29 페이지의 제 3 장 『JText Connector Meta Object 사용』을 참조하십시오.

표준 구성 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 등록 정보의 문서에 대해서는 81 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

표 2에서는 부록 A의 구성 등록 정보에 대해 이 커넥터에 고유한 정보를 제공합니다.

표 2. 이 커넥터에 고유한 등록 정보

등록 정보	참고
CharacterEncoding	이 커넥터가 Java 기반이기 때문에 이 등록 정보를 사용하지 않습니다.
Locale	커넥터가 국제화되었으므로, 이 등록 정보의 값을 변경할 수 있습니다. 현재 지원되는 로케일을 판별하려면 커넥터에 대한 릴리스 정보를 참조하십시오.

커넥터 특정 등록 정보

커넥터 특정 구성 등록 정보는 런타임 시 커넥터에 필요한 정보를 제공합니다. 또한 커넥터를 재코드화 및 재빌드하지 않으면서 커넥터 내의 논리 또는 정적 정보의 변경 방법을 제공합니다.

22 페이지의 표 3에는 커넥터에 대한 커넥터 특정 구성 등록 정보가 나열되어 있습니다. 등록 정보에 대한 설명은 다음 절을 참조하십시오.

표 3. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수 여부
ArchivingEnabled	true 또는 false	true	예
EventLog	파일의 이름 및 위치	event.log	아니오
EventRecovery	abort 또는 retry	retry	예
FTPPollFrequency	폴 주기의 수		아니오
GenerateTemplate	<i>BOName</i>		아니오
OutputLog	요청 처리 중에 각 수신 <i>Business Object</i> 에 대한 다음 순번을 등록하는 파일	Output.Log	아니오
PollQuantity	각 폴에서 처리되는 이벤트 수	25	아니오

ArchivingEnabled

아카이브를 사용합니다. 이 등록 정보가 true로 설정되는 경우, 이벤트 파일은 지정된 확장자를 사용하여 아카이브 디렉토리에 아카이브됩니다. 이 등록 정보가 false로 설정되는 경우, 이벤트 파일은 아카이브되지 않습니다. 이 경우에 커넥터는 모든 이벤트를 통합 브로커에 전송한 후 파일을 삭제합니다. 자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오.

기본값은 true입니다.

EventLog

커넥터가 생성하는 이벤트에 대한 파일 저장 위치를 제공합니다. 이 파일은 제품이 설치되는 connectors 디렉토리의 JText 서브디렉토리에 있습니다.

기본값은 event.log입니다.

EventRecovery

복구 작동을 지정합니다. 이 등록 정보가 retry로 설정되는 경우, 커넥터는 event.log 파일을 사용하여 실패한 이벤트를 복구합니다. 이 등록 정보가 abort로 설정되는 경우, 커넥터는 실패한 이벤트가 발생할 때 종료합니다. 자세한 정보는 69 페이지의 『이벤트 로그 파일』을 참조하십시오.

기본값은 retry입니다.

FTPPollFrequency

커넥터가 FTP 서버를 폴링하는 빈도를 표준 폴 주기의 수로 판별합니다. 예를 들어, PollFrequency가 10000으로 설정되고 FTPPollFrequency가 6으로 설정되는 경우, 커넥터는 10초마다 로컬 이벤트 디렉토리를 폴링하고 60초마다 원격 디렉토리를 폴링합니다. 커넥터는 사용자가 이 등록 정보에 대해 값을 지정하는 경우에만 FTP 폴링을 수행합니다. FTPPollFrequency가 0 또는 공백으로 평가되는 경우, 커넥터는 FTP 폴링을 수행하지 않습니다. 기본적으로 커넥터가 폴링을 수행하지 않습니다.

이 등록 정보에 대한 기본값은 없습니다.

GenerateTemplate

커넥터 시작 후 지원되는 각 Business Object에 대한 템플릿을 커넥터가 생성할 수 있도록 합니다. 이 등록 정보에 대한 구문은 *BOName;BOName*이며, 여기서 특정 Business Object의 이름이 *BOName*으로 대체됩니다. 예를 들어, 고객 Business Object 및 항목 Business Object에 대해 각각 하나씩, 두 개의 템플릿을 생성하려면 *Customer;Item*을 지정하십시오. 자세한 정보는 65 페이지의 『테스트를 위한 샘플 Business Object 생성』을 참조하십시오.

이 등록 정보에 대한 기본값은 없습니다.

OutputLog

커넥터가 요청 처리 중에 Business Object 유형마다 고유한 출력 파일을 작성하는 데 사용하는 순번을 저장하는 파일 이름을 지정합니다. 파일의 형식은 다음과 같습니다.

BusinessObjectName = *NextSequenceNumber*

여기서 *BusinessObjectName*은 요청 Business Object의 이름이며, *NextSequenceNumber*는 가장 최근에 수신된 Business Object의 순번으로서 하나씩 증가됩니다. 예를 들어, 커넥터가 고객 및 항목 Business Object를 처리 중인 경우, 출력 로그 파일은 다음을 포함할 수 있습니다.

```
Customer = 12  
Item = 2
```

이 파일은 커넥터가 11개의 고객 및 1개의 항목을 이미 처리했음을 나타냅니다. 다음 고객 및 항목 Business Object는 각각 *Customer_12.out* 및 *Item_2.out* 파일에 기록됩니다. 주문 Business Object 요청을 수신하면 커넥터는 새 행을 출력 로그 파일에 추가하고 Business Object를 *Order_1.out* 파일에 기록합니다.

*FileSeqEnabled*가 true로 설정되는 경우, 커넥터는 이 순번을 사용하여 각 Business Object에 대해 작성하는 출력 파일 이름을 고유하게 지정합니다. 커넥터는 Business Object의 이름 또는 *OutputFileName* Meta Object 속성에 이름이 지정되는 파일에 *밀줄()* 및 순번을 추가하여 각 출력 파일의 이름을 지정합니다. 출력 로그가 사용자가 읽을 수 있는 형식으로 저장되기 때문에 표준 문서 편집기를 사용하여 파일을 읽거나 그 값을 재설정할 수 있습니다.

OutputFileName 속성에 대한 자세한 정보는 39 페이지의 『출력 파일의 이름 지정』을 참조하십시오. 출력 로그에 대한 자세한 정보는 45 페이지의 『요청 처리 지정』을 참조하십시오. 생성된 파일 이름 리턴에 대한 정보는 46 페이지의 『파일 이름 리턴』을 참조하십시오.

기본값은 *Output.Log*입니다.

PollQuantity

각 폴에 대해 처리할 이벤트 수를 지정합니다. 커넥터 폴 메소드는 이벤트 레코드의 지정된 수를 검색하며 단일 폴에서 이를 처리합니다. 폴마다 복수 이벤트를 처리하면 응용프로그램이 다수의 이벤트를 생성할 때 성능을 향상시킬 수 있습니다. 그러나 폴 메소드가 이벤트를 처리하는 동안 통합 브로커 요청이 차단되기 때문에 이벤트 수를 너무 높게 설정하지 마십시오. 각 폴 호출이 긴 시간을 소요하는 경우, 통합 브로커 요청 조작을 지연시킵니다. 자세한 정보는 64 페이지의 『JText Connector의 성능 조정』을 참조하십시오.

기본값은 25입니다.

다중 커넥터 인스턴스 작성

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

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

새 디렉토리 작성

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

```
ProductDir\connectors\connectorInstance
```

여기서 `connectorInstance`는 고유하게 커넥터 인스턴스를 식별합니다.

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

```
ProductDir\repository\connectorInstance
```

Business Object 정의 작성

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

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

2. 초기 커넥터용 파일은 다음 디렉토리에 있어야 합니다.

ProductDir\repository\initialConnectorInstance

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

커넥터 정의 작성

Connector Configurator에서 커넥터 인스턴스의 구성 파일(커넥터 파일)을 작성합니다. 이를 수행하려면 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 24 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.

지원되는 Business Object 추가

기본적으로, JText Connector는 MO_JTextConnector_Default Meta Object를 지원 합니다. 커넥터를 완전히 구성하려면 Connector Configurator를 사용하여 지원되는 Business Object 목록에 다른 필수 Business Object를 추가하십시오. 커넥터를 사용하는 방법에 따라서 다음 Business Object의 전부 또는 다수를 추가할 필요가 있습니다.

- Data Handler에 대한 Meta Object(MO_JTextConnector_Default Meta Object의 EventDataHandler 및 OutputDataHandler 속성에 지정). 기본적으로, 이들 속성은 NameValue Meta Object를 표시하는

MO_DataHandler_DefaultNameValueConfig Data Handler Meta Object를 지정합니다. 자세한 정보는 61 페이지의 『Data Handler 지정』을 참조하십시오.

- MO_JTextConnector_BusObjName -- 특정 Business Object에 대한 Meta Object를 작성하는 경우. 자세한 정보는 61 페이지의 『특정 Business Object에 대한 JText Meta Object 작성』을 참조하십시오.
- 파일에서 읽거나 파일에 기록할 Business Object 자세한 정보는 4 페이지의 『JText Connector가 사용하는 Business Object』를 참조하십시오.

커넥터 시작

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

ProductDir\connectors\connName

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

표 4. 커넥터의 시작 스크립트

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

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

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

- 명령행에서 커넥터를 시작할 경우에는 다음을 수행하십시오.

- Windows 시스템의 경우:

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

- UNIX 기반 시스템의 경우:

```
connector_manager_connName -start
```

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

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

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

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

- System Manager를 시작할 때 실행되는 Adaptor Monitor(WebSphere Business Integration Adapters 제품 전용)에서 다음을 수행할 수 있습니다.
이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.
- System Monitor(WebSphere InterChange Server 제품 전용)에서 다음을 수행할 수 있습니다.
이 도구를 사용하여 커넥터를 로드, 활성화, 비활성화, 일시정지, 종료 또는 삭제할 수 있습니다.
- Windows 시스템에서 Windows 서비스로 시작하도록 커넥터를 구성할 수 있습니다.
이 경우, Windows 시스템이 시동하거나(자동 서비스인 경우) Windows 서비스를 통해 서비스를 시작할 때(수동 서비스인 경우) 커넥터가 시작됩니다.

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

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

커넥터 중지

커넥터를 중지하는 방법은 다음과 같이 커넥터를 시작한 방법에 따라 다릅니다.

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

```
connector_manager_connName -stop
```

여기서 *connName*은 커넥터의 이름입니다.

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

제 3 장 JText Connector Meta Object 사용

Meta Object는 커넥터 또는 Data Handler가 사용하는 구성 정보를 포함하는 WebSphere Business Integration Adapters의 Business Object입니다. JText Connector는 해당 Business Object 유형에 대해 연관된 JText Meta Object가 지원되는 각각의 Business Object를 포함할 것을 요구합니다. 이 최상위 레벨 Meta Object는 최소한 하나 이상의 하위 Meta Object를 포함합니다.

- 커넥터는 최상위 레벨 JText Meta Object를 사용하여 구성 정보(예: 데이터 변환에 사용할 Data Handler, Business Object 이벤트, 아카이브 및 출력 디렉토리의 경로, 이벤트, 아카이브 및 출력 파일에 대한 파일 확장자, 커넥터가 FTP 시스템에서 파일을 처리하는 경우 필요한 정보와 커넥터가 출력 파일에 대한 고유한 파일 ID를 생성하는지 여부)를 확보합니다.
- 커넥터는 하위 Meta Object를 사용하여 Business Object와 문자열 또는 바이트 배열 사이에서 데이터를 변환할 때 사용할 Data Handler에 대한 구성 값을 지정합니다. 기본적으로, 최상위 레벨 Meta Object는 데이터를 변환할 NameValue Data Handler를 지정합니다.

커넥터가 지원하는 각 Business Object에 대해 상이한 구성 정보를 제공하기 위해 각각에 대해 사용자 정의 최상위 레벨 JText Meta Object를 작성할 수 있습니다. 각 최상위 레벨 Meta Object가 고유 Data Handler Meta Object를 지정하기 때문에 커넥터는 각 유형의 Business Object를 서로 다른 형식으로 처리할 수 있습니다. Data Handler Meta Object는 새 데이터 형식을 도입하거나 기존 형식을 변경하는 경우 Business Object 정의를 편집하거나 커넥터 자체를 수정할 필요성을 제거합니다.

Meta Object는 시작 시에 메모리에 로드되어 구성 정보가 커넥터에 사용 가능하게 합니다. Meta Object가 처리를 위해 통합 브로커에 전송되지 않음에 유의하십시오. 이들은 커넥터의 작동에만 영향을 줍니다.

이 장에서는 JText Meta Object를 사용하여 JText Connector를 구성하는 방법에 대해 설명합니다. Data Handler Meta Object 사용에 대한 자세한 정보는 *Data Handler 안내서*를 참조하십시오. 이 장의 주제는 다음과 같습니다.

- 30 페이지의 『JText Meta Object 이름 지정 규칙』
- 30 페이지의 『JText Meta Object 구조』
- 41 페이지의 『공통 구성 작업』

JText Meta Object 이름 지정 규칙

최상위 레벨 JText Meta Object의 이름에는 기본 최상위 레벨 Meta Object의 이름인 MO_JTextConnector_Default에서 보는 것처럼 세 가지 구성요소가 있습니다. 최상위 레벨 JText Meta Object 이름의 구성요소는 다음과 같습니다.

- MO_는 Meta Object를 나타내는 접두부입니다.
- ConnectorInstanceName_은 커넥터 인스턴스(예: JText)의 이름을 지정합니다. 이 이름은 다중 커넥터 인스턴스 사용을 지원하기 위해 구성 가능합니다. 예를 들어, JText2라는 커넥터는 MO_JText2Connector_Default라는 Meta Object를 포함할 수 있습니다.
- Default는 연관된 Business Object의 이름을 지정합니다. 특정 Business Object에 대한 Meta Object를 작성하려면 Customer라는 Business Object에 대한 MO_JTextConnector_Customer에서와 같이 문자열 Default를 Business Object의 이름으로 변경하십시오. Meta Object 이름에 추가 구성요소와 밑줄을 포함할 수 있습니다. 예를 들어, Oracle_Customer Business Object는 MO_JTextConnector_Oracle_Customer Meta Object와 연관됩니다. 커넥터는 대응하는 Business Object 특정 Meta Object가 존재하지 않을 때마다 기본 Meta Object를 사용합니다.

특정 Business Object에 대한 Meta Object 작성 정보는 61 페이지의 『특정 Business Object에 대한 JText Meta Object 작성』을 참조하십시오

JText Meta Object 구조

JText Meta Object는 계층 구조를 포함합니다. 기본 최상위 레벨 Meta Object는 MO_JTextConnector_Default로 이름 지정됩니다. 최상위 레벨 Meta Object의 두 속성인 EventHandler 및 OutputDataHandler는 커넥터가 사용하는 Data Handler에 대한 구성 정보를 제공하는 하위 Meta Object를 표시합니다. 커넥터는 Data Handler를 사용하여 Business Object와 문자열 또는 바이트 배열 사이의 데이터를 변환합니다.

기본적으로, 이들 속성은 모두 동일한 Data Handler Meta Object (MO_DataHandler_DefaultNameValueConfig)를 지정합니다. 이 Data Handler Meta Object는 NameValue Data Handler를 호출하여 실제로 데이터를 변환합니다. 즉, 제공되는 기본 구성에서 이벤트 및 출력 파일 변환이 동일한 Data Handler를 사용하도록 지정합니다. Data Handler 인스턴스화에 대한 자세한 정보는 *Data Handler* 안내서를 참조하십시오.

주: 포맷터 사용이 Data Handler 사용에 유리하도록 거부되었기 때문에 이전에 포맷터를 표시했던 EventFormat 및 OutputFormat 속성이 MO_JTextConnector_Default Meta Object에서 제거되었습니다. 포맷터를 사용하려면 다음을 수행해야 합니다.

- 최상위 레벨 Meta Object에 EventFormat 및 OutputFormat 속성 추가
- 이들 속성의 유형으로 적당한 Business Object 지정
- EventHandler 및 OutputHandler 속성의 유형을 String으로 변경

포맷터 사용에 대한 정보는 JText Connector 3.0.0 또는 2.3.0 릴리스의 문서를 참조하십시오.

그림 5는 기본 JText Meta Object에 대한 계층 구조와 각 속성 이름 및 유형을 보여줍니다.

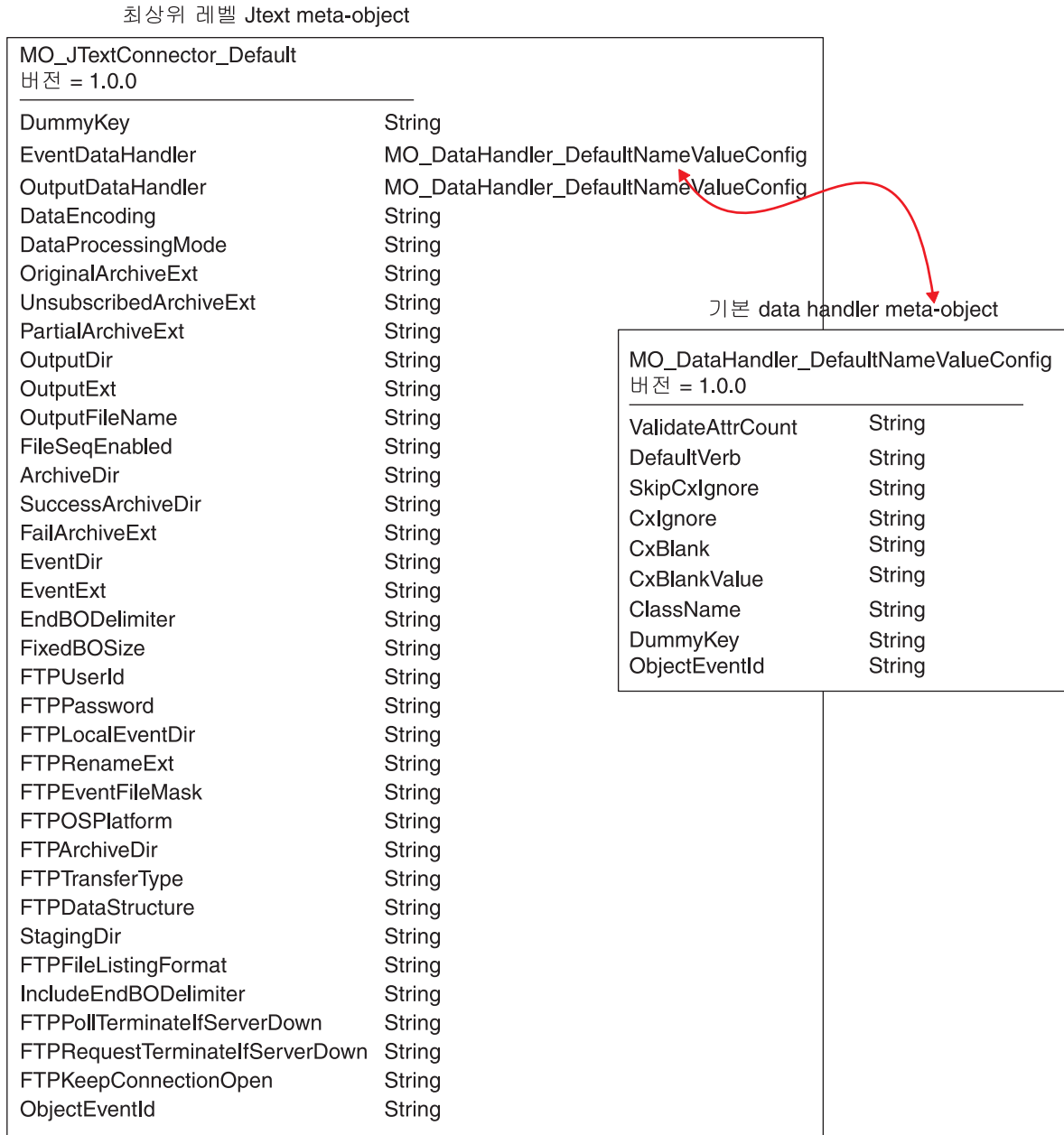


그림 5. JText Meta Object의 계층 구조

사용자 정의 Meta Object 작성

최상위 레벨 JText Meta Object로서의 MO_JTextConnector_Default는 커넥터에 대한 구성 정보 및 하위 Meta Object를 포함합니다. 커넥터가 처리하는 각 유형의 Business Object에 대해 별도의 최상위 레벨 Meta Object를 작성할 수 있습니다. 이들 사용자 정의 Meta Object는 Data Handler의 유형을 구성하기 위해 동일하거나 다른 하위 Meta Object를 포함할 수 있습니다. 예를 들어, 고객 및 항목 Business Object에 대해 서로

다르게 처리를 구성하려면 MO_JTextConnector_Customer 및 MO_JTextConnector_Item Meta Object를 작성하고 상이한 Data Handler Meta Object를 포함하도록 이들 최상위 레벨 Meta Object를 설계하십시오.

초기화 시, 커넥터는 통합 브로커에서 지원되는 Meta Object 및 Business Object의 목록을 검색합니다. 이들 오브젝트의 이름에서 커넥터는 어떤 Business Object가 자체 소유의 연관된 최상위 레벨 Meta Object를 포함하는지를 판별합니다. 런타임 시에 커넥터는 요청 Business Object의 이름을 지원되는 Meta Object 중 하나와 일치시켜서 적당한 구성 정보를 찾습니다.

예를 들어, 커넥터가 다음 Meta Object를 지원한다고 가정하십시오.

- MO_JTextConnector_Default
- MO_JTextConnector_Customer
- MO_JTextConnector_Item

그리고 다음 Business Object를 지원한다고 가정하십시오.

- 고객
- 항목
- 주문

통합 브로커가 고객 Business Object 요청을 전송하면 커넥터는 MO_JTextConnector_Customer Meta Object에 지정된 구성 정보를 사용합니다. 통합 브로커가 주문 Business Object 요청을 전송하면 커넥터는 MO_JTextConnector_Default Meta Object에 지정된 구성 정보를 사용합니다.

MO_JTextConnector_Default 속성

이 절에서는 MO_JTextConnector_Default Meta Object에 있는 속성에 대해 설명합니다.

주: 속성의 DefaultValue 등록 정보에 있는 모든 값은 대소문자를 구분합니다. 디렉토리 정보는 디렉토리의 절대 경로를 지정해야 합니다.

34 페이지의 표 5 및 후속 절에서는 MO_JTextConnector_Default Meta Object에 있는 각 속성의 기능에 대해 설명합니다. 기타 정보 외에, 이 표는 각 단순 속성의 DefaultValue 등록 정보에 대해 제공되는 값도 포함합니다. 제품이 제공하는 값을 사용자 고유의 값으로 바꿀 수 있습니다.

표 5. MO_JTextConnector_Default Meta Object 정의의 속성

속성 이름	설명
ArchiveDir	아카이브 디렉토리의 절대 경로를 지정합니다. 이 디렉토리는 이미 존재해야 합니다. 제공된 기본값은 다음과 같습니다. UNIX: /tmp/JTextConn/Default/archive Windows:C:\temp\JTextConn\Default\Archive
DataEncoding	DataEncoding은 Business Object 문자열을 읽고 쓰는 데 사용할 인코딩입니다. 이 등록 정보가 정적 Meta Object에서 지정되지 않은 경우, 커넥터는 특정 인코딩을 사용하지 않고 Business Object 문자열을 읽거나 쓰도록 시도합니다. 이 속성에 대한 모든 Java 지원 인코딩 세트를 지정할 수 있습니다.
DataProcessing Mode	이 속성은 2진 파일을 읽고 쓸 수 있는 새로운 유연성을 제공합니다. 이 MO 등록 정보를 Binary으로 설정하면 BO에 적합한 Data Handler 인터페이스를 바이트 배열로 호출하거나 그 반대 방향으로 변환하면서 파일 시스템에서 2진 파일을 읽고 쓸 수 있습니다. 이 등록 정보에 대한 일반적인 설정은 Text입니다. Text 모드로, BO에서 문자열 또는 그 반대 방향의 Data Handler 인터페이스를 사용합니다. 이 등록 정보를 설정하지 않는 경우, 기본값은 Text입니다. Binary 모드는 getB0(byte[]) 및 getByteArrayFromB0() 메소드를 올바르게 구현하는 Data Handler에서만 사용해야 합니다.
DummyKey	모든 Business Object 정의의 한 속성이 Key 등록 정보를 사용 가능하게 하는 요구사항을 충족시키기 위해 이 속성이 존재합니다.
EndB0Delimiter	입력 파일 내에서 Business Object를 구분하는 분리문자를 지정합니다. EndB0Delimiter 속성에 대한 자세한 정보는 49 페이지의 『특정 Business Object 폴링』을 참조하십시오. 구성 시 기본값을 제공하지 않고 DataProcessingMode를 Text로 설정하는 경우, 등록 정보의 기본값은 <EndB0:BOName>입니다. DataProcessingMode가 2진인 경우, 등록 정보의 기본값은 FF01입니다.
EndOfFileDelimiter	DataProcessingMode가 2진이고 FTPDataStructure가 Record인 경우, EndB0Delimiter 및 EndOfFileDelimiter를 모두 사용합니다. 이 등록 정보는 레코드 파일의 파일 마커 끝에 사용되는 16진 바이트로 설정됩니다. 설정하지 않는 경우, 기본값, FF02가 사용됩니다.
EventDataHandler	속성이 이벤트 처리(Business Object 문자열이 Business Object로 변환됨)에 사용할 Data Handler에 대한 구성값을 제공하는 하위 Meta Object를 표시합니다. 제공되는 기본값은 MO_DataHandler_DefaultNameValueConfig입니다.

표 5. MO_JTextConnector_Default Meta Object 정의의 속성 (계속)

속성 이름	설명
EventDir	<p>이벤트 디렉토리의 절대 경로를 지정합니다. 이 디렉토리는 이미 존재해야 합니다. 상이한 Business Object에 대한 Meta Object를 작성하고 모두에 대해 동일한 EventDir 경로를 지정하는 경우, 각 Meta Object의 EventExt 속성에 대해 고유한 값을 지정해야 합니다. 자세한 정보는 42 페이지의 『이벤트 디렉토리 및 확장자 지정』을 참조하십시오. 커넥터가 이벤트 처리를 위해 원격 FTP 파일 시스템을 사용하도록 구성하려면 이 속성에 FTP URL을 지정하십시오. 선택적으로 이 속성을 사용하여 URL에 다음 추가 정보를 지정할 수 있습니다.</p> <ul style="list-style-type: none"> FTP 서버에 연결하고 FTP 조작을 수행할 특권이 있는 사용자의 ID 및 암호. EventDir에 지정되지 않은 경우, FTPUserId 및 FTPPassword에 지정되어 있어야 합니다. FTP 포트. EventDir에 지정되지 않은 경우, 커넥터는 기본 FTP 포트를 사용합니다. 원격 이벤트 디렉토리. EventDir에 지정되지 않은 경우, 커넥터는 FTP 서버에 대한 연결이 설정된 디렉토리에서 이벤트 파일을 폴링합니다. <p>EventDir 속성에 FTP 정보를 지정하기 위한 구문은 다음과 같습니다.</p> <pre>ftp://[UserId:password@]FTPserver[:port][RemoteEventDirectory]</pre> <p>자세한 정보는 53 페이지의 『원격 이벤트 처리』를 참조하십시오. EventDir 속성에 로컬 파일 정보를 지정하려면 해당 파일의 전체 경로를 사용하십시오. 또는 FILE URL을 사용할 수 있으며, 이는 다음 형식을 사용합니다.</p> <pre>[file://]FullPathname</pre> <p>제공된 기본값은 다음과 같습니다.</p> <pre>UNIX: /tmp/JTextConn/Default/event</pre> <pre>Windows:C:\temp\JTextConn\Default\Event</pre>
EventExt	<p>이벤트 공고에 사용되는 파일의 확장자를 지정합니다. 값이 지정되지 않으면 JText Connector는 파일 확장자가 없는 파일을 폴링합니다. 자세한 정보는 49 페이지의 『복수 이벤트 파일 또는 복수 이벤트 디렉토리 지정』을 참조하십시오.</p> <p>주: 커넥터가 확장자와 상관없이 단일 이벤트 디렉토리의 모든 파일을 폴링하도록 지정하기 위해 이 속성에 별표(*)를 사용하는 것은 더 이상 지원되지 않습니다. 제공되는 기본값은 in입니다.</p>
FailArchiveExt	<p>성공적으로 처리되지 않은 Business Object를 아카이브하는 데 사용되는 파일 확장자를 지정합니다. 자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오. 제공되는 기본값은 fail입니다.</p>
FileSeqEnabled	<p>각 Business Object를 개별 파일에 출력하는 파일 이름 순서를 지정합니다. 파일 이름은 고유한 순번을 포함합니다. 자세한 정보는 45 페이지의 『요청 처리 지정』을 참조하십시오. 제공되는 기본값은 true입니다.</p>
FixedBOSize	<p>이 Meta Object 등록 정보가 올바른 값으로 표시되면 EndBODelimiter 등록 정보를 대체하여, 일반 분리문자 기반 BO 구문 분석에 대한 대안을 사용자에게 제공합니다.</p>

표 5. MO_JTextConnector_Default Meta Object 정의의 속성 (계속)

속성 이름	설명
FTPArchiveDir	<p>FTP 서버의 아카이브 디렉토리의 절대 경로를 지정합니다. 이 디렉토리는 이미 존재해야 합니다. 이 속성을 사용하여 아카이브를 지정하기 위한 여러 옵션이 있습니다.</p> <ul style="list-style-type: none"> 이 속성에 대한 값을 지정하고 FTPRenameExt 속성에 대한 값을 지정하지 않으면 커넥터가 이벤트 파일 이름에 시간 소인을 추가하고 이 파일을 이 속성에 지정된 FTP 서버 아카이브 디렉토리로 이동시킵니다. 이 속성 및 FTPRenameExt 속성 모두에 대한 값을 지정하면 커넥터가 처리된 이벤트 파일의 이름을 시간 소인 및 FTPRenameExt에 지정된 값으로 바꾸고, 이 속성에 지정된 FTP 서버 아카이브 디렉토리로 해당 파일을 이동시킵니다. 이 속성과 FTPRenameExt 속성 모두에 대한 값을 지정하지 않으면 커넥터는 처리된 이벤트 파일을 아카이브하지 않고 삭제합니다. 이 속성에는 값을 지정하지 않고 FTPRenameExt 속성에 대한 값을 지정하면 커넥터가 시간 소인 및 FTPRenameExt에 지정된 값을 추가하여 처리된 이벤트 파일의 이름을 변경하고, 이 파일을 EventExt 속성에 지정된 디렉토리로 이동시킵니다. 이 속성에 대해 /(슬래시)를 지정하고 FTPRenameExt 속성에 대한 값을 지정하지 않으면 커넥터는 처리된 이벤트 파일을 FTP 서버의 루트 디렉토리로 이동시킵니다. 이 속성에 대해 /(슬래시)를 지정하고 FTPRenameExt 속성에 대한 값을 지정하면 커넥터가 처리된 이벤트 파일의 이름을 FTPRenameExt에 지정된 확장자로 바꾸고 이 파일을 FTP 서버의 루트 디렉토리로 이동시킵니다. <p>자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오. 이 속성에 대해 제공되는 기본값은 없습니다.</p>
FTPDataStructure	<p>이 속성의 유형은 문자열입니다. 사용자는 원격 사이트에서 파일을 가져오거나 여기에 저장하도록 FTP 데이터 구조(File 또는 Record)를 지정할 수 있습니다. 값을 지정하지 않으면, Jtext가 'File'을 기본값으로 사용합니다.</p>
FTPEventFileMask	<p>이벤트 처리를 위해 원격 FTP 파일의 마스크 또는 접두부를 지정하는 데 필요한 임베드된 와일드 카드 문자를 사용합니다. UNIX 또는 Windows 시스템에 적용되는 동일한 이름 지정 표준을 준수하지 않는 메인프레임에서 파일 마스크를 식별하기 위해서만 이 속성에 대한 값을 지정하십시오. 파일 이름에 와일드 카드 문자를 사용하면 이벤트 처리를 위해 복수 파일을 지정할 수 있습니다. 예를 들어, ACT.Z1UC.INPT*. 형식을 사용하여 복수 이벤트 파일을 지정할 수 있습니다. 자세한 정보는 57 페이지의 『메인프레임에서 파일 식별: 선택적 구성』을 참조하십시오. 제공되는 기본값은 없습니다.</p>
FTPFileListingFormat	<p>파일에서 읽을 때 JText Connector가 파일 정보가 표시되리라고 예상해야 하는 형식을 지정합니다. 이것은 날짜 및 시간 정보가 파일 형식 정보 내에서 다른 순서로 저장될 수 있는 다른 로케일에서 파일을 읽을 수 있게 합니다. 커넥터가 사용자 로케일의 형식을 사용할 수 있도록 구성하려면, 파일 속성이 발생하는 순서를 표시하는 세미콜론으로 구분된 일련의 문자를 지정하십시오. 아래 목록은 가능한 문자를 해당 파일 속성과 연관시키는 목록입니다.</p> <p>P Permission L Links U User G Group S Size D Date M Month T Time N Name</p> <p>이 속성에 대해 적합한 값으로 P:L:U:G:S:D:M:T:N이 될 수 있습니다.</p>

표 5. MO_JTextConnector_Default Meta Object 정의의 속성 (계속)

속성 이름	설명
FTPKeepConnectionOpen	이 속성의 Default Value 등록 정보를 값 true로 설정하여 JText Connector가 FTP 사이트와의 연결을 유지보수하게 하십시오. 속성이 값 true로 설정된 경우, 커넥터가 종료하거나 FTP 서버가 연결 자체를 닫을 때(예를 들어, 구성된 제한시간으로 인해)에만 커넥터가 연결을 닫습니다. 제한시간으로 인해 FTP 서버가 연결을 닫았을지도 모를 상황을 처리하기 위해 원격 조작을 수행할 때마다 연결이 여전히 활동 상태인지를 커넥터가 검사합니다. 연결이 닫히면 커넥터가 다시 설정됩니다. 이 속성의 Default Value 등록 정보를 값 false로 설정하여 JText Connector가 조작을 수행할 때마다 FTP 서버와의 연결을 열고 조작이 종료되면 연결을 닫게 하십시오. 커넥터가 연결을 활동 상태로 유지하도록 구성하면 FTP 사이트에서 요청 처리 수행 시 커넥터의 성능을 향상시킬 수 있습니다.
FTPLocalEventDir	커넥터가 FTP 사이트로부터 이벤트 파일을 다운로드하는 로컬 시스템 디렉토리를 지정합니다. 커넥터가 FTP를 사용하여 이벤트를 처리할 수 있게 하려면 이 속성에 대한 값을 지정해야 합니다. 자세한 정보는 54 페이지의 『로컬 디렉토리 지정』을 참조하십시오. 제공되는 기본값은 없습니다.
FTPPlatform	원격 FTP 서버가 MVS 플랫폼인 원격 FTP 파일 시스템을 사용하도록 커넥터를 구성하는 경우에만 이 속성을 사용하십시오. 이 경우, 이 속성의 값을 MVS로 지정하십시오. 대소문자 구분은 중요하지 않습니다. 자세한 정보는 53 페이지의 『원격 FTP 파일 시스템 지정』을 참조하십시오. 제공되는 기본값은 없습니다.
FTPPassword	FTP 서버에 연결하고 FTP 조작을 수행할 특권이 있는 사용자의 암호를 지정합니다. 암호가 EventDir 또는 OutputDir 속성에 지정된 URL에 포함되는 경우, 이 속성에 대한 값을 지정할 필요가 없습니다. 자세한 정보는 54 페이지의 『FTP URL 및 로그인 정보 지정』을 참조하십시오. 이 속성에 대해 제공되는 기본값은 없습니다.
FTPPollTerminateIfServerDown	이벤트의 FTP 사이트를 폴링하도록 구성되었으며 FTP 사이트가 사용 불가능할 때 커넥터의 작동을 지정합니다. FTPPollTerminateIfServerDown 속성의 Default Value 등록 정보가 값 true로 설정되고 커넥터가 폴 호출을 시도할 때 FTP 사이트가 사용 불가능하면, 커넥터가 종료합니다. FTPPollTerminateIfServerDown 속성의 Default Value 등록 정보가 값 false로 설정되고 커넥터가 폴 호출을 시도할 때 FTP 사이트가 사용 불가능하면, 커넥터가 종료하지 않습니다. 제공되는 기본값은 없습니다.
FTPRenameExt	커넥터가 원격 FTP 파일을 폴링한 후, 이름을 바꾸는 데 사용하는 파일 확장자 또는 접미부를 지정합니다. 파일의 이름을 바꾸면 커넥터가 다음 폴 주기에서 동일한 파일을 폴링하지 못하게 됩니다. 또는 처리된 이벤트 파일의 이름을 바꾸고 아카이브 디렉토리로 해당 파일을 이동하도록 커넥터를 구성할 수 있습니다. 자세한 정보는 FailArchiveExt 속성을 참조하십시오. 자세한 정보는 57 페이지의 『메인프레임에서 파일 식별: 선택적 구성』을 참조하십시오. 제공되는 기본값은 없습니다.
FTPRequestTerminateIfServerDown	요청 처리를 수행하고 FTP 사이트와 통신하도록 구성되었으며 FTP 사이트가 사용 불가능할 때 커넥터의 작동을 지정합니다. FTPRequestTerminateIfServerDown 속성의 Default Value 등록 정보가 값 true로 설정되고 커넥터가 요청 처리를 수행할 때 FTP 사이트가 사용 불가능하면, 커넥터가 종료합니다. FTPRequestTerminateIfServerDown 속성의 Default Value 등록 정보가 값 false로 설정되고 커넥터가 요청 처리를 수행할 때 FTP 사이트가 사용 불가능하면, 커넥터가 종료하지 않습니다. 기본값은 제공되지 않습니다. .
FTPTransferType	이 JText Meta Object 등록 정보는 이벤트 및 요청 처리 모두에서 사용됩니다. 이 등록 정보에 가능한 값은 Binary 및 ASCII입니다. 이 등록 정보는 FTP 서버에서 원격으로 파일을 저장 또는 검색할 때 JText가 사용할 전송 유형을 지정합니다. 등록 정보가 존재하지 않는 경우, 어댑터 작동의 기본값은 Binary입니다.

표 5. MO_JTextConnector_Default Meta Object 정의의 속성 (계속)

속성 이름	설명
FTPUserId	FTP 서버에 연결하고 FTP 조작성을 수행할 특권이 있는 사용자의 이름을 지정합니다. UserId가 EventDir 또는 OutputDir 속성에 지정된 URL에 포함되는 경우, 이 속성에 대한 값을 지정할 필요가 없습니다. 커넥터는 EventDir 속성(이벤트 처리 중) 또는 OutputDir 속성(요청 처리 중)에서 FTP URL을 찾지 못하는 경우, 이 속성을 무시합니다. 자세한 정보는 54 페이지의 『FTP URL 및 로그인 정보 지정』을 참조하십시오. 이 속성에 대해 제공되는 기본값은 없습니다.
IncludeEndBODelimiter	EndBODelimiter Meta Object 속성에 지정된 값이 JText Connector에 의해 파일에 작성된 문자열에 포함되는지 여부를 지정합니다. 이 속성의 Default Value 등록 정보가 true로 설정된 경우, 커넥터는 파일 작성 시 EndBODelimiter 속성에 지정된 값을 포함합니다. 이 속성의 Default Value 등록 정보가 false로 설정된 경우, 커넥터는 파일 작성 시 EndBODelimiter 속성에 지정된 값을 포함하지 않습니다.
ObjectEventID	Meta Object에서 커넥터가 사용하지 않지만 통합 브로커에 필요한 위치 표시기입니다. 이 속성은 Meta Object의 마지막 속성이어야 합니다. 제공되는 기본값은 없습니다.
OriginalArchiveExt	원래 이벤트 파일을 아카이브하는 데 사용되는 파일 확장자를 지정하며, 이는 Business Object가 처리에 실패하거나 등록되지 않은 경우, 참조용으로 전체 이벤트 파일을 보존합니다. 자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오. 제공되는 기본값은 orig입니다.
OutputDataHandler	속성이 서비스 호출 요청(Business Object가 Business Object 문자열로 변환됨)에 사용할 Data Handler에 대한 구성값을 제공하는 하위 Meta Object를 표시합니다. 제공되는 기본값은 MO_DataHandler_DefaultNameValueConfig입니다.
OutputDir	출력 디렉토리의 절대 경로를 지정합니다. 이 디렉토리는 이미 존재해야 합니다. 커넥터가 요청 처리를 위해 원격 FTP 파일 시스템을 사용하도록 구성하려면 이 속성에 FTP URL을 지정하십시오. 선택적으로 이 속성을 사용하여 URL에 다음 추가 정보를 지정할 수 있습니다. <ul style="list-style-type: none"> FTP 서버에 연결하고 FTP 조작성을 수행할 특권이 있는 사용자의 사용자 ID 및 암호. EventDir에 지정되지 않은 경우, FTPUserId 및 FTPPassword에 지정되어 있어야 합니다. FTP 포트. OutputDir에 지정되지 않은 경우, 커넥터는 기본 FTP 포트를 사용합니다. 원격 출력 디렉토리. OutputDir에 지정되지 않은 경우, 커넥터는 기본 연결 디렉토리(연결이 설정되는 FTP 서버의 디렉토리)에 요청 파일을 로드합니다. OutputDir 속성에 FTP 정보 지정을 위한 구문은 다음과 같습니다. ftp://[UserId:password@]FTPserver[:port]. 자세한 정보는 58 페이지의 『원격 요청 처리』를 참조하십시오. OutputDir 속성에 로컬 파일 정보를 지정하려면 해당 파일의 전체 경로를 사용하십시오. 또는 FILE URL을 사용할 수 있으며, 이는 다음 형식을 사용합니다. [file://]FullPathname 제공되는 기본값은 다음과 같습니다. UNIX: /tmp/JTextConn/Default/out Windows:c:\temp\JTextConn\Default\Out
OutputExt	요청 처리에 사용되는 파일의 확장자를 지정합니다. 제공되는 기본값은 out입니다. 주: OutputFileName이 확장자를 포함하지 않지만 OutputExt 속성이 확장자를 포함하는 경우, 출력 파일은 파일 이름과 확장자를 모두 사용하여 생성됩니다. 이들 속성이 모두 확장자를 포함하지 않는 경우, 출력 파일은 확장자 없이 생성됩니다.

표 5. MO_JTextConnector_Default Meta Object 정의의 속성 (계속)

속성 이름	설명
OutputFileName	<p>커넥터가 요청 처리 중에 수신되는 Business Object를 기록하는 출력 파일의 이름과 경로를 지정합니다. OutputDir 속성이 유효한 출력 디렉토리를 포함하는 경우, 출력 파일은 지정된 디렉토리에 생성됩니다. 자세한 정보는 39 페이지의 『출력 파일의 이름 지정』을 참조하십시오.</p> <p>주: OutputFileName 및 OutputExt 속성이 확장자를 포함하지 않는 경우, 출력 파일은 확장자 없이 생성됩니다. 제공되는 기본값은 Native입니다.</p>
PartialArchiveExt	<p>성공적으로 처리된 Business Object(이벤트 파일이 복수 Business Object를 포함할 때 모든 프로세스가 성공하지는 않음)를 아카이브하는 데 사용되는 파일 확장자를 지정합니다. 자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오. 제공되는 기본값은 partial입니다.</p>
StagingDir	<p>OutputDir 속성에 의해 지정된 디렉토리로 이동하기 전에 커넥터가 파일을 기록해야 하는 디렉토리를 지정합니다. JText Connector가 파일을 출력하는 디렉토리를 다른 소프트웨어 프로세스에서 모니터링하고 조작할 수 있는 환경을 처리하도록 설계되었습니다(예: 커넥터에서 작성한 파일을 발견하여 다른 위치로 이동시키는 FTP 프로세스). 이와 같은 상황에는 완전히 기록되기 전에 외부 프로세스가 파일을 이동시킬 수 있는 위험이 있습니다. StagingDir 속성에서 스테이징 디렉토리를 지정할 수 있으므로, 커넥터는 전체 파일을 스테이징 디렉토리에 기록한 후 완료 시 출력 디렉토리로 이동시켜, 외부 프로세스가 미완료 파일을 선택하는 위험을 줄입니다.</p> <p>파일 이동 조작에 대한 다른 운영 체제의 접근을 수용하기 위해 동일한 파일 시스템이나 드라이브에 스테이징 디렉토리 및 출력 디렉토리가 있는 것이 좋습니다.</p> <p>제공되는 기본값은 없습니다.</p>
SuccessArchiveExt	<p>성공적으로 처리된 모든 Business Object를 아카이브하는 데 사용되는 파일 확장자를 지정합니다. 자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오. 제공되는 기본값은 success입니다.</p>
UnsubscribedArchiveExt	<p>등록되지 않은 모든 Business Object를 아카이브하는 데 사용되는 파일 확장자를 지정합니다. 자세한 정보는 43 페이지의 『이벤트 아카이브 지정』을 참조하십시오. 제공되는 기본값은 unsub입니다.</p>

주: FTPTransferType, FTPDataStructure, DataProcessingMode,

EndOfFileDelimiter 및 FixedBOSize 속성은 제공되는 JText Meta Object의 일부가 아닙니다. 이 속성을 사용하려면, Meta Object에 명시적으로 추가하고 해당 기본값을 설정해야 합니다.

출력 파일의 이름 지정

출력 파일의 이름을 지정하는 방법에는 세 가지가 있습니다.

- OutputFileName 속성 사용

커넥터가 동일한 유형의 각 Business Object를 고유한 순번이 있는 개별 파일에 기록하거나 복수 Business Object를 지정된 이름이 있는 단일 파일에 추가하기 원할 때 이 속성을 사용하십시오.

- 동적 하위 Meta Object 사용

각 유형의 Business Object에 대해 출력 파일 이름을 동적으로 생성하거나 커넥터가 생성하는 출력 파일의 이름을 리턴할 때 동적 하위 Meta Object를 사용하십시오. 세부사항은 5 페이지의 『동적 하위 Meta Object 사용』을 참조하십시오.

OutputFileName 속성을 사용하여 출력 파일의 이름을 지정하는 방법에는 여러 가지가 있습니다.

- OutputFileName이 문자열 Native로 설정되고 FileSeqEnabled 속성이 true로 설정되는 경우, 커넥터는 수신 Business Object의 이름에서 파생된 이름, OutputExt 속성에서 파생된 확장자 및 OutputDir 속성에서 파생되는 경로가 있는 고유한 파일로 Business Object 문자열을 전송합니다. 이 경우, 커넥터의 기본 작동은 동일한 유형의 각 Business Object를 고유한 순번이 있는 개별 파일에 기록하는 것입니다. 커넥터가 동일한 유형의 Business Object를 수신할 때마다 출력 파일을 겹쳐쓰게 하려면 FileSeqEnabled 속성을 false로 설정하십시오.
- OutputFileName이 Native 이외의 문자열로 설정되고 FileSeqEnabled 속성이 true로 설정되는 경우, 커넥터는 다음 방법 중 하나로 출력 파일의 값을 처리합니다.
 - OutputFileName이 절대 경로를 포함하는 경우(출력 파일의 파일 이름과 확장자 포함. 예: OutputFileName= C:\temp\Out\test.out), 커넥터는 이 속성만을 사용하여 출력 파일을 생성합니다. 이 경우, 커넥터의 기본 작동은 동일한 유형의 각 Business Object를 지정된 이름 및 고유한 순번이 있는 개별 파일에 기록하는 것입니다.
 - OutputFileName이 전체 경로와 파일 이름을 포함하지만 확장자를 포함하지 않고 OutputExt 속성이 값을 포함하는 경우(예: OutputFileName= C:\temp\Out\test 및 OutputExt=out), 커넥터는 속성의 값을 모두 사용하여 출력 파일을 생성합니다. 이 경우, 커넥터는 C:\temp\Out\test_1.out 파일을 생성합니다.
 - OutputFileName이 확장자를 제외한 전체 경로와 파일 이름을 포함하고 OutputExt 속성이 값을 포함하지 않는 경우, 커넥터는 확장자가 없는 출력 파일을 생성합니다. 이 경우, 커넥터는 C:\temp\Out\test_1 파일을 생성합니다.
 - OutputFileName이 파일 이름만을 포함하고 경로 또는 확장자를 포함하지 않으며, OutputDir 속성이 값을 포함하는 경우, 커넥터는 OutputDir에 의해 지정된 디렉토리에 출력 파일을 생성합니다. OutputExt가 값을 포함하는 경우, 커넥터는 또한 해당 값을 사용합니다. 그렇지 않으면, 확장자 없이 파일 이름을 작성합니다.

주: 커넥터가 둘 이상의 Business Object 유형을 처리 중이고 OutputFileName이 Native 이외의 문자열로 설정되는 경우, 각 Business Object에는 고유한 출력 파일 이름을 지정하는 자체 최상위 레벨 Meta Object가 있어야 합니다. 예를 들어, 고객 Business Object가 사용하는 Meta Object는 MO_JTextConnector_Customer이고, 항목이 사용하는 Meta Object는 MO_JTextConnector_Item일 수 있습니다. 이 Meta Object 각각에 있는 OutputFileName 속성값을 고유한 값으로 설정하십시오.

- 커넥터가 지정된 이름이 있는 단일 파일에 복수 Business Object를 추가하게 하려면 OutputFileName에 대한 값을 지정하고 FileSeqEnabled 속성을 false로 설정하십시오.
- 커넥터가 동일한 유형의 Business Object를 수신할 때마다 출력 파일을 겹쳐쓰게 하려면 동적 하위 Meta Object를 사용하십시오. InFileName 속성에 절대 경로와 파일 이름을 지정하고 FileWriteMode 속성을 “o”로 설정하십시오. 동적 하위 Meta Object 사용에 대한 자세한 정보는 5 페이지의 『동적 하위 Meta Object 사용』을 참조하십시오.

Native는 예약어입니다.

자세한 정보는 45 페이지의 『요청 처리 지정』을 참조하십시오.

공통 구성 작업

이 절에서는 가장 공통적인 구성 작업에 대해 설명합니다.

- 41 페이지의 『이벤트 공고 지정』
- 43 페이지의 『이벤트 아카이브 지정』
- 45 페이지의 『요청 처리 지정』
- 49 페이지의 『복수 이벤트 파일 또는 복수 이벤트 디렉토리 지정』
- 49 페이지의 『특정 Business Object 폴링』
- 53 페이지의 『원격 FTP 파일 시스템 지정』
- 61 페이지의 『Data Handler 지정』
- 61 페이지의 『특정 Business Object에 대한 JText Meta Object 작성』
- 62 페이지의 『동일한 파일에서 상이한 유형의 복수 Business Object 읽기』
- 62 페이지의 『ObjectEventID 속성에 대한 값 지정』
- 62 페이지의 『JText Connector의 2차 인스턴스 설정』
- 64 페이지의 『JText Connector의 성능 조정』
- 65 페이지의 『테스트를 위한 샘플 파일 생성』
- 65 페이지의 『테스트를 위한 샘플 Business Object 생성』

이벤트 공고 지정

이 절에서는 다음에 대해 설명합니다.

- 42 페이지의 『이벤트 디렉토리 및 확장자 지정』
- 42 페이지의 『폴링 작동 구성』

이벤트 디렉토리 및 확장자 지정

둘 이상의 Business Object 유형을 처리하기 위해 커넥터에 전송하고 각 Business Object 유형에 고유한 최상위 레벨 Meta Object가 있는 경우, 사용자가 EventDir 및 EventExt 속성에 대해 지정하는 값의 조합이 각 Business Object에 대한 각 디렉토리/확장자 쌍에 대해 고유해야 합니다.

즉, 두 Business Object 유형에 대해 동일한 이벤트 디렉토리를 지정하는 경우, 이 Business Object에 대해 상이한 이벤트 확장자를 지정해야 합니다. 두 Business Object 유형에 대해 동일한 확장자를 지정하는 경우, 이 Business Object에 대해 상이한 이벤트 디렉토리를 지정해야 합니다.

예를 들어, 고객 및 항목 Business Object에 대한 구성값을 제공하기 위해 각각 MO_JTextConnector_Customer 및 MO_JTextConnector_Item Meta Object를 작성했다고 가정하십시오. 동일한 디렉토리에서 양 Business Object에 대한 입력 파일을 찾으도록 커넥터에 지시하는 경우(EventDir 속성에 동일한 경로를 지정), EventExt 속성에 대해 상이한 값을 지정하여 입력 파일을 고유하게 식별해야 합니다.

그러므로 EventDir 속성이 고객 및 항목 모두에 대해 C:\temp\event로 평가되는 경우, 이들 두 Business Object에 대한 EventExt 속성값은 달라야 합니다(예를 들어 고객 입력 파일의 경우 in, 항목의 경우 inp).

폴링 작동 구성

폴링 작동을 구성하려면 다음 단계를 수행하십시오.

1. MO_JTextConnector_Default Meta Object에 대해 다음 속성을 구성하십시오.
 - EventDir -- 해당 파일이 이벤트 공고를 트리거하는 기존 디렉토리의 절대 경로를 지정합니다.
 - EventExt -- 커넥터는 전달되는 기본 확장자가 in인 파일을 찾습니다. 이 속성을 사용하여 다른 확장자를 지정하는 경우, 커넥터는 지정된 확장자를 찾습니다. 이 속성을 비워둘 경우, 커넥터는 확장자가 없는 파일을 폴링합니다.
 - EventHandler -- 이벤트 공고 중에 데이터 변환에 사용할 Data Handler를 지정합니다.
2. Connector Configurator를 사용하여 다음 커넥터 등록 정보를 구성하십시오.
 - PollFrequency -- 간격 빈도를 지정합니다.
 - PollQuantity -- 각 폴링 간격에 대한 이벤트 수를 지정합니다.
 - PollEndTime -- 이벤트 폴링을 완료할 시간을 지정합니다.
 - PollStartTime -- 이벤트 폴링을 시작할 시간을 지정합니다.
3. 이벤트 디렉토리에 대한 읽기 권한을 설정하십시오.

이벤트 아카이브 지정

이벤트 파일의 Business Object 전부 또는 일부가 성공적으로 처리되는지에 따라 JText Connector는 성공적으로 처리된 Business Object에 대한 아카이브 파일 작성 시 상이한 확장자를 사용합니다. 커넥터는 또한 처리에 실패하거나 등록되지 않은 Business Object를 다르게 이름 지정된 아카이브 파일에 기록합니다.

이 절에서는 다음에 대해 설명합니다.

- 『로컬 아카이브 파일 이름』
- 44 페이지의 『로컬 아카이브 구성』

로컬 아카이브 파일 이름

아카이브 확장자 속성에 대해 제공되는 기본값을 유지하는 경우, 커넥터는 아래 표시된 것처럼 이름 지정되는 아카이브 파일을 작성합니다.

- 이벤트 파일에 하나의 Business Object가 있는 경우

JText Connector가 단일 Business Object를 포함하는 이벤트 파일을 처리한 후 아카이브 디렉토리에 다음 파일 중 하나를 작성합니다.

- 성공적으로 처리된 Business Object를 아카이브하기 위한 `filename_timestamp.success`
- 성공적으로 처리되지 않은 Business Object를 아카이브하기 위한 `filename_timestamp.fail`
- 등록하지 않은 Business Object를 아카이브하기 위한 `filename_timestamp.unsub`

Business Object가 처리에 실패하거나 등록되지 않은 경우 커넥터는 또한 `filename_timestamp.orig` 파일을 작성하며, 이 파일은 커넥터가 원래 수신한 대로 이벤트 파일을 보존합니다.

- 이벤트 파일에 복수 Business Object가 있고 모두가 성공적으로 처리되는 경우
JText Connector가 복수 Business Object가 있는 이벤트 파일을 성공적으로 처리한 후 아카이브 디렉토리에 `filename_timestamp.success`를 작성합니다.
- 이벤트 파일에 복수 Business Object가 있고 그 중 일부가 등록되지 않거나 처리에 실패하는 경우

JText Connector가 복수 Business Object를 포함하는 이벤트 파일을 처리한 후 아카이브 디렉토리에 다음의 모든 파일을 작성할 수 있습니다.

- 처리에 성공한 모든 Business Object를 아카이브하기 위한 `filename_timestamp.partial`
- 처리에 실패한 모든 Business Object를 아카이브하기 위한 `filename_timestamp.fail`
- 커넥터가 등록하지 않은 모든 Business Object를 아카이브하기 위한 `filename_timestamp.unsub`

- 커넥터가 원래 수신한 그대로 이벤트 파일을 보존하기 위한 `filename_timestamp.orig`

예를 들어, LegacyApp.in 파일이 다음 네 개의 Business Object를 포함한다고 가정하십시오.

- 계약 - 성공적으로 처리됨
- 고객 - 형식화에 실패함
- 주문 - 성공적으로 처리됨
- 항목 - 커넥터가 등록하지 않음

이 경우, 커넥터는 아카이브 디렉토리에 다음 파일을 작성합니다.

- LegacyApp_timestamp.partial - 계약과 주문을 포함합니다.
- LegacyApp_timestamp.fail - 고객을 포함합니다.
- LegacyApp_timestamp.unsub - 항목을 포함합니다.
- LegacyApp_timestamp.orig - 계약, 고객, 주문 및 항목을 포함합니다.

로컬 아카이브 구성

아카이브를 위해 커넥터를 구성하려면 다음 단계를 따르십시오.

1. MO_JTextConnector_Default Meta Object에 대해 다음 속성을 구성하십시오.
 - ArchiveDir -- 커넥터가 처리된 후 이벤트(처리 상태를 표시하는 파일 확장자 보유)를 보관할 기존 로컬 또는 FTP 서버 디렉토리의 절대 경로를 지정합니다.
 - SuccessArchiveExt -- 성공적으로 처리된 Business Object를 포함하는 파일에 대해 확장자를 지정합니다(모든 Business Object가 성공적으로 처리되는 경우).
 - PartialArchiveExt -- 성공적으로 처리된 모든 Business Object를 포함하는 파일에 대해 확장자를 지정합니다(이벤트 파일에 있는 Business Object의 일부가 성공적으로 처리되지 않는 경우).
 - UnsubscribedArchiveExt -- 커넥터가 등록하지 않는 Business Object를 포함하는 파일에 대한 확장자를 지정합니다.
 - OriginalArchiveExt -- 이벤트 파일에 포함된 모든 Business Object를 보존하는 파일에 대한 확장자를 지정합니다.
 - FailArchiveExt -- 처리에 실패한 Business Object를 포함하는 파일에 대한 확장자를 지정합니다.
2. Connector Configurator를 사용하여 ArchivingEnabled 커넥터 등록 정보를 구성하십시오.
3. 아카이브 디렉토리에 대한 쓰기 권한을 설정하십시오.

요청 처리 지정

JText Connector가 이름이 동적으로(각 Business Object 인스턴스에서) 또는 정적으로(Meta Object를 통해) 지정되는 파일에 Business Object를 기록하도록 할 수 있습니다. 또한 커넥터가 정적으로 생성하는 각 파일 이름을 리턴하도록 할 수도 있습니다. 이 기능은 고유한 순번을 사용하여 생성되는 파일 이름을 얻는 데 유용합니다. 이 절에는 다음 부속 절이 포함되어 있습니다.

- 45 페이지의 『동적 파일 이름 지정』
- 45 페이지의 『정적 파일 이름 지정』
- 46 페이지의 『파일 이름 리턴』
- 47 페이지의 『로컬과 원격 처리의 차이점』
- 47 페이지의 『출력 파일 구성』

동적 파일 이름 지정

커넥터가 각 유형의 Business Object에 대해 출력 파일 이름을 동적으로 생성하게 하려면 동적 하위 Meta Object를 작성하십시오. 다음 경우에 하위 Meta Object를 사용하십시오.

- 출력 파일의 이름을 지정하거나 생성되는 파일 이름의 이름을 수신하는 경우
- 출력 파일에 추가 또는 겹쳐쓰기 여부를 지정하는 경우

중요: ICS를 통합 브로커로 사용하는 경우, 동적 하위 Meta Object를 작성하여 커넥터가 출력 파일 이름을 생성 또는 리턴할 수 있도록 하는 것 외에, 맵 또는 협업 논리를 수정하여 동적 하위 Meta Object의 InFileName 속성에 각 Business Object에 대한 파일 이름 및 경로와, 필요한 경우 고유한 순번을 삽입해야 합니다.

자세한 정보는 5 페이지의 『동적 하위 Meta Object 사용』을 참조하십시오.

커넥터가 Meta Object를 처리하는 방법에 대한 정보는 12 페이지의 『요청 처리』를 참조하십시오.

동적으로 생성되는 출력 파일 이름을 사용하기 위한 커넥터 구성에 대한 정보는 47 페이지의 『출력 파일 구성』을 참조하십시오.

정적 파일 이름 지정

Meta Object를 사용하여 출력 파일의 이름을 지정할 때 변경사항이 효력을 가지려면 커넥터를 다시 시작해야 합니다. 커넥터가 제공된 유형의 모든 Business Object를 단일 파일에 추가하거나 각 Business Object에 대해 별도의 파일을 작성하는지 여부를 지정할 수 있습니다.

제공되는 기본 구성을 사용할 때 커넥터는 처리하는 각 Business Object에 대해 출력 파일을 작성합니다. 출력 파일을 수신 Business Object에 대해 이름 지정하고 순번을

추가하여 이름을 고유하게 작성합니다. 확장자 .out을 부여합니다. 예를 들어, 고객 및 항목 Business Object를 수신하는 경우, 커넥터는 데이터를 Customer_1.out 및 Item_1.out 출력 파일에 기록합니다. 생성되는 출력 파일의 이름 얻기에 대한 정보는 46 페이지의 『파일 이름 리턴』을 참조하십시오. 순번을 저장하는 파일에 대한 정보는 23 페이지의 『OutputLog』를 참조하십시오.

Meta Object를 사용하여 출력 파일의 이름을 구성하려면 다음을 수행하십시오.

1. MO_JTextConnector_Default Meta Object에 대해 다음 속성을 구성하십시오.
 - OutputDir -- 요청 처리 시 커넥터가 파일을 기록할 기존 디렉토리의 절대 경로를 지정합니다. 자세한 정보는 47 페이지의 『출력 파일 구성』을 참조하십시오.
 - FileExt -- 제공된 기본 구성을 변경하려는 경우, 이 속성을 사용하여 사용자 고유 확장자를 지정하여 커넥터에서 out 확장자로 파일을 작성할 수 있도록 하십시오.
 - FileSeqEnabled -- true로 설정하여 커넥터가 파일 당 하나의 Business Object를 고유한 순번과 함께 출력하도록 합니다. false로 설정하여 커넥터가 겹쳐쓰거나 추가하여 제공된 유형의 모든 Business Object를 단일 파일에 출력하도록 합니다. 겹쳐쓰기 또는 추가 작동 구성에 대한 정보는 47 페이지의 표 6을 참조하십시오.
 - OutputFileName -- 커넥터에서 각 Business Object에 대한 고유한 파일을 생성하거나 파일의 데이터를 겹쳐쓰지 않고 단일 출력 파일에 Business Object를 추가하도록 하려면, 출력 파일의 전체 경로 및 파일 이름을 지정하십시오.
커넥터가 동일한 유형의 Business Object를 수신할 때마다 출력 파일을 겹쳐쓰게 하려면 OutputFileName에 대한 값을 지정하지 마십시오.
이들 각 경우에 FileSeqEnabled를 false로 설정하십시오.
겹쳐쓰기 또는 추가 작동 구성에 대한 정보는 47 페이지의 표 6을 참조하십시오.
2. 출력 디렉토리에 대한 쓰기 권한을 설정하십시오.

주: 커넥터가 상이한 Business Object에 대해 서로 다른 데이터 형식이나 파일 이름 지정 규칙을 사용할 경우 특정 Business Object에 대한 Meta Object를 작성해야 합니다.

파일 이름 리턴

커넥터가 생성하는 파일의 이름을 리턴하게 하려면 다음을 수행하십시오.

- Meta Object를 사용하여 경로와 파일 이름을 지정하고 커넥터가 각 출력 파일에 대한 고유한 순번을 생성하게 하십시오. 자세한 정보는 45 페이지의 『정적 파일 이름 지정』을 참조하십시오.
- 동적 하위 Meta Object를 사용하여 커넥터가 생성하는 각 파일의 이름을 리턴하게 하십시오. 5 페이지의 『동적 하위 Meta Object 사용』의 단계를 따르지만, InFileName 속성에 대한 값을 지정하지 마십시오. 커넥터가 동적 하위 Meta Object가

OutFileName=CxIgnore를 지정하는 Business Object를 수신할 때, 커넥터는 최상위 레벨 Meta Object의 구성을 기본으로 파일 이름을 작성하고 전체 경로 및 파일 이름을 InFileName 속성의 값으로 리턴합니다.

주: 커넥터는 FTP 서버에서 파일을 처리할 때에도 InFileName 속성을 로컬 경로만으로 채웁니다.

중요: ICS를 통합 브로커로 사용하는 경우, 동적 하위 Meta Object를 작성하여 커넥터가 출력 파일 이름을 생성 또는 리턴할 수 있도록 하는 것 외에, 맵 또는 협업 논리를 수정하여 동적 하위 Meta Object의 InFileName 속성에 각 Business Object에 대한 파일 이름 및 경로와, 필요한 경우 고유한 순번을 삽입해야 합니다.

로컬과 원격 처리의 차이점

커넥터는 파일을 로컬로 처리하는 것과 거의 동일한 방법으로 파일을 원격으로 처리합니다. 그러나 약간의 차이는 있습니다.

- 동적으로 이벤트를 처리하고 파일 이름을 생성할 때 커넥터는 동적 하위 Meta Object의 InFileName 속성을 로컬 경로 이름으로만 채우고 원격 경로로 채우지는 않습니다.
- 요청을 처리할 때 커넥터가 동적 파일 이름 지정에 대해 구성되지 않고 FileSeqEnabled가 false로 평가되고 출력 파일이 이미 존재하는 경우,
 - 로컬로 처리하는 경우에는 커넥터가 기존 파일을 겹쳐씁니다.
 - 원격으로 처리하는 경우 커넥터는 예외를 발생시킵니다.
- 커넥터를 사용하여 FTP 서버에서 원격으로 파일을 처리하는 경우, 로컬 이벤트를 처리를 위해 표준 아카이브 확장 속성을 구성할 뿐 아니라 FTPArchiveDir 및 FTPRenameExt 속성 또한 구성할 수 있습니다. 이 속성을 사용하면 처리의 성공 여부와 상관없이 원격으로 아카이브되는 파일의 이름을 바꾸고 이동시킬 수 있습니다.

자세한 정보는 55 페이지의 『원격 아카이브 지정』을 참조하십시오.

출력 파일 구성

표 6에서는 출력 파일에 대한 가능한 구성 옵션을 설명합니다.

표 6. 출력 파일 지정

원하는 출력 조건	속성/등록 정보 필수 구성	속성/등록 정보 값
제공된 유형의 각 Business Object는 절대 경로 및 파일 이름이 Business Object의 속성으로부터 런타임 시 파생되는 파일에 추가됩니다.	동적 하위 Meta Object 사용 AppSpecificInfo(Business Object 레벨에서) 동적 하위 Meta Object의 경우, OutFileName FileWriteMode	cw_mo_JTextConfig = DynChildMOName 사용자가 지정하는 경로 이름 및 파일 이름 a 또는 append

표 6. 출력 파일 지정 (계속)

원하는 출력 조건	속성/등록 정보 필수 구성	속성/등록 정보 값
제공된 유형의 각 Business Object는 절대 경로 및 파일 이름이 Business Object의 속성으로부터 런타임 시 파생되는 출력 파일을 겹쳐줍니다.	동적 하위 Meta Object 사용 AppSpecificInfo(Business Object 레벨에서) 동적 하위 Meta Object의 경우, OutFileName FileWriteMode	cw_mo_JTextConfig = DynChildMOName 사용자가 지정하는 경로 이름 및 파일 이름 o 또는 overwrite 사용자가 지정하는 경로 이름
제공된 유형의 각 Business Object는 이름이 Business Object의 이름과 생성되는 고유한 순번으로부터 파생되는 고유한 파일에 기록됩니다.	OutputDir FileSeqEnabled	true
커넥터가 생성하는 각 파일의 이름을 리턴합니다. 제공된 유형의 각 Business Object는 이름이 Business Object의 이름과 생성되는 고유한 순번으로부터 파생되는 고유한 파일에 기록됩니다.	동적 하위 Meta Object 사용 AppSpecificInfo(Business Object 레벨에서) InFileName(동적 하위 Meta Object에서) FileWriteMode(동적 하위 Meta Object에서) Meta Object 구성 사용: MO_JTextConnector_ businessobjectname: OutputDir	Native cw_mo_JTextConfig = DynChildMOName CxIgnore N/A 사용자가 지정하는 경로 이름
제공된 유형의 모든 Business Object는 사용자가 지정하는 이름의 단일 파일에 추가됩니다.	FileSeqEnabled OutputFileName FileSeqEnabled OutputFileName	true Native false 사용자가 지정하는 경로 이름 및 파일 이름
제공된 유형의 각 Business Object는 이름이 사용자가 지정한 이름과 고유한 순번이 있는 고유한 파일에 기록됩니다.	FileSeqEnabled	true
커넥터가 둘 이상인 유형의 Business Object를 처리 중이고 OutputFileName이 Native 이외의 문자열로 설정되는 경우, 각 비즈니스에는 고유한 최상위 레벨 Meta Object가 있어야 합니다. 자세한 정보는 39 페이지의 『출력 파일의 이름 지정』을 참조하십시오.	OutputFileName	사용자가 지정하는 경로 이름 및 파일 이름
제공된 유형의 각 Business Object는 이름이 Business Object의 이름에서 파생되는 출력 파일을 겹쳐줍니다.	OutputDir FileSeqEnabled OutputFileName	사용자가 지정하는 경로 이름 false Native

표 6. 출력 파일 지정 (계속)

원하는 출력 조건	속성/등록 정보 필수 구성	속성/등록 정보 값
커넥터가 생성하는 각 파일의 이름을 리턴합니다. 제공된 유형의 각 Business Object는 이름이 사용자가 지정한 이름과 고유한 순번이 있는 고유한 파일에 기록됩니다.	동적 하위 Meta Object 사용 AppSpecificInfo(Business Object 레벨에서) InFileName(동적 하위 Meta Object에서) FileWriteMode(동적 하위 Meta Object에서) Meta Object 구성 사용: MO_JTextConnector_ businessobjectname: FileSeqEnabled OutputFileName	cw_mo_JTextConfig = DynChildMOName CxIgnore N/A true 사용자가 지정하는 경로 이름 및 파일 이름

복수 이벤트 파일 또는 복수 이벤트 디렉토리 지정

지정된 확장자가 있는 파일만을 선택하도록 커넥터를 구성할 수 있습니다. 또한 복수 디렉토리에서 파일을 선택하도록 커넥터를 구성할 수도 있습니다.

중요: 커넥터가 확장자와 상관없이 단일 이벤트 디렉토리의 모든 파일을 폴링하도록 지정하기 위해 EventExt 속성에 별표(*)를 사용하는 것은 더 이상 지원되지 않습니다.

각 Business Object 유형에 대해 별도의 이벤트 디렉토리를 지정하려면 다음 단계를 수행하십시오.

1. 지원되는 각 Business Object에 대해 별도의 Meta Object를 작성하십시오. 예를 들어, MO_JTextConnector_Customer 및 MO_JTextConnector_Item을 작성하십시오. 자세한 정보는 61 페이지의 『특정 Business Object에 대한 JText Meta Object 작성』을 참조하십시오.
2. 각 Meta Object의 EventDir 속성에 적절한 디렉토리를 지정하십시오.

주: JText Connector는 파일의 위치와 상관없이 가장 오래된 것부터 가장 최신 것까지 시간 소인의 순서대로 이벤트 파일을 처리합니다. 즉, JText Connector는 시간 소인의 연대순으로 개별 디렉토리에 있는 파일을 처리합니다.

특정 Business Object 폴링

JText Connector의 구성은 사용자의 모든 이벤트 파일이 단일 디렉토리에 있는지, 이벤트 파일이 모두 동일한 확장자를 갖고 있는지, 이벤트 파일이 단일 Business Object 또는 복수 Business Object를 포함하는지, 이벤트 파일이 하나의 유형 또는 복수 유형의 Business Object를 포함하는지 및 단일 행 또는 복수 행에 각 Business Object를 표시하는지 여부에 따라서 다릅니다.

이 절에서는 다음에 대해 설명합니다.

- 『EndBODelimiter 구문 분석 메소드 사용』
 - 51 페이지의 『인쇄 불가능한 문자를 EndBODelimiter로 사용』
- 53 페이지의 『FixedBOSize 구문 분석 메소드 사용』

EndBODelimiter 구문 분석 메소드 사용

EndBODelimiter Meta Object 속성에 대한 값이 지정되지 않으면 커넥터는 다음을 수행합니다.

- 이벤트 파일이 <EndB0:BOName>으로 Business Object 문자열을 구분한다고 예상합니다.
- 출력 파일에 Business Object 문자열을 기록할 때 분리문자로서 <EndB0:BOName>을 지정합니다.

이벤트 파일이 하나의 Business Object만을 포함하는 경우, 이 속성에 대해 EOF(파일 끝)를 지정할 수 있습니다.

EndBODelimiter 속성값을 비어 있지 않은 문자열로 설정하는 경우, 해당 문자열이 모든 파일에 대한 Business Object 분리문자인 것으로 가정됩니다. 값이 설정되지 않거나 지워지는 경우, 커넥터는 분리문자가 <EndB0:BOName>이라고 가정합니다.

중요: DataProcessingMode를 binary로 설정하고 EndBODelimiter에 지정된 값이 없는 경우, JText는 기본값 EndBODelimiter를 FF01(2바이트)로 설정하고 EndOfFileDelimiter를 FF02(2바이트)로 설정합니다.

표 7에서는 분리문자 옵션을 설명합니다.

표 7. EndBODelimiter 속성 사용

조건	분리문자	참고
파일이 Business Object 유형이 하나 이상인 Business Object 문자열을 하나 이상 포함하거나 또는 파일이 동일한 유형의 Business Object의 복수 Business Object 문자열을 포함하며, 각 문자열이 여러 행에서 실행됩니다.	<EndB0:BOName>또는 EOL 또는 사용자 지정 Business Object 값	<ul style="list-style-type: none">• Business Object 문자열 사이에 줄 바꿈기가 있을 때마다 세미콜론으로 분리된 EOL을 지정하십시오.• EOL과 함께 사용자 정의 분리문자를 지정하십시오. 사용자 정의 분리문자는 EOL과 함께 사용될 때 항상 첫 번째 요소여야 합니다. 올바른 예: customEndB0;EOL;EOL. 올바르지 않은 예: EOL;customEndB0;EOL

표 7. EndBODelimiter 속성 사용 (계속)

조건	분리문자	참고
각 파일이 단 하나의 Business Object 문자열을 포함합니다.	EOL F 또는 사용자 지정 값	<ul style="list-style-type: none"> Business Object 문자열 사이에 줄 바꿈기가 있을 때마다 세미콜론으로 분리된 EOL을 지정하십시오. 입력 문자열에 필요한 경우, EOL 및 EOF와 함께 사용자 지정 분리문자를 지정하십시오. 사용자 정의 분리문자는 EOL과 함께 사용될 때 항상 첫 번째 요소여야 합니다. 올바른 예: customEndBO;EOL;EOL. 올바르지 않은 예: EOL;customEndBO;EOL
파일이 해당 하나씩, 복수의 Business Object 문자열을 포함합니다.	EOL	
파일이 동일한 유형의 Business Object의 복수 Business Object 문자열을 포함합니다. 각 문자열은 Business Object 문자열 사이에 분리자가 없고 여러 행에서 실행됩니다.	None	<p>제공되는 기본 Meta Object 또는 사용자 정의 Meta Object를 사용할 수 있습니다.</p> <p>주: 이 옵션은 서비스 호출 요청 중에만 사용할 수 있고 이벤트 공고에는 사용할 수 없습니다. 다른 어떠한 분리문자와 함께라도 이 분리문자를 사용하지 마십시오.</p>

주: 소스 파일이 빈 행을 포함하는 경우, 커넥터는 해당 행을 무시합니다.

인쇄 불가능한 문자를 EndBODelimiter로 사용: 복수 디렉토리에서 파일을 폴링하려면 지원되는 각 Business Object에 대해 Meta Object를 작성해야 합니다. 사용자가 각 Meta Object의 EndBODelimiter 속성에 지정하는 값은 소스 파일이 단일 Business Object 또는 복수 Business Object를 포함하는지 여부에 따라 다릅니다.

- 단일 Business Object를 포함하는 파일
전체 데이터 파일이 단 하나의 Business Object 문자열을 포함하는 경우 EOF를 EndBODelimiter로 지정할 수 있습니다.
- 복수 Business Object를 포함하는 파일
입력 파일이 Business Object 분리문자로서 줄 바꿈기만이 있는 복수 Business Object를 포함하는 경우, EndBODelimiter 속성에 문자열 EOL을 지정하십시오. 이 경우, 소스 파일은 동일한 유형의 복수 Business Object를 표시하는 문자열을 포함합니다.

중요: 복수 Business Object 유형을 포함하는 파일에서 폴링하려면

MO_JTextConnector_Default Meta Object를 사용하고, EventExt 및 EventDir 속성이 이 이벤트 파일이 있는 디렉토리를 올바르게 지시하는지 확인해야 합니다. 별도의 이벤트 파일에 표시되거나 이벤트 파일이 상이한 디렉토리에 있는 Business Object 유형을 폴링하려면 각 유형에 대해 별도의 최상위 레벨 Meta Object를 작성해야 합니다. EventExt 및 EventDir 속성을 사용하여 적당한 디렉토리를 지시하십시오.

상이한 유형의 복수 Business Object를 포함하는 파일을 풀링할 때 사용자 정의 Data Handler를 사용하려면 62 페이지의 『동일한 파일에서 상이한 유형의 복수 Business Object 읽기』를 참조하십시오.

이름/값 형식을 사용할 때 이벤트 파일이 Business Object를 복수 행으로 분할하는 경우, EOL Business Object 분리문자를 지정할 수 없습니다. 자세한 정보는 *Data Handler* 안내서를 참조하십시오.

다음 예는 상이한 이벤트 파일 형식에 사용할 분리문자에 대해합니다.

- 네 개의 Business Object 문자열을 포함하고 인쇄 불가능한 문자 EOL을 Business Object 분리문자의 끝으로 사용하는 파일:

```
Sample_B0~Create~1~TableGenKey5~strange~TextConnector_924055528_0
Sample_B0~Create~2~TableGenKey5~strange~TextConnector_924055528_0
Sample_B0~Create~3~TableGenKey5~strange~TextConnector_924055528_0
Sample_B0~Create~4~TableGenKey5~strange~TextConnector_924055528_0
```

- 네 개의 Business Object 문자열을 포함하고 사용자 지정 값과 인쇄 불가능한 문자 EOL을 Business Object 분리문자의 끝으로 사용하는 파일(즉, CustomEndB0;EOL):

```
Sample_B0~Create~1~TableGenKey5~strange~TextConnector_924055528_0CustomEndB0
Sample_B0~Create~2~TableGenKey5~strange~TextConnector_924055528_0CustomEndB0
Sample_B0~Create~3~TableGenKey5~strange~TextConnector_924055528_0CustomEndB0
Sample_B0~Create~4~TableGenKey5~strange~TextConnector_924055528_0CustomEndB0
```

- 네 개의 Business Object 문자열을 포함하고 인쇄 불가능한 문자 EOL;EOL을 Business Object 분리문자의 끝으로 사용하는 파일:

```
Sample_B0~Create~1~TableGenKey5~strange~TextConnector_924055528_0
Sample_B0~Create~2~TableGenKey5~strange~TextConnector_924055528_0
Sample_B0~Create~3~TableGenKey5~strange~TextConnector_924055528_0
Sample_B0~Create~4~TableGenKey5~strange~TextConnector_924055528_0
```

- 네 개의 Business Object 문자열을 포함하고 None을 Business Object 분리문자의 끝으로 사용하는 파일:

```
Sample_B0~Create~1~TableGenKey5~strange~TextConnector_924055528_0Sample_B0
~Create~2~TableGenKey5~strange~TextConnector_924055528_0Sample_B0~Create~3
~TableGenKey5~strange~TextConnector_924055528_0Sample_B0~Create~4
~TableGenKey5~strange~TextConnector_924055528_0
```

주: 커백터는 EOL 및 EOF 분리문자를 제외하고는 사용자가 지정하는 문자열을 대소문자 구분합니다.

사용자의 Meta Object 작성에 대한 자세한 정보는 61 페이지의 『특정 Business Object에 대한 JText Meta Object 작성』을 참조하십시오.

FixedBOSize 구문 분석 메소드 사용

이 Meta Object 등록 정보는 다음 경우에만 유효합니다.

1. 이벤트 처리를 수행하는 경우.
2. DataProcessingMode를 Binary로 설정하는 경우.

이 Meta Object 등록 정보가 올바른 값으로 표시되면 EndBODelimiter 등록 정보를 대체하여, 일반 분리문자 기반 Business Object 구문 분석에 대한 대안을 사용자에게 제공합니다. 이 등록 정보를 사용하면 커넥터가 고정 바이트 수와 단일 Business Object 를 상관시킬 수 있습니다. 예를 들어 파일이 300바이트로 구성되고 FixedBOSize 등록 정보를 100으로 설정한 경우, JText Adapter는 2진 사용 가능 Data Handler를 통해 이 세 가지 100바이트 길이 패킷을 변환하여 ICS로 전송합니다.

FixedBOSize 및 EndBODelimiter 모두 값 세트를 갖는 경우, Jtext는 파일 구문 분석에 FixedBOSize를 선택하고 EndBODelimiter는 무시합니다.

원격 FTP 파일 시스템 지정

이 절에서는 이벤트 및 요청 처리를 위해 원격 FTP 파일 시스템을 사용하도록 JText Connector를 구성하는 방법에 대해 설명합니다.

중요: 커넥터가 원격 FTP 파일 시스템을 사용할 수 있게 하려면 EventDir 속성(이벤트 처리의 경우) 또는 OutputDir 속성(요청 처리의 경우)에 FTP URL을 지정해야 합니다. 또한 커넥터를 사용하여 FTP 조작을 수행하기 전에 모든 방화벽 문제도 분석해야 합니다.

이 절에서는 다음에 대해 설명합니다.

- 『원격 이벤트 처리』
- 58 페이지의 『원격 요청 처리』
- 60 페이지의 『FTP 전송을 위한 커넥터 구성에 대한 참고사항』

원격 이벤트 처리

이벤트 처리를 위해 원격 FTP 파일 시스템을 사용하도록 커넥터를 구성하려면 FTP URL, FTP 로그인 정보, 커넥터가 원격 디렉토리에서 이벤트 파일을 다운로드하는 로컬 디렉토리, 아카이브 정보 및 FTP 서버가 사용 불가능할 때 커넥터가 작동하는 방법과 관련된 정보를 지정해야 합니다. 다음 절에서는 이러한 모든 구성뿐만 아니라 추가 선택적 구성도 설명합니다.

- 54 페이지의 『FTP URL 및 로그인 정보 지정』
- 54 페이지의 『로컬 디렉토리 지정』
- 55 페이지의 『원격 아카이브 지정』
- 56 페이지의 『원격 폴링 지정』
- 56 페이지의 『커넥터가 원격 사이트에서 이벤트를 처리하는 방법』

- 57 페이지의 『메인프레임에서 파일 식별: 선택적 구성』
- 57 페이지의 『이벤트 처리를 위한 구성 조作的 요약』

FTP URL 및 로그인 정보 지정: 커넥터는 EventDir Meta Object 속성에 지정된 디렉토리에서 이벤트를 폴링합니다. 이벤트 처리를 위해 원격 FTP 파일 시스템을 사용하도록 커넥터를 구성하려면 이 속성의 값으로 FTP URL을 지정하십시오. FTP URL은 IETF 표준을 준수해야 합니다.

URL에 FTP 서버를 지정하는 것 외에, 선택적으로 EventDir Meta Object 속성에 다음 정보도 지정할 수 있습니다.

- FTP 서버에 연결하고 FTP 조작을 수행할 특권이 있는 사용자의 이름 -- EventDir에 사용자 이름을 지정하지 않는 경우, FTPUserId Meta Object 속성에 지정하십시오.
- FTP 서버에 연결하고 FTP 조작을 수행할 특권이 있는 사용자의 암호 -- EventDir에 암호를 지정하지 않는 경우, FTPPassword Meta Object 속성에 지정하십시오.
- 포트 번호 -- 포트가 EventDir에 지정되지 않는 경우, 커넥터는 기본 포트를 사용합니다.
- 원격 이벤트 디렉토리 -- EventDir에 원격 이벤트 디렉토리를 지정하지 않는 경우, 커넥터는 FTP 서버에 대한 연결이 설정되는 디렉토리에서 이벤트 파일을 폴링합니다.

중요: 정적 최상위 레벨 Meta Object나 동적 하위 Meta Object에서 FTP 값을 지정할 수 있습니다. 사용자 이름과 암호가 어떠한 Meta Object 속성에도 지정되지 않는 경우, 커넥터는 FTP 서버에 연결하려 시도할 때 종료합니다. 자세한 정보는 5 페이지의 『동적 하위 Meta Object 사용』을 참조하십시오.

아래 예는 EventDir 속성값에 대한 세 가지 형식을 설명합니다.

필수 값만이 있는 URL:

ftp://ftp.companyA.com

선택적 사용자 이름과 포트 번호 값이 있는 URL:

ftp://companyA:admin@ftp.companyA.com:1433

선택적 사용자 이름, 포트 번호 및 원격 이벤트 디렉토리 값이 있는 URL:

ftp://companyA:admin@ftp.companyA.com:1433/temp/JTextConn/Default/Event

로컬 디렉토리 지정: FTP URL 및 관련 로그인 정보를 지정하는 것 외에, 사용자는 커넥터가 원격 디렉토리에서 이벤트 파일을 다운로드하는 로컬 디렉토리의 위치도 지정해야 합니다. 로컬 디렉토리를 지정하려면 FTPLocalEventDir Meta Object 속성을 사용하십시오.

중요: 커넥터가 EventDir에서 적당한 FTP URL을 찾지만 FTPLocalEventDir Meta Object 속성을 찾지 못하거나 이 속성에 대해 유효하지 않거나 공백인 값을 찾는 경우, 커넥터는 시작하지 않습니다. 커넥터는 로컬로 실행하도록 구성될 때 FTPLocalEventDir 속성을 평가하지 않습니다.

원격 아카이브 지정: 커넥터가 원격 아카이브를 처리하는 방법 지정에는 여러 가지 옵션이 있습니다. 원격 아카이브 디렉토리를 지정하려면 FTPArchiveDir Meta Object 속성을 사용하십시오. 이 속성은 FTP 서버에 있는 아카이브 디렉토리의 절대 경로를 지정합니다. 이 디렉토리는 이미 존재해야 합니다. 이 속성을 사용하여 아카이브를 지정하기 위한 여러 옵션이 있습니다.

- FTPArchiveDir 속성에 대한 값을 지정하고 FTPRenameExt 속성에 대한 값을 지정하지 않는 경우, 커넥터가 이벤트 파일 이름에 시간 소인을 추가하고 이 파일을 FTPArchiveDir 속성에 지정된 원격 FTP 서버 아카이브 디렉토리로 이동시킵니다.
- FTPArchiveDir 속성 및 FTPRenameExt 속성 모두에 대한 값을 지정하면 커넥터가 처리된 이벤트 파일의 이름을 바꿔 시간 소인을 추가하고 FTPRenameExt를 무시한 후 FTPArchiveDir 속성에 지정된 FTP 서버 아카이브 디렉토리로 해당 파일을 이동시킵니다.
- FTPArchiveDir 또는 FTPRenameExt 속성에 대한 값을 지정하지 않으면, 커넥터는 처리된 이벤트 파일을 아카이브하지 않고 삭제합니다.
- FTPArchiveDir 속성에 대한 값을 지정하지 않고 FTPRenameExt 속성에 대한 값을 지정하면, 커넥터가 처리된 이벤트 파일의 이름을 FTPRenameExt에 지정된 값으로 바꾸고 EventDir 속성에 지정된 디렉토리로 이동시킵니다.

원격 FTP 서버에 대한 시간 소인이 있는 파일 이름 지정: 중복 파일 이름을 피하기 위한 시간 소인을 제공함으로써 순차 데이터 세트를 사용하는 호스트 파일 시스템(MVS)에 대한 지원이 향상되었습니다. MVS는 데이터 세트 또는 레코드 세트 이름에 "_"와 같은 특수 문자를 지원하지 않습니다. Windows 또는 Unix 플랫폼의 경우, 파일 아카이브 시 원래 파일 이름의 시간 소인을 사용하며, 이를 통해 아카이브 폴더에 파일 이름이 중복되지 않아 기존 파일을 겹쳐쓰지 않게 됩니다.

MVS 시스템에 다음 형식을 사용하여 이 한계를 극복합니다.

이벤트 파일: Test.in

아카이브 파일: Test.TSyyyyMM.TSDDHHMM.TSSsSss

각 요소에 대한 설명은 다음과 같습니다.

yyyy -- 년
MM -- 월
DD -- 날짜
HH -- 시간
MM -- 분
Ss -- 초

Sss -- 밀리초

MVS 플랫폼의 경우 데이터 세트 또는 레코드 세트 분리자는 "."(점)이며 데이터 세트 또는 레코드 세트에서 허용되는 최대 '.'(점) 수는 6입니다. 데이터 세트 또는 레코드 세트 이름은 "."(점) 당 여덟 자를 초과해서는 안 되며 총 문자 수는 44자를 초과해서는 안 됩니다. 다음은 이 형식의 파일 이름 예입니다.

FTPRenameExt -- ARCHIVE

아카이브 파일 -- (SAMPLE).ARCHIVE.TS200304.TS290535.TS42234

주: 아카이브 시 PDS 구성원의 이름을 시간 소인으로 바꿀 수 없으므로 JText Aapter는 MVS에서 PDS를 지원하지 않습니다.

원격 폴링 지정: 22 페이지의 『FTPPollFrequency』 구성 등록 정보를 사용하여 표준 폴 주기의 수로 측정된 커넥터가 FTP 서버를 폴링하는 빈도를 설정할 수 있습니다. 이 설정은 커넥터가 다음 폴링 주기를 시작할 때 여전히 로컬 이벤트 디렉토리에서 파일을 읽고 있는 경우에 유용합니다.

예를 들어, PollFrequency가 10000으로 설정되고 FTPPollFrequency가 6으로 설정되는 경우, 커넥터는 10초마다 로컬 이벤트 디렉토리를 폴링하고 60초마다 원격 디렉토리를 폴링합니다. 커넥터는 사용자가 이 등록 정보에 대해 값을 지정하는 경우에만 FTP 폴링을 수행합니다. FTPPollFrequency가 0 또는 공백으로 평가되는 경우, 커넥터는 FTP 폴링을 수행하지 않습니다.

자세한 정보는 64 페이지의 『JText Connector의 성능 조정』을 참조하십시오.

커넥터가 원격 사이트에서 이벤트를 처리하는 방법: 원격 사이트에서 이벤트를 폴링할 때 커넥터는 다음 단계를 수행합니다.

1. Meta Object 속성 또는 기본값으로부터 서버 이름, 포트 번호, 사용자 이름, 암호 및 원격 이벤트 디렉토리를 얻습니다.
2. 원격 FTP 사이트에 대한 연결을 설정하여 원격 이벤트 디렉토리에서 이벤트 파일을 가져옵니다.
3. 원격 디렉토리에서 FTPLocalEventDir Meta Object 속성에 지정되는 로컬 디렉토리로 이벤트 파일을 다운로드합니다.

주: 커넥터가 FTP를 사용하여 이벤트를 처리할 수 있게 하려면 이 속성에 값이 있어야 합니다.

4. 로컬 디렉토리를 폴링합니다.

그림 6은 로컬 및 원격 이벤트 처리를 보여줍니다.

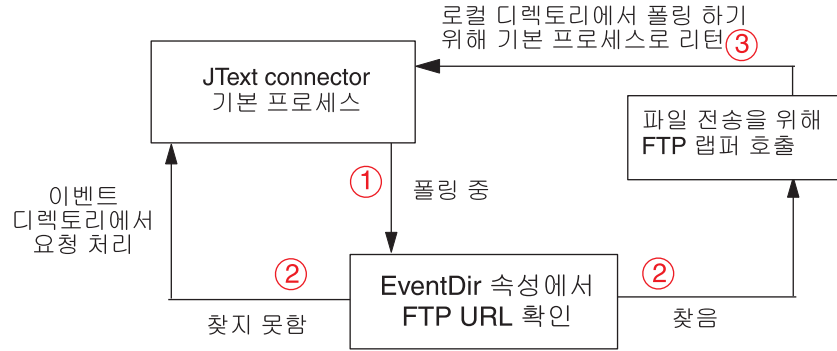


그림 6. 로컬 및 원격 이벤트 공고 조작

메인프레임에서 파일 식별: 선택적 구성: FTPEventFileMask 속성을 사용하여 UNIX 또는 Windows 시스템에 적용되는 동일한 이름 지정 표준을 따르지 않는 메인프레임에서 파일 확장자를 식별하십시오. 이 속성에 대한 값이 제공되지 않는 경우, 커넥터는 EventExt 속성에 지정되는 값을 사용합니다.

FTPEventFileMask에 대한 값을 지정할 때 와일드 카드 문자를 포함할 수 있습니다. 다음 예는 이 속성에 대해 가능한 몇 가지 형식을 보여줍니다.

```
ACT.Z1UC.*
ACT.*.INPT
*.Z1UC.INPT
```

커넥터가 FTPEventFileMask에 지정되는 기준을 충족시키는 원격 사이트에서 둘 이상의 파일을 찾는 경우 다음을 수행합니다.

1. 지정된 모든 원격 이벤트 파일을 FTPLocalEventDir 속성에 지정되는 디렉토리에 다운로드합니다.
2. FTPRenameExt Meta Object 속성에 지정되는 값으로 원격 파일 확장자의 이름을 바꿉니다. 파일 이름을 바꾸면 커넥터가 다음 폴 주기에서 동일한 파일을 폴링하지 못하게 됩니다.
3. FTP 서버에서 연결을 끊습니다.
4. FTPEventFileMask Meta Object 속성에 지정되는 디렉토리에서 해당 파일을 로컬로 처리합니다.

이벤트 처리를 위한 구성 조작의 요약: 이벤트 처리를 위해 원격 FTP 파일 시스템을 사용하도록 커넥터를 구성하려면 다음 구성 값을 지정하십시오.

- EventDir Meta Object 속성에 FTP URL을 지정하십시오. 선택적으로 FTP 서버에 연결하고 FTP 조작을 수행할 특권이 있는 사용자의 이름과 암호를 지정하십시오.
- EventDir Meta Object 속성에 로그인 이름과 암호를 지정하지 않는 경우, FTPUserId 및 FTPPassword Meta Object 속성에서 지정하십시오.

- EventDir Meta Object 속성에 포트를 지정하지 않는 경우, 커넥터는 기본 FTP 포트를 사용합니다.
- FTPLocalEventDir Meta Object 속성을 사용하여 커넥터가 FTP 사이트에서 이벤트를 파일을 다운로드하는 로컬 시스템 디렉토리를 지정하십시오.
- UNIX 또는 Windows 시스템에 적용되는 동일한 이름 지정 표준을 따르지 않는 메인프레임에서는 FTPEventFileMask Meta Object 속성을 사용하여 폴링할 파일을 식별하십시오.
- 원격 시스템이 MVS일 때 MVS FTP 서버에 대해 작업하도록 커넥터를 구성하려면 FTPOsPlatform 속성에 MVS를 지정하십시오.

원격 요청 처리

요청 처리를 위해 원격 FTP 파일 시스템을 사용하도록 커넥터를 구성하려면 FTP URL, FTP 로그인 정보 및 커넥터가 로컬 디렉토리에서 요청 파일을 업로드하는 원격 디렉토리를 지정해야 합니다. 다음 절에서는 이러한 모든 구성뿐만 아니라 추가 선택적 구성도 설명합니다.

- 『FTP URL 및 로그인 정보 지정』
- 59 페이지의 『커넥터가 원격 사이트로의 서비스 호출 요청을 처리하는 방법』
- 59 페이지의 『요청 처리를 위한 구성 조작의 요약』

FTP URL 및 로그인 정보 지정: 커넥터는 OutputDir Meta Object 속성에 지정된 디렉토리에 서비스 호출 요청 파일을 업로드합니다. 요청 처리를 위해 원격 FTP 파일 시스템을 사용하도록 커넥터를 구성하려면 이 속성의 값으로 FTP URL을 지정하십시오. FTP URL은 IETF 표준을 준수해야 합니다.

FTP URL 외에, 선택적으로 OutputDir Meta Object 속성에 다음 정보를 지정할 수 있습니다.

- FTP 서버에 연결하고 FTP 조작을 수행할 특권이 있는 사용자의 이름 -- OutputDir에 사용자 이름을 지정하지 않는 경우, FTPPassword Meta Object 속성에 지정하십시오.
- FTP 서버에 연결하고 FTP 조작을 수행할 수 있는 특권이 있는 갖는 사용자의 암호 -- OutputDir에 암호를 지정하지 않은 경우, FTPPassword Meta Object 속성에 지정하십시오.
- 포트 번호 -- 포트가 EventDir에 지정되지 않는 경우, 커넥터는 기본 포트를 사용합니다.
- 원격 출력 디렉토리 -- OutputDir에 원격 출력 디렉토리를 지정하지 않는 경우, 커넥터는 기본 연결 디렉토리(연결이 설정되는 FTP 서버의 디렉토리)에 요청 파일을 로드합니다.

중요: 정적 최상위 레벨 Meta Object나 동적 하위 Meta Object에서 FTP 값을 지정할 수 있습니다. 사용자 이름과 암호가 어떠한 Meta Object 속성에도 지정되지 않는 경우, 커넥터는 예외를 생성하여 종료합니다. 자세한 정보는 5 페이지의 『동적 하위 Meta Object 사용』을 참조하십시오.

아래 예는 OutputDir 속성값에 대한 세 가지 형식을 설명합니다.

필수 값만이 있는 URL:

ftp://ftp.companyA.com

선택적 사용자 이름과 포트 번호 값이 있는 URL:

ftp://companyA:admin@ftp.companyA.com:1433

선택적 사용자 이름, 포트 번호 및 원격 출력 디렉토리 값이 있는 URL:

ftp://companyA:admin@ftp.companyA.com:1433/temp/JTextConn/Default/Out

커넥터가 원격 사이트로의 서비스 호출 요청을 처리하는 방법: 커넥터가 FTP 처리에 대해 구성되고 서비스 호출 요청을 수신할 때 다음 단계를 수행합니다.

1. Meta Object 속성 또는 기본값으로부터 서버 이름, 포트 번호, 사용자 이름 및 암호를 얻습니다.
2. 원격 FTP 사이트에 대한 연결을 설정하여 로컬 디렉토리에서 서비스 호출 요청 파일을 찾습니다.
3. 로컬 디렉토리에서 원격 디렉토리로 요청 파일을 업로드합니다.
4. 원격 서버에서 연결을 끊습니다.

그림 7은 로컬 및 원격 요청 처리를 보여줍니다.

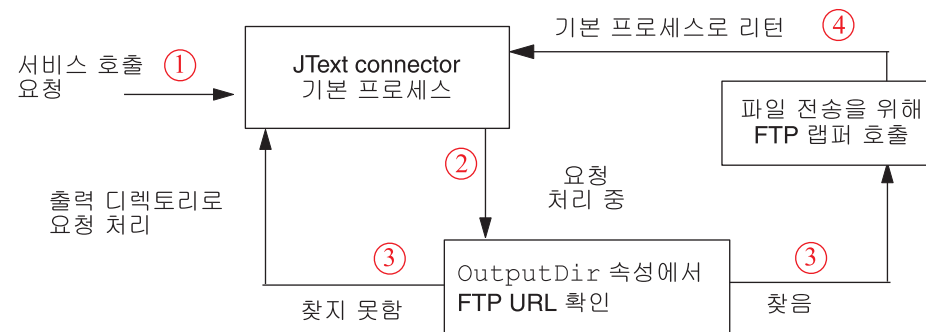


그림 7. 로컬 및 원격 요청 조작

요청 처리를 위한 구성 조작의 요약: 요청 처리를 위해 원격 FTP 파일 시스템을 사용하도록 커넥터를 구성하려면 다음 구성값을 지정하십시오.

- OutputDir Meta Object 속성에 FTP URL을 지정하십시오. 선택적으로 FTP 서버에 연결하고 FTP 조사를 수행할 특권이 있는 사용자의 이름과 암호를 지정하십시오.
- OutputDir Meta Object 속성에 로그인 이름과 암호를 지정하지 않는 경우, FTPUserId 및 FTPPassword Meta Object 속성에서 지정하십시오
- OutputDir Meta Object 속성에 포트를 지정하지 않는 경우, 커넥터는 기본 FTP 포트를 사용합니다.
- 원격 시스템이 MVS일 때 MVS FTP 서버에 대해 작업하도록 커넥터를 구성하려면 FTPOsPlatform 속성에 MVS를 지정하십시오.

FTP 전송을 위한 커넥터 구성에 대한 참고사항

다음 기능이 데이터의 FTP 전송에 적용됩니다.

- 커넥터는 FTP 조사를 수행할 때 2진 모드의 데이터 전송을 사용합니다.
- EventDir 또는 OutputDir Meta Object 속성이 ftp://로 시작하지 않는 경우, 커넥터는 데이터의 FTP 전송을 사용하지 않습니다.
- 이벤트 처리 중에 이벤트 Business Object가 InFileName 속성과 함께 동적 하위 Meta Object를 포함하는 경우, 커넥터는 원격 시스템의 경로가 아닌 FTPLocalEventDir에 지정되는 파일의 전체 경로로 이 속성을 채웁니다.
- EventExt 및 FTPRenameExt Meta Object 속성에 입력되는 값은 동일할 수 없습니다. 두 값이 동일한 경우 커넥터는, 이전에 이미 선택된 파일을 계속 선택합니다.
- 커넥터는 FTP가 지원하지 않는 파일 크기를 지원하지 않습니다.
- FTP 사이트의 플랫폼에 따라서 파일 이름, 확장자 및 기타 구성요소에 대한 대소문자 구분을 고려해야 합니다.
- 원격 FTP 사이트로부터의 파일 전송은 커넥터 성능에 영향을 줄 수 있습니다.
- 데이터가 원격 FTP 사이트와의 사이에서 교환되는 경우, 데이터가 끊어진 네트워크 연결 또는 유사한 문제점으로 인해 손상되거나 유실될 수 있습니다.
- 통합 브로커는 어떠한 유형의 연결 캐시 또는 풀도 유지보수하지 않습니다. 연결은 각 폴링 주기 및 요청 처리에 대해 열리고 닫힙니다. 네트워크 대기시간 및 커넥터의 제어를 받지 않는 기타 구성이 커넥터 성능에 영향을 줄 수 있습니다.
- FTPLocalEventDir Meta Object 속성에 지정되는 값은 FTP 값을 지정하지 않는 모든 Meta Object의 EventDir Meta Object 속성의 값으로 지정될 수 없습니다. 이 제한은 커넥터가 동일한 디렉토리에서 완전히 다른 유형의 처리가 필요한 상이한 유형의 Business Object에 지정되는 값을 사용하지 못하게 합니다.
- FTP URL의 끝에 지정되는 원격 이벤트 디렉토리 또는 출력 디렉토리가 존재하지 않는 경우, 커넥터는 FTP 사이트와 상호작용할 때 종료합니다. 커넥터 시작 시에 종료하지 않습니다.

- FTP 서버에서의 파일 처리를 위해 커넥터를 구성할 때 NetComponents API에 필요한 UNIX 설정을 사용하도록 FTP 서버를 구성해야 합니다.

Data Handler 지정

JText Connector가 사용할 Data Handler를 지정하려면 다음 단계를 수행하십시오.

1. JText Connector가 통신하는 응용프로그램이 사용하는 형식을 판별하십시오. 제공된 모든 형식 유형에 대해 하나의 Data Handler 클래스만이 등록할 수 있음에 유의하십시오.
2. 최상위 레벨 JText Meta Object의 다음 하위 오브젝트를 구성하십시오.
 - EventHandler -- 이벤트 처리에 사용할 Data Handler Meta Object를 지정합니다(Business Object 문자열 또는 바이트 배열을 Business Object 변환으로).
 - OutputDataHandler -- 요청 처리에 사용할 Data Handler Meta Object를 지정합니다(Business Object를 Business Object 문자열 또는 바이트 배열 변환으로).

지정된 Data Handler 변경

제공되는 기본값에서 Data Handler를 (제공되는 다른 기본값 또는 사용자 정의 Data Handler로) 변경하려면 다음을 수행하십시오.

- 커넥터가 EventHandler 및 OutputDataHandler 속성에 기본값으로 지정되는 Business Object를 지원하는지 검증하십시오.
- Data Handler를 포함하는 클래스 또는 Jar 파일이 커넥터가 시작될 때 클래스 경로에 포함되는지 검증하십시오. 전달된 Data Handler를 사용하거나 사용자 정의 Data Handler를 CustDataHandler.jar 파일에 추가하는 경우(Data Handler 안내서에서 권장된 대로), 파일은 전달된 시작 스크립트(start_JText.bat 또는 connector_manager_JText.sh)에 포함되어 있습니다.

Data Handler 작성에 대한 자세한 정보는 *Data Handler* 안내서를 참조하십시오.

특정 Business Object에 대한 JText Meta Object 작성

특정 Business Object에 대한 JText Meta Object를 작성할 때 특정 Business Object를 식별하기 위해 Meta Object의 이름을 바꾸십시오. 예를 들어, 고객 및 항목 Business Object에 대한 Meta Object를 작성하려면 Meta Object 이름을 MO_JTextConnector_Customer 및 MO_JTextConnector_Item으로 지정할 수 있습니다.

팁: 파일에 기록할 모든 Business Object에 정확하게 동일한 구성이 있을 때 기본 Meta Object를 사용하십시오. 즉, 모든 텍스트 파일이 동일한 이벤트 디렉토리에 상주하고 동일한 출력 디렉토리에 기록되고 동일한 Data Handler를 사용하고 동일한 파일 확장자가 있습니다. (또는 동일한 파일에 저장됩니다.) 커넥터가 상이한 Business

Object 또는 요청에 대해 서로 다른 처리를 사용해야 하거나 특정 처리 지시사항이 폴링을 위해 필요한 경우, 사용자의 Meta Object를 작성하십시오. 특정 Business Object에 대해 별도의 Meta Object를 작성하는 경우, 커넥터는 통합 브로커 요청 및 등록 전달 조작 모두에 대해 사용자의 Meta Object를 사용합니다.

사용자가 Meta Object를 작성하지 않는 모든 Business Object는 기본 MO_JTextConnector_Default Meta Object에 있는 값에 의해 구성됩니다. 이 기본 Meta Object의 Business Object 정의에 대한 정보는 \repository\JText 디렉토리를 참조하십시오.

동일한 파일에서 상이한 유형의 복수 Business Object 읽기

텍스트 파일이 상이한 유형의 복수 Business Object를 포함하는 경우, MO_JTextConnector_Default Meta Object를 사용해야 하고 EventExt 및 EventDir 속성이 이 이벤트 파일이 위치하는 디렉토리를 올바르게 가리키는지 확인해야 합니다. 파일의 각 Business Object는 동일한 분리문자에 의해 구분되어야 합니다.

제공되는 Data Handler가 입력 문자열로부터 각 Business Object의 이름을 판별할 수 있습니다. 즉, 기본 최상위 레벨 JText Meta Object와 제공되는 Data Handler를 사용할 때 복수 유형을 포함하는 한 파일에 있는 Business Object의 각 유형을 식별하기 위해 <EndBO:BOName> 분리문자를 사용할 필요가 없습니다.

Business Object 문자열을 Business Object로 변환하는 사용자 정의 Data Handler를 개발하는 경우, 그것이 입력 문자열의 Business Object 유형을 해석할 수 있는지 확인하십시오.

ObjectEventID 속성에 대한 값 지정

Business Object 문자열에 ObjectEventID 속성을 추가할 필요가 없습니다. 이벤트 공고 Business Object의 경우, ID가 커넥터에 의해 채워지지 않으면 커넥터 프레임워크가 이들 Business Object 속성을 채웁니다.

서비스 호출 요청 Business Object의 경우, ObjectEventID는 무시되거나 파일에 기록되는 문자열에 포함됩니다. ObjectEventID 속성이 출력 파일에 포함되는지 여부는 사용되는 Data Handler에 따라 다릅니다.

JText Connector의 2차 인스턴스 설정

JText Connector의 2차 인스턴스를 설정하려면 다음 단계를 수행하십시오.

1. JText Connector 디렉토리 및 해당 저장소 디렉토리의 사본을 작성하고 이름을 바꾸십시오. 예를 들어, 2차 커넥터 정의의 이름을 JText2로 지정한다고 가정하십시오. 2차 디렉토리를 작성한 후 디렉토리 구조는 다음과 유사합니다.

\connectors\JText
\connectors\JText2
\repository\JText
\repository\JText2

2. JText Connector에 대한 모든 Meta Object(그 중 최소한 둘 이상이 있어야 함)를 복사하고 Business Object의 이름을 수정하십시오. 예를 들어, JText2 커넥터의 경우 이름을 MO_JTextConnector_BOName에서 MO_JText2Connector_BOName으로 변경하십시오.

Meta Object를 복사하는 방법에는 두 가지가 있습니다.

- MO_JText2Connector_BOName Meta Object 및 해당 하위를 포함하는 텍스트 파일을 작성하십시오. 문서 편집기의 검색 및 바꾸기 옵션을 사용하여 MO_JTextConnector_를 MO_JText2Connector_로 바꾸십시오.
- Business Object Designer를 사용하여 한번에 하나씩 Meta Object를 복사하십시오.

중요: Business Object Designer에서 Business Object 정의를 조작하기 전에 \repository\ReposVersion.txt 파일의 맨 위에서 모든 정의 파일의 맨 위로 텍스트를 복사해야 합니다.

3. Connector Configurator에서, 커넥터의 정의를 복사하고 이름을 JText2Connector로 바꾸십시오. 지원되는 Meta Object 및 Business Object를 변경하십시오.
4. 모든 새 정의 파일을 저장소에 복사하십시오. Business Object Designer를 사용하여 Business Object 정의를 저장소에 복사하려면 파일 메뉴에서 서버에 저장 하위 메뉴를 선택하십시오. 이 외에, ICS에서 다음 단계를 수행하여 운영 체제로부터 Business Object 정의를 저장소에 복사하십시오.
 - a. \repository\ReposVersion.txt 파일의 맨 위에서 모든 정의 파일의 맨 위로 텍스트를 복사하십시오.
 - b. 다음 repos_copy 명령을 사용하여 새 Meta Object에 복사하십시오.

```
repos_copy -sServerName -iFileName
```
5. 통합 브로커의 관리 유틸리티를 새로 고쳐 새 Business Object를 확인하십시오.
6. UNIX의 경우, JText Connector에 대한 기존 커넥터 관리자 스크립트 사본을 작성하고 JText2를 참조하도록 매개변수를 변경하십시오. Windows의 경우, JText Connector에 대한 기존 바로 가기의 사본을 작성하고 JText2를 참조하도록 매개변수를 변경하며, JText 디렉토리보다는 JText2 디렉토리를 가리키도록 수정하십시오.
7. 새 커넥터에 대해 새 MQ 대기열을 추가하십시오. 추가 방법에 대한 자세한 정보는 *Implementation Guide for WebSphere MQ Integrator Broker* 또는 *System Administration Guide*를 참조하십시오.
8. 통합 브로커를 다시 시작하십시오.

9. UNIX에서 커넥터 관리자 스크립트를 실행하십시오. Windows에서는 새 바로 가기를 누르십시오.

JText Connector의 성능 조정

JText Connector의 폴링 성능을 조정하려면 아래에서 설명하는 대로 다음 커넥터 구성 등록 정보를 설정하십시오.

- **PollQuantity** - 이 등록 정보는 커넥터가 이벤트 폴링에 대한 단일 호출에서 통합 브로커로 전달할 수 있는 비즈니스 오브젝트의 최대수를 설정합니다. PollQuantity를 높은 값으로 설정하는 경우, 커넥터는 하나의 폴에서 더 많은 Business Object를 제출하려 시도합니다. 이것은 성능을 향상시킬 수 있고 내부 대기열 및 메모리 사용을 지우는 데 도움이 됩니다.

그러나 커넥터가 많은 양의 Business Object를 통합 브로커에 게시할 수 있게 하면 다른 비즈니스 통합 구성요소에 영향을 줄 수 있습니다. 예를 들어, 메시지 대기열 시스템이 기본값으로 설정된 경우, JText Connector가 시스템을 통해 많은 대형 Business Object를 전송하면 대기열이 빨리 채워질 수 있습니다. 그러므로 성능을 조정할 때 PollQuantity에 대한 최적 성능 설정이 있음을 기억하십시오.

- **PollFrequency** - 이 커넥터 구성 등록 정보는 폴링 조치 사이의 시간을 지정합니다. 이 등록 정보를 더 긴 시간으로 설정하면 이벤트 처리 중에 커넥터가 느려집니다. 더 짧은 시간으로 설정하면 이벤트가 빨리 선택되고 Business Object로 변환되고 전달되도록 합니다.

즉, 커넥터는 각 폴 호출 중에 새 파일을 선택합니다. 커넥터가 자주 폴링하지 않는 경우, 이벤트 디렉토리에서 발생하는 파일을 전달하는 데 더 오래 걸립니다. 커넥터가 자주 폴링하는 경우, 커넥터가 더 자주 파일을 선택하고 더 자주 파일을 전달합니다.

그러나 커넥터가 이벤트를 더 자주 폴링할수록 요청을 처리하는 데 더 적은 시간이 소요됩니다. 주로 요청 처리를 위해 커넥터를 사용하는 경우, 커넥터를 주로 이벤트 처리에 사용하는 경우보다 PollFrequency를 더 낮은 값으로 설정하십시오.

위에서 논의한 PollQuantity 구성 등록 정보에서와 같이, PollFrequency를 아주 길거나 짧은 시간 같은 극한값으로 설정하는 것이 다른 비즈니스 통합 구성요소의 성능에 영향을 줄 수 있습니다.

- **FTPPollFrequency** - 이 커넥터 구성 등록 정보는 커넥터가 FTP 서버를 폴링하는 빈도를 표준 폴 주기의 수로 지정합니다. 예를 들어, PollFrequency가 10000으로 설정되고 FTPPollFrequency가 6으로 설정되는 경우, 커넥터는 10초마다 로컬 이벤트 디렉토리를 폴링하고 60초마다 원격 디렉토리를 폴링합니다. 커넥터는 사용자가 이 등록 정보에 대해 값을 지정하는 경우에만 FTP 폴링을 수행합니다. FTPPollFrequency가 0 또는 공백으로 평가되는 경우, 커넥터는 FTP 폴링을 수행하지 않습니다.

요약하면, 폴링에서 성능을 향상시키기 위한 최상의 접근 방법은 PollQuantity, PollFrequency 및 FTPPollFrequency를 서로 보완하도록 설정하는 것입니다.

테스트를 위한 샘플 파일 생성

JText Connector가 기대하는 입력 파일과 유사한 파일을 생성하기 원할 수 있습니다. 이 파일은 소스 응용프로그램에서 출력 형식을 설정하는 데 도움을 줄 수 있습니다. 샘플 파일은 또한 테스트에 사용할 수 있습니다.

ICS에서 입력 파일과 유사한 파일을 생성하는 가장 쉬운 방법은 다음과 같습니다.

1. 파일에 기록할 Business Object를 입력으로 하고 목적지에 전송하는 pass-through 협업을 작성합니다.
2. 소스 포트를, 해당 Business Object를 지원하고 Test Connector에 의해 에뮬레이션될 수 있는 커넥터에 바인드합니다.
3. 목적지 포트를 JText Connector에 바인드합니다.
4. Business Object에 대한 샘플 값을 Test Connector에 입력하고 해당 Business Object를 JText Connector에 전송합니다. JText Connector가 구성된 형식으로 출력 파일에 값을 기록합니다.

이 프로세스는 단일 파일에 기록된 복수 Business Object를 볼 수 있게 하는데, 이 파일을 테스트 중에 입력으로 사용할 수 있습니다.

테스트를 위한 샘플 Business Object 생성

JText Connector가 기대하는 것과 유사한 Business Object를 생성하기 원할 수 있습니다. 테스트 중에 사용할 값으로 Business Object를 채울 수 있습니다.

커넥터가 Business Object 템플릿을 자동으로 생성하게 하려면 GenerateTemplate 구성 등록 정보를 사용하십시오. 커넥터가 지원하는 각 Business Object에 대한 정의를 생성할 수 있습니다.

커넥터는 GenerateTemplate 등록 정보의 값을 사용하여 커넥터가 시작할 때 직렬화된 Business Object의 인스턴스를 작성합니다. 직렬화된 Business Object는 Data Handler가 작성하는 Business Object의 문자열 표시입니다. Connector Configurator를 사용하여 이 등록 정보에 대한 Business Object의 이름을 지정하십시오.

이 등록 정보에 대한 구문은 *BOName;BOName*으로, 여기서 특정 Business Object 이름의 이름이 *BOName*으로 대체됩니다. 대소문자 구분이 중요합니다. 둘 이상의 Business Object를 지정하려면 *Customer;Item*에서와 같이 이름을 세미콜론으로 구분하십시오. 종료 구두점은 필요하지 않습니다. 이들 Business Object에 대한 템플릿은 사용자가 다음에 커넥터를 시작할 때 작성됩니다.

생성된 템플리트는 Business Object의 정의에서 Business Object의 속성에 대해 설정되는 제공된 기본값을 포함합니다. 속성에 대해 제공되는 기본값이 없는 경우, 기본값은 무시되거나(CxIgnore를 사용하여) 공백으로 남습니다(CxBlank를 사용하여). 각 단일 카디널리티 하위 Business Object에 대해 하나의 하위 Business Object가 작성되고 다중 카디널리티 Business Object에 대해 하위 Business Object의 동일한 인스턴스 두 개가 작성됩니다.

지정된 Business Object에 대해 템플리트 생성을 시작하려면 커넥터를 시작하십시오. 커넥터는 출력 파일과 동일한 파일에 템플리트를 기록합니다. 이 기능을 사용하지 않으려는 경우, GenerateTemplate 등록 정보를 공백으로 두십시오.

제 4 장 JText Connector 문제점 해결

이 장은 JText Connector의 문제점을 진단하는 데 도움이 되는 다음 정보를 포함합니다.

- 『오류 메시지 로깅』
- 『Meta Object 이름 지정에 대한 문제점』
- 68 페이지의 『이벤트 트리거링에 대한 문제점』
- 68 페이지의 『JText 장애 핸들링』
- 69 페이지의 『이벤트 로그 파일』
- 69 페이지의 『장애 복구』
- 71 페이지의 『Business Object 분리문자 오류 복구』
- 72 페이지의 『등록 오류 복구』
- 72 페이지의 『형식화 오류 복구』
- 72 페이지의 『전송 오류 복구』
- 73 페이지의 『Data Handler 및 지원되는 Business Object』

오류 메시지 로깅

오류 메시지는 표준 커넥터 로그 파일 STDOUT 또는 LogFileName 표준 커넥터 등록 정보에 의해 지정된 파일에됩니다.

오류는 또한 이벤트 로그 파일에도 로그됩니다. 이벤트 로그 파일에 대한 자세한 정보는 69 페이지의 『이벤트 로그 파일』을 참조하십시오.

Meta Object 이름 지정에 대한 문제점

커넥터 시작 중에 다음 오류 메시지는 Meta Object 이름이 커넥터 인스턴스 이름에 대응하지 않음을 의미합니다.

```
Wrong subscription: JText_Customer doesn't have supporting MO: this BO is unsubscribed."
```

Meta Object 이름이 커넥터 인스턴스의 이름에 대응하지 않는 경우, Meta Object는 커넥터가 지원하는 Business Object를 인식하지 않습니다. 이를 방지하려면 커넥터 인스턴스와 대응하도록 Meta Object의 이름을 지정하십시오. 예를 들어, MO_JText2Connector_Default라는 Meta Object는 JText2 커넥터가 지원하는 Business Object를 인식합니다.

이벤트 트리거링에 대한 문제점

커넥터는 다음 분리문자 문제점이 있는 이벤트 파일을 무시합니다.

- 최상위 레벨 Meta Object의 EndBODelimiter 속성이 유효한 값(예: 더하기 부호(+)
또는 파이프 기호(|))으로 설정되지만 이벤트 파일이 각 Business Object의 끝에 지
정된 분리문자를 포함하지 않습니다.
- 커넥터가 EndBO:BOName Business Object 분리문자를 찾도록 구성되지만 이벤트 파
일이 이 분리문자를 포함하지 않습니다. 커넥터는 다음을 명시하는 경고 메시지를 로
그합니다.

```
Unable to create Workunits from file filename. Check EndBODelimiter in the file.
```

위의 경우 모두에서 파일은 어떠한 변경사항도 없이 이벤트 디렉토리에 남아 있습니다.

커넥터는 또한 파일에 액세스하거나 파일이 열리거나 닫히는 중에 장치 장애가 발생할
때 변경 없이 이벤트 디렉토리에 파일을 보관합니다. 예를 들어, 시스템이 메모리가 부
족할 때 파일에 액세스하려고 시도하는 경우 커넥터는 해당 파일을 무시합니다.

JText 장애 핸들링

JText Connector의 경우 다음 유형의 오류가 발생할 수 있습니다.

표 8. JText 오류 유형

오류의 유형	설명
Business Object 분리문자 장애	최상위 레벨 Meta Object의 EndBODelimiter 속성이 유효한 값으로 설정 되고 이벤트 파일이 각 Business Object의 끝에 지정된 분리문자를 포함하 지만 데이터 자체가 해당 텍스트에 분리문자 값을 사용하는 경우, Business Object 분리문자 장애가 발생합니다. 커넥터가 텍스트에서 분리문자 값을 받 건할 때 포맷터에 부분 Business Object 문자열을 전송하고, 해당 포맷터는 처리에 실패합니다. 이 경우에 커넥터는 이벤트를 <i>filename_timestamp.fail</i> 파일에 기록하는데, 여기에는 분리문자 장애가 발생한 모든 Business Object 에 대한 레코드를 포함합니다.
등록 오류	커넥터가 Business Object 분리문자를 찾고 Business Object 이름을 검색 할 수 있지만 Business Object가 등록되지 않은 경우에 발생할 수 있습니 다. 이 경우에 이벤트가 <i>filename_timestamp.unsub</i> 파일에 전송되는데, 이 파일에는 등록되지 않은 모든 Business Object에 대한 레코드가 들어 있습 니다.
형식화 오류	커넥터가 입력 Business Object 이름과 대응하지 않는 Business Object 이 름이 있는 분리문자를 찾거나 Business Object 파일의 형식이 Meta Object 의 형식과 일치하지 않는 경우에 발생할 수 있습니다. 이벤트가 <i>filename_timestamp.fail</i> 파일에 전송되는데, 이 파일에는 형식화에 실 패한 모든 Business Object에 대한 레코드가 들어 있습니다.
전송 오류	통합 브로커가 작동 중지될 때 커넥터가 Business Object를 전송하려고 시 도하는 경우에 발생할 수 있습니다. 전송 조작이 실패하는 경우 이벤트가 <i>filename_timestamp.fail</i> 파일에 전송되는데, 이 파일에는 성공적으로 전 송되지 않은 모든 Business Object에 대한 레코드를 포함합니다.

이벤트 로그 파일

커넥터는 성공적으로 처리된 Business Object에 대한 정보를 event.log 파일에 로그합니다. 커넥터가 이벤트 파일의 모든 Business Object를 처리하기 전에 작동 중지되는 경우 커넥터는 복구 중에 이 로그 파일을 사용하여 각 Business Object를 한 번만 통합 브로커에 전송합니다.

로그 파일의 형식은 다음과 같습니다.

```
EventFileName ::1,2,n
```

여기서 EventFileName은 현재 이벤트 파일의 이름이고, 각 번호는 해당 파일에 있는 성공적으로 처리된 Business Object의 순번을 나타냅니다.

예를 들어, 커넥터가 Customer.in 파일에 있는 처음 네 개의 Business Object 중 세 개를 성공적으로 처리했으며 두 번째 Business Object가 처리에 실패했다고 가정합니다. 또한 커넥터가 아직 Customer.in 처리를 완료하지 않았다고도 가정합니다. 이 경우에 event.log 파일은 UNIX에서 다음과 유사할 수 있습니다.

```
$ProductDir/JText/Event/Customer.in :: 1,3,4
```

Windows에서는 다음과 유사할 수 있습니다.

```
C:\JText\Event\Customer.in :: 1,3,4
```

전체 Customer.in 파일을 처리하기 전에 커넥터가 정지된 경우, 시작 시에 커넥터는 로그 파일의 정보를 사용하여 처리를 중지한 시점에서 이벤트 파일 처리를 재개합니다. 커넥터는 복구할 이벤트 파일의 이름과 마지막 Business Object 순번을 확보하기 위해 로그를 읽습니다. 그런 다음 커넥터는 이벤트 파일에서 그 순번이 로그 파일의 마지막 번호보다 큰 모든 Business Object를 통합 브로커로 전송하기 시작합니다. 예를 들어, 위의 파일이 제공될 때 커넥터는 Customer.in 파일에 있는 다섯 번째 Business Object 처리를 시작합니다.

커넥터는 성능을 향상시키기 위해 로그 파일의 내용을 메모리에 보관합니다. 새 항목으로 갱신하기 위해서만 디스크의 파일에 액세스합니다. 커넥터는 복구 시간에만 로그 파일을 읽습니다.

커넥터가 복구 프로세스에서 event.log 파일을 사용하는 방법에 대한 정보는 『장애 복구』를 참조하십시오.

장애 복구

주: 다음 복구 단계는 디스크 장애가 발생하거나 디스크가 가득 찬 경우에는 적용되지 않습니다.

이벤트 공고 중 장애를 복구하기 위해 커넥터는 다음을 수행합니다.

1. 커넥터는 이벤트 파일의 Business Object 문자열을 처리합니다. 항목을 성공적으로 처리할 때 커넥터는 해당 항목을 event.log 파일에 로그합니다. 또한 아카이브 디렉토리(ArchiveDir Meta Object 속성에 지정)의 파일에도 항목을 기록합니다.
 - 이벤트 파일에 있는 모든 Business Object가 처리에 성공한 경우, 커넥터는 성공적으로 처리된 Business Object를 SuccessArchiveExt 속성에 지정되는 확장자와 함께 아카이브 파일에 아카이브합니다.
 - 이벤트 파일에 있는 임의의 Business Object가 처리에 실패한 경우, 커넥터는 성공적으로 처리된 Business Object를 PartialArchiveExt 속성에 지정되는 확장자와 함께 아카이브 파일에 아카이브합니다.
 - SuccessArchiveExt 속성에 지정된 파일에 Business Object를 기록한 후, 임의의 Business Object가 처리에 실패하는 경우 커넥터는 이 파일의 확장자를 PartialArchiveExt에 지정되는 확장자로 변경합니다.

이들 확장자에 대해 제공되는 기본값은 .success 및 .partial입니다.

2. 오류가 발생하는 경우 커넥터는 다음을 수행합니다.
 - 등록 오류 -- 이 커넥터는 아카이브 디렉토리에 UnsubscribedArchiveExt Meta Object 속성에 지정된 확장자가 있는 아카이브 파일을 작성합니다. 이 확장자에 대해 제공되는 기본값은 .unsub입니다.
 - 형식화 오류 또는 전송 오류 -- 이 커넥터는 아카이브 디렉토리에 FailArchiveExt Meta Object 속성에 지정된 확장자가 있는 아카이브 파일을 작성합니다. 이 확장자에 대해 제공되는 기본값은 .fail입니다.
 - Business Object 분리문자 오류 -- 이 커넥터는 아카이브 디렉토리에 FailArchiveExt 속성에 지정된 확장자가 있는 아카이브 파일을 작성합니다. 또한 이벤트 파일을 아카이브 디렉토리로 이동하고 그 확장자를 OriginalArchiveExt에 지정된 확장자로 변경하여 이벤트 파일을 백업합니다.

커넥터는 실패한 Business Object를 event.log에 로그하지 않습니다.

3. 커넥터가 이벤트 파일의 모든 Business Object를 처리한 후 event.log 파일을 지우고 다음 이벤트 파일에서 해당 파일에 항목을 기록하기 시작합니다.
4. 커넥터가 이벤트 파일의 모든 Business Object를 처리하기 전에 작동 중지하는 경우, event.log의 정보를 사용하여 복구 프로세스 중에 처리를 시작할 위치를 판별합니다. 백업할 경우, 커넥터는 로그 파일에 임의의 항목 존재 여부를 검사합니다.
 - 항목이 없는 경우, 커넥터는 이벤트 파일의 모든 Business Object를 통합 브로커에 전송합니다.
 - 항목이 있는 경우, 커넥터는 이 정보를 사용하여 처리를 중지한 시점에서 이벤트 파일 처리를 재개합니다. 커넥터는 복구할 이벤트 파일의 이름과 마지막 Business Object 순번을 확보하기 위해 로그를 읽습니다. 그런 다음 커넥터는 이벤트 파일에서 그 순번이 로그 파일의 마지막 번호보다 큰 모든 Business Object를 통

합 브로커로 전송합니다. 예를 들어, 이벤트 파일이 15개의 Business Object를 포함하고 로그 파일의 마지막 순번이 8인 경우, 커넥터는 마지막 7개의 Business Object를 통합 브로커에 전송합니다.

로그 파일 사용은 커넥터가 동일한 이벤트를 여러 번 통합 브로커에 전송하는 것을 방지합니다. 커넥터는 로그 파일을 메모리에 보관하여 성능을 향상시킵니다. 커넥터는 디스크의 파일에만 액세스하여 새 항목으로 이를 갱신하고, 복구 시간에 로그 파일만을 읽습니다.

EventRecovery 구성 등록 정보를 retry로 설정하는 경우, 커넥터는 시작 시에 자동으로 이전에 처리된 파일로부터 미해결 이벤트를 복구합니다. 그러나 이 등록 정보를 abort로 설정하는 경우, 커넥터는 복구할 이벤트가 있으면 시작 중에 종료합니다.

5. 이벤트 공고 프로세스 중에 발생한 오류를 복구하려면 커넥터를 다시 시작해야 합니다. 그러나 다시 시작하기 전에 다음을 수행하십시오.
 - 커넥터가 실패하여 등록되지 않은 Business Object에 대해 작성한 파일을 검사합니다. 적절하게 수정하여 커넥터가 시작할 때 Business Object 문자열이 성공적으로 처리될 수 있도록 합니다.
 - 아카이브 디렉토리에서 이벤트 디렉토리로 적절한 파일을 복사하고 모든 .fail 또는 .unsub 확장자를 EventExt 속성에 지정된 확장자(기본적으로 .in)로 변경합니다. 레코드 보관을 용이하게 하려면 이들 파일의 이름을 의미있는 형태로 바꾸십시오. 예를 들어, Customer.unsub의 이름을 Customer_unsub_resubmit.in으로 바꾸십시오.
 - 발생한 장애의 유형에 따라서 추가 단계를 수동으로 수행하여 복구해야 합니다.

다음 지침은 발생한 오류의 유형에 따라 조치를 취할 복구 단계를 판별하는 데 도움이 될 수 있습니다.

Business Object 분리문자 오류 복구

커넥터는 Business Object를 아카이브 디렉토리의 아카이브 파일에 기록하고 FailArchiveExt Meta Object 속성에 지정되는 확장자를 부여합니다. 이러한 장애에 대한 복구를 처리하려면 다음을 수행하십시오.

1. 이벤트 파일이 Business Object 분리문자를 포함하고 분리문자가 올바르게 분리문자 값을 데이터 자체에 텍스트로서 포함하고 있지 않는지 확인하십시오. 분리문자의 사용이 맞지 않는 경우 이를 수정하십시오.
2. 커넥터의 로그 파일(LogFileName 구성 속성에 지정됨)을 검토하여 프로세스가 실패한 다른 이유가 있는지 판별하십시오.
3. 파일을 아카이브 디렉토리에서 이벤트 디렉토리로 복사하고 .fail 확장자를 EventExt 속성에 지정된 확장자(기본적으로 .in)로 변경하십시오. 레코드 보관을 용

이하계 하려면 파일의 이름을 의미있는 형태로 바꾸십시오. 예를 들어, Customer.fail의 이름을 Customer_delimiter_error.in으로 바꾸십시오.

등록 오류 복구

커넥터는 아카이브 디렉토리에 있는 파일에 Business Object를 기록하며 UnsubscribedArchiveExt Meta Object 속성에 지정된 확장자를 부여합니다. 이러한 장애에 대한 복구를 처리하려면 다음을 수행하십시오.

1. 아카이브된 파일을 열고, 해당 Business Object 문자열을 찾은 후 Business Object 이름 및 Verb가 등록되었는지 검증하십시오. 필요한 경우, 적절한 정정을 수행하십시오.
2. 통합 브로커가 실행 중인지 확인하십시오.
3. 파일을 아카이브 디렉토리에서 이벤트 디렉토리로 복사하고 .unsub 확장자를 EventExt 속성에 지정된 확장자(기본적으로 .in)로 변경하십시오. 레코드 보관을 용이하게 하려면 파일의 이름을 의미있는 형태로 바꾸십시오. 예를 들어, Customer.unsub의 이름을 Customer_unsub_resubmit.in으로 바꾸십시오.

형식화 오류 복구

커넥터는 아카이브 디렉토리에 있는 파일에 Business Object를 기록하며 FailArchiveExt Meta Object 속성에 지정된 확장자를 부여합니다. 이러한 장애에 대한 복구를 처리하려면 다음을 수행하십시오.

1. 아카이브된 파일을 열고 다음을 검증하십시오.
 - Business Object 문자열 형식이 Meta Object에서 예상되는 형식과 일치하는지 검증하십시오. 불일치가 있는 경우, Meta Object 또는 Business Object 문자열의 형식 유형을 변경하십시오.
 - Business Object 문자열의 형식화 구문이 올바른지 검증하십시오. 올바르지 않은 경우, 정정하십시오.
2. 파일을 아카이브 디렉토리에서 이벤트 디렉토리로 복사하고 .fail 확장자를 EventExt 속성에 지정된 확장자(기본적으로 .in)로 변경하십시오. 레코드 보관을 용이하게 하려면 파일의 이름을 의미있는 형태로 바꾸십시오. 예를 들어, Customer.fail의 이름을 Customer_fail_formatting.in으로 바꾸십시오.

전송 오류 복구

커넥터는 아카이브 디렉토리에 있는 파일에 Business Object를 기록하며 FailArchiveExt Meta Object 속성에 지정된 확장자를 부여합니다. 이러한 장애에 대한 복구를 처리하려면 다음을 수행하십시오.

1. 비즈니스 통합 시스템의 모든 구성요소가 실행 중인지 검증하십시오.
2. 파일을 아카이브 디렉토리에서 이벤트 디렉토리로 복사하고 .fail 확장자를 EventExt 속성에 지정된 확장자(기본적으로 .in)로 변경하십시오. 레코드 보관을 용

- 이하계 하려면 파일의 이름을 의미있는 형태로 바꾸십시오. 예를 들면 Customer.fail의 이름을 Customer_fail_sending.in으로 바꾸십시오.
3. 커넥터를 다시 시작하십시오.

Data Handler 및 지원되는 Business Object

커넥터가 Data Handler가 구성되지 않았음을 표시하는 오류를 리턴하는 경우, Data Handler에 대한 Meta Object가 지원되는 Business Object 목록에 포함되는지 검증하십시오. 커넥터가 리턴하는 가장 공통적인 오류는 BOPrefix가 설정되지 않았음을 표시합니다.

DHFormatter에 대한 지원되는 Business Object 목록은 다음을 포함해야 합니다.

- MO_JTextConnector_Default
- MO_JTextConnector_BusObjName(특정 Business Object에 대해 작성된 Meta Object)
- 파일에서 읽거나 파일에 기록할 Business Object
- Data Handler에 대한 Meta Object(MO_JTextConnector_Default Meta Object의 DataHandlerConfigMO 속성에 지정됨)

제 5 장 JText Connector로 이주 또는 업그레이드

이 장에서는 JText Connector 4.0.x 버전에서 5.3.x 버전으로, 그리고 3.2.0 버전에서 4.0.x 버전으로의 업그레이드 방법에 대해 설명합니다. 또한 Text Connector에서 JText Connector로 업그레이드하는 방법에 대해서도 설명합니다.

주: JText Connector 버전 4.4.x 및 4.3.x는 특정 구성 변경사항을 포함하지 않습니다. 이는 선택적 구성 변경사항만을 포함하며 새 옵션 활용을 원하지 않을 경우에는 이 변경사항 모두 필요하지 않습니다. 자세한 정보는 vii 페이지의 『이 릴리스의 새로운 기능』 절을 참조하십시오.

이 장의 주제는 다음과 같습니다.

- 『업그레이드 시나리오』
- 『버전 4.0.x에서 5.3.x로 업그레이드』
- 77 페이지의 『버전 3.2.0에서 4.0.x로 업그레이드해야 하는 이유』
- 77 페이지의 『버전 4.0.x로 업그레이드』
- 79 페이지의 『Text Connector로부터 업그레이드해야 하는 이유』
- 79 페이지의 『JText Connector로 업그레이드』

업그레이드 시나리오

JText Connector의 4.0.x 릴리스에서 4.1.x 릴리스로 업그레이드하려는 경우, 75 페이지의 『업그레이드 시나리오』 지시사항을 따르십시오.

JText Connector의 3.2.0 릴리스에서 4.1.x 릴리스로 업그레이드하려는 경우, 77 페이지의 『버전 4.0.x로 업그레이드』 및 75 페이지의 『업그레이드 시나리오』 지시사항을 따르십시오.

Text Connector에서 JText Connector로 업그레이드하려는 경우, 79 페이지의 『JText Connector로 업그레이드』 지시사항을 따르십시오.

버전 4.0.x에서 5.3.x로 업그레이드

이 절에서는 다음에 대해 설명합니다.

- 76 페이지의 『Meta Object 변경』
- 76 페이지의 『구조 변경사항』
- 77 페이지의 『Jar 파일 변경사항』

Meta Object 변경

MO_JTextConnector_Default Meta Object는 아카이브 처리의 유연성을 확장하는 세 개의 새로운 속성(PartialArchiveExt, UnsubscribedArchiveExt 및 OriginalArchiveExt)을 포함합니다.

이들 속성을 JText 최상위 레벨 Meta Object에 추가하고 그에 대한 값을 구성해야 합니다. Business Object Designer를 사용하여 새 속성을 추가하고 사용자의 기본값을 제공하며 정의에 변경사항을 저장하십시오.

세 개의 새 속성을 MO_JTextConnector_Default에서 사용자 정의한 각 Meta Object에 추가하십시오. 예를 들어, 고객 및 항목 Business Object에 대한 사용자의 Meta Object를 작성한 경우 이들 Meta Object에 새 속성을 추가하고 사용자의 기본값을 제공하며 저장소에 변경사항을 저장하십시오.

자세한 정보는 34 페이지의 표 5를 참조하십시오.

구조 변경사항

이 절에서는 다음에 대해 설명합니다.

- 『이름 지정 규칙 변경사항』
- 77 페이지의 『구성 등록 정보 변경사항』

이름 지정 규칙 변경사항

커넥터의 구조에 대한 변경사항은 포맷터 Meta Object의 이름이 JText 최상위 레벨 Meta Object에 대한 이름 지정 규칙을 준수하지 않아야 합니다.

커넥터의 일부 이전 릴리스는 MO_JTextConnector_DHFormatter로 이름 지정된 DHFormatter Meta Object를 제공합니다. 이 이름은 더 이상 유효하지 않습니다. 이 Meta Object에 대해 허용 가능한 이름은 두 번째 위치에 Connector가 지정되지 않아야 합니다. 예를 들어, MO_JText_Default_DHFormatter는 이 Meta Object에 대한 유효한 이름입니다.

포맷터 Meta Object가 JText 최상위 레벨 Meta Object와 동일한 이름 지정 규칙을 사용하는 경우, 해당 이름을 변경하고 이 Meta Object를 표시하는 모든 Business Object 속성의 유형을 변경해야 합니다. 이를 변경하려면 다음을 수행하십시오.

1. 기존 포맷터 Meta Object의 이름을 새 Meta Object 이름으로 바꾸십시오.
2. 최상위 레벨 Meta Object에서 포맷터 Meta Object를 표시하는 모든 속성의 유형을 이전 Meta Object 이름에서 새 이름으로 변경하십시오.
3. 커넥터의 구성을 편집하여 지원되는 Business Object 목록에서 이전 Meta Object를 제거하고 목록에 새 포맷터 Meta Object를 추가하십시오.
4. 통합 브로커 저장소에서 기존 포맷터 Meta Object를 제거하십시오.

자세한 정보는 30 페이지의 『JText Meta Object 이름 지정 규칙』을 참조하십시오.

구성 등록 정보 변경사항

InRecoveryWindow, OutRecoveryWindow 및 EventRecoveryEnabled 커넥터 특정 구성 등록 정보는 더 이상 존재하지 않습니다. 새 등록 정보인 EventRecovery가 추가되었습니다.

커넥터의 새 아카이브 기능을 활용하려면 다음을 수행하십시오.

1. Connector Configurator에서 JText Connector를 편집하여 이전 등록 정보를 제거하고 새 등록 정보를 추가하십시오.
2. 새 등록 정보의 값을 "abort" 또는 "retry"로 설정하십시오.

이 등록 정보에 대한 자세한 정보는 22 페이지의 『EventRecovery』를 참조하십시오.

Jar 파일 변경사항

CwJTFormatter.jar 파일은 더 이상 제공되지 않습니다. 제품이 제공하는 모든 포맷터는 CwJText.jar 파일로 이동되었습니다.

CwJTFormatter.jar 파일에 저장한 사용자 정의 포맷터를 사용하는 경우, 다음 중 하나를 수행하십시오.

- 사용자 정의 포맷터를 새 jar 파일로 이주시키십시오. 해당 jar을 start_JText.bat 또는 start_JText.sh 파일에 지정된 클래스 경로 내에 포함시키십시오.
- 클래스 경로에 이전 CwJTFormatter.jar 파일을 배치하십시오. CwJTFormatter.jar 이 클래스 경로의 CwJText.jar 다음에 있는지 확인하십시오.

버전 3.2.0에서 4.0.x로 업그레이드해야 하는 이유

JText Connector의 버전 4.0.x에서는 커넥터를 구성하기 위해 필요한 Meta Object의 구조가 획기적으로 단순화되었으며, 따라서 구성 프로세스를 단순화합니다.

커넥터의 초기 버전이 계층 구조의 세 레벨과 최소한 10개 이상의 상이한 Meta Object를 포함한 Meta Object 구조를 사용하는 반면, 4.0.x 버전은 두 개의 Meta Object와 계층 구조의 두 레벨만을 사용합니다. 이 새 버전은 사용자가 커넥터를 구성하는 방법을 변경할 뿐 커넥터의 기능을 변경하지는 않습니다.

버전 4.0.x로 업그레이드

새 Meta Object가 이전 버전과 동일한 구성 데이터를 사용하기 때문에 업그레이드 시 어떠한 구성값 변경도 필요하지 않습니다. 그러나 새 Meta Object가 훨씬 더 적은 Meta Object의 상이하게 이름 지정된 속성에 데이터를 저장하기 때문에 업그레이드 시 다음 조작이 필요합니다.

- 새 Meta Object를 작성합니다.
- 각각의 새로운 Meta Object에 있는 각 속성의 DefaultValue 등록 정보 값을 사용자의 기존 Meta Object에 있는 사용자 정의 기본값으로 바꿉니다.
- 저장소에서 모든 구식 Meta Object를 제거합니다.

IBM WBIS Support는 위의 조작을 자동화하는 유틸리티를 제공합니다. 이들 조작을 수동으로 실행하려면 다음 단계를 수행하십시오.

1. repos_copy 유틸리티를 사용하여 저장소의 백업을 작성하십시오. 예를 들어, 다음 명령은 Server1 저장소의 모든 내용을 출력 파일 InterChangeRepository.out에 백업합니다.

```
repos_copy -oInterChangeRepository.out -sServer1 -pmpassword
```

2. 사용자의 기존 최상위 레벨 Meta Object의 각각에 대해 새로 제공되는 최상위 레벨 Meta Object인 MO_JTextConnector_Default와 동일한 속성이 있는 새 Meta Object를 작성하십시오. 예를 들어, 이전 이름 지정 규칙 (MO_JText_Customer_Connector)에 따라서 이름 지정된 고객 Business Object에 대한 사용자의 Meta Object를 작성한 경우, 새 이름 지정 규칙 (MO_JTextConnector_Customer)을 사용하는 고객에 대해 새 Meta Object를 작성하십시오.
3. 원래 Meta Object의 값을 기본으로 새 Meta Object의 기본값을 설정하십시오. 원래 Meta Object와 새 Meta Object의 속성 사이의 대응에 대해서는 표 9를 참조하십시오.
4. System Manager를 사용하여 저장소에서 Meta Object 정의의 원래 세트를 삭제하고 방금 작성한 것과 MO_JText_Default_DHFormatter만을 보관하십시오.

표 9에서는 원래 Meta Object의 이름을 포함하여 원래 속성과 새 속성 사이의 대응을 설명합니다. 복수 Meta Object가 원래 속성을 포함하는 반면,

MO_JTextConnector_Default Meta Object는 모든 새 속성을 포함합니다.

표 9. 새 속성에 대한 원래 Meta Object 및 속성의 대응

원래 Meta Object 이름	원래 속성 이름	새 속성 이름
MO_JText_BOName_Connector	DummyKey	N/A
MO_JText_BOName_ArchiveDir	Directory	ArchiveDir
MO_JText_BOName_ArchiveFileExt	Success	SuccessArchiveExt
	Fail	PartialArchiveExt
		FailArchiveExt
		UnsubscribedArchiveExt
MO_JText_BOName_EventDir	Directory	EventDir
	FileExt	EventExt
MO_JText_BOName_OutputDir	Directory	OutputDir
	FileExt	OutputExt
	FileSequencingEnabled	FileSeqEnabled
MO_JText_BOName_FormatType		N/A
MO_JText_BOName_ServicePolicy	OutputFileName	OutputFileName
	EndBODelimiter	EndBODelimiter

표 9. 새 속성에 대한 원래 Meta Object 및 속성의 대응 (계속)

원래 Meta Object 이름	원래 속성 이름	새 속성 이름
MO_JText_BOName_FormatService	EventService OutputService	EventFormat OutputFormat

78 페이지의 표 9에는 다음 정보가 포함되어 있지 않습니다.

- MO_JText_BOname_FormatType Meta Object에 대응하는 속성
이벤트 및 출력 파일에 사용할 형식을 구성하기 전에 MO_JText_BOname_FormatType Meta Object의 모든 가능한 형식을 나열하는 데 필요한 커넥터의 이전 버전입니다. 새 Meta Object 구조에서는 이벤트 및 출력 파일에 사용될 형식만을 구성하면 됩니다. 이 변경사항은 MO_JText_BOname_FormatType Meta Object에 대한 대응하는 속성이 없으므로 78 페이지의 표 9에 표시됩니다.
- 개별 포맷터에 대한 Meta Object
최상위 레벨 Meta Object에는 포맷터 Meta Object를 포함하는 두 개의 속성이 있습니다. 포맷터 Meta Object에는 원래 구조에서와 동일한 속성이 있으며 동일한 방법으로 사용됩니다. 네 개의 포맷터 중 세 개가 거부되었기 때문에 유일한 관련 포맷터 Meta Object는 MO_JText_BOname__DHFormatter입니다.

Text Connector로부터 업그레이드해야 하는 이유

사이트가 현재 Text Connector를 사용하여 응용프로그램과 통합 브로커 사이에서 통신하는 경우, JText Connector로 업그레이드해야 하는 이유는 다음과 같습니다.

- 성능. Text Connector는 한번에 하나의 파일만을 처리하며, 이는 큰 파일이나 다수의 파일을 처리할 때 성능을 저해할 수 있습니다.
- 형식 사용가능성. Text Connector는 소수의 형식 유형을 처리합니다.
- 수정의 용이성. Text Connector는 수정이 보다 어렵습니다.

대조적으로, JText Connector는 다음을 수행하도록 구성할 수 있습니다.

- 한번에 여러 파일을 처리합니다.
- 특정 Business Object에 대한 여러 위치를 검색하므로 성능을 증가시킵니다.
- 광범위한 형식 유형을 수용합니다.

JText Connector로 업그레이드

Text Connector에서 JText Connector로 업그레이드하려면 다음을 수행하십시오.

1. 제품 CD에서 JText 디렉토리를 %CROSSWORLD%connectors 디렉토리에 복사하십시오.
2. 명령 프롬프트 창을 열고 repos_copy를 사용하여 두 파일(CN_JText.txt 및 MO_JText_Default.txt)을 저장소에 추가하십시오.

3. ArchiveDir, EventDir 및 OutputDir Meta Object 속성에 대해 지정된 디렉토리가 작성되었는지 확인하십시오. 이들 디렉토리가 작성되지 않은 경우, 작성하십시오.
4. Meta Object 속성을 구성하십시오.
5. 원하는 Business Object에 등록하십시오.

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 10은 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 10. 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 10. 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 10. 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 10. 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 `true`로 설정할 때, JMS 사용 커넥터는 중복된 이벤트가 전달 대기열로 전달되지 않게 보장할 수 있습니다. 이 기능을 사용하려면 응용프로그램 특정 코드에서 Business Object의 **ObjectEventId** 속성으로 설정된 커넥터에 고유한 이벤트 ID가 있어야 합니다. 이는 커넥터 개발 중에 수행됩니다.

또한 이 등록 정보는 `false`로 설정될 수 있습니다.

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

FaultQueue

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

기본값은 CONNECTORNAME/FAULTQUEUE입니다.

JvmMaxHeapSize

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

기본값은 128m입니다.

JvmMaxNativeStackSize

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

기본값은 128k입니다.

JvmMinHeapSize

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

기본값은 1m입니다.

jms.FactoryClassName

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

기본값은 CxCommon.Messaging.jms.IBMMQSeriesFactory입니다.

jms.MessageBrokerName

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

기본값은 crossworlds.queue.manager입니다.

jms.NumConcurrentRequests

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

기본값은 10입니다.

jms.Password

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

기본값은 없습니다.

jms.UserName

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

기본값은 없습니다.

ListenerConcurrency

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

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

Locale

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

로케일 이름의 형식은 다음과 같습니다.

ll_TT.codeset

여기서

ll

2문자 언어 코드(보통 소문자)

TT

2문자 국가 또는 지역 코드(보통 대문자)

codeset

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

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

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

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

LogAtInterchangeEnd

RepositoryDirectory가 <REMOTE>인 경우에만 적용 가능합니다.

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

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

MaxEventCapacity

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

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

MessageFileName

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

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

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

MonitorQueue

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

기본값은 CONNECTORNAME/MONITORQUEUE입니다.

OADAutoRestartAgent

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

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

기본값은 false입니다.

OADMaxNumRetry

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 1000입니다.

OADRetryTimeInterval

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 10입니다.

PollEndTime

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

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

PollFrequency

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

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

기본값은 10000입니다.

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

PollQuantity

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

PollStartTime

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

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

RequestQueue

Business Object를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTOR/REQUESTQUEUE입니다.

RepositoryDirectory

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

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

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

ResponseQueue

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

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

RestartRetryCount

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

기본값은 3입니다.

RestartRetryInterval

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

기본값은 1입니다.

SourceQueue

DeliveryTransport가 JMS이고 ContainerManagedEvents가 지정된 경우에만 적용 가능합니다.

JMS 이벤트 저장소를 사용하는 JMS 사용 커넥터에 대해 보증된 이벤트 전달을 지원하는 커넥터 프레임워크의 JMS 소스 대기열을 지정합니다. 자세한 정보는 88 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.

- 99 페이지의 『Connector Configurator 개요』
- 100 페이지의 『Connector Configurator 시작』
- 101 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 104 페이지의 『새 구성 파일 작성』
- 107 페이지의 『구성 파일 등록 정보 설정』
- 115 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(101 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, 102 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(106 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 102 페이지의 『새 템플리트 작성』을 참조하십시오.
- 기존 파일을 사용하려면 기존 템플리트를 수정하여 새 이름으로 저장하기만 하면 됩니다.

새 템플리트 작성

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

템플리트를 작성하려면 다음을 수행하십시오.

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

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

일반 특성 지정

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

- 일반
 - 등록 정보 유형
 - 갱신 메소드
 - 설명

- 플래그
표준 플래그
- 사용자 정의 플래그
플래그

등록 정보의 일반 특성을 선택한 후, **값** 탭을 누르십시오.

값 지정

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

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

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

기본값 열을 사용하면 임의의 값을 기본값으로 지정할 수 있습니다.

값 범위는 등록 정보 값 대화 상자에서 입력한 범위를 표시합니다.

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

종속성 설정

일반 및 값 탭에서 변경을 수행한 후, 다음을 누르십시오. 종속성 - 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. 완료를 누르십시오. Connector Configurator가 Connector Configurator를 설치한 \bin 디렉토리의 \data\app에 XML 문서로 입력한 정보를 저장합니다.

새 구성 파일 작성

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

브로커를 선택하려면 다음을 수행하십시오.

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

다음 작업도 수행할 수 있습니다.

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

커넥터 특정 템플릿에서 구성 파일 작성

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

1. 파일 > 새로 작성 > 커넥터 구성을 누르십시오.

2. 다음 필드가 있는 새 커넥터 대화 상자가 나타납니다.

- 이름

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

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

- 시스템 연결

ICS, WebSphere Message Brokers 또는 WAS를 누르십시오.

- 커넥터 특정 등록 정보 템플릿

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

사용하려는 템플릿을 선택하고 확인을 누르십시오.

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

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

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

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

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

기존 파일 사용

다음 중 하나 이상의 형식으로 기존 파일을 사용할 수 있습니다.

- 커넥터 정의 파일.

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

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

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

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

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

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

System Manager 프로젝트에서 커넥터 구성을 열려면 다음 단계를 따르십시오.

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

구성 파일 완료

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

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

1. 표준 등록 정보 탭에서 브로커 유형 등록 정보에 대한 값 필드를 선택하십시오. 드롭 다운 메뉴에서 ICS, WMQI 또는 WAS를 선택하십시오.
2. 표준 등록 정보 탭에 선택한 브로커와 연관된 등록 정보가 표시됩니다. 지금 파일을 저장하거나 110 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. 108 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

82 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>
```

주의사항

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

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

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.

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

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

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

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

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

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

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

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

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

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

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

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

프로그래밍 인터페이스 정보

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

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

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

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

상표 및 서비스표

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

IBM

IBM 로고

AIX

CrossWorlds

DB2

DB2 Universal Database

Domino

Lotus

Lotus Notes

MQIntegrator

MQSeries

Tivoli

WebSphere

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

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

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

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



WebSphere Business Integration Adapter Framework V2.4.0

IBM