



Adapter for WebSphere MQ Workflow 사용자 안내서

버전 2.5.0



Adapter for WebSphere MQ Workflow 사용자 안내서

버전 2.5.0

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 133 페이지의 『주의사항』의 정보를 읽으십시오.

2003년 12월 19일

이 문서 개정판은 새 개정판에서 별도로 명시하지 않는 한 WebSphere MQ Workflow 버전 2.5.0 및 모든 후속 릴리스와 수정판에 적용됩니다.

이 문서에 대한 의견이 있으면 ibmkspoe@kr.ibm.com으로 전자 우편을 보내 주십시오. 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
이 책을 사용하기 전에	v
관련 서적	v
인쇄 규칙	vi
이 릴리스의 새로운 기능	vii
릴리스 2.5.0의 새로운 기능	vii
릴리스 2.4.x의 새로운 기능	vii
릴리스 2.3의 새로운 기능	vii
릴리스 2.2.x의 새로운 기능	viii
릴리스 2.1.x의 새로운 기능	viii
릴리스 1.2.x의 새로운 기능	viii
제 1 장 개요	1
커넥터 구조	1
응용프로그램 커넥터 통신 메소드	2
이벤트 처리	6
보증된 이벤트 전달	9
Business Object 요청	10
Verb 처리	10
로케일 종속 데이터 처리	17
제 2 장 커넥터 설치 및 구성	19
어댑터 환경	19
전제조건	20
설치 작업 개요	21
어댑터 및 관련 파일 설치	21
설치된 파일 구조	21
XML API를 사용하기 위해 WebSphere MQ Workflow 업그레이드	23
커넥터 구성	24
보증된 이벤트 전달 사용 가능	31
최상위 레벨 Business Object 및 내용 구성	35
Meta Object 구성	37
시작 파일 구성	52
커넥터의 다중 인스턴스 작성	53
커넥터 시작	54
커넥터 중지	56
제 3 장 WebSphere MQ Workflow 응용프로그램 수정	57
UPES 구성	57
제 4 장 Business Object 개발	65

커넥터 Business Object 구조	65
샘플 Business Object 정의	66
오류 처리	73
추적.	75
제 5 장 FDLBORGEN 유틸리티를 사용한 Business Object 정의 작성	77
FDLBORGEN 유틸리티 정보	77
FDLBORGEN 유틸리티를 사용하기 전에	78
FDLBORGEN 구문	78
FDLBORGEN 예제	79
변환 참고사항	79
제 6 장 문제점 해결	81
시작 시 문제점	81
이벤트 처리	82
부록 A. 커넥터의 표준 구성 등록 정보	83
신규 및 삭제된 등록 정보	83
표준 커넥터 등록 정보 구성.	83
표준 등록 정보 요약	85
표준 구성 등록 정보	88
부록 B. Connector Configurator	101
Connector Configurator 개요.	101
Connector Configurator 시작.	102
System Manager에서 Configurator 실행	103
커넥터 특정 등록 정보 템플릿 작성	103
새 구성 파일 작성	106
기존 파일 사용.	108
구성 파일 완료.	109
구성 파일 등록 정보 설정	109
구성 파일 저장.	116
구성 파일 변경.	117
구성 완료	117
국제화된 환경에서 Connector Configurator 사용	117
부록 C. 자습서	119
전제조건	119
설치 전 점검 목록.	119
환경 설정	120
샘플, 템플릿, 어댑터 및 맵 구성	121
시나리오 실행	125
주의사항	133
프로그래밍 인터페이스 정보	134
상표 및 서비스표	135

이 책의 정보

IBM^(R) WebSphere^(R) Business Integration Adapter 포트폴리오는 앞선 e-business 기술, 엔터프라이즈 응용프로그램, 레거시 및 메인프레임 시스템에 대한 통합 연결성을 제공합니다. 제품 세트에는 비즈니스 프로세스 통합용 구성요소를 사용자 정의, 작성 및 관리하는 데 필요한 도구 및 템플릿이 포함되어 있습니다.

이 책은 Adapter for WebSphere MQ Workflow의 설치, 구성 및 Business Object 개발에 대해 설명합니다.

이 책의 사용자

이 책은 고객의 사이트에서 제품을 지원 및 관리하는 시스템 관리자, 개발자 및 컨설턴트를 위한 것입니다.

이 책을 사용하기 전에

이 책의 사용자는 WebSphere Business Integration System, Business Object 및 협업 개발, WebSphere MQ Workflow 응용프로그램과 WebSphere MQ Workflow 런타임 및 빌드타임 구성요소에 대해 잘 알고 있어야 합니다.

관련 서적

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

다음 사이트에서 관련 문서를 설치할 수 있습니다.

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

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

- WebSphere InterChange Server와 함께 어댑터를 사용하는 경우에는 다음 웹 사이트를 참조하십시오.

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

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

- WebSphere Message Broker에 대한 자세한 정보는 다음 사이트를 참조하십시오.

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

- WebSphere Application Server에 대한 자세한 정보는 다음 사이트를 참조하십시오.

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

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

인쇄 규칙

이 문서에서는 다음과 같은 규칙을 사용합니다.

<code>courier</code> 글꼴	명령 이름, 파일 이름, 사용자가 입력하는 정보 또는 시스템이 화면에 인쇄하는 정보와 같은 리터럴값을 표시합니다.
굵은체 기울임	처음 나타나는 새 용어를 표시합니다.
파란색 윤곽선	변수 이름 또는 상호 참조를 표시합니다.
{ }	온라인으로 메뉴얼을 볼 때만 볼 수 있는 파란색 윤곽선은 상호 참조 하이퍼링크를 표시합니다. 참조 오브젝트로 이동하려면 윤곽선 내부를 누르십시오.
[]	구문 행에서 옵션을 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 복수 옵션을 쉼표로 구분하여 입력할 수 있음을 의미합니다.
/, \	이름 지정 규칙에서 꺾쇠 괄호는 이름의 개별 요소를 묶어 이들을 서로 구별합니다(예: <code><server_name><connector_name>tmp.log</code>).
%text% 및 \$text	이 문서에서는 디렉토리 경로 규칙으로 백슬래시(\)를 사용합니다. UNIX 설치의 경우, 백슬래시를 슬래시(/)로 대체합니다. 모든 제품 경로 이름은 사용자 시스템에 제품을 설치하는 디렉토리과 관련됩니다.
<i>ProductDir</i>	퍼센트 부호(%) 사이의 텍스트는 Windows text 시스템 변수값 또는 사용자 변수값을 표시합니다. UNIX 환경에서의 동일한 표기법은 \$text 로서, text UNIX 환경 변수 값을 표시합니다.
	IBM WebSphere Business Integration Adapters 제품이 설치된 디렉토리를 표시합니다.

이 릴리스의 새로운 기능

릴리스 2.5.0의 새로운 기능

어댑터 설치 정보는 이 안내서에서 삭제되었습니다. 해당 정보의 새 위치는 2장을 참조하십시오.

이 릴리스로 시작할 때 Adapter for WebSphere MQ Workflow는 더 이상 Microsoft Windows NT에서 지원되지 않습니다.

릴리스 2.4.x의 새로운 기능

이제 어댑터에서 WebSphere Application Server를 통합 브로커로 사용할 수 있습니다.

시작 스크립트, 파일 이름 및 디렉토리에서 이름 MQWorkflow가 WebSphereMQWorkflow로 바뀌었습니다.

이제 다음 플랫폼에서 어댑터를 실행할 수 있습니다.

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

이 어댑터 릴리스에서는 IBM WebSphere MQ Workflow 3.3.2 또는 3.4가 필요하며, 더 이상 Java API 통신 모드를 지원하지 않습니다.

릴리스 2.3의 새로운 기능

2003년 3월에 갱신되었습니다. 이름 “CrossWorlds”는 더 이상 전체 시스템을 설명하거나 구성요소 또는 도구 이름을 수정하는 데 사용되지 않으며, 그렇지 않은 경우는 이전과 거의 동일합니다. 예를 들어 “CrossWorlds System Manager”는 이제 “System Manager”이며 “CrossWorlds InterChange Server”는 이제 “WebSphere InterChange Server”입니다.

보증된 이벤트 전달 기능이 향상되었습니다. 추가 정보는 31 페이지의 『보증된 이벤트 전달 사용 가능』을 참조하십시오.

릴리스 2.2.x의 새로운 기능

이 릴리스에서는 커넥터가 MQ Workflow 3.3.2 XML API Verb 처리를 지원합니다. 자세한 정보는 12 페이지의 『XML API Verb 처리』 및 23 페이지의 『XML API를 사용하기 위해 WebSphere MQ Workflow 업그레이드』를 참조하십시오. 기능을 지원하기 위해 새로운 커넥터 특정 등록 정보가 추가되었습니다. 28 페이지의 『JavaCorbaApi』를 참조하십시오.

릴리스 2.1.x의 새로운 기능

커넥터는 국제화되었습니다. 자세한 정보는 17 페이지의 『로케일 종속 데이터 처리』 및 83 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

이 안내서는 ICS에서 이 어댑터의 사용에 대한 정보를 제공합니다.

주: 보증된 이벤트 전달 기능을 사용하려면 ICS 릴리스 4.1.1.2를 설치해야 합니다.

릴리스 1.2.x의 새로운 기능

IBM WebSphere Business Integration Adapter for MQ Workflow에는 Connector for MQ Workflow가 포함됩니다. 이 어댑터는 ICS(InterChange Server) 통합 브로커와 함께 작동합니다. 통합 브로커는 이기종 응용프로그램 세트 통합을 수행하는 응용 프로그램으로, 데이터 라우팅을 포함하는 서비스를 제공합니다. 어댑터 구성요소는 다음과 같습니다.

- MQ Workflow 특정 응용프로그램 구성요소
- 샘플 Business Object
- 다음으로 구성된 IBM WebSphere Adapter Framework.
 - 커넥터 프레임워크
 - 개발 도구(Business Object Designer 및 IBM CrossWorlds System Manager 포함)
 - API(CDK 포함)

이 매뉴얼에서는 ICS에서의 이 어댑터 사용에 대한 정보를 제공합니다.

중요: 커넥터는 국제화되지 않았으므로, ISO Latin-1 데이터만이 처리될 것으로 보증할 수 없는 경우 ICS 4.1.1에서 커넥터를 실행하지 마십시오.

커넥터는 이제 AIX 4.3.3 Patch Level 9에서 사용 가능합니다.

제 1 장 개요

Connector for WebSphere MQ Workflow는 WebSphere Business Integration Adapter for WebSphere MQ의 런타임 구성요소입니다. WebSphere MQ Workflow는 IBM 워크플로우 관리 시스템입니다.

커넥터를 사용하면 WebSphere Integration Broker가 WebSphere MQ Workflow와 Business Object를 교환할 수 있습니다. WebSphere MQ Workflow는 이 책에서 설명하는 커넥터 등의 외부 응용프로그램 및 WebSphere MQ Workflow 클라이언트 프로세스(노드)와 통신합니다.

이 장에서는 커넥터 구성요소 및 Business Integration System 구조를 설명합니다. 여기에서는 다음 주제를 다룹니다.

- 『커넥터 구조』
- 2 페이지의 『응용프로그램 커넥터 통신 메소드』
- 6 페이지의 『이벤트 처리』
- 9 페이지의 『보증된 이벤트 전달』
- 10 페이지의 『Business Object 요청』
- 10 페이지의 『Verb 처리』
- 17 페이지의 『로케일 종속 데이터 처리』

커넥터 구조

커넥터는 응용프로그램 특정 구성요소 및 커넥터 프레임워크로 구성됩니다. 응용프로그램 특정 구성요소에는 특정 응용프로그램으로 조정된 코드가 있습니다. 코드가 모든 커넥터에 공통인 커넥터 프레임워크는 통합 브로커와 특정 응용프로그램 구성요소 간의 매개자로 작용합니다. 커넥터 프레임워크는 통합 브로커와 특정 응용프로그램 구성요소 간에 다음과 같은 서비스를 제공합니다.

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

Connector for WebSphere MQ Workflow는 데이터 구조를 교환하고 관련 프로세스를 제어함으로써 협업과 WebSphere MQ Workflow 노드를 연결합니다. 커넥터는 외부 응용프로그램이지만 WebSphere MQ Workflow 시스템에서는 내부 “노드”와 같은 기능을 수행합니다. 노드와 마찬가지로, 커넥터는 프로세스 지향 기능을 수행합니다.

WebSphere MQ Workflow 노드는 지정 대기열에서 커넥터에 대한 요청을 발행하도록 구성됩니다. 커넥터는 이러한 대기열에서 WebSphere MQ Workflow 요청 메시지

를 폴링하고 검색합니다. 커넥터는 DOM(Document Object Model) 구문 분석기 및 XML Data Handler를 사용하여 데이터를 추출한 후 요청 및 해당 협업에 적절한 Business Object로 변환합니다. 반대로, 커넥터는 협업에서 Business Object 요청을 수신하여 MQ Workflow 프로세스 요청으로 변환한 후 MQ Workflow의 XML 입력 대기열에 발행합니다. 또한 커넥터는 MQ Workflow 서버에 직접 바인드하여 MQ Workflow 프로세스에 대한 보다 강력한 제어를 제공할 수 있습니다.

커넥터에서 MQ Workflow 시스템의 XML 메시지 API를 사용하도록 구성하십시오. MQ Workflow 시스템의 XML API는 메시지 요청의 비동기 및 동기 처리를 위한 것입니다. 사용자는 MQ Workflow 노드가 다른 노드인 것처럼 커넥터와 통신할 수 있도록 허용하는 워크플로우 조치를 트리거하는 데 사용되는 IBM MQ Workflow 프로그램인 UPES(User-Defined Program Execution Server)를 작성합니다. 커넥터는 XML API를 사용하여 협업에서 Business Object를 요청하는 메시지에 응답하고 MQ Workflow의 조치를 요청하는 메시지를 전송합니다.

MQ Workflow와 교환된 모든 XML 메시지는 단일 DTD(Document Type Definition)인 WfMessage를 따릅니다. DOM(Document Object Model) 구문 분석기가 WfMessage에서 워크플로우 데이터 구조를 추출하고 나면, 커넥터는 WfMessage를 사용하여 데이터 구조를 유지하기 위한 컨테이너 오브젝트를 작성합니다.

주: 하나의 MQ Workflow 프로세스가 다른 입력 및 출력 데이터 구조를 가질 수는 있지만, 협업과 커넥터 간의 모든 주어진 트랜잭션은 한 가지 오브젝트 유형만을 포함할 수 있습니다. 이러한 제한사항을 피하기 위해, Connector for MQ Workflow에서는 하나의 요청 오브젝트와 하나 이상의 응답 오브젝트를 갖는 하위로서의 하나의 컨테이너 오브젝트가 구성될 필요가 있습니다. 이 사항에 대한 자세한 정보는 37 페이지의 『Meta Object 구성』을 참조하십시오.

폴링 시, 커넥터는 WfMessage에 포함되어 있는 데이터 구조를 식별함으로써 작성해야 할 최상위 레벨 오브젝트를 알 수 있습니다. 특히, <boprefix> 구성 등록 정보에 추가되는 데이터 구조 이름은 작성될 최상위 레벨 오브젝트를 판별합니다. 이 최상위 레벨 오브젝트의 첫 번째(관련된 비 메타 데이터) 하위 오브젝트는 데이터 구조로 채워집니다. 상위 컨테이너 오브젝트에 지정되는 verb는 WfMessage의 ProgramParameters 필드를 기준으로 합니다. 상위 컨테이너 오브젝트는 InterChange Server에 게시됩니다.

응용프로그램 커넥터 통신 메소드

소개에서 설명한 대로, connector for MQ Workflow는 XML API 통신 모드를 지원합니다.

커넥터는 XML API를 사용하여 MQ Workflow의 조치를 트리거하는 메시지를 전송할 수 있으며 커넥터 폴링 시 MQ Workflow로부터의 동기 요청을 처리할 수 있습니다. 커넥터는 요청 MQ 메시지가 발행되는 고정 대기열을 폴링하고 내용을 처리하며 응

답 메시지(필요 시 두 번째 대기열로)를 리턴합니다. 커넥터에 발행된 XML 메시지는 비즈니스 내용 외에, 실행할 협업, 사용할 verb 및 다른 처리 정보를 표시합니다.

XML API를 사용하려면 커넥터의 입력 대기열을 지정하는 UPES를 구성해야 합니다.

커넥터 및 UPES

XML 메시지 API 사용 시, 커넥터는 새 이벤트를 확인하기 위해 MQ Workflow를 직접 폴링하지 않습니다. 커넥터 노드와 같이 외부 대기열에 요청을 발행하도록 MQ Workflow 노드를 구성해야 합니다. 그런 다음 커넥터가 이러한 외부 대기열을 폴링할 수 있습니다.

UPES를 통해 외부 대기열에 요청을 발행하도록 MQ Workflow 노드를 구성해야 합니다. UPES는 MQ Workflow 서버에서 요청을 승인하도록 설계된 프로그램입니다. 따라서 UPES는 서버와 상호작용하여 모든 추가 데이터를 검색하고 결과를 다시 전달할 수 있습니다. MQ Workflow는 XML 메시지로의 요청 변환을 처리하며, 그 반대의 경우도 마찬가지입니다.

그림 1에서 보여주는 것처럼 커넥터는 MQ Workflow 시스템의 UPES 모드와 같은 기능을 수행합니다. 실제로, 커넥터는 MQ Workflow 서버에 투명합니다.

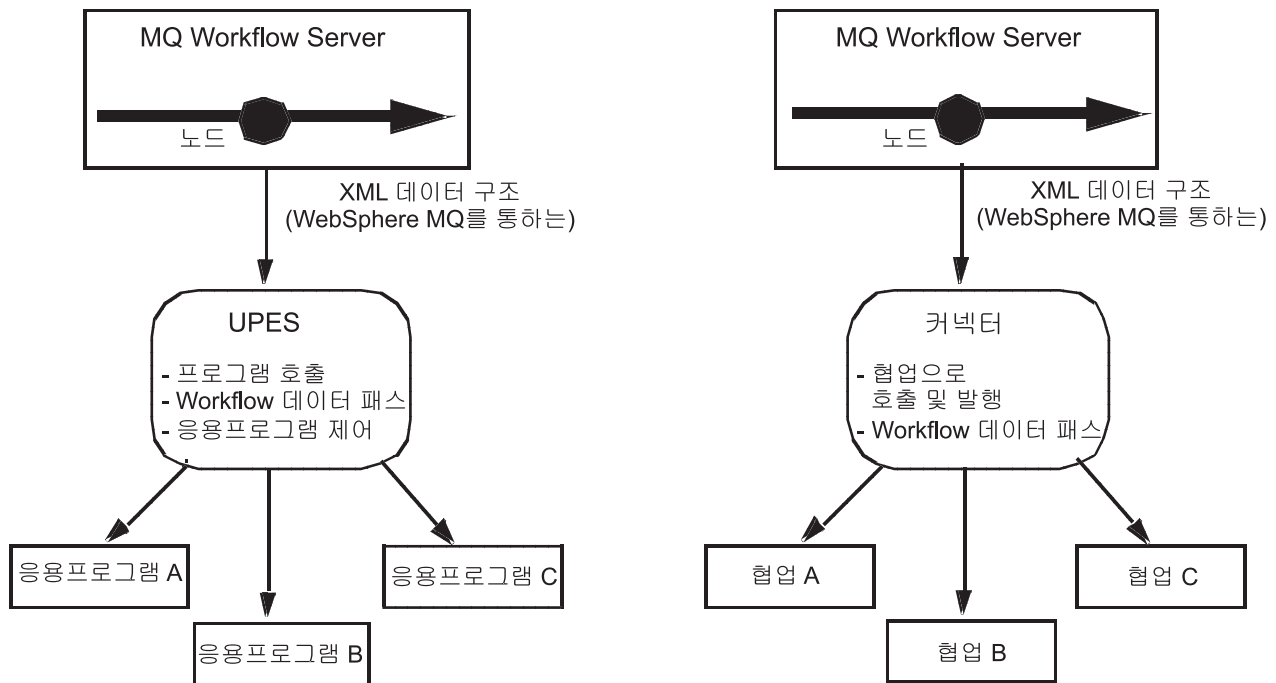


그림 1. MQ Workflow UPES로서의 커넥터 기능

커넥터 시작 요청: XML API

커넥터가 협업을 대신하여 메시지를 수신하는 경우, 커넥터는 MQ Workflow 서버의 XML 입력 대기열로 XML 요청 메시지를 발행합니다. 선택적으로(동기적으로), 커넥터는 MQ Workflow 서버가 리턴하는 응답 메시지를 기다립니다. 서버는 워크플로우 프로세스를 트리거하고 필요한 대로 응답을 발행합니다.

그림 2는 커넥터를 통한 협업에서 MQ Workflow로의 메시지 요청 통신을 보여줍니다.

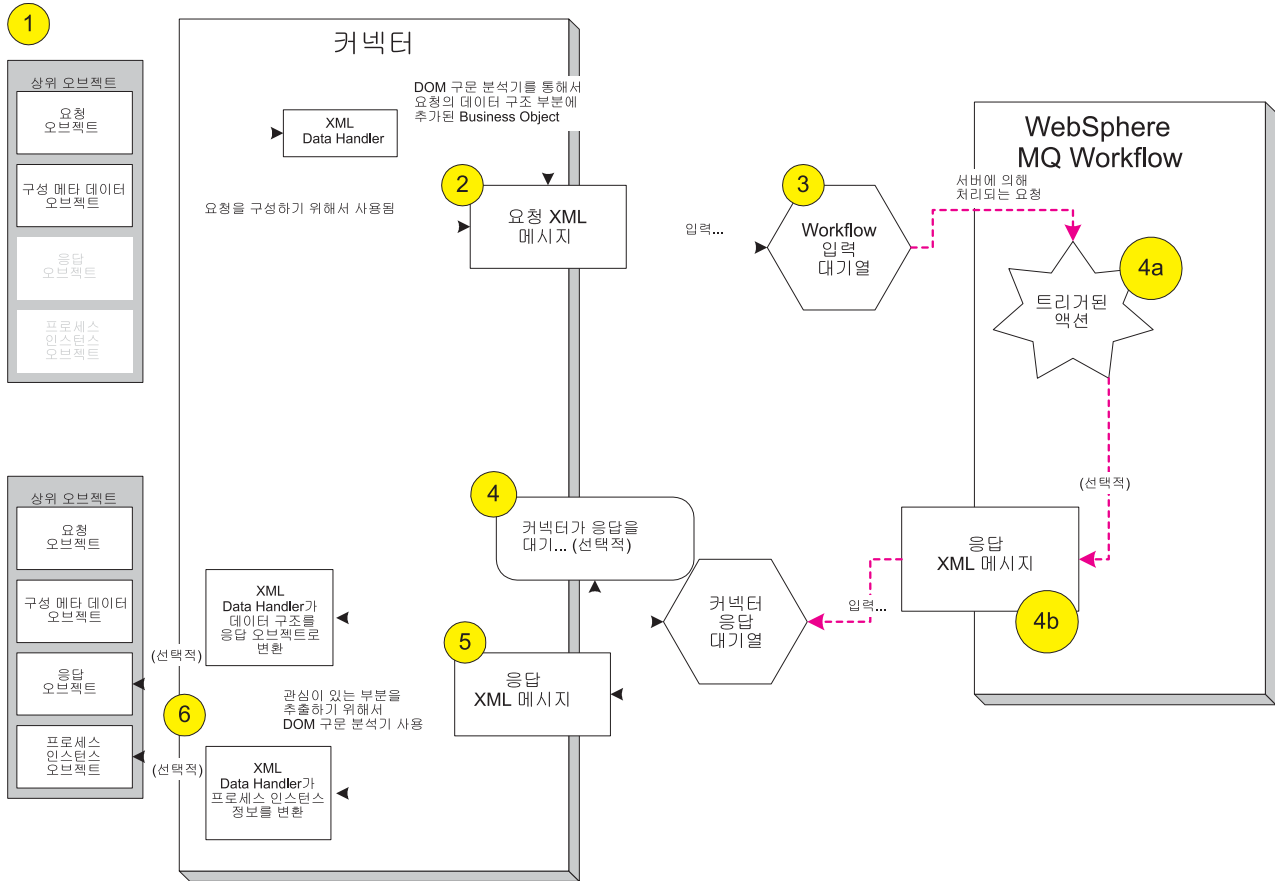


그림 2. XML API 통신 모드: 커넥터 시작 요청

1. 커넥터는 MQ Workflow에 지정된 Business Object에 대한 협업에서 요청을 수신합니다.
2. 커넥터는 XML Data Handler를 사용하여 상위 Business Object에 포함되어 있는 요청 오브젝트를 XML로 변환합니다. XML 배열 Business Object는 DOM 구문 분석기를 사용하여 MQ Workflow 메시지 DTD를 따르는 보다 큰 XML 메시지로 통합됩니다. 작성 및 실행할 워크플로우 프로세스에 대한 정보는 상위 Business Object에 포함되어 있는 구성 Meta Object에 요약되어 있습니다.
3. 커넥터는 MQ Workflow 서버의 XML 입력 대기열에 요청을 게시합니다.

- MQ Workflow 서버는 Business Object 내용에서 지정하는 대로 요청을 수신하고 조치를 수행합니다. 병렬 실행 시, 커넥터는 a) 성공적으로 리턴하거나(예정 응답이 없는 경우) 또는 b) 선택적으로 응답 메시지를 기다립니다.
- 커넥터에서 발행된 요청 메시지에 따라, MQ Workflow 서버는 동시 실행 프로세스의 PID(Process Instance Identifier)만을 포함하는 응답을 즉시 리턴할 수 있습니다. 또는 MQ Workflow 서버는 프로세스가 완료될 때까지 기다린 후 결과 출력 데이터 구조를 리턴합니다.
- 커넥터는 XML DOM 구문 분석기를 사용하여 응답 메시지에서 출력 데이터 구조 및 프로세스 ID를 추출합니다. 필연적으로 이들 구조는 XML Data Handler를 사용하는 Business Object로 변환되며 상위 Business Object에 추가됩니다. 그런 다음 이 오브젝트는 대기 협업으로 리턴됩니다.

MQ Workflow 시작 요청

그림 3은 커넥터에 대한 MQ Workflow 시작 요청을 보여줍니다.

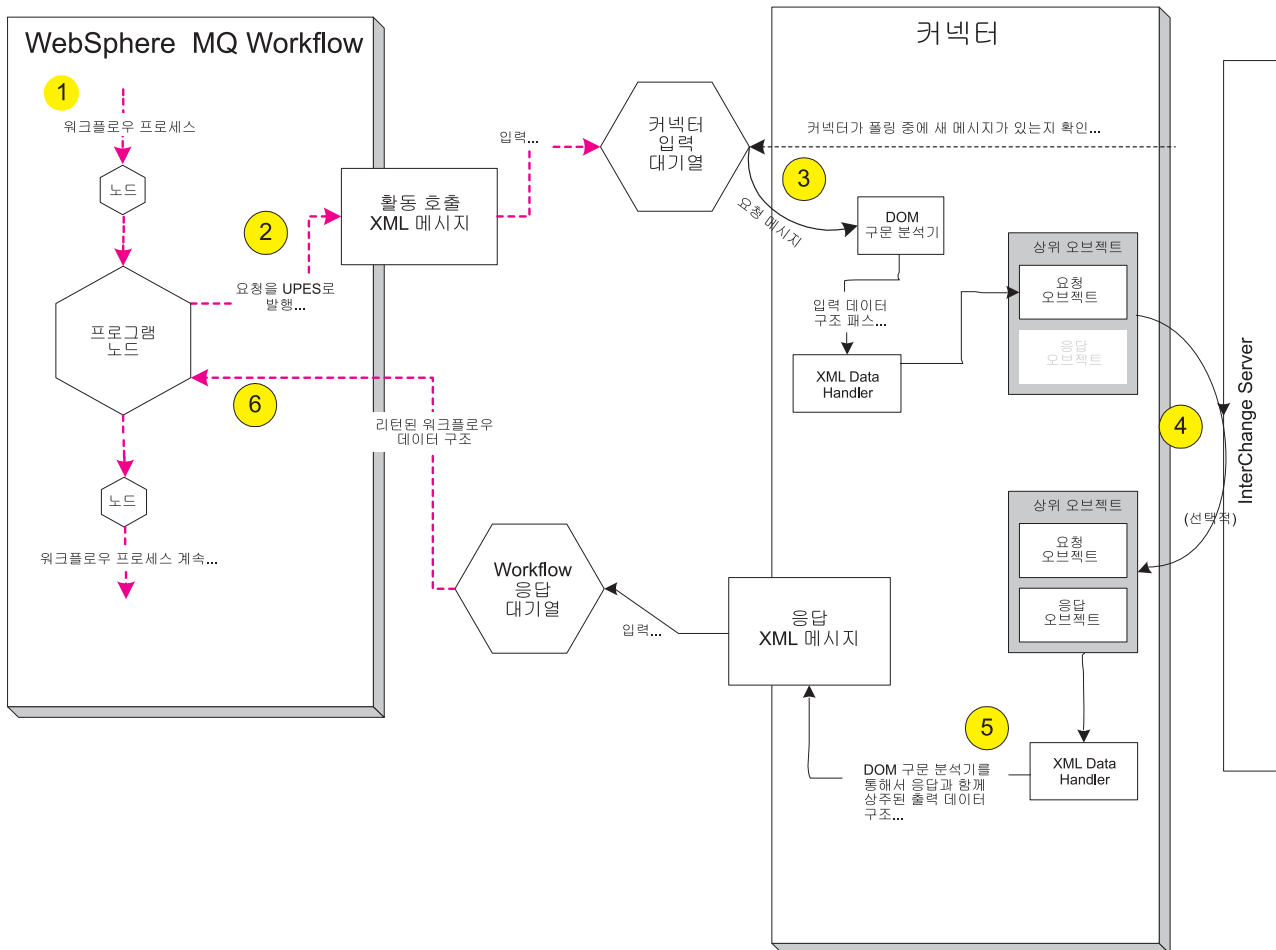


그림 3. MQ Workflow 시작 요청

1. 워크플로우 구현 시, UPES는 UPES로 전송된 메시지가 커넥터 입력 대기열로 받
행되는 것으로 정의됩니다.
2. MQ 서버는 협업으로부터의 조치를 요청하기 위해 요청 데이터 구조를 UPES로 라
우트합니다(위의 “프로그램 노트”에 표시된 대로). 그런 다음 MQ Workflow는 XML
기반 프로그램 호출 메시지를 지정 커넥터 입력 대기열로 전송합니다. 메시지에는
워크플로우 데이터 구조가 포함됩니다.
3. 커넥터는 대기열 폴링 시 메시지를 검색하고, XML DOM 구문 분석기를 사용하여
내용 구문을 분석하며, XML Data Handler를 사용하여 입력 데이터 구조를 Business
Object로 변환합니다.
4. UPES(프로그램 노트) 정의 시 사용자가 지정한 대로 요청이 비동기로 실행되면, 커
넥터는 Business Object를 모든 대기 협업에 게시하고 응답을 MQ Workflow 서
버로 되돌려 보내지 않습니다.
5. 요청이 동기 실행되고 협업 이름이 명령 매개변수에 지정되면(62 페이지의 그
림 20 참조), 커넥터는 Business Object를 특정 협업으로 보내고 리턴을 기다립니
다. 커넥터는 리턴을 기본으로 하는 적당한 응답 메시지를 구성합니다. 응답은 임의
의 Business Object 변경을 포함합니다. 그런 다음 응답은 MQ Workflow 서버로
전송됩니다.
6. 요청이 동기 실행되고 협업 이름이 명령 매개변수에 지정되지 않으면(62 페이지
의 그림 20 참조), 커넥터는 Business Object를 모든 대기 협업에 게시하고 응답을
MQ Workflow 서버로 전송하지 않습니다. 비록 이 사례가 위의 비동기 경우와 유
사하지만, MQ Workflow 서버는 여전히 커넥터로부터 응답을 기다립니다. 그러므
로 서비스 호출 요청을 이용하여 관련 응답을 커넥터로 전송하는 것은 협업의 책임
입니다.
7. 요청 MQ Workflow UPES(프로그램 노트)는 응답 코드 및 결과 비즈니스 데이터
구조를 수신합니다.

이벤트 처리

MQ Workflow UPES가 커넥터 입력 대기열에 요청을 게시하는 경우, 이벤트가 발생
합니다. 커넥터는 입력 대기열을 폴링하여 이벤트를 발견합니다. 이 섹션에서는 이벤트
처리 프로세스에 대해 설명합니다.

이벤트 공고

커넥터는 이벤트 발견 후 아래에서 설명하는 단계를 수행합니다.

1. 커넥터가 DOM 구문 분석기로 XML 메시지를 로드합니다.
2. 커넥터는 이 메시지가 WfMessage인지 검증하고 인식 템플릿을 확인합니다.
3. 커넥터는 메시지에 포함되어 있는 입력 데이터 구조를 식별한 후 `<boprefix><data
structure name>` 유형의 Business Object를 작성합니다. 예를 들어, 커넥터 구

성 등록 정보인 *boprefix*의 값이 *WfRequest_*이고 커넥터가 *MyCustomer*라는 이름의 데이터 구조를 포함하는 메시지를 수신하는 경우, 커넥터는 해당 구조가 최상위 레벨 Business Object *WfRequest_MyCustomer*의 하위를 준수하는 것으로 예상합니다. 커넥터는 XML Data Handler를 사용하여 XML 데이터 구조를 Business Object로 변환합니다.

4. 커넥터는 *WfMessage*에서 *ProgramParameters* 요소를 찾습니다. *ProgramParameters* 필드에는 실제 MQ Workflow에서 사용자가 지정한 응용프로그램 매개변수가 포함됩니다.
5. Data Handler에서 작성한 Business Object가 *verb*를 지정하지 않은 경우, 커넥터는 요청 메시지의 *ProgramParameters* 부분에서 지정한 데이터를 사용하여 *verb*를 설정합니다.
6. 커넥터는 MQ Workflow가 해당 요청에 대한 응답을 예상하는지 여부를 판별하기 위해, XML 문서의 *WfMessageHeader* 컨테이너에 있는 *ResponseRequired* 요소의 값이 *yes*, *no* 또는 *iferror*인지 검증합니다. *yes* 값은 MQ Workflow가 요청 상태에 대한 커넥터의 응답 메시지 대기가 활성화되어 있음을 표시합니다.

주: *ResponseRequired* 요소는 MQ Workflow Buildtime 구성의 프로그램 활동 등록 정보에 있는 모드에 해당합니다(64 페이지의 그림 22의 Workflow 서버 정의 참조). 비동기 모드를 지정하는 경우에는 *ResponseRequired=no*이고, 동기 모드를 지정하는 경우에는 *ResponseRequired=yes*입니다.

7. 응답이 예상되면(*ResponseRequired=yes*), 커넥터는 응답을 처리하는 방법을 결정하기 위해 *WfMessage*의 *ProgramParameters* 요소에 지정된 데이터를 평가합니다.
8. 협업이 *ProgramParameters*에 지정되면, 커넥터는 *executeCollaboration()* 메소드를 사용하여 요청을 협업으로 보냅니다. 이 메소드는 협업에 대한 동기 요청이므로, 메소드 호출로부터의 리턴은 시간을 필요로 합니다. 메소드는 해당 인수에 응답 오브젝트를 리턴합니다. 커넥터는 응답 오브젝트를 채우게 함으로써 응답 메시지를 생성하고 응답을 MQ Workflow 서버로 보냅니다.
9. 협업이 *ProgramParameters*에 지정되지 않으면, 커넥터는 *gotApplEvent()* 메소드를 사용하여 응답을 모든 등록 협업에 게시합니다. 커넥터는 MQ Workflow 활동 정보(예: *ActImplCorrelID*)를 포함하고 있는 Meta Object와 함께 *gotApplEvent()* 메소드의 인수에서 보여지는 Business Object를 채우게 합니다. 메소드는 협업에 응답을 게시하고, 커넥터는 즉시 메소드 호출로부터 리턴을 수신합니다. 메소드 호출이 리턴하는 응답 오브젝트가 없으므로, 생성 또는 MQ Workflow 서버로 전송되는 응답 메시지는 없습니다. 대신에 협업은 해당 응답 오브젝트를 서비스 호출 요청을 이용하여 MQ Workflow 서버로 보내야 합니다. 응답 오브젝트는 MQ Workflow 활동 정보(예: *ActImplCorrelID*)가 들어 있는 Meta Object를 포함해야 합니다. 따라서 커넥터는 서비스 호출 요청을 기본으로 하는 적

당한 응답 메시지를 구성합니다. 응답 메시지는 MQ Workflow 활동 정보와 응답 Business Object뿐 아니라 협업에서부터의 리턴 코드를 포함합니다. 응답 메시지는 MQ Workflow 서버로 전송됩니다. 메시지의 비즈니스 내용과 무관한 문제로 인해 8 또는 9단계에서 오류가 발생하면, 커넥터는 오류를 기록하고 응답 WfMessage를 생성하지 않습니다.

10. 메시지는 아카이브, 오류 또는 비등록 대기열에 선택적으로 아카이브됩니다.

검색

커넥터는 정기적으로 메시지에 대한 해당 입력 대기열을 폴링합니다. 커넥터가 메시지를 발견하면, 대기열에서 메시지를 검색하여 DOM 구문 분석기 및 XML Data Handler로 전달하여 WfMessage 내용을 확보합니다. 커넥터는 WfMessage에서 추출된 데이터 구조를 사용하여 verb로 적절한 Business Object를 생성합니다. 이벤트 실패 시나리오는 73 페이지의 『오류 처리』를 참조하십시오.

커넥터는 입력 대기열에 대한 트랜잭션 세션을 처음에 열어 메시지를 처리합니다. 이러한 트랜잭션 접근 방법은 커넥터가 Business Object 제출에는 성공하지만 대기열 내 트랜잭션 확약에 실패하여 Business Object가 협업으로 두 번 전송될 수 있는 경우(거의 발생하지 않음)를 허용합니다. 이러한 문제점을 피하기 위해, 커넥터는 모든 메시지를 진행 중인 대기열로 이동시킵니다. 여기에서 메시지는 처리가 완료될 때까지 보류됩니다. 커넥터가 처리 중 예기치 않게 종료한 경우, 메시지는 원래 입력 대기열로 복귀되는 대신 진행 중인 대기열에 남습니다.

주: 커넥터는 1) 메시지가 Business Object로 변환된 경우 2) Business Object가 InterChange Server에 전송되는 경우 3) 리턴값이 수신되는 경우에 진행 중 대기열에서 메시지를 제거합니다.

복구

초기화 시, 커넥터는 커넥터 종료 때문에 완전히 처리되지 않은 메시지에 대한 진행 중인 대기열을 검사합니다. 커넥터 구성 등록 정보 InDoubtEvents는 시작 시 실패, 재처리, 무시 또는 오류 로그와 같은 메시지의 복구를 처리하는 네 가지 옵션 중 하나를 지정하게 합니다.

FailOnStartup

커넥터가 FailOnStartup 옵션으로 초기화 시 진행 중 대기열에서 메시지를 발견하는 경우, 오류를 기록하고 즉시 종료됩니다. 메시지를 점검하고 적절한 조치를 수행하여 이 메시지를 완전히 삭제하거나 다른 대기열로 이동시키는 것은 사용자 또는 시스템 관리자의 책임입니다.

재처리

재처리 옵션 사용 시, 커넥터가 초기화 중에 진행 중인 대기열에서 메시지를 찾은 경우 차후 폴링 중 처음에 이들 메시지를 처리합니다. 진행 중인 대기열의 모든 메시지가 처리되면, 커넥터가 입력 대기열에서 메시지 처리를 시작합니다.

무시

무시 옵션 사용 시, 커넥터가 초기화 중에 진행 중인 대기열에서 메시지를 찾은 경우 커넥터는 메시지를 무시하지만 종료하지 않습니다. 커넥터는 입력 대기열에서 메시지 처리를 시작합니다.

오류 로그

오류 로그 옵션 사용 시, 커넥터가 초기화 중에 진행 중인 대기열에서 메시지를 찾은 경우 커넥터는 오류를 로그하지만 종료하지 않습니다. 커넥터는 입력 대기열에서 메시지 처리를 시작합니다.

아카이브

커넥터 등록 정보 ArchiveQueue가 지정되고 유효한 대기열을 식별하는 경우, 커넥터는 정상적으로 처리된 모든 메시지의 사본을 아카이브 대기열에 놓습니다. ArchiveQueue를 정의하지 않은 경우, 처리 완료 메시지는 처리 후 버려집니다. 비등록 메시지 및 오류 메시지에 대한 자세한 정보는 65 페이지의 제 4 장 『Business Object 개발』, 73 페이지의 『오류 처리』를 참조하십시오.

보증된 이벤트 전달

보증된 이벤트 전달 기능을 사용하면 커넥터 프레임워크에서 이벤트가 유실되지 않아 커넥터의 이벤트 저장소, JMS 이벤트 저장소 및 목적지의 JMS 대기열 간에 이벤트가 두 번 전송되지 않도록 할 수 있습니다. JMS를 사용하려면 커넥터의 DeliveryTransport 표준 등록 정보를 JMS로 구성해야 합니다. 따라서 구성된 커넥터는 JMS 전송을 사용하며, 커넥터와 통합 브로커 간의 모든 후속 통신은 이 전송을 통해 발생합니다. JMS 전송으로 메시지는 결국 목적지에 전달됩니다. 이 역할은, 트랜잭션 대기열 세션이 시작되면 확약이 발행될 때까지 메시지가 캐시화하고 장애가 발생하거나 롤백이 발행되면 메시지를 버리는 것입니다.

주: 보증된 이벤트 전달 기능을 사용하지 않으면 커넥터가 이벤트를 공개한 시간(커넥터가 pollForEvents() 메소드 내에서 gotAppEvent() 메소드를 호출한 시간)과 커넥터가 이벤트 레코드를 삭제하여(또는 “이벤트 게시” 상태로 갱신하여) 이벤트 저장소를 갱신한 시간 사이에 작은 창에서 장애가 발생할 수 있습니다. 이 창에서 장애가 발생하면 이벤트는 전송되지만 이벤트 레코드는 이벤트 저장소에 “진행 중”인 상태로 남아 있습니다. 커넥터가 다시 시작될 때 커넥터는 이벤트 저장소에 남아 있는 이 이벤트 레코드를 발견하여 전송합니다. 따라서 이벤트는 두 번 전송됩니다.

JMS 이벤트 저장소의 사용 여부에 관계없이 JMS 사용 가능 커넥터에 보증된 이벤트 전달 기능을 구성할 수 있습니다. 보증된 이벤트 전달을 위해 커넥터를 구성하려면 31 페이지의 『보증된 이벤트 전달 사용 가능』을 참조하십시오.

커넥터 프레임워크가 ICS 통합 브로커로 Business Object를 전달할 수 없는 경우, 오 브젝트는 FaultQueue(UnsubscribedQueue 및 ErrorQueue가 아닌)에 놓이고 상태 표시기 및 문제점에 대한 설명을 기록합니다. FaultQueue 메시지는 MQRFH2 형식으로 작성됩니다.

Business Object 요청

Business Object 요청은 InterChange Server가 커넥터에 Business Object를 전송할 때 처리됩니다. 커넥터는 XML Data Handler를 사용하여 Business Object를 MQ Workflow WfMessage 형식 메시지로 변환하여 MQ Workflow로 발행합니다.

WfMessage에는 Business Object 및 처리 정보가 포함되며, MQ 메시지에 MQMD(MQ Message Descriptor) 머리글로 요약됩니다. MQMD 머리글에는 커넥터가 채우는 UserID 필드가 포함됩니다. (기본적으로 커넥터는 커넥터 구성 등록 정보 ApplicationUserID의 값을 전달합니다.) MQ Workflow는 MQ Message의 MQMD 머리글에 있는 UserID 필드를 사용하여 커넥터가 전송한 요청에 대한 권한 검사를 수행합니다.

주: WfMessage 데이터 구조를 채우는 하위 구성 Meta Object에 UserID 속성을 정의하는 경우, 이 UserID 속성은 MQ Workflow에 발행된 Business Object 인스턴스의 커넥터 구성 등록 정보에 정적으로 정의되어 있는 값을 대체합니다.

주: 사용자는 MQ Workflow에서 워크플로우 프로세스를 시작할 수 있는 권한을 가질 수 있으나, WebSphere MQ를 통해 메시지를 발행하기 위해서는 계속 권한이 필요합니다. 메시지 발행을 위한 사용자 권한 부여 방법에 대해서는 WebSphere MQ 문서를 참조하십시오.

Verb 처리

커넥터의 역할은 워크플로우의 데이터 구조와 Business Object를 연결하는 것입니다. Business Object에 설정되어 있는 verb에 따라 조치를 수행하는 것은 MQ Workflow의 책임입니다. 커넥터는 워크플로우가 내용을 성공적으로 수신하는 것만을 보증할 수 있습니다. 따라서 커넥터는 MQ Workflow 제품의 Business Object에 영향을 줄 수 없습니다. 실제로 MQ Workflow 특성을 고려할 때, Business Object에 대한 지속 데이터 구조가 존재하지 않을 수 있습니다. 이는 지속 데이터 구조가 후속 워크플로우의 트리거로서만 작동하기 때문입니다.

BOHandler(Business Object Handler)는 verb에 관계없이 모든 Business Object를 동일한 방식으로 처리합니다. 커넥터는 DOM 구문 분석기를 사용하여 WfMessage를 구성합니다.

커넥터는 최상위 레벨 Business Object의 응용프로그램 특정 텍스트를 검사하고 cw_mo_wfptcfg=XXX 양식의 이름-값 쌍을 정의했는지 확인합니다. XXX로 식별되는 하위 Meta Object의 구문을 분석하고 값이 해석됩니다.

실행되는 템플리트는 Meta Object 속성 ProcessTemplateName으로 식별됩니다. 이 템플리트는 결과 포함뿐 아니라 전체 명령을 MQ Workflow에 지정하는 데 필요한 구조를 제공합니다. ProcessInstanceName 속성을 지정한 경우, 커넥터는 기존 인스턴스를 실행합니다. 그렇지 않은 경우 템플리트의 새 인스턴스를 작성합니다. 템플리트는 하위 Meta Object의 UserId 속성으로 식별되는 사용자에게 부여되는 권한 하에서 실행됩니다. 속성을 지정하지 않는 경우, 커넥터 구성 등록 정보 ApplicationUserID의 값이 대신 사용됩니다. 템플리트에 대한 자세한 정보는 57 페이지의 제 3 장 『WebSphere MQ Workflow 응용프로그램 수정』에 있는 35 페이지의 『최상위 레벨 Business Object 및 내용 구성』을 참조하십시오.

커넥터는 기존 작업을 제어하는 요청에 대해 다음 Business Object 용어를 지원합니다.

- Delete
- Suspend
- Terminate
- Restart
- Resume

커넥터는 MQ Workflow에서 메시지를 수신할 때마다 WfMessage의 ProgramParameters 값에서 Business Object의 verb를 식별합니다.

ProgramParameters의 텍스트에는 메시지에 포함되어 있는 Business Object의 verb를 지정하는 name=value 쌍이 포함되어야 합니다. 예를 들어, Delete 용어를 지정하기 위해 요소 텍스트에는 name=value 쌍인 verb=Delete가 포함됩니다.

반대로, 커넥터는 요청 처리에 사용할 verb를 MQ Workflow에 지정하지 않습니다. MQ Workflow에 Business Object를 발행하는 경우, 커넥터는 Business Object의 verb를 무시합니다. 대신, 커넥터는 Business Object를 XML로 변환하고 내용을 MQ Workflow의 WfMessage로 통합합니다. Business Object가 발행되는 워크플로우는 수행된 조치(협업으로 Business Object에 지정된 verb가 아님)를 판별합니다.

XML API Verb 처리

주: MQ Workflow 버전 3.3.2 이상에서는 verb 처리를 위해 XML API를 권장합니다.

MQ Workflow 커넥터 XML API를 사용하면 협업이 워크플로우 프로세스의 상태를 모니터링하고 제어할 수 있습니다. 커넥터가 워크플로우 조작을 성공적으로 제어하면, 프로세스 인스턴스 오브젝트(MO_MQWorkflow_ProcessInstance)를 프로세스의 세부사항으로 채웁니다. 커넥터는 app-text가 ProcessInstance인 오브젝트를 MO_MQWorkflow_ProcessInstance의 인스턴스로 간주합니다.

커넥터는 Business Object를 XML로 변환하고 내용을 MQ Workflow의 WfMessage로 통합합니다. Business Object에 지정된 verb가 프로세스 인스턴스에서 수행되는 조치를 판별합니다.

XML API를 사용하면, 커넥터가 MO_MQWorkflow_ProcessInstance에 대해 다음 verb를 지원합니다.

Delete

커넥터가 MQ Workflow에서 지정된 프로세스 인스턴스를 삭제합니다. 프로세스 인스턴스는 준비, 완료됨, 종료됨 상태 중 하나여야 합니다. 프로세스가 없거나 삭제할 수 없으면, 커넥터가 BON_FAIL을 리턴합니다. 아니면, 커넥터가 새 상태로 채워진 MO_MQWorkflow_ProcessInstance 오브젝트를 리턴합니다.

Suspend

커넥터는 워크플로우 프로세스를 일시중단하기 위한 요청을 발행하고, 프로세스가 존재하지 않거나 일시중단될 수 없는 경우 BON_FAIL을 리턴합니다. 프로세스 인스턴스는 실행 중 상태여야 합니다. deep 옵션이 true이면, 자동이 아닌 모든 서브프로세스도 일시중단됩니다. 프로세스가 없거나 일시중단될 수 없으면, 커넥터가 BON_FAIL을 리턴합니다. 아니면, 커넥터가 새 상태로 채워진 MO_MQWorkflow_ProcessInstance 오브젝트를 리턴합니다.

Terminate

커넥터가 프로세스 인스턴스와 자동이 아닌 모든 서브프로세스를 종료합니다. 실행 중이고 체크아웃되고 일시중단된 모든 활동이 종료됩니다. 프로세스는 실행 중, 일시중단됨, 일시중단 중 상태중 하나여야 합니다. 프로세스가 없거나 종료될 수 없으면, 커넥터가 BON_FAIL을 리턴합니다. 아니면, 커넥터가 새 상태로 채워진 MO_MQWorkflow_ProcessInstance 오브젝트를 리턴합니다.

Restart

커넥터가 워크플로우 프로세스 인스턴스를 다시 시작하라는 요청을 발행합니다. 완료되거나 종료된 최상위 레벨 프로세스 인스턴스만이 다시 시작될 수 있습니다. 프로세스가

없거나 다시 시작될 수 없으면, 커넥터가 BON_FAIL을 리턴합니다. 아니면, 커넥터가 새 상태로 채워진 MO_MQWorkflow_ProcessInstance 오브젝트를 리턴합니다.

Resume

커넥터가 일시중단된 처리나 일시중단된 프로세스 인스턴스를 재개하라는 요청을 발행합니다. deep 옵션이 true이면, 자동이 아닌 모든 서브프로세스도 재개됩니다. 프로세스가 없거나 재개될 수 없으면, 커넥터가 BON_FAIL을 리턴합니다. 아니면, 커넥터가 새 상태로 채워진 MO_MQWorkflow_ProcessInstance 오브젝트를 리턴합니다.

비동기 요청

구성 Meta Object의 ResponseTimeout 속성이 0 미만인 경우, 커넥터는 응답을 기다리지 않고 MQ Workflow 서버에 요청을 발행합니다. 처리 실행 중 오류가 발생하는 경우 협업에 알릴 수 있는 방법이 없습니다. 그림 4는 샘플 비동기 요청을 보여줍니다.

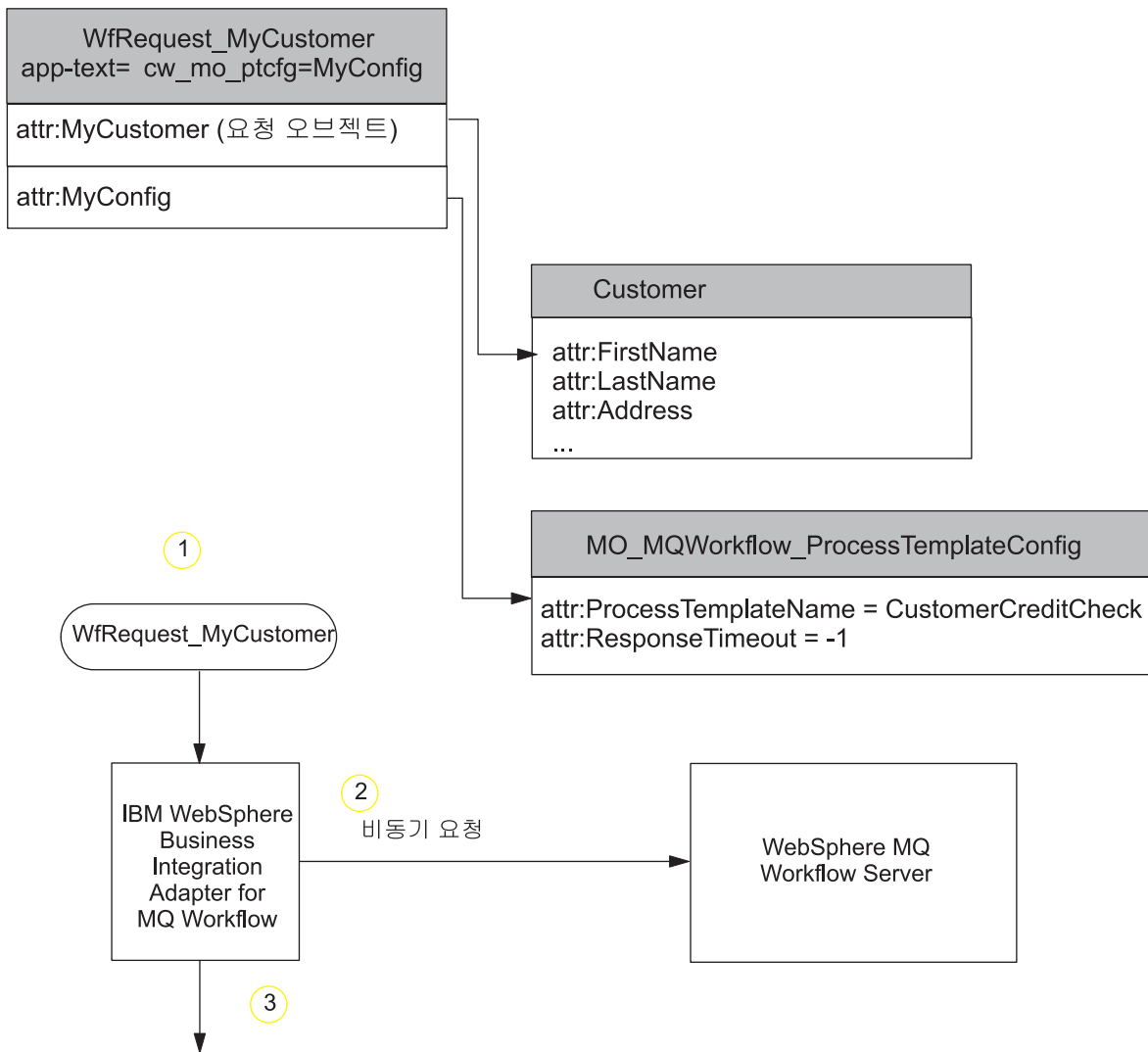


그림 4. MQ Workflow로의 샘플 비동기 커넥터 요청

1. 커넥터는 음수 “ResponseTimeout” 속성을 갖는 최상위 Business Object WfRequest_MyCustomer를 수신합니다.
2. 커넥터는 프로세스로 전달될 데이터 구조로서 “MyCustomer” 오브젝트를 포함하고 있는 MQ Workflow 서버로 요청을 발행합니다.
3. 커넥터는 응답을 기다리지 않고 성공적으로 리턴합니다. 커넥터가 MQ Workflow 서버의 XML 입력 대기열에 메시지를 두는 데 실패하는 경우에만 오류가 발생합니다.

프로세스 인스턴스 ID에 대한 비동기 요청

하위 Meta Object에 음수가 아닌 ResponseTimeout이 제공되고 ExecutionMode 속성이 *Asynchronous*인 경우, 커넥터는 요청을 발행하고 협업으로 프로세스 인스턴스 ID를 리턴합니다. 프로세스 인스턴스 ID를 성공적으로 수신했다고 해서 해당 워크플로우 프로세스가 성공적으로 완료된 것을 의미하지는 않습니다. 협업은 프로세스 인스턴스 ID에 대해 “Retrieve”를 수행하여 상태를 판별해야 합니다. 이는 장기 트랜잭션에 유용합니다. 그림 5는 이 프로세스를 보여줍니다.

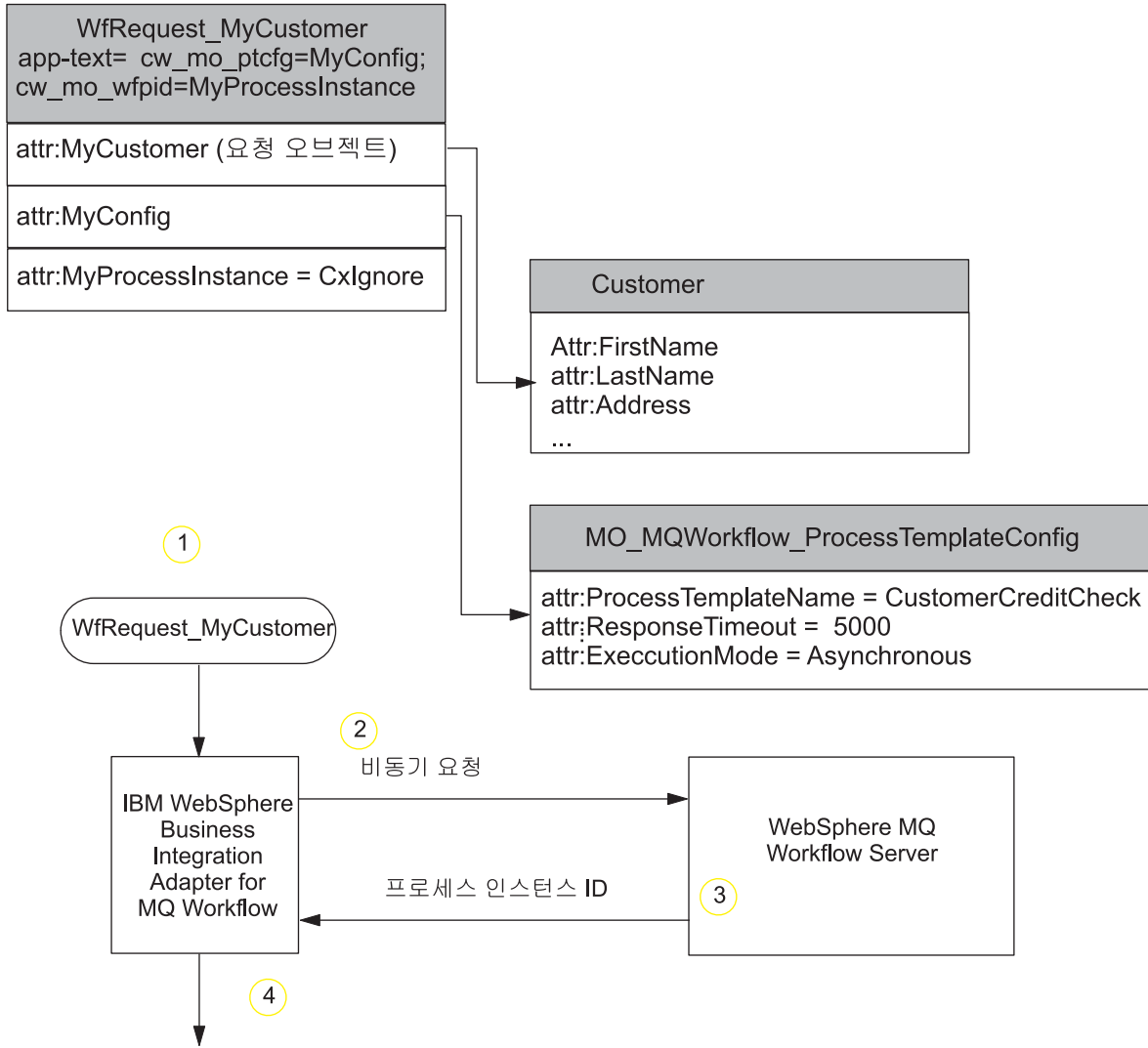


그림 5. 프로세스 인스턴스 ID에 대한 샘플 비동기 커넥터 요청

1. 커넥터는 *ResponseTimeout* 속성에 음수가 아닌 값을 지정하고 *ExecutionMode* 속성에 *Asynchronous* 값을 지정하는 최상위 Business Object인 *WfRequest_MyCustomer*를 수신합니다.
2. 커넥터는 프로세스 템플릿 *CustomerCreditCheck*의 입력 데이터 구조로서 *MyCustomer* 오브젝트를 포함하는 MQ Workflow 서버에 요청 메시지를 발행합니다. 커넥터는 응답을 기다립니다(최대 5000밀리초).
3. 새 MQ Workflow 프로세스 인스턴스 *CustomerCreditCheck*가 인스턴스화됩니다. 워크플로우 프로세스가 완료되기 전에 응답 메시지가 커넥터로 리턴됩니다. 메시지에는 프로세스 인스턴스 ID만이 포함됩니다.
4. 커넥터는 프로세스 인스턴스 정보로 *MyProcessInstance* 오브젝트를 채웁니다. MQ Workflow가 오류 메시지를 리턴하는 경우, 커넥터는 *BON_FAIL*을 리턴하고 워크플로우 메시지에서 제공되는 오류 메시지를 전송합니다.

동기 요청

하위 Meta Object에 음수가 아닌 ResponseTimeout 속성이 제공되고 ExecutionMode 속성이 Synchronous인 경우, 커넥터는 동기 요청을 발행합니다. 워크플로우 프로세스가 완료된 후 요청이 성공적으로 리턴됩니다. 동기 요청 처리는 협업이 협업을 시작하는 MQ Workflow 프로세스의 성공 또는 실패를 알리는 것을 보증합니다. 단기 트랜잭션의 경우, 동기 처리는 빠른 피드백을 생성하는 효율적인 방법입니다. 그림 6은 동기 요청을 보여줍니다.

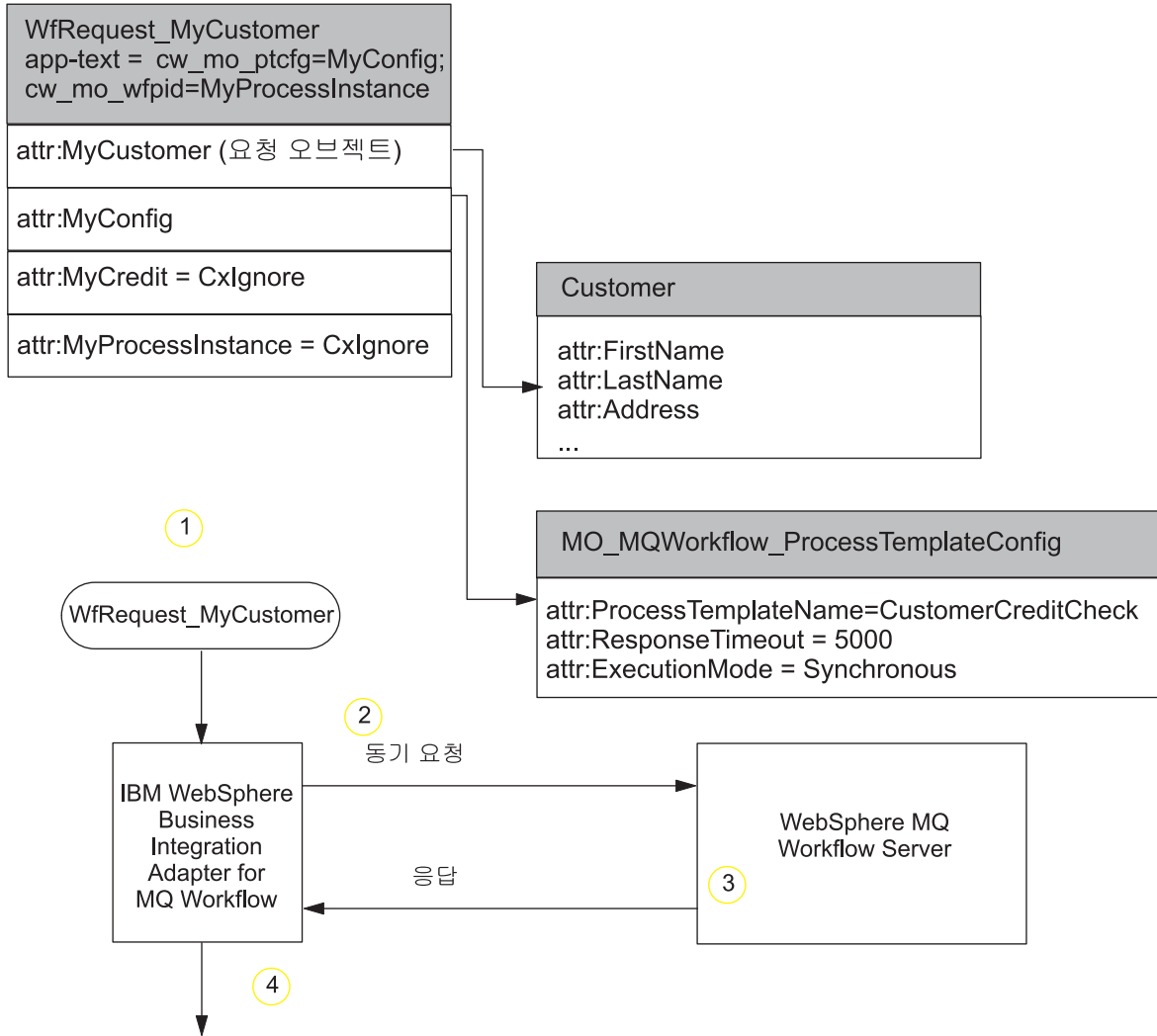


그림 6. MQ Workflow로의 샘플 동기 커넥터 요청

1. 커넥터는 WfRequest_MyCustomer 최상위 Business Object를 수신하고, 구성 Meta Object는 ResponseTimeout 속성에 음수가 아닌 값을 지정하며 ExecutionMode 속성에 Synchronous 값을 지정합니다.
2. 커넥터는 프로세스 템플릿 CustomerCreditCheck의 입력 데이터 구조로서 MyCustomer 오브젝트를 포함하는 MQ Workflow 서버에 요청 메시지를 발행합니다. 커넥터는 응답을 기다립니다(최대 5000밀리초).

3. 워크플로우 프로세스 CustomerCreditCheck의 새 인스턴스가 MQ Workflow에서 인스턴스화됩니다. 워크플로우 프로세스가 완료되면 응답 메시지가 커넥터로 리턴됩니다.

주: 장기 동기 트랜잭션의 경우, ResponseTimeout에 대한 값이 전체 프로세스가 실행될 수 있을 정도로 충분한지 확인하십시오.

4. 커넥터는 프로세스에서 리턴된 데이터 구조를 응답 오브젝트 MyCreditCheck로 변환합니다. 또한 커넥터는 프로세스 인스턴스 정보로 MyProcessInstance 오브젝트를 채웁니다. MQ Workflow가 오류 메시지를 리턴하는 경우, 커넥터는 BON_FAIL을 리턴하고 MQ Workflow 메시지에서 제공되는 오류 메시지를 전송합니다.

로케일 종속 데이터 처리

커넥터가 2바이트 문자 세트를 지원하고 지정된 언어로 메시지 텍스트를 전달하도록 국제화되었습니다. 커넥터가 한 문자 코드를 사용하는 위치에서 다른 문자 세트를 사용하는 위치로 데이터를 전송할 때 커넥터는 문자 변환을 수행하여 데이터 의미를 유지합니다.

JVM(Java Virtual Machine) 내의 Java 런타임 환경은 Unicode 문자 코드 세트에 데이터를 표시합니다. Unicode는 가장 잘 알려진 문자 코드 세트(단일 바이트와 다중 바이트 모두)로 된 문자의 인코딩을 포함합니다. WebSphere Business Integration System의 대부분의 구성요소는 Java로 작성됩니다. 따라서 데이터가 대부분의 통합 구성요소 사이에서 전송될 때 문자 변환은 필요하지 않습니다.

해당 국가 또는 지역에서 해당 언어로 오류 및 정보 메시지를 로그하려면 사용자 환경에 대한 Locale 표준 구성 등록 정보를 구성하십시오. 구성 등록 정보에 대한 자세한 정보는 83 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.

제 2 장 커넥터 설치 및 구성

이 장에서는 IBM WebSphere Business Integration adapter for WebSphere MQ Workflow를 설치하고 구성하는 방법을 설명합니다. 여기에는 다음과 같은 섹션이 들어 있습니다.

- 『어댑터 환경』
- 20 페이지의 『전제조건』
- 21 페이지의 『설치 작업 개요』
- 21 페이지의 『어댑터 및 관련 파일 설치』
- 21 페이지의 『설치된 파일 구조』
- 23 페이지의 『XML API를 사용하기 위해 WebSphere MQ Workflow 업그레이드』
- 24 페이지의 『커넥터 구성』
- 31 페이지의 『보증된 이벤트 전달 사용 가능』
- 35 페이지의 『최상위 레벨 Business Object 및 내용 구성』
- 37 페이지의 『Meta Object 구성』
- 52 페이지의 『시작 파일 구성』
- 53 페이지의 『커넥터의 다중 인스턴스 작성』
- 54 페이지의 『커넥터 시작』
- 56 페이지의 『커넥터 중지』

어댑터 환경

어댑터를 설치, 구성 및 사용하기 전에 해당 환경 요구사항을 이해해야 합니다. 요구사항은 다음 섹션에 나열됩니다.

- 28 페이지의 『MQSeries CCSID 커넥터 등록 정보의 변경』
- 20 페이지의 『어댑터 플랫폼』
- 20 페이지의 『세계화』

브로커 호환성

어댑터가 사용하는 어댑터 프레임워크는 어댑터가 통신하는 통합 브로커(또는 브로커)의 버전과 호환 가능해야 합니다. Adapter for WebSphere MQ Workflow 버전 2.5는 다음 어댑터 프레임워크 및 통합 브로커에서 지원됩니다.

- 어댑터 프레임워크:
WebSphere Business Integration Adapter Framework 버전 2.3.1 및 2.4.

- 통합 브로커:
 - WebSphere InterChange Server, 버전 4.1.1, 4.2, 4.2.1, 4.2.2
 - WebSphere MQ Integrator, 버전 2.1.0
 - WebSphere MQ Integrator Broker, 버전 2.1.0
 - WebSphere Business Integration Message Broker, 버전 5.0
 - WebSphere가 있는 WebSphere Application Server Enterprise, 버전 5.0.2
 - Studio Application Developer Integration Edition, 버전 5.0.1

예외사항은 릴리스 정보를 참조하십시오.

주: 통합 브로커 설치 지시사항 및 해당 전제조건은 다음 안내서를 참조하십시오.

WebSphere ICS(InterChange Server)는 *UNIX용 IBM WebSphere InterChange Server 시스템 설치 안내서* 또는 *Windows용 IBM WebSphere InterChange Server 시스템 설치 안내서*를 참조하십시오.

WebSphere 메시지 브로커는 *Implementing Adapters with WebSphere Message Brokers*를 참조하십시오.

WebSphere Application Server는 *Implementing Adapters with WebSphere Application Server*를 참조하십시오.

어댑터 플랫폼

어댑터를 설치하기 전에 시스템에 다음 소프트웨어가 설치되고 구성되어 있어야 합니다.

운영 체제:

다음 응용프로그램 플랫폼 중 하나에서 실행됩니다.

- AIX 5.1, AIX 5.2
- Solaris 7.0, Solaris 8.0
- HP UX 11i
- Windows 2000

제 3 자 소프트웨어:

IBM WebSphere MQ Workflow 버전 3.4

세계화

이 어댑터는 DBCS(double-byte character set)가 사용 가능하도록 설정되어 있습니다.

전제조건

Windows 2000 서버에 클라이언트를 설정해야 합니다. 이 작업을 수행하려면 *Windows 2000용 WebSphere MQ 빠른 시작* 문서를 참조하십시오.

설치 작업 개요

Connector for WebSphere MQ Workflow를 설치하려면 다음 작업을 수행해야 합니다.

- 통합 브로커 설치 WebSphere Business Integration System 설치와 통합 브로커 시작을 포함하는 이 작업은 사용자 브로커 및 운영 체제의 설치 문서에서 설명됩니다.
- 어댑터 및 관련 파일 설치 이 작업은 소프트웨어 패키지에서 어댑터의 파일을 사용자 시스템으로 설치하는 것을 포함합니다. 『어댑터 및 관련 파일 설치』를 참조하십시오.

어댑터 및 관련 파일 설치

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

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

설치된 파일 구조

다음 섹션에서는 설치 후 제품의 경로 및 파일 이름에 대해 설명합니다.

Windows 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

유틸리티는 커넥터 에이전트를 *ProductDir\connectors\WEBSPHEREMQWORKFLOW* 디렉토리에 설치하고 커넥터 에이전트의 바로 가기를 시작 메뉴에 추가합니다.

표 1에서는 커넥터가 사용하는 Windows 파일 구조에 대해 설명하며 설치 프로그램을 통한 커넥터 설치 선택 시 자동으로 설치되는 파일을 보여줍니다. 샘플 설치, 구성 및 실행에 대한 지시사항은 101 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

표 1. 커넥터를 위해 설치된 Windows 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors\WebSphereMQWorkflow\CWMQWorkflow.jar	WebSphere MQ Workflow 커넥터가 사용하는 클래스 파일을 포함합니다.
connectors\WebSphereMQWorkflow\utilities\FdlBorgen.jar	FDLBORGEN 유틸리티가 사용하는 클래스 파일을 포함합니다.
repository\WebSphereMQWorkflow\CN_WebSphereMQWorkflow.txt	커넥터의 저장소 정의
repository\WebSphereMQWorkflow\WebSphereMQWorkflow_MetaObjects.txt	커넥터가 사용하는 Meta Object의 저장소 정의
connectors\messages\WebSphereMQWorkflowConnector.txt	커넥터 메시지 파일
connectors\WebSphereMQWorkflow\start_WebSphereMQWorkflow.bat	커넥터의 시작 스크립트(2000)
connectors\WebSphereMQWorkflow\utilities\fdlborgen.bat	FDLBORGEN 유틸리티의 시작 스크립트(NT/2000)
connectors\WebSphereMQWorkflow\Samples\WebSphereMQWorkflow_Samples.fdl	샘플 워크플로우가 들어 있는 워크플로우 정의 파일
connectors\WebSphereMQWorkflow\Samples\Sample_MQWF_Order_Collaborations.in	샘플 협업의 저장소 정의

표 1. 커넥터를 위해 설치된 Windows 파일 구조 (계속)

<i>ProductDir</i> 의 서브디렉토리	설명
connectors\WebSphereMQWorkflow\Samples\Sample_MQWF_Order_Connectors.in	샘플 오브젝트의 저장소 정의
connectors\WebSphereMQWorkflow\Samples\Sample_MQWF_Order_Objects.in	샘플 오브젝트의 저장소 정의
connectors\WebSphereMQWorkflow\Samples\collaboration_classes\SampleItemOrderSync.class	샘플 협업 클래스
connectors\WebSphereMQWorkflow\Samples\collaboration_classes\SampleItemSync.class	샘플 협업 클래스
connectors\WebSphereMQWorkflow\Samples\collaboration_classes\SampleWorkflowProcessControl.class	샘플 협업 클래스
connectors\WebSphereMQWorkflow\Samples\collaboration_classes\SampleItemActivityImpl.class	샘플 협업 클래스
connectors\WebSphereMQWorkflow\Samples\map_classes\MQWF_Sample_GB0toResponse.class	샘플 맵 클래스
connectors\WebSphereMQWorkflow\Samples\map_classes\MQWF_Sample_RequesttoGB0.class	샘플 맵 클래스

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

UNIX 커넥터 파일 구조

설치 프로그램은 커넥터와 연관된 표준 파일을 사용자 시스템으로 복사합니다.

유틸리티는 커넥터 에이전트를 *ProductDir/connectors/WEBSPHEREMQWORKFLOW* 디렉토리에 설치합니다.

표 2에서는 커넥터가 사용하는 UNIX 파일 구조에 대해 설명하며 설치 프로그램을 통한 커넥터 설치 선택 시 자동으로 설치되는 파일을 보여줍니다. 샘플 설치, 구성 및 실행에 대한 지시사항은 101 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

표 2. 커넥터에 설치된 UNIX 파일 구조

<i>ProductDir</i> 의 서브디렉토리	설명
connectors/WebSphereMQWorkflow/CWMQWorkflow.jar	WebSphere MQ Workflow 커넥터가 사용하는 클래스 파일을 포함합니다.
connectors/WebSphereMQWorkflow/utilities/FdlBorgen.jar	FDLBORGEN 유틸리티가 사용하는 클래스 파일을 포함합니다.
repository/WebSphereMQWorkflow/CN_WebSphereMQWorkflow.txt	커넥터의 저장소 정의
repository/WebSphereMQWorkflow/WebSphereMQWorkflow_MetaObjects.txt	커넥터가 사용하는 Meta Object의 저장소 정의
connectors/messages/WebSphereMQWorkflowConnector.txt	커넥터 메시지 파일
connectors/WebSphereMQWorkflow/start_WebSphereMQWorkflow.sh	커넥터의 시스템 시작 스크립트. 이 스크립트는 일반 커넥터 관리자 스크립트에서 호출됩니다. 시스템 관리자의 커넥터 구성 화면을 누르면, 설치 프로그램은 이 커넥터 관리자 스크립트의 사용자 정의된 랩퍼를 작성합니다. 커넥터를 시작하고 중지하려면 이 사용자 정의된 랩퍼를 사용하십시오.
connectors/WebSphereMQWorkflow/utilities/fdlborgen.sh	FDLBORGEN 유틸리티의 시작 스크립트
connectors/WebSphereMQWorkflow/Samples/WebSphereMQWorkflow_Samples.fdl	샘플 워크플로우가 들어 있는 워크플로우 정의 파일
connectors/WebSphereMQWorkflow/Samples/Sample_MQWF_Order_Collaborations.in	샘플 협업의 저장소 정의
connectors/WebSphereMQWorkflow/Samples/Sample_MQWF_Order_Connectors.in	샘플 오브젝트의 저장소 정의
connectors/WebSphereMQWorkflow/Samples/Sample_MQWF_Order_Objects.in	샘플 오브젝트의 저장소 정의
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleItemOrderSync.class	샘플 협업 클래스
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleItemSync.class	샘플 협업 클래스
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleWorkflowProcessControl.class	샘플 협업 클래스
connectors/WebSphereMQWorkflow/Samples/collaboration_classes/SampleItemActivityImpl.class	샘플 협업 클래스
connectors/WebSphereMQWorkflow/Samples/map_classes/MQWF_Sample_GB0toResponse.class	샘플 맵 클래스
connectors/WebSphereMQWorkflow/Samples/map_classes/MQWF_Sample_RequesttoGB0.class	샘플 맵 클래스

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

XML API를 사용하기 위해 WebSphere MQ Workflow 업그레이드

XML API를 사용하기 위해 WebSphere MQ Workflow를 업그레이드하려면(XML API에 대한 자세한 정보는 12 페이지의 『XML API Verb 처리』 참조) 다음 작업을 수행해야 합니다.

1. WebSphere MQ Workflow 3.3.2를 설치하십시오.
2. Suspend, Terminate와 Resume뿐 아니라 Delete와 Restart verb를 지원하는 협업을 작성하십시오. XML API는 이 verb 모두를 사용하는 것을 지원합니다.
3. 커넥터 특정 등록 정보인 JavaCorbaApi를 추가하고 그 값을 false로 설정하십시오.
4. 다음 속성을 MO_MQWorkflow_ProcessInstance Business Object에 추가하십시오.

표 3. 커넥터 특정 구성 등록 정보

이름	유형	응용프로그램 특정 정보
ProcTemplValidFromDate	String	ProcTemplValidFromDate; type=pcdata;
ProcInstSuspensionTime	String	ProcInstSuspensionTime; type=pcdata;
ProcInstSuspensionExpirationsTime	String	ProcInstSuspensionExpirationsTime; type=pcdata;
ExternalProcessContext	String	ExternalProcessContext; type=pcdata

주: 플로우를 사용하고 있는 경우 InterChange 서버에서 모니터링할 때 이 속성을 정의되지 않은 상태로 두십시오(아래의 7번 참조).

또는

Repository 폴더의 repos_copy MQWorkflowMetaObject.txt 및 CNMQWorkflow.txt에 추가하십시오.

5. Suspend 및 Resume Verb를 처리하려면 다음 응용프로그램 특정 정보를 MO_MQWorkflow_ProcessInstnace에 추가하십시오.

```
Suspend    deep=true;
Resume     deep=true;
```

deep=true이면, 자동이 아닌 모든 서브프로세스도 일시중단되거나 재개됩니다. deep=false이면, 같은 서브프로세스가 무시됩니다. deep의 기본값은 false입니다.

6. 워크플로우 프로세스의 상태를 제어하거나 모니터링하려면 MQ_MQWorkflow_ProcessInstance 오브젝트의 ProcInstName(ProcInstID 대신)을 사용하십시오.

- ExternalProcessContext가 오브젝트에 정의되지 않은 상태로 있으면 InterChange 서버는 ObjectEvent ID를 통해 플로우 모니터링 정보를 어댑터로 전달합니다.

커넥터 구성

커넥터에는 두 가지 유형의 구성 등록 정보, 표준 구성 등록 정보 및 어댑터 특정 구성 등록 정보가 있습니다. 어댑터를 실행하기 전에 이 등록 정보 값을 설정해야 합니다.

Connector Configurator를 사용하여 다음과 같이 커넥터 등록 정보를 구성하십시오.

- Connector Configurator 및 단계별 프로시저에 대한 설명은 101 페이지의 부록 B 『Connector Configurator』를 참조하십시오.
- 표준 커넥터 등록 정보에 대한 설명은 『표준 커넥터 등록 정보』 및 83 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오.
- 커넥터 특정 등록 정보에 대한 설명은 『커넥터 특정 등록 정보』를 참조하십시오.

표준 커넥터 등록 정보

표준 구성 등록 정보는 모든 커넥터가 사용하는 정보를 제공합니다. 이 등록 정보에 대한 문서는 83 페이지의 부록 A 『커넥터의 표준 구성 등록 정보』를 참조하십시오. Connector Configurator의 구성 등록 정보를 설정하는 경우, BrokerType 등록 정보를 사용하여 브로커를 지정해야 합니다. 이 등록 정보가 설정되면 브로커와 관련된 등록 정보가 Connector Configurator 창에 나타납니다. 자세한 정보는 101 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

커넥터 특정 등록 정보

커넥터 특정 구성 등록 정보는 Connector for WebSphere MQ Workflow에 필요한 정보를 제공합니다. 커넥터 특정 등록 정보는 또한 에이전트의 재코드화 및 재빌드없이 커넥터에서 정적 정보 또는 논리를 변경하는 방법을 제공합니다.

표 4에는 커넥터의 커넥터 특정 등록 정보가 나열되어 있습니다. 등록 정보의 설명에 대해서는 다음 섹션을 참조하십시오.

표 4. 커넥터 특정 구성 등록 정보

이름	가능한 값	기본값	필수
ApplicationPassword	로그인 암호	password	아니오
ApplicationUserID	로그인 사용자 ID	ADMIN	아니오
ArchiveQueue	정상적으로 처리된 메시지의 사본이 전송되는 대기열	MQWFCONN.ARCHIVE	아니오
MQSeriesCCSID	대기열 관리자 연결에 대한 문자-세트	null	아니오
MQSeriesChannel	MQ 서버 커넥터 채널	FMCQM.CL.TCP	예
DataHandlerClassName	Data Handler 클래스 이름	com.crossworlds. .DataHandlers.text.xml	아니오
DataHandlerConfigMO	Data Handler Meta Object	MO_DataHandler_Default	예

표 4. 커넥터 특정 구성 등록 정보 (계속)

이름	가능한 값	기본값	필수
DataHandlerMimeType	MIME 유형의 파일	text/xml	아니오
ErrorQueue	오류를 포함하는 메시지 대기열	MQWFCONN.ERROR	아니오
MQSeriesHostName	WebSphere MQ Workflow 대기열 관리자를 포함하는 시스템 이름	공백	예
InDoubtEvents	FailOnStartup 재처리 IgnoreLogError	Reprocess	아니오
InputQueue	WebSphere MQ Workflow 요청을 풀링할 때 기열	CWLDINPUTQ	예
InProgressQueue	진행 중 이벤트 대기열	MQWFCONN.IN_PROGRESS	예
JavaCorbaApi	값이 true로 설정되면 Java CORBA API 사용 가능	false	아니오
OutputQueue	WebSphere MQ Workflow에 요청을 발행하기 위한 대기열	FMC.FMCGRP.EXE. XML	예
PollQuantity	입력 대기열에서 검색할 메시지 수	1	아니오
MQSeriesPort	WebSphere MQ (MQSeries) 리스너에 설정된 포트	14000	예
ReplyToQueue	커넥터가 요청을 발행할 때 응답 메시지가 전달되는 대기열	MQWFCONN.REPLYTO	아니오
UnsubscribedQueue	등록되지 않은 메시지가 전송되는 대기열	MQWFCONN.UNSUBSCRIBE	아니오
MQSeriesQueueManager	워크플로우의 대기열 관리자	FMC(공백인 경우 기본 대기열 관리자가 사용됨)	아니오
BOPrefix	등록 전송의 경우, 데이터 구조 이름이 이 접두부에 추가되어 최상위 레벨 Object의 이름을 판별합니다.	MQWF_	아니오
WorkflowSystemName	커넥터가 워크플로우 프로세스의 직접 제어를 위해 바인드하는 WebSphere MQ Workflow 시스템 이름	FMCSYS	아니오
WorkflowSystemGroup	워크플로우 프로세스 직접 제어를 위해 연결을 설정하는 WebSphere MQ Workflow 시스템 그룹 이름	FMCGRP	아니오
WorkflowAgentLocatorPolicy	로컬 또는 원격 WebSphere MQ Workflow 서버 연결을 지정합니다. LOC 커넥터가 로컬 시스템에서 실행 중인 서버에 연결됩니다. OSA 커넥터가 WebSphere MQ Workflow 서버에 원격으로 연결됩니다.	LOC	아니오
WorkflowAgentName	WebSphere MQ Workflow CORBA 에이전트 이름입니다.		아니오

ApplicationPassword

WebSphere MQ에 로그인하기 위해 UserID와 함께 사용되는 암호

기본값 = password

ApplicationPassword가 공백이거나 제거된 경우, 커넥터는 WebSphere MQ Workflow에서 제공하는 기본 암호를 사용합니다.

ApplicationUserID

커넥터가 WebSphere MQ Workflow 서버로 전달하여 연결 권한을 부여합니다(Java 직접 바인딩 API). 이 등록 정보는 또한 다음과 같은 경우에 사용됩니다.

- WebSphere MQ Workflow 및 WebSphere MQ Workflow 프로세스 실행을 통한 두 가지 메시지 전송에 모두 권한을 부여하기 위해 사용됩니다.
- 사용자가 메타 오브젝트(XML API)에 UserID 속성 지정에 실패하는 경우, WebSphere MQ Workflow 응용프로그램에 대한 요청 조작 시 커넥터에서 사용됩니다.

ApplicationUserID가 공백이거나 제거된 경우, 커넥터는 WebSphere MQ Workflow 에서 제공하는 기본 사용자 ID를 사용합니다.

이 등록 정보나 Meta Object 속성 UserID에서 사용자 ID를 지정할 수 있습니다. 어느 경우이나, ApplicationUserID는 다음과 같아야 합니다.

- WebSphere MQ Workflow 응용프로그램에 정의되고 모든 필수 권한(예를 들어, 지정된 워크플로우를 실행하기 위해)을 갖습니다.
- 로컬 운영 체제에 정의됩니다.
- 사용자 로컬 시스템에서 그룹 mqm의 구성원으로 WebSphere MQ 메시지가 권한 아래에서 전송될 수 있습니다.

주: ApplicationUserID는 WebSphere MQ Workflow에서 사용되는 WebSphere MQ 서버 연결 채널의 MCA 등록 정보에 지정되지 않습니다. 기본값으로 WebSphere MQ Workflow는 이 등록 정보의 사용자 fmc를 지정하여, 사용자 fmc의 권한으로 전송되는 WebSphere MQ Workflow 응용프로그램 및 어댑터 사이의 모든 메시지가 교환되도록 합니다. WebSphere MQ 서버 연결 채널 등록 정보에서 이 값을 지워 이 커넥터 특정 등록 정보에서 사용자가 지정하는 ApplicationUserID로 메시지가 전송될 수 있게 하십시오.

기본값 = ADMIN.

ArchiveQueue

정상적으로 처리된 메시지의 사본이 전송되는 대기열입니다.

기본값 = MQWFCNN.ARCHIVE

MQSeriesChannel

커넥터가 WebSphere MQ와 통신하기 위해 사용하는 WebSphere MQ Workflow 서버 커넥터 채널입니다.

기본값 = FMCQM.CL.TCP

채널이 공백이거나 제거된 경우, 커넥터는 WebSphere MQ Workflow에서 제공하는 기본 서버 채널을 사용합니다.

DataHandlerClassName

메시지를 Business Object로 변환하거나 역으로 변환할 때 사용할 Data Handler 클래스입니다.

기본값 = com.crossworlds.DataHandlers.text.xml

DataHandlerConfigMO

구성 정보를 제공하기 위해 Data Handler로 전달되는 Meta Object입니다.

기본값 = MO_DataHandler_Default

DataHandlerMimeType

특정 MIME 유형에 근거한 Data Handler를 요청하게 합니다.

기본값 = text/xml

ErrorQueue

처리할 수 없는 메시지가 전송되는 대기열입니다.

기본값 = MQWFCNN.ERROR

MQSeriesHostName

WebSphere MQ Workflow를 호스트하는 서버 이름

기본값 = 공백

InDoubtEvents

예기치 않은 커넥터 종료로 인해 완전히 처리되지 않은 진행 중 이벤트를 처리하는 방식을 지정합니다. 초기화 중에 진행 중인 대기열에서 이벤트를 찾은 경우 취할 수 있는 네 가지 조치 중 하나를 선택하십시오.

- FailOnStartup. 오류를 로그하고 즉시 종료합니다.
- Reprocess. 우선 나머지 이벤트를 처리한 후, 입력 대기열에서 메시지를 처리합니다.
- Ignore. 진행 중인 대기열에서 모든 메시지를 무시합니다.
- LogError. 오류를 로그하지만 종료하지 않습니다.

기본값 = Reprocess

InputQueue

새 메시지에 대해 커넥터가 폴링하는 메시지 대기열입니다.

기본값 = CWLDINPUTQ

InProgressQueue

진행 중 이벤트 대기열입니다.

기본값 = MQWFCONN.IN_PROGRESS

JavaCorbaApi

WebSphere MQ Workflow 3.2.2와 함께 사용하려면 Java CORBA API를 사용 가능하게 해야 합니다. false이면, 커넥터가 WebSphere MQ Workflow 3.3.2 이상과 함께 사용하기 위한 XML API를 지원하지 않습니다. WebSphere MQ Workflow 3.4 이상을 사용 중인 경우, 이 등록 정보는 false여야 합니다.

기본값 = false

MQSeriesCCSID

WebSphere MQ Workflow의 대기열 관리자 연결에는 CCSID를 사용합니다. 이 값은 WebSphere MQ Workflow에 대한 대기열 관리자의 CCSID 등록 정보와 일치해야 합니다.

Default = blank(왼쪽 공백일 경우 819로 봄)

선택된 문자를 지원하려면 CCSID를 변경해야 합니다. 변경할 경우, WebSphere MQ Workflow 대기열의 CCSID뿐만 아니라 CCSID 커넥터 특정 등록 정보도 변경해야만 합니다.

MQSeries CCSID 커넥터 등록 정보의 변경:

1. System Manager의 MQWF Connector를 두 번 누르십시오. Connector Designer -- MQWorkflowConnector가 열립니다.
2. 응용프로그램 구성 등록 정보 탭을 누르십시오.
3. MQSeriesCCSID 등록 정보에 새 값을 입력하십시오(예: "943").
4. 커넥터를 다시 시작하십시오.
5. ICS를 다시 시작하십시오(권장).

MQSeries CCSID 대기열 등록 정보의 변경:

1. 명령 프롬프트에서 RUMMQSC FMCQM을 실행하십시오.
2. ALTER QMGR CCSID(new_value)를 입력한 후 Enter를 누르십시오.
3. END를 입력한 후 Enter를 누르십시오.

OutputQueue

MQSeries Workflow로 요청을 발행하는 대기열입니다.

기본값 = FMC.FMCGRP.EXE.XML

PollQuantity

입력 대기열에서 검색할 메시지 수입니다.

기본값 =1

MQSeriesPort

MQSeries(WebSphere MQ) 리스너에 설정된 포트입니다.

기본값 = 14000

ReplyToQueue

커넥터가 요청을 발행할 때 응답 메시지가 전달되는 대기열입니다.

주: 메시지를 WebSphere MQ Workflow 응용프로그램으로 전송 중인 경우, 커넥터는 커넥터가 응답을 기대하는지의 여부와는 상관없이 아웃바운드 메시지의 머릿글에 ReplyToQueue 필드를 채웁니다. 이렇게 하면, 올바르게 않은 비즈니스 데이터가 MQ Workflow 응용프로그램에 전송될 때 문제점을 식별하는데 도움을 줍니다.

기본값 = MQWFCONN.REPLYTO

UnsubscribedQueue

등록되지 않은 메시지가 전송되는 대기열입니다. 커넥터는 다음과 같은 경우 이 대기열 등록 정보에 메시지를 전송합니다.

- 커넥터가 MQSTR 또는 FMCXML 형식이 아닌 메시지를 검색합니다.
- 커넥터가 WfMessage를 검색하지만 메시지 이름 유형이 ActivityImplInvoke가 아니거나 General Error입니다.
- 커넥터가 등록 전송 처리 시 데이터 구조의 최상위 레벨 Business Object를 찾을 수 없습니다.

기본값 = MQWFCONN.UNSUBSCRIBE

MQSeriesQueueManager

WebSphere MQ Workflow의 대기열 관리자입니다.

기본값 = FMC(공백인 경우 기본 대기열 관리자가 사용됨)

BOPrefix

등록 전송의 경우 이 접두부에 추가되는 데이터 구조 이름입니다. 데이터 구조는 트랜잭션을 위한 최상위 레벨 Business Object 이름을 판별합니다.

기본값 = MQWF_

WorkflowSystemName

커넥터에 워크플로우 프로세스 직접 제어가 필요한 경우 연결이 설정될 WebSphere MQ Workflow 시스템 이름입니다.

기본값 = FMCSYS

WorkflowSystemGroup

커넥터가 워크플로우 프로세스 직접 제어를 필요로 하는 경우 연결이 설정될 WebSphere MQ Workflow 시스템 그룹 이름

기본값 = FMCGRP

WorkflowAgentLocatorPolicy

커넥터가 *WorkflowSystemName* 및 *WorkflowSystemGroup* 등록 정보에 의해 식별되는 WebSphere MQ Workflow 서버 연결을 설정하는 방법을 지정합니다. 가능한 값은 다음과 같습니다.

- LOC. 커넥터가 로컬 시스템에서 실행하는 WebSphere MQ Workflow 서버에 연결됩니다.
- OSA. 커넥터가 IBM Java ORB(Object Request Broker)를 사용하여 WebSphere MQ Workflow 서버에 원격으로 연결됩니다. WebSphere MQ Workflow는 IBM Java ORB 클라이언트를 지원하도록 구성되어야 합니다. 자세한 정보는 *WebSphere MQ Workflow 설치 안내서* 및 *WebSphere MQ Workflow Programming Guide*를 참조하십시오.

기본값 = LOC

주: IBM Java ORB를 사용하여 클라이언트 연결을 지원하려면 `start_MQWorkflow.bat`(Windows) 또는 `start_MQWorkflow.sh`(UNIX) 파일을 수정해야 합니다. 적절한 `start_MQWorkflow` 파일을 열어 Step 3으로 시작하는 주석과 모든 다음 행이 표시될 때까지 아래로 화면이동한 후 제공된 지시사항에 표시되는 대로 경로를 조정하십시오. 이렇게 하면 초기화 시 WebSphere MQ Workflow 클라이언트 라이브러리가 올바른 IBM Java ORB 라이브러리를 로드한 후 사용하도록 할 수 있습니다. 이 수정은 InterChange Server와의 통신에 영향을 주지 않습니다.

WorkflowAgentName

WebSphere MQ Workflow CORBA 에이전트 이름입니다.

기본값 = 없음

보증된 이벤트 전달 사용 가능

다음 방법 중 하나로 JMS 사용 가능 커넥터의 보증된 이벤트 전달 기능을 구성할 수 있습니다.

- 커넥터가 JMS 이벤트 저장소(JMS 소스 대기열로서 구현됨)를 사용하는 경우, 커넥터 프레임워크가 JMS 이벤트 저장소를 관리할 수 있습니다. 자세한 정보는 『JMS 이벤트 저장소를 사용하는 커넥터의 보증된 이벤트 전달』을 참조하십시오.
- 커넥터가 JMS가 아닌 이벤트 저장소(예를 들어, JDBC 테이블, 전자 우편 편지함 또는 텍스트 파일로 구현됨)를 사용하는 경우, 커넥터 프레임워크는 중복된 이벤트가 발생하지 않도록 보장하기 위해 JMS 모니터 대기열을 사용할 수 있습니다. 자세한 정보는 33 페이지의 『JMS가 아닌 이벤트 저장소를 사용하는 커넥터의 보증된 이벤트 전달』을 참조하십시오.

JMS 이벤트 저장소를 사용하는 커넥터의 보증된 이벤트 전달

JMS 사용 가능 커넥터가 JMS 대기열을 사용하여 자체 이벤트 저장소를 구현하는 경우, 커넥터 프레임워크는 “컨테이너” 역할을 수행하여 JMS 이벤트 저장소(JMS 소스 대기열)를 관리할 수 있습니다. 커넥터는 단일 JMS 트랜잭션을 통해 소스 대기열에서 메시지를 제거한 후 목적지 대기열에 지정할 수 있습니다. 이 섹션에서는 JMS 이벤트 저장소를 사용하는 JMS 사용 가능 커넥터의 보증된 이벤트 전달 기능 사용 방법에 대한 다음 정보를 제공합니다.

- 『MS 이벤트 저장소를 사용하는 커넥터의 기능 사용 가능』
- 33 페이지의 『이벤트 폴링 효과』

MS 이벤트 저장소를 사용하는 커넥터의 기능 사용 가능

JMS 이벤트 저장소를 사용하는 JMS 사용 가능 커넥터의 보증된 이벤트 전달 기능을 사용 가능하게 하려면 커넥터 구성 등록 정보의 값을 표 5에 표시되어 있는 대로 설정하십시오.

표 5. JMS 이벤트 저장소를 사용하는 커넥터의 보증된 이벤트 전달 커넥터 등록 정보

커넥터 등록 정보	값
DeliveryTransport	JMS
ContainerManagedEvents	JMS
PollQuantity	이벤트 저장소의 단일 폴에서 처리할 이벤트의 수

표 5. JMS 이벤트 저장소를 사용하는 커넥터의 보증된 이벤트 전달 커넥터 등록 정보 (계속)

커넥터 등록 정보	값
SourceQueue	<p>커넥터 프레임워크가 풀하고 처리할 이벤트를 검색하는 JMS 소스 대기열(이벤트 저장소)의 이름</p> <p>주: 소스 대기열과 다른 JMS 대기열은 동일한 대기열 관리자의 일부여야 합니다. 커넥터의 응용프로그램이 서로 다른 대기열 관리자에 저장되어 있는 이벤트를 생성하는 경우, 원격 대기열 관리자에 대한 원격 대기열 정의를 정의해야 합니다. 그런 다음 WebSphere MQ가 원격 대기열에서 JMS 사용 가능 커넥터가 통합 브로커로의 전송에 사용하는 대기열 관리자로 이벤트를 전송할 수 있습니다. 원격 대기열 정의 구성 방법에 대한 정보는 IBM WebSphere MQ 문서를 참조하십시오.</p>

커넥터 구성뿐만 아니라, JMS 저장소의 이벤트와 Business Object 간에 변환하는 Data Handler 또한 구성해야 합니다. 이 Data Handler 정보는 표 6에서 요약하는 커넥터 구성 등록 정보로 구성됩니다.

표 6. 보증된 이벤트 전달을 위한 Data Handler 등록 정보

Data Handler 등록 정보	값	필수
MimeType	Data Handler가 처리하는 MIME 유형. 이 MIME 유형은 호출할 Data Handler를 식별합니다.	예
DHClass	Data Handler를 구현하는 Java 클래스의 전체 이름	예
DataHandlerConfigMName	MIME 유형과 해당 Data Handler를 연관시키는 최상위 레벨 Meta Object의 이름	선택적

주: Data Handler 구성 등록 정보는 기타 커넥터 구성 등록 정보와 함께 커넥터 구성 파일에 있습니다.

보증된 이벤트 전달을 사용하기 위해 JMS 이벤트 저장소를 사용하는 커넥터를 구성하는 경우, 표 5 및 표 6에서 설명하는 대로 커넥터 등록 정보를 설정해야 합니다. 이 커넥터 구성 등록 정보를 설정하려면 Connector Configurator 도구를 사용하십시오. Connector Configurator는 해당 표준 등록 정보 탭의 표 5에 커넥터 등록 정보를 표시합니다. 또한 Data Handler 탭의 표 6에 커넥터 등록 정보를 표시합니다.

주: Connector Configurator는 DeliveryTransport 커넥터 구성 등록 정보를 JMS로 설정하고 ContainerManagedEvents를 JMS로 설정하는 경우에만 해당 Data Handler 탭의 필드를 활성화합니다.

Connector Configurator에 대한 정보는 101 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

이벤트 폴링 효과

커넥터에서 ContainedManagedEvents를 JMS로 설정하여 보증된 이벤트 전달을 사용하는 경우, 이 기능을 사용하지 않는 커넥터와 약간 다르게 작동합니다. 커넥터 프레임워크는 컨테이너 관리 이벤트를 제공하기 위해 다음 단계에 따라 이벤트 저장소를 폴합니다.

1. JMS 트랜잭션을 시작하십시오.
2. 이벤트 저장소에서 JMS 메시지를 읽으십시오.

이벤트 저장소는 JMS 소스 대기열로 구현됩니다. JMS 메시지는 이벤트 레코드가 들어 있습니다. JMS 소스 대기열의 이름은 SourceQueue 커넥터 구성 등록 정보에서 확보합니다.

3. Data Handler를 호출하여 이벤트를 Business Object로 변환하십시오.

커넥터 프레임워크는 32 페이지의 표 6의 등록 정보로 구성된 Data Handler를 호출합니다.

4. WebSphere MQ Integrator Broker가 통합 브로커인 경우, 구성된 와이어 형식(XML)에 따라 Business Object를 메시지로 변환하십시오.

5. 결과 메시지를 JMS 목적지 대기열로 전송하십시오.

WebSphere ICS 통합 브로커를 사용하는 경우, JMS 목적지 대기열로 전송된 메시지는 Business Object입니다. WebSphere MQ Integrator Broker를 사용하는 경우, JMS 목적지 대기열로 전송된 메시지는 XML 메시지(Data Handler가 생성)입니다.

6. JMS 트랜잭션을 확약하십시오.

JMS 트랜잭션이 확약되면, JMS 목적지 대기열에 메시지가 기록되고 동일한 트랜잭션의 JMS 소스 대기열에서 제거됩니다.

7. 루프 1 - 6단계를 반복합니다. PollQuantity 커넥터 등록 정보가 이 루프가 반복되는 횟수를 결정합니다.

중요: ContainerManagedEvents 등록 정보를 JMS로 설정한 커넥터는 이벤트 폴링을 수행하는 pollForEvents() 메소드를 호출하지 않습니다. 커넥터의 기본 클래스에 pollForEvents() 메소드가 포함되는 경우, 이 메소드는 호출되지 않습니다.

JMS가 아닌 이벤트 저장소를 사용하는 커넥터의 보증된 이벤트 전달

JMS 사용 가능 커넥터가 이벤트 저장소(예: JDBC 이벤트 테이블, 전자 우편 편지함 또는 텍스트 파일)를 사용하는 경우, 커넥터 프레임워크는 중복된 이벤트 제거를 사용하여 중복된 이벤트가 발생하지 않도록 보장할 수 있습니다. 이 섹션에서는 JMS가 아닌 이벤트 저장소를 사용하는 JMS 사용 가능 커넥터의 보증된 이벤트 전달 기능 사용 방법에 대한 다음 정보를 제공합니다.

- 34 페이지의 『JMS가 아닌 이벤트 저장소를 사용하는 커넥터의 기능 사용 가능』

- 33 페이지의 『이벤트 폴링 효과』

JMS가 아닌 이벤트 저장소를 사용하는 커넥터의 기능 사용 가능: JMS가 아닌 이벤트 저장소를 사용하는 JMS 사용 가능 커넥터의 보증된 이벤트 전달 기능을 사용 가능하게 하려면 커넥터 구성 등록 정보의 값을 표 7에 표시되어 있는 대로 설정해야 합니다.

표 7. JMS가 아닌 이벤트 저장소를 사용하는 커넥터의 보증된 이벤트 전달 커넥터 등록 정보

커넥터 등록 정보	값
DeliveryTransport	JMS
DuplicateEventElimination	true
MonitorQueue	커넥터 프레임워크가 처리된 Business Object의 ObjectEventId를 저장하는 JMS 모니터 대기열의 이름

보증된 이벤트 전달을 사용하도록 커넥터를 구성하는 경우, 표 7에서 설명하는 대로 커넥터 등록 정보를 설정해야 합니다. 이 커넥터 구성 등록 정보를 설정하려면 Connector Configurator 도구를 사용하십시오. 표준 등록 정보 탭에 이 커넥터 등록 정보를 표시합니다. Connector Configurator에 대한 정보는 101 페이지의 부록 B 『Connector Configurator』를 참조하십시오.

이벤트 폴링 효과: 커넥터에서 DuplicateEventElimination을 true로 설정하여 보증된 이벤트 전달을 사용하는 경우, 이 기능을 사용하지 않는 커넥터와 약간 다르게 작동합니다. 커넥터 프레임워크는 중복 이벤트 제거를 제공하기 위해 JMS 모니터 대기열을 사용하여 Business Object를 추적합니다. JMS 모니터 대기열의 이름은 MonitorQueue 커넥터 구성 등록 정보에서 확보합니다.

커넥터 프레임워크가 응용프로그램 특정 구성요소에서 Business Object를 수신하고 나면(pollForEvents() 메소드의 getAppEvent()를 호출하여), 현재 Business Object(getAppEvents()에서 수신)가 중복 이벤트를 표시하는지 판별해야 합니다. 커넥터 프레임워크는 이러한 판별을 하기 위해 다음과 같이 JMS 모니터 대기열에서 Business Object를 검색하여 ObjectEventId를 현재 Business Object의 ObjectEventId와 비교합니다.

- 이 두 가지 ObjectEventIds가 동일한 경우, 현재 Business Object는 중복 이벤트를 나타냅니다. 이러한 경우, 커넥터 프레임워크는 현재 Business Object가 표시하는 이벤트를 무시하여, 이 이벤트를 통합 브로커로 전송하지 않습니다.
- 이 ObjectEventIds가 동일하지 않은 경우, Business Object는 중복 이벤트를 표시하지 않습니다. 이러한 경우, 커넥터 프레임워크는 현재 Business Object를 JMS 모니터 대기열로 복사한 후 JMS 전달 대기열로 전달합니다(모두 동일한 JMS 트랜잭션의 일부로). JMS 전달 대기열의 이름은 DeliveryQueue 커넥터 구성 등록 정보에서 확보합니다. 제어는 커넥터의 pollForEvents() 메소드로 리턴됩니다 (getAppEvent() 메소드를 호출한 후).

JMS 사용 가능 커넥터가 중복 이벤트 제어를 지원하도록 하려면, 커넥터의 pollForEvents() 메소드에 다음 단계가 포함되어야 합니다.

- JMS가 아닌 이벤트 저장소에서 검색된 이벤트 레코드에서 Business Object를 작성하려면, 이벤트 레코드의 고유한 이벤트 ID를 Business Object의 ObjectEventId 속성으로 저장해야 합니다.

응용프로그램은 이 이벤트 ID를 생성하여 이벤트 저장소의 이벤트 레코드를 고유하게 식별합니다. 이벤트를 통합 브로커로 전송하고 이 이벤트 레코드의 상태가 변경되기 전에 커넥터의 작동이 중지되는 경우, 이 이벤트 레코드는 진행 중 상태로 이벤트 저장소에 그대로 남아 있게 됩니다. 커넥터가 다시 작동하면, 진행 중 이벤트를 복구해야 합니다. 커넥터가 폴링을 재개하면, 이벤트 저장소에 계속 남아 있는 이벤트 레코드에 대해 Business Object를 생성합니다. 그러나 이미 전송된 Business Object와 새 Business Object 모두 해당 ObjectEventIds와 동일한 이벤트를 갖고 있어, 커넥터 프레임워크가 새 Business Object를 중복으로 인식하여 통합 브로커로 전송할 수 없습니다.

- 커넥터 복구 시, 커넥터가 새 이벤트에 대한 폴링을 시작하기 전에 진행 중 이벤트를 처리해야 합니다.

커넥터가 시작 시 진행 중 이벤트를 폴 준비 상태로 변경해야 폴링 메소드가 재처리할 이벤트 레코드를 선택합니다.

최상위 레벨 Business Object 및 내용 구성

메타 데이터는 WebSphere MQ Workflow의 WfMessage 구조에 있는 Business Object 데이터와 함께 임베드됩니다. 이 구조는 커넥터와 XML 메시지 API를 사용하는 WebSphere MQ Workflow 간의 모든 요청 및 응답의 기본이 됩니다. 모든 메시지 구조는 다음 그림 7에 표시됩니다.

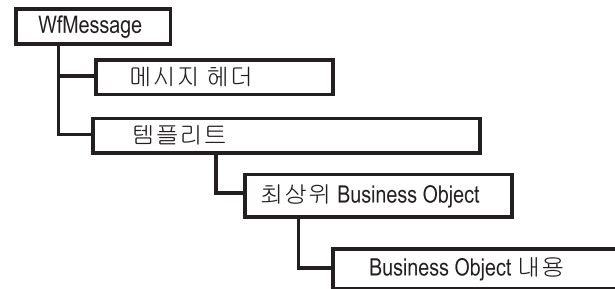


그림 7. WebSphere MQ Workflow 메시지 구조

XML API의 명령 및 리턴값은 “템플릿”가 둘러쌉니다. 이 템플릿은 결과 포함뿐만 아니라 전체 명령을 WebSphere MQ Workflow에 지정하는 데 필요한 구조를 제공합니다. 요청된 조치에 따른 템플릿 유형 변경사항 및 대부분의 경우 비즈니스 내

용은 템플리트의 하위 요소에 포함됩니다. 비즈니스 내용을 식별하려면 커넥터가 각 템플리트를 고유하게 인식해야 합니다. 템플리트 이름이 다르므로 하위 요소의 이름도 다릅니다.

커넥터는 다음과 같은 세 가지 템플리트 및 해당 연관 응답 구조를 처리할 수 있습니다.

ProcessTemplateExecute

동기 또는 비동기로 프로세스를 실행하기 위해 커넥터가 WebSphere MQ Workflow 서버로 전송합니다. 비동기로 프로세스가 실행되는 경우, WebSphere MQ Workflow에 의해 발행되는 응답은 없습니다. 동기 프로세스를 실행하는 경우, 워크플로우 프로세스가 완료된 후에만 응답이 리턴됩니다. 워크플로우 입력 데이터 구조를 나타내는 Business Object는 ProcInstInputData 하위 요소에 포함됩니다.

ProcessTemplateExecuteResponse

커넥터가 발행하는 동기 요청에 대한 응답으로 WebSphere MQ Workflow에 의해 전송됩니다. 워크플로우 프로세스 결과인 Business Object는 ProcInstOutputData 하위 요소에 포함됩니다. 더 이상 활성화되지 않고 워크플로우를 제어하기 위해 더 이상 사용될 수 없는 경우에도 PID(Process Instance Identifier)가 리턴됩니다.

ProcessTemplateCreateAndStartInstance

비동기로 프로세스를 처리하기 위해 커넥터가 WebSphere MQ Workflow 서버로 전송합니다. ProcessTemplateExecute 템플리트와 달리, 활성화 PID(Business Object 대신)를 포함하는 커넥터에 즉시 응답이 발행됩니다. 이 PID는 후에 워크플로우 프로세스를 제어하기 위해 사용할 수 있습니다. 워크플로우에 지정되는 데이터 구조를 나타내는 Business Object는 ProcInstInputData 하위 요소에 포함됩니다.

ProcessTemplateCreateAndStartInstanceResponse

커넥터가 전송하는 요청에 대한 응답으로 WebSphere MQ Workflow에 의해 전송됩니다. Business Object 없이도 PID가 리턴됩니다(워크플로우가 비동기로 실행되는 것으로 가정되므로).

ActivityImplInvoke

비즈니스 내용이 Interchange Server에 게시되도록 요청하기 위해 WebSphere MQ Workflow에 의해 커넥터로 전송됩니다. Business Object는 ProgramInputData 하위 요소에 포함됩니다. WebSphere MQ Workflow는 동기 요청을 위해 워크플로우로 리턴된 비즈니스 내용의 기본값이 들어 있는 ProgramOutputDataDefault 추가 하위 요소를 포함할 수 있습니다.

ActivityImplInvokeResponse

이벤트 폴링 시 처리되는 동기 요청을 완료하기 위해 커넥터가 WebSphere MQ Workflow로 리턴합니다. 협업에 의해 리턴되는 Business Object는 ProgramOutputData 하위 요소에 추가됩니다.

처리된 템플릿 구조에 따라 커넥터는 다음 XML 하위 요소 중 한 요소에서 비즈니스 내용을 검색 또는 추가해야 합니다.

- ProcInstInputData
- ProcInstOutputData
- ProgramInputData
- ProgramOutputData

Meta Object 구성

Business Object는 데이터 구조를 나타내는 하나 이상의 하위 오브젝트를 보유하는 최상위 레벨 Business Object로 구성됩니다. 이는 각 Business Object가 실제 비즈니스 내용이 들어 있는 하나 이상의 하위 오브젝트와, 메타 데이터가 들어 있는 일부 하위 오브젝트를 보유함을 의미합니다.

최상위 레벨 Business Object는 교환될 모든 오브젝트의 래퍼입니다. 최상위 레벨 Business Object는 래퍼로서 비즈니스 데이터를 보유하지 않으며 교환 시작 및 완료에 필요한 오브젝트를 작성하는 데 필요한 이름-값 쌍 및 응용프로그램 특정 텍스트를 제공합니다. 이름-값 쌍은 요청 또는 필수 Business Object 작성에 필요한 메타 데이터를 지정하는 하위 속성을 표시합니다.

그림 8은 다음 경우에 최상위 레벨 Business Object 및 하위 오브젝트 관계를 보여줍니다.

- 커넥터가 이벤트를 폴링하고 WebSphere MQ Workflow에서 요청을 수신하는 경우
- WebSphere MQ Workflow 모드가 비동기 또는 동기이며 협업 이름이 WebSphere MQ Workflow 명령행 매개변수에 지정되어 있는 경우(62 페이지의 그림 20 참조)

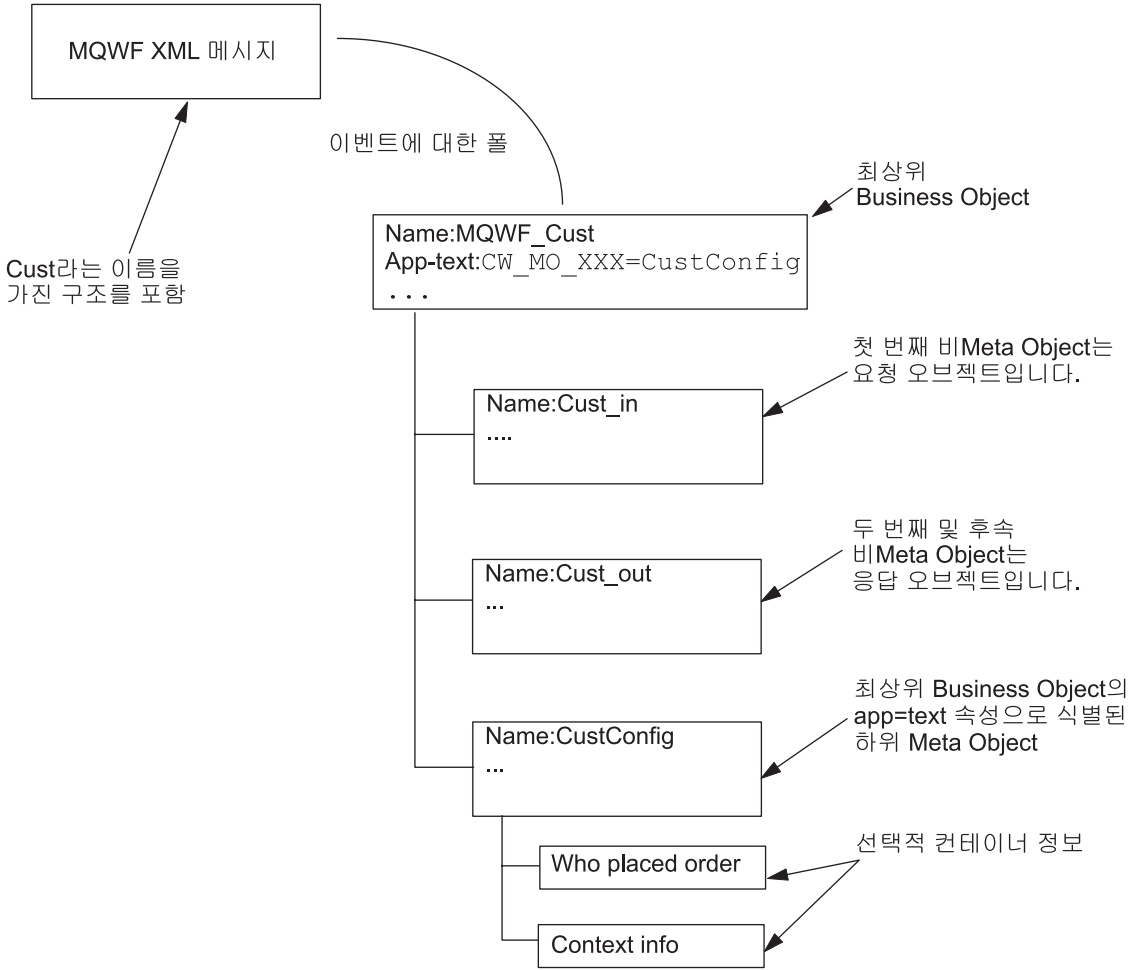


그림 8. 이벤트 풀링 시나리오에서의 최상위 레벨 Business Object 및 하위(Business Object 및 Meta Object)

그림 9는 다음 경우에 최상위 레벨 Business Object 및 하위 오브젝트 관계를 보여줍니다.

- 커넥터가 이벤트를 풀링하고 WebSphere MQ Workflow에서 요청을 수신하는 경우
- WebSphere MQ Workflow 모드는 동기이며 협업 이름은 WebSphere MQ Workflow 명령행 매개변수에 지정되어 있지 않는 경우(62 페이지의 그림 20 참조)

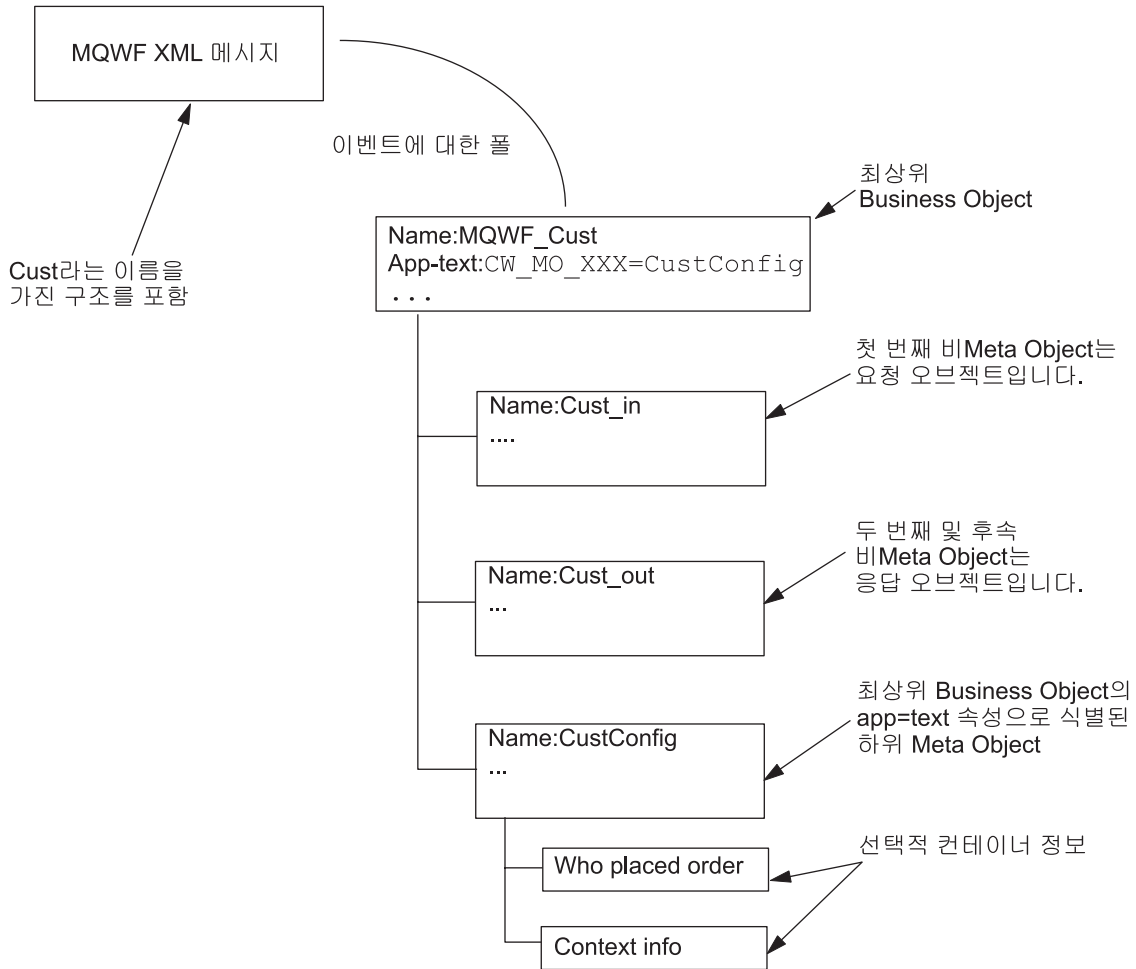


그림 9. 이벤트 폴링 시나리오에서의 최상위 레벨 Business Object 및 하위 Meta Object

그림 10은 다음 경우에 최상위 레벨 Business Object와 하위 오브젝트의 관계를 보여 줍니다.

- 커넥터가 WebSphere MQ Workflow에서 협업 대신 요청을 발행하는 경우
- cw_mo_wfactivityresponse 태그가 최상위 레벨 Business Object의 응용프로그램 특정 정보(app-text)에 지정되어 있지 않은 경우

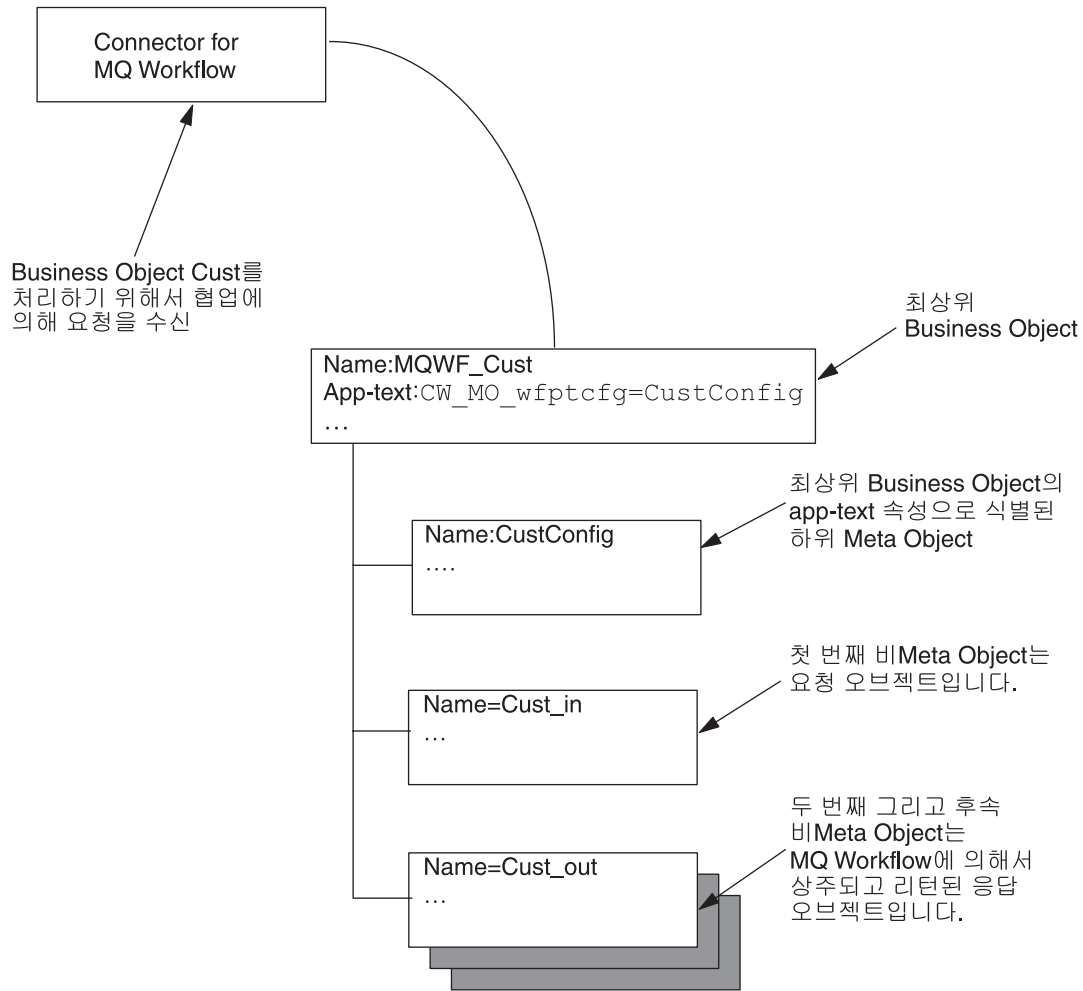


그림 10. 요청 처리 시나리오에서의 최상위 레벨 Business Object 및 하위(Business 및 Meta) Object

그림 11은 다음 경우에 최상위 레벨 Business Object와 하위 오브젝트의 관계를 보여 줍니다.

- 커넥터가 WebSphere MQ Workflow에서 협업 대신 요청을 발행하는 경우
- cw_mo_wfactivityresponse 태그가 최상위 레벨 Business Object의 응용프로그램 특정 정보(app-text)에 지정되어 있는 경우(참조)

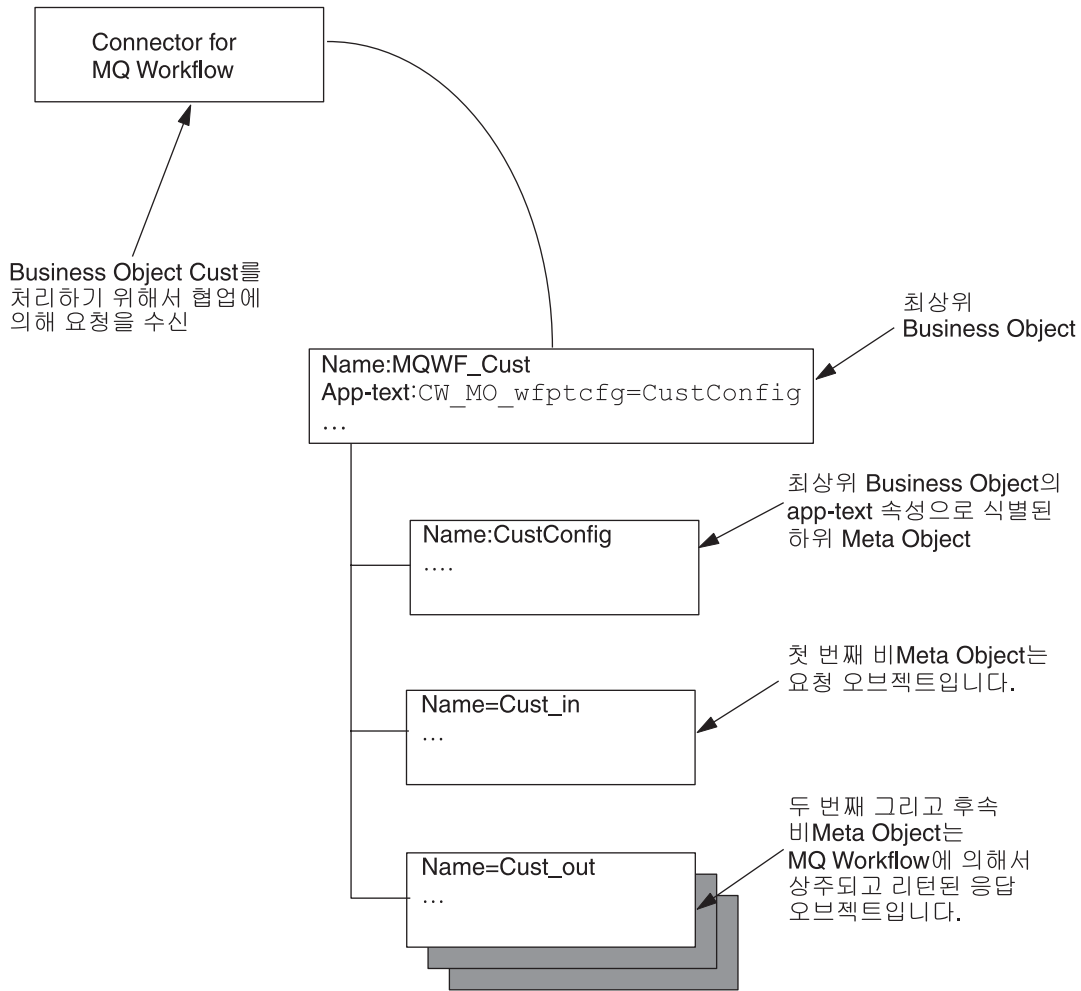


그림 11. 요청 처리 시나리오에서의 최상위 레벨 Business Object 및 하위(Business 및 Meta) Object

WebSphere MQ Workflow 프로세스가 다른 입력 및 출력 데이터 구조를 가질 수는 있지만, 협업 및 커넥터 관련 트랜잭션은 한 가지 오브젝트 유형만을 포함합니다. 이러한 제한사항을 피하려면, 최상위 레벨 Business Object는 하위 오브젝트로 하나의 요청 오브젝트와 하나 이상의 응답 오브젝트로 구성되어야 합니다. 특히, `<boprefix>` 구성 등록 정보에 추가되는 데이터 구조 이름은 작성되는 최상위 레벨 Business Object를 판별합니다. 이 최상위 레벨 Business Object의 첫 번째(관련된 비 메타 데이터) 하위 오브젝트는 데이터 구조로 채워집니다.

하위 Meta Object와 비즈니스 내용 하위 오브젝트를 구별하려면 최상위 레벨 Business Object의 응용프로그램 특정 정보에 다음과 같은 태그가 포함되어야 합니다.

`cw_mo_tag=child meta-object_attribute_name`

주: 다중 메타 데이터 태그는 세미콜론으로 분리되어야 합니다. 분리문자 주위의 공백은 무시됩니다. (예를 들어, `cw_mo_foo = bar`은 `cw_mo_foo=bar`와 동일합니다.)

tag는 다음 중 한 가지입니다.

- wfptcfg WebSphere MQ Workflow 프로세스 템플릿 메타 데이터 --
『MO_MQWorkflow_ProcessTemplateConfig』 참조
- wfcontainer WebSphere MQ Workflow 컨테이너 정보 --
43 페이지의 『MO_MQWorkflow_ContainerInfo』 참조
- wfpid WebSphere MQ Workflow 프로세스 인스턴스 정보 --
49 페이지의 『MO_MQWorkflow_ProcessInstance』 참조
- wfactivityresponse WebSphere MQ Workflow 활동 정보 --
51 페이지의 『MO_MQWorkflow_ActivityResponse』 참조

cw_mo_로 시작하는 모든 응용프로그램 특정 정보는 구성 또는 동적 메타 데이터용으로 예약되어 있습니다. 따라서 커넥터 에이전트가 Business Object를 수신하면, Business Object 자체의 응용프로그램 특정 정보를 단순히 확인하여 Business Object의 런타임 메타 데이터가 포함되었는지를 즉시 판별할 수 있습니다. Data Handler 또한 Business Object 레벨에서 응용프로그램 특정 정보를 확인하여 직렬화 또는 직렬화 해제 프로세스에 어떤 하위 오브젝트가 포함되고 제외되었는지 판별합니다.

예를 들어, 구성 Meta Object를 필요로 하는 WfRequest_MyCustomer 최상위 레벨 Business Object를 고려하십시오. MyConfig 유형의 오브젝트 구성 속성을 지정할 수 있습니다. 커넥터가 Meta Object를 인식하고 Data Handler가 상위 오브젝트 배열 시 “MyConfig”를 통합하지 못하도록 하려면, cw_mo_wfptcfg=MyConfig 태그 양식으로 WRequest_MyCustomer 오브젝트에 응용프로그램 특정 정보를 추가하십시오.

WebSphere MQ Workflow의 요청 또는 응답 구성 시 커넥터는 템플릿을 사용하여 Business Object를 구성합니다.

MO_MQWorkflow_ProcesTemplateConfig

커넥터가 작성 및 실행될 WebSphere MQ Workflow 프로세스에 대한 정보를 제공하기 위해서는 Meta Object가 최상위 레벨 Business Object에 포함되어야 합니다. 이 WebSphere MQ Workflow에는 사용할 프로세스 템플릿, 응답 필요 여부, 결과를 리턴하기 전에 프로세스가 완료될 때까지 Business Object가 대기해야 하는지 여부에 대한 정보가 포함됩니다. 커넥터는 이 정보를 Meta Object에 저장함으로써 요청 WebSphere MQ Workflow 프로세스의 응용프로그램 특정 정보를 동적으로 구성할 수 있습니다. 이 Meta Object(또는 이에 상응하는 오브젝트)는 모든 협업 요청에 필요합니다.

커넥터는 다음과 같이 최상위 레벨 Business Object의 응용프로그램 특정 정보를 읽고 이름값 쌍을 찾습니다.

```
cw_mo_wfptcfg=xxx
```

여기서 xxx는 메타 데이터를 지정하는 하위 속성 이름입니다. 표 8에서는 속성 이름 및 설명을 보여줍니다.

표 8. MO_MQWorkflow_ProcessTemplateConfig Meta Object 속성

속성 이름	설명	승인값
ProcessTemplateName(필수) ProcessInstanceName	실행할 WebSphere MQ Workflow 템플릿 이름. 기본값 = 없음 실행할 WebSphere MQ Workflow 인스턴스 이름. 비동기 실행 모드에는 적용되지 않습니다.	모두 모두
KeepName	기본값이 공백으로 남게 되면 프로세스 템플릿의 새 인스턴스가 작성됩니다. 프로세스 사용 후 버리기 여부를 표시하는 플래그	true 또는 false
UserID	기본값 = false 프로세스 실행 권한을 갖고 있는 사용자를 식별합니다. 이것은 커넥터 특정 등록 정보인 ApplicationUserID와 동일합니다. 이 속성에 대한 중요한 제한사항은 26 페이지의 『ApplicationUserID』를 참조하십시오.	모두
ResponseTimeout	기본값 = 커넥터 구성 등록 정보 ApplicationUserID의 값 WebSphere MQ Workflow에서 응답을 기다리는 시간(밀리초). 양수 정수 값은 커넥터의 응답 대기 시간을 필요로 합니다. 음수값은 일단 WebSphere MQ Workflow 입력 대기열에 요청이 발행되면 커넥터가 성공적으로 리턴하도록 합니다.	
TimeoutFatal	기본값 = -1 WebSphere MQ Workflow에 의해 응답이 수신되지 않는 경우, 커넥터가 InterChange Server로 BON_APPRESPONSETIMEOUT을 리턴하고 커넥터 에이전트를 종료합니다.	true 또는 false
ExecutionMode	기본값 = false(ResponseTimeout이 0보다 작은 경우에는 적용되지 않음) 협업과 관련하여 프로세스의 비동기 또는 동기 실행 여부를 판별합니다. 이 모드가 Asynchronous인 경우, 프로세스 템플릿의 새 인스턴스가 작성 및 실행됩니다. 추적 목적으로 PID가 협업으로 리턴됩니다. 이 모드가 Synchronous인 경우, 프로세스 템플릿의 인스턴스(기존 또는 새 인스턴스)가 실행됩니다. 워크플로우 프로세스가 완료되면 결과 Business Object가 협업으로 리턴됩니다.	Asynchronous 또는 Synchronous
	기본값 = Synchronous	

MO_MQWorkflow_ContainerInfo

WebSphere MQ Workflow에서 발행된 활동 호출(ActivityImplInvoke) 메시지는 Business Object 외에 컨테이너 정보를 포함할 수 있습니다. 이 컨테이너 정보는 하위 Meta Object MO_MQWorkflow_ContainerInfo(정의된 경우)로 맵핑되고 등록 협업으로 공개될 수 있습니다. 이 정보에는 프로세스 시작 환경 및 조건에 대한 WebSphere MQ Workflow에서 제공하는 정보가 포함됩니다.

주: MO_MQWorkflow_ContainerInfo 메타 데이터는 정보용으로만 사용해야 합니다. 협업은 요청 프로세스를 시작하는 사용자 역할 또는 프로세스 모델을 기반으로 이 정보를 무시하거나 다른 조치를 수행하도록 선택할 수 있습니다.

커넥터는 다음과 같이 최상위 레벨 Business Object의 응용프로그램 특정 정보를 읽고 이름-값 쌍을 찾습니다.

```
cw_mo_wfcontainer=XXX
```

여기서 XXX는 최상위 레벨 Business Object 정보로 채울 하위 속성 이름입니다. 이 정보에는 커넥터에 대한 값이 없으며 Business Object 처리에 사용되지 않습니다. 이 정보를 Business Object의 일부로서 커넥터에 전달해도 효과는 없습니다.

MO_MQWorkflow_ContainerInfo 정의는 다음과 같습니다.

```
[ReposCopy]
    Version = 3.1.0
[End]
[BusinessObjectDefinition]
    Name = MO_MQWorkflow_ProcessInfo
    Version = 1.0.0

    [Attribute]
    Name = Role
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = true
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
[End]
    [Attribute]
    Name = Organization
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
[End]
    [Attribute]
    Name = ProcessAdministrator
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
[End]
    [Attribute]
    Name = Duration
    Type = String
    Cardinality = 1
    MaxLength = 255
    IsKey = false
    IsForeignKey = false
    IsRequired = false
    IsRequiredServerBound = false
[End]
```

```

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

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]
[BusinessObjectDefinition]
Name = MO_MQWorkflow_ActivityInfo
Version = 1.0.0

[Attribute]
Name = Priority
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = MembersOfRoles
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = CoordinatorOfRole
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false

```

```

IsRequiredServerBound = false
[End]
[Attribute]
Name = Organization
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = OrganizationType
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = LowerLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = UpperLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = People
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PersonToNotify
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false

```



```

IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration2
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]
[BusinessObjectDefinition]
Name = MO_MQWorkflow_ContainerInfo
Version = 1.0.0

[Attribute]
Name = PROCESS_INFO
Type = MO_MQWorkflow_ProcessInfo
ContainedObjectVersion = 1.0.0

```

```

Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ACTIVITY_INFO
Type = MO_MQWorkflow_ActivityInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ACTIVITY
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PROCESS
Type = String
Cardinality = 1
MaxLength = 1
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PROCESS_MODEL
Type = String
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false

```

```

[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

```

MO_MQWorkflow_ProcessInstance

선택적으로, PID(Process Instance ID)는 프로세스 실행에 대한 응답으로 WebSphere MQ Workflow에 의해 리턴됩니다. 워크플로우 프로세스 작성 또는 실행 완료할 때마다, 커넥터는 이 오브젝트를 프로세스 세부사항으로 채웁니다. 프로세스가 시작 협업과 병렬로 실행하는 경우, 협업은 PID를 사용하여 프로세스 인스턴스를 제어할 수 있습니다(협업에 비동기로 프로세스를 실행한 경우).

커넥터는 다음과 같이 상위 레벨 Business Object의 응용프로그램 특정 정보를 읽고 이름-값 쌍을 찾습니다.

```

cw_mo_wfpid

=XXX

```

여기서 XXX는 프로세스 인스턴스 메타 데이터가 들어 있어야 하는 하위 속성 이름입니다. 오브젝트는 XML Data Handler가 설정한 요구사항을 따라야 합니다. 또한 하위 오브젝트의 속성 이름은 커넥터에 대한 의미값을 가져야 합니다. 오브젝트의 응용프로그램 특정 정보에는 “ProcessInstance”가 포함되어야 합니다.

모든 사용자 정의 오브젝트가 이 샘플에서 파생되도록 적극 권장됩니다.

표 9. MO_MQWorkflow_ProcessTemplateInstance Meta Object 속성

속성 이름	설명	승인값
ProcInstID	프로세스 인스턴스를 식별하는 1차 키	모두
ProcessInstName	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	
ProcInstParentName	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	
ProcInstTopLevelName	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	모두

표 9. MO_MQWorkflow_ProcessTemplateInstance Meta Object 속성 (계속)

속성 이름	설명	승인값
ProcInstDescription	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	정수
ProcInstState	프로세스 상태	SuspendedResumedTerminated
LastStateChangeTime	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	Asynchronous 또는 Synchronous
LastModificationTime	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	
ProcTempID	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	
ProcTempIName	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	
Icon	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	
Category	WebSphere MQ Workflow Programming Guide를 참조 하십시오.	

MO_MQWorkflow_ActivityRequest

WebSphere MQ Workflow 프로세스 인스턴스가 활동 호출(ActivityImplInvoke) 메시지를 발행하면 다음 조건이 충족됩니다.

- WebSphere MQ Workflow 모드는 동기입니다(64 페이지의 그림 22 참조).
- 협업 이름이 WebSphere MQ Workflow 명령행 매개변수에 지정되어 있지 않습니다(62 페이지의 그림 20 참조).

커넥터는 ActivityImplInvoke 메시지에서 활동 정보를 받습니다. 이 활동 정보는 MO_MQWorkflow_ActivityRequest 하위 Meta Object로 맵핑된 후 등록된 협업을 공개합니다.

커넥터는 다음과 같이 최상위 레벨 Business Object의 응용프로그램 특정 정보를 읽고 이름값 쌍을 찾습니다.

```
cw_mo_wfactivityrequest
```

```
=XXX
```

여기서 XXX는 최상위 레벨 Business Object 정보로 채울 하위 속성 이름입니다. 이 정보에는 커넥터에 대한 값이 없으며 Business Object 처리에도 사용되지 않습니다. 표 10에서는 속성 이름 및 설명을 보여줍니다.

표 10. MO_MQWorkflow_ActivityRequest 속성

속성 이름	설명	승인값
ActImplCorrelID	다음 ActivityImplInvokeResponse 메시지와 함께 프로세스 인스턴스에 의해 발행되는 ActivityImplInvoke 메시지를 서로 연관시키는 ID. 협업은 응답을 커넥터로 보 내기 위해 ActImplCorrelID 를 사용합니다.	모두
Starter	프로세스 인스턴스를 시작하는 사용자 ID	모두
ProcTempID	프로세스 템플릿의 ID.	모두
ProgramName	프로세스 인스턴스에 의해 호출되는 프로그램 이름	모두
ResponseRequired	프로세스(또는 인스턴스)가 ActivityImplInvokeResponse 메 시지를 예상했는지를 식별합니다.	예 또는 아니오 또는 IfError
ExternalProcessContext	프로세스 인스턴스의 프로세스 컨텍스트	모두

MO_MQWorkflow_ActivityResponse

요청과 응답을 상호 연관시키려면 커넥터에는 적절한 정보를 가진 Meta Object를 포함하는 최상위 레벨 Business Object가 필요합니다. 이 Meta Object는 ActivityImplInvoke 메시지와 연관된 ActImplCorrelID, 프로세스 인스턴스와 연관된 사용자 ID 및 프로세스 인스턴스에 링크된 리턴 코드와 관련있는 정보를 포함하고 있습니다. 협업이 ActivityImplInvokeResponse와 ActivityImplInvoke 메시지를 연관시킬 때 이 Meta Object가 필요합니다.

커넥터는 다음과 같이 최상위 레벨 Business Object의 응용프로그램 특정 정보를 읽고 이름값 쌍을 찾습니다.

```
cw_mo_wfactivityrequest
```

```
=XXX
```

여기서 XXX는 메타 데이터를 지정하는 하위 속성 이름입니다. 표 11에서는 속성 이름 및 설명을 보여줍니다.

표 11. MO_MQWorkflow_ActivityResponse 속성

속성 이름	설명	승인값
ActImplCorrelID	프로세스 인스턴스가 ActivityImplInvokeResponse 메시지와 연관되어 호출되는 요청과 응답을 서로 관련시키는 ID.	모두

표 11. MO_MQWorkflow_ActivityResponse 속성 (계속)

속성 이름	설명	승인값
Starter	ActivityImplInvoke 메시지와 연관되어 프로세스 인스턴스를 시작하는 사용자 ID	모두
ReturnCode	프로세스 인스턴스를 발행하는 리턴 코드	정수

응용프로그램 특정 정보

상위 Business Object 레벨의 응용프로그램 특정 정보는 이름-값 쌍 형식으로 구성되며 세미콜론으로 분리됩니다. 공백은 무시됩니다. 예를 들면, 다음과 같습니다.

```
cw_mo_wfptcfg=CUST_Config;cw_mo_pid=CUST_IN_Nieman
```

시작 파일 구성

Connector for WebSphere MQ Workflow를 시작하려면 시작 파일을 구성해야 합니다.

Windows

Windows용 커넥터의 구성을 완료하려면 다음과 같이 start_WebSphereMQWorkflow.bat 파일을 수정해야 합니다.

1. start_WebSphereMQWorkflow.bat 파일을 여십시오.
2. STEP 1로 시작하는 섹션으로 이동하여 사용자 MQSeries Java 클라이언트 라이브러리 위치를 지정하십시오.
3. STEP 2로 시작하는 섹션으로 화면이동하여 사용자 Workflow Java 클라이언트 라이브러리 위치를 지정하십시오.
4. IBM Java ORB를 통해 WebSphere MQ Workflow에 연결하려는 경우, 3단계로 시작하는 섹션으로 화면이동하여 IBM Java ORB 라이브러리 위치를 지정한 후 이 행의 주석을 없애십시오.

UNIX

UNIX용 커넥터 구성을 완료하려면 다음과 같이 start_WebSphereMQWorkflow.sh 파일을 수정해야 합니다.

1. start_WebSphereMQWorkflow.sh 파일을 여십시오.
2. STEP 1로 시작하는 섹션으로 이동하여 사용자 MQSeries Java 클라이언트 라이브러리 위치를 지정하십시오.
3. STEP 2로 시작하는 섹션으로 화면이동하여 사용자 Workflow Java 클라이언트 라이브러리 위치를 지정하십시오.

4. IBM Java ORB(Object Request Broker)를 통해 WebSphere MQ Workflow에 연결하려는 경우, 3단계로 시작하는 섹션으로 화면이동하여 IBM Java ORB 라이브러리 위치를 지정한 후 이 행의 주석을 없애십시오.

커넥터의 다중 인스턴스 작성

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

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

새 디렉토리 작성

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

```
ProductDir\connectors\connectorInstance
```

여기서 connectorInstance는 고유하게 커넥터 인스턴스를 식별합니다.

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

```
ProductDir\Repository\connectorInstance
```

Business Object 정의 작성

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

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

```
ProductDir\Repository\initialConnectorInstance
```

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

커넥터 정의 작성

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

1. 초기 커넥터의 구성 파일(커넥터 정의)을 복사한 후 이름을 바꾸십시오.
2. 커넥터 인스턴스에 지원되는 Business Object(및 연관된 Meta Object)가 올바르게 나열되는지 확인하십시오.
3. 커넥터 등록 정보를 적절하게 사용자 정의하십시오.

시작 스크립트 작성

시작 스크립트를 작성하려면 다음을 수행하십시오.

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

`dirname`

2. 시작 스크립트를 53 페이지의 『새 디렉토리 작성』에서 작성한 커넥터 디렉토리에 배치하십시오.
3. 시작 스크립트 단축 아이콘을 작성하십시오(Windows 전용).
4. 초기 커넥터의 단축 아이콘 텍스트를 복사한 후 새 커넥터 인스턴스의 이름과 일치하도록 초기 커넥터의 이름을 변경하십시오(명령행에서).

이제 통합 서버에서 두 커넥터 인스턴스를 동시에 실행할 수 있습니다.

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

커넥터 시작

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

`ProductDir\connectors\connName`

여기서 `connName`은 커넥터를 식별합니다. 시작 스크립트의 이름은 표 12에 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.

표 12. 커넥터의 시작 스크립트

운영 체제	시작 스크립트
UNIX 기반 시스템	<code>connector_manager_connName</code>
Windows	<code>start_connName.bat</code>

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

- Windows 시스템의 경우, 시작 메뉴에서 다음을 수행하십시오.

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

- 명령행에서 커넥터를 시작할 경우에는 다음을 수행하십시오.

- Windows 시스템의 경우:

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

- UNIX 기반 시스템의 경우:

```
connector_manager_connName -start
```

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

- WebSphere InterChange Server의 경우 *brokerName*에 ICS 인스턴스의 이름을 지정하십시오.
- WebSphere Message Broker(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 message broker 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, 시스템 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, 시스템 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, 시스템 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, 시스템 종료 또는 삭제할 수 있습니다.

- Windows 시스템에서 Windows 서비스로 시작하도록 커넥터를 구성할 수 있습니다. 이 경우, Windows 시스템이 종료되면 커넥터가 중지됩니다.

제 3 장 WebSphere MQ Workflow 응용프로그램 수정

WebSphere MQ Workflow 응용프로그램의 경우, 노드는 작업 완료 프로세스 단계에 있습니다. 각 노드는 역할(부서 책임자)에 해당하는 특정 활동(예: 주문 승인)과 연관됩니다.

노드는 요청을 발행하고 다른 노드 또는 WebSphere MQ Workflow의 외부 응용프로그램에 응답할 수 있습니다. 노드가 Connector for WebSphere MQ Workflow와 통신할 수 있도록 하려면, 먼저 UPES(User-Defined Program Execution Server)를 지정해야 합니다.

이 장에서는 Connector for WebSphere MQ Workflow와 상호작용할 수 있는 UPES(User-Defined Program Execution Server) 작성 방법에 대해 설명합니다. 여기에서는 다음과 같은 주제를 다룹니다.

- 『UPES 구성』

UPES 구성

이 장에서는 UPES(User-defined Program Execution Server) 정의 및 구성 방법에 대해 설명합니다. 워크플로우 노드는 UPES를 통해 Connector for WebSphere MQ Workflow에 요청을 발행할 수 있습니다.

주: UPES를 구성하기 전에 시스템에 WebSphere MQ Workflow Buildtime 환경을 설치해야 합니다.

1. MQ Workflow Buildtime 응용프로그램을 시작한 후 네트워크 탭을 누르십시오.

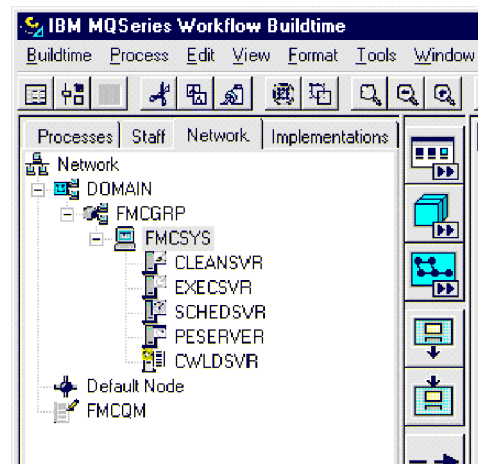


그림 12. WebSphere MQ Workflow Buildtime: 네트워크 보기

2. 메뉴 표시줄에서 시스템 > 새 사용자 정의 프로그램 실행 서버를 선택하십시오.

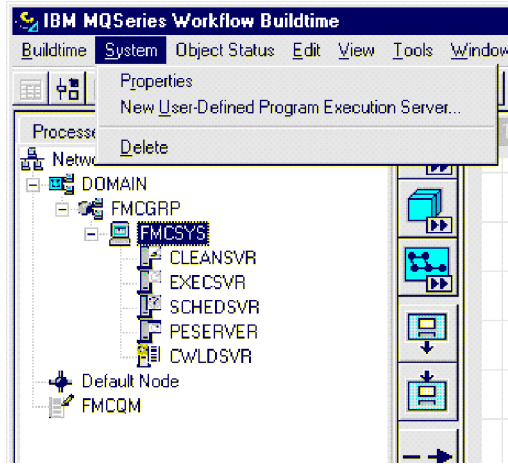


그림 13. WebSphere MQ Workflow Buildtime: 새 UPES 선택

3. 대화 상자에서 UPES의 고유한 이름(예: CWLDSVR)을 입력하십시오.

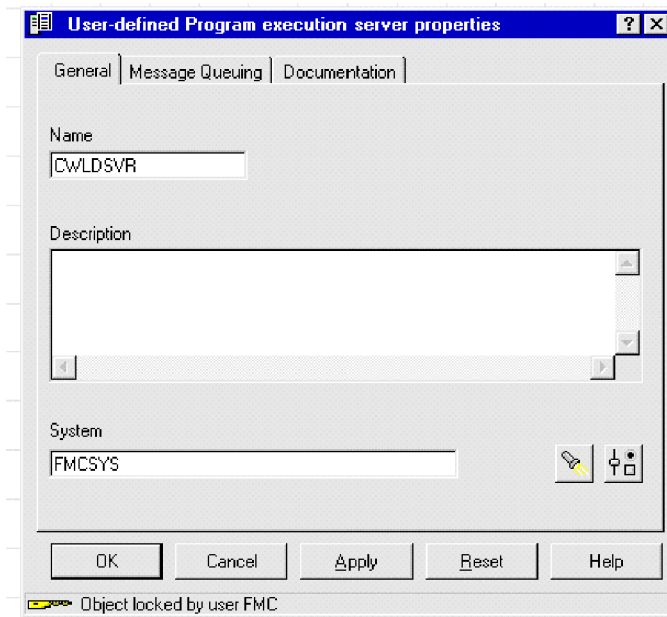


그림 14. WebSphere MQ Workflow Buildtime: 새 UPES 이름 지정

4. 메시지 대기열 탭을 누르고 커넥터용 대기열 관리자 및 입력 대기열 이름을 입력 하십시오.

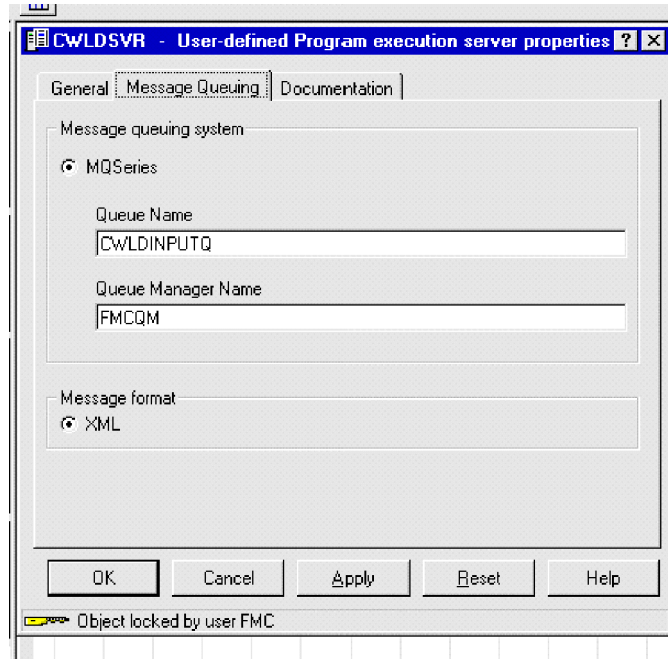


그림 15. WebSphere MQ Workflow Buildtime: 메시지 대기열 구성

5. 구현 탭을 누르십시오.

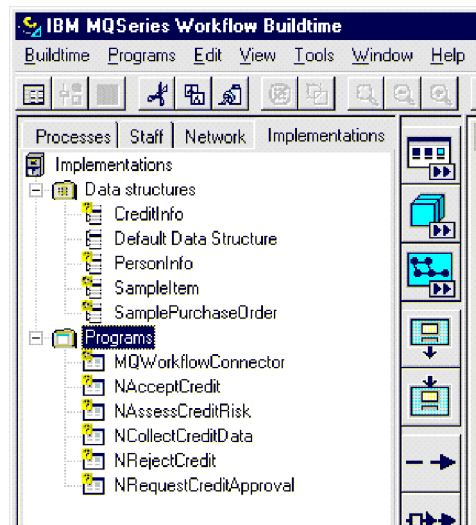


그림 16. WebSphere MQ Workflow Buildtime: 구현 보기

6. 메뉴 표시줄에서 프로그램 > 새 프로그램을 선택하십시오.

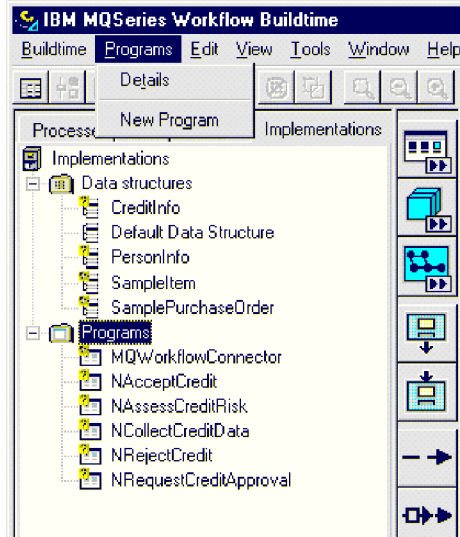


그림 17. WebSphere MQ Workflow Buildtime: 새 프로그램 선택

7. 프로그램 이름을 지정하십시오. 각 노드 대 협업 관계에 개별 UPES 프로그램을 정의해야 하므로, 협업과 동일한 이름을 사용하려고 할 수 있습니다.

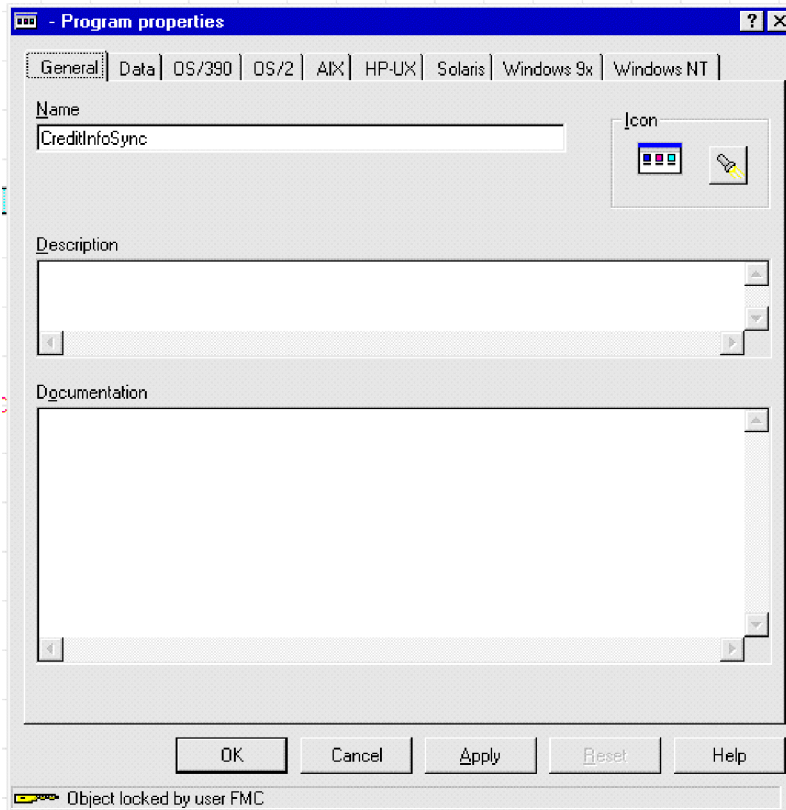


그림 18. WebSphere MQ Workflow Buildtime: 새 프로그램 이름 지정

8. 데이터 탭을 누르고 프로그램 또는 협업에 의해 승인될 데이터 구조를 지정하십시오. “프로그램 무인 실행 가능” 상자를 선택했는지 확인하십시오.

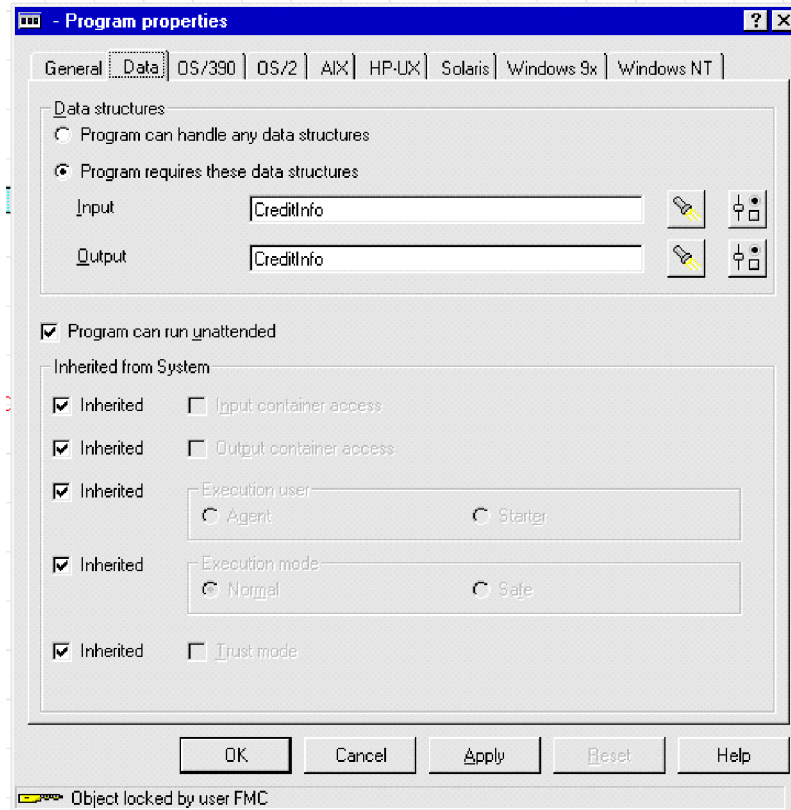


그림 19. WebSphere MQ Workflow Buildtime: 데이터 구조 지정

9. Windows 탭을 누르고 실행할 기존 프로그램을 입력하십시오.

주: 지정하는 프로그램이 실행되지 않는 경우에도 WebSphere MQ Workflow에서는 해당 프로그램을 정의해야 합니다.

워크플로우 설계 시 ICS에 데이터 구조를 게시할 때 사용할 verb 또는 협업을 표시하는 두 가지 명령행 매개변수를 지정해야 합니다. 커넥터에서는 이 매개변수가 이름-값 형식을 따라야 하며, 복수 이름-값 쌍은 세미콜론으로 분리되어야 합니다. 현재 verb와 collab, 두 가지 값을 지정할 수 있습니다. 예를 들어, 워크플로우 데이터 구조가 커넥터로 발행된 후 CreditInfoSync 협업의 Update verb로 처리되도록 지정하려면, 프로그램 매개변수는 verb=Update; collab=CreditInfoSync와 동일해야 합니다. 협업 이름이 지정되지 않은 경우(verb=Update 프로그램 매개변수), 데이터 구조는 모든 등록된 협업을 표시합니다.

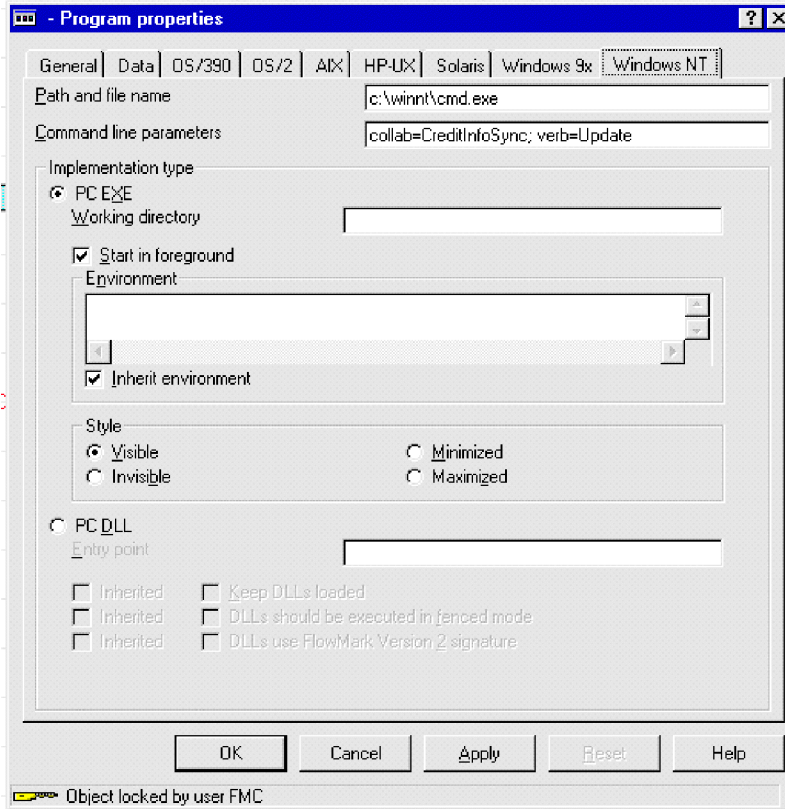


그림 20. WebSphere MQ Workflow Buildtime: 명령행 매개변수 지정

10. WebSphere MQ Workflow 커넥터에 대한 프로그램 노드 발행 요청을 작성하려면 새 프로그램 노드를 작성한 후 프로그램 이름을 지정하십시오(7단계에 정의되어 있는 대로 지정).

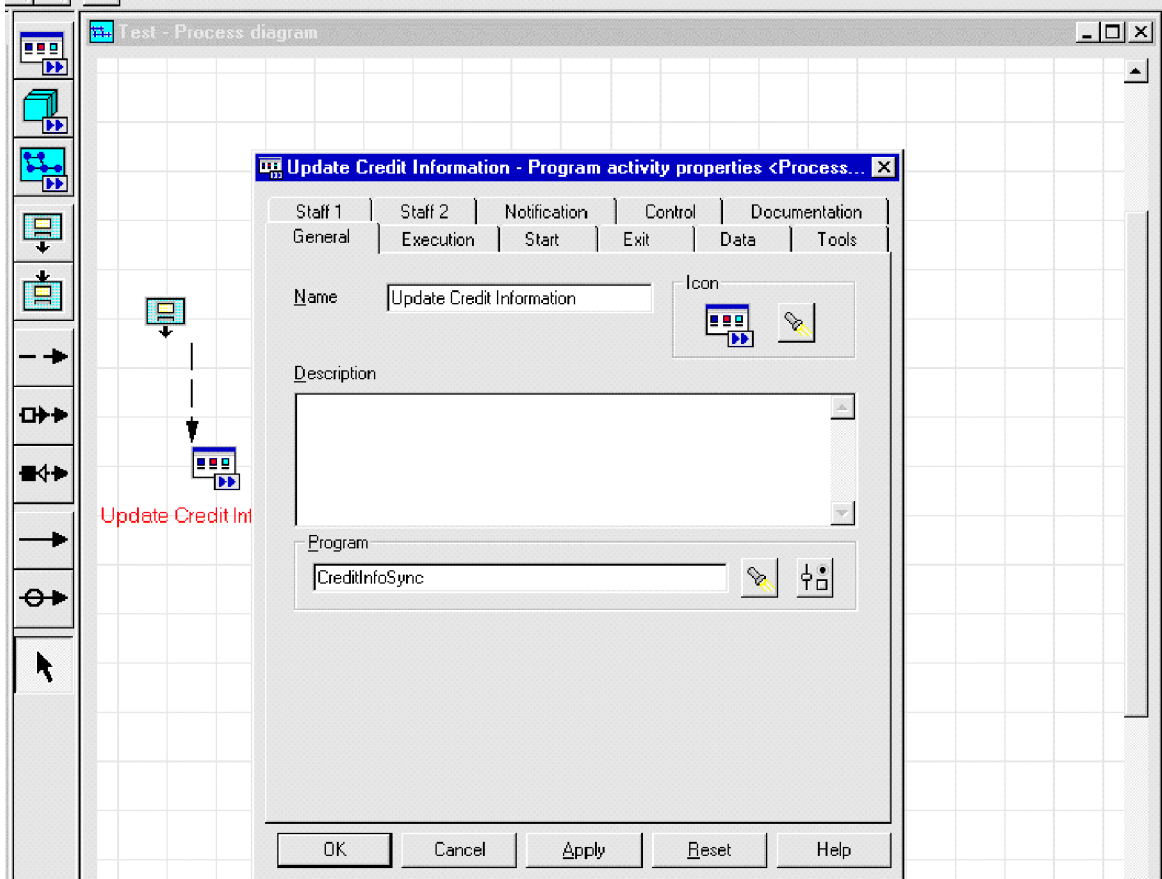


그림 21. WebSphere MQ Workflow Buildtime: 새 프로그램 노드 작성

11. 프로그램 실행 서버(CWLDSVR.FMCSYS.FMGRP)를 정의하고 요청 유형으로 동기 또는 비동기를 선택하십시오.

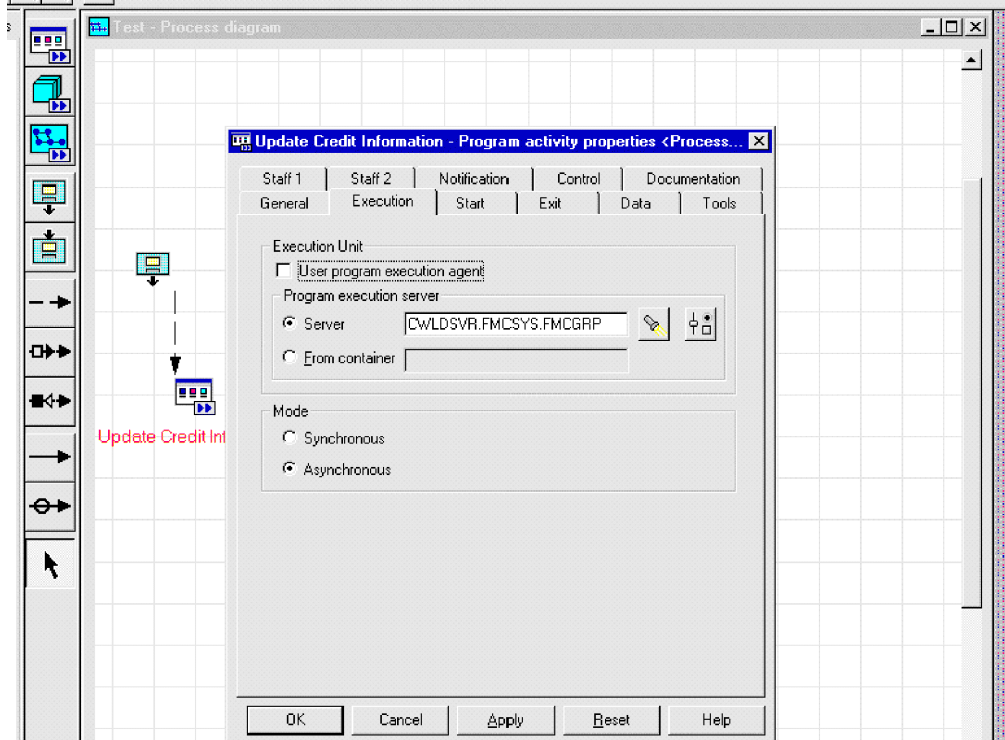


그림 22. WebSphere MQ Workflow Buildtime: 프로그램 서버 정의

이 노트가 WebSphere MQ Workflow 내에 도달하면, 워크플로우 데이터 구조를 포함하는 메시지가 커넥터에 발행됩니다. 커넥터는 비즈니스 내용을 처리하고 내용의 오류 또는 변경사항을 리턴합니다.

제 4 장 Business Object 개발

- 『커넥터 Business Object 구조』
- 66 페이지의 『샘플 Business Object 정의』
- 73 페이지의 『오류 처리』
- 75 페이지의 『추적』

Connector for WebSphere MQ Workflow를 설치 및 구성한 후 Business Object를 작성해야 합니다. 이 작업은 다음 작업을 통해 수행할 수 있습니다.

- 이 릴리스에서 제공하는 샘플 Business Object 수정
- WebSphere MQ Workflow 내보내기 파일에서 Business Object 정의 파일 압축 풀기

이 장에서는 새 Business Object 구현을 위한 안내서로 사용할 수 있는 정보를 제공합니다. 또한 사용자가 수정할 수 있는 샘플 Business Object를 보여줍니다. WebSphere MQ Workflow에서 Business Object 정의 파일 압축을 풀려면 77 페이지의 제 5 장 『FDLBORGEN 유틸리티를 사용한 Business Object 정의 작성』을 참조하십시오.

커넥터 Business Object 구조

커넥터는 메타 데이터 구동 커넥터입니다. Business Object의 경우, 메타 데이터는 응용프로그램에 대한 데이터로서, Business Object 정의에 저장되고 응용프로그램과 커넥터가 상호작용하는 데 도움이 됩니다. 메타 데이터 구동 커넥터는 커넥터에서 하드 코딩된 명령어가 아닌 Business Object 정의에서 인코딩된 메타 데이터에 따라 자체 지원하는 각 Business Object를 처리합니다.

Business Object 메타 데이터에는 Business Object 구조, 해당 속성 등록 정보 설정 및 해당 응용프로그램 특정 정보가 포함됩니다. 커넥터는 메타 데이터 구동 방식이므로 커넥터 코드를 수정할 필요 없이 새 Business Object 또는 수정된 Business Object를 처리할 수 있습니다. 그러나 커넥터의 구성 Data Handler는 Business Object 구조를 가정합니다. 따라서 WebSphere MQ Workflow용 Business Object 작성 또는 수정 시, 수정사항은 커넥터가 따르도록 설계된 규칙과 일치해야 하며, 그렇지 않은 경우 커넥터는 새 Business Object 또는 수정된 Business Object를 올바르게 처리할 수 없습니다.

XML Data Handler에서 부과한 Business Object 외의 Business Object 구조에 대한 요구사항이 있습니다. 이 사항에 대한 자세한 정보는 37 페이지의 『Meta Object 구성』을 참조하십시오. 커넥터가 처리하는 Business Object는 InterChange Server에서 허용하는 어떤 이름이라도 가질 수 있습니다.

커넥터는 대기열에서 메시지를 검색하고 메시지 내용으로 Business Object(최상위 Business Object 및 메타 데이터에서 정의함)를 채우려고 시도합니다. 엄밀히 말해서, 커넥터는 Business Object 구조를 제어하지도 영향을 주지도 않습니다. 그러한 기능은 커넥터의 Data Handler 요구사항이자 메타 데이터 정의의 기능입니다. Business Object 검색 및 전달 시 커넥터의 주요 역할은 Business Object로의 메시지 프로세스(및 반대 프로세스)에서 오류를 모니터링하는 것입니다.

샘플 Business Object 정의

이 섹션에서는 샘플 Business Object 정의를 제공합니다. Cardinality 및 IsKey 등과 같은 Business Object 속성에 관한 특정 정보는 *Connector Development Guide for Java*를 참조하십시오.

```
[BusinessObjectDefinition]
Name = MQWF_SampleItem
Version = 1.0.0
AppSpecificInfo = cw_mo_wfcontainer=ContainerInfo
```

```
[Attribute]
Name = Input_Item
Type = MQWF_Structure_SampleItem
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = SampleItem;type=pcdata;
IsRequiredServerBound = false
[End]
```

```
[Attribute]
Name = ContainerInfo
Type = MO_MQWorkflow_ContainerInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
```

```
[Attribute]
Name = Output_Item
Type = MQWF_Structure_SampleItem
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = SampleItem;type=pcdata;
```

```

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

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

[BusinessObjectDefinition]
Name = MQWF_Structure_SampleItem
Version = 1.0.0
AppSpecificInfo = SampleItem

[Attribute]
Name = Name
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
AppSpecificInfo = Name;type=pcdata;
DefaultValue =
[End]

[Attribute]
Name = Price
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = Price;type=pcdata;
DefaultValue =
[End]

```

```
[Attribute]
Name = Stock
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = Stock;type=pcdata;
DefaultValue =
[End]
```

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

```
[Verb]
Name = Create
[End]
```

```
[Verb]
Name = Retrieve
[End]
```

```
[Verb]
Name = Update
[End]
```

```
[Verb]
Name = Delete
[End]
[End]
```

```
[BusinessObjectDefinition]
Name = MQ_MQWorkflow_ProcessInfo
Version = 1.0.0
```

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

```
[Attribute]
Name = Organization
Type = String
```

```

Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ProcessAdministrator
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

[BusinessObjectDefinition]
Name = MO_MQWorkflow_ActivityInfo
Version = 1.0.0

```

```

[Attribute]
Name = Priority
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = MembersOfRoles
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = CoordinatorOfRole
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Organization
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = OrganizationType
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = LowerLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false

```



```

IsRequiredServerBound = false
[End]
[Attribute]
Name = UpperLevel
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = People
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PersonToNotify
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = Duration2
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false

```

```

IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

[BusinessObjectDefinition]
Name = MO_MQWorkflow_ContainerInfo
Version = 1.0.0

[Attribute]
Name = PROCESS_INFO
Type = MO_MQWorkflow_ProcessInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ACTIVITY_INFO
Type = MO_MQWorkflow_ActivityInfo
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 1
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ACTIVITY
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false

```

```

[End]
[Attribute]
Name = PROCESS
Type = String
Cardinality = 1
MaxLength = 255
IsKey = true
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = PROCESS_MODEL
Type = String
Cardinality = 1
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]
[Attribute]
Name = ObjectEventId
Type = String
MaxLength = 255
IsKey = false
IsForeignKey = false
IsRequired = false
IsRequiredServerBound = false
[End]

[Verb]
Name = Create
[End]

[Verb]
Name = Delete
[End]

[Verb]
Name = Retrieve
[End]

[Verb]
Name = Update
[End]
[End]

```

오류 처리

커넥터가 생성하는 모든 오류 메시지는 메시지 파일에 저장됩니다. (파일 이름은 LogFileName 표준 커넥터 구성 등록 정보에 의해 판별됩니다.) 각 오류에는 다음과 같이 오류 번호와 오류 메시지가 순서대로 표시됩니다.

```

메시지 번호
메시지 텍스트

```

커넥터는 다음 섹션에 설명되어 있는 대로 특정 오류를 처리합니다.

응용프로그램 시간 종료

다음과 같은 경우에 BON_APPRESPONSETIMEOUT 오류 메시지가 리턴됩니다.

- 메시지 검색 시 커넥터가 WebSphere MQ Workflow 대기열 관리자 연결을 설정할 수 없는 경우
- 커넥터가 Business Object를 메시지로 변환하지만 연결 유실로 인해 전송 대기열로 전송할 수 없는 경우
- 커넥터가 메시지를 발행하지만 변환 등록 정보 TimeoutFatal을 true로 하여 Business Object 응답을 기다리는 중 시간이 종료된 경우
- 동기 API를 사용하여 WebSphere MQ Workflow와 통신 시 커넥터 시간이 종료된 경우

비등록 메시지

커넥터는 다음과 같은 경우 UnsubscribedQueue 등록 정보가 지정한 대기열에 메시지를 전달합니다.

- 커넥터가 MQSTR 또는 FMXML 형식을 가지고 있지 않은 메시지를 검색한 경우
- 커넥터가 WfMessage를 검색하지만 메시지 이름 유형이 ActivityImplInvoke가 아니거나 General Error인 경우
- 커넥터가 등록 전송 처리 시 데이터 구조의 최상위 레벨 Business Object를 발견할 수 없는 경우

주: UnsubscribedQueue를 정의하지 않은 경우 미등록 메시지를 버립니다.

XML 구조 오류

커넥터는 다음과 같은 경우 ErrorQueue 등록 정보가 지정한 대기열에 메시지를 전송합니다.

- XML 문서 양식이 잘 구성되지 않은 경우
- WfMessage 구조가 완벽하지 않은 경우

Data Handler 변환

커넥터는 다음과 같은 경우 ErrorQueue 등록 정보가 지정한 대기열에 메시지를 전송합니다.

- Data Handler가 Business Object를 XML로 변환하거나 XML을 Business Object로 변환할 수 없는 경우

오류 및 WfMessage 문서

커넥터는 커넥터에 의해 보고된 표준 오류뿐만 아니라 WebSphere MQ Workflow 자체적으로 발행한 오류에도 응답합니다. 커넥터가 발행한 메시지를 동기식으로 처리하는 동안 오류가 발생하는 경우, WebSphere MQ Workflow는 Exception 요소를 포함하는 WfMessage를 리턴합니다. 이 요소의 텍스트는 WebSphere MQ Workflow 응용 프로그램에 대한 요청 조작 실패로 커넥터가 InterChange Server로 리턴되는 오류 메시지를 지정합니다.

커넥터가 WebSphere MQ Workflow가 발행한 WfMessage의 동기식 처리에 실패하는 경우(앞에서 설명한 이유로 인해), 커넥터는 Exception 요소를 포함하는 응답 WfMessage에 WebSphere MQ Workflow를 전송하려고 시도합니다. 커넥터는 장애를 일으킨 이벤트에 대한 상세 설명으로 이 요소를 채웁니다. WebSphere MQ Workflow는 적절한 사용자 개입이 이루어질 수 있도록 이 메시지를 사용하여 실패한 워크플로우 상태를 갱신합니다.

일단 커넥터가 원래 WebSphere MQ Workflow 서버가 발행한 요청에 대해 응답을 발행하면, 커넥터는 승인을 기다리지 않습니다. WebSphere MQ Workflow는 그러한 경우에만 오류를 보고하므로 커넥터는 승인을 수신할 수 없습니다. 이 문제점을 피하기 위해, 커넥터는 응답에 따라 발생하는 모든 오류 메시지가 커넥터의 입력 대기열에 발행되도록 자체 응답 메시지에 지정합니다. 오류 발생 시, 커넥터는 결국 이런 방식으로 pollForEvents 동안 통지를 받게 됩니다. 오류가 기록되지만 더 이상의 조치는 수행되지 않습니다. 그러한 오류는 WebSphere MQ Workflow가 불안정하거나 잘못된 것으로 판별한 데이터 구조를 Business Object가 생성한 데 따른 결과로 가정됩니다.

추적

추적은 커넥터 작동 후 바로 ON 상태로 할 수 있는 선택적 디버깅 기능입니다. 기본적으로 추적 메시지는 STDOUT에 기록됩니다. 추적 메시지 구성에 대한 자세한 정보는 57 페이지의 제 3 장 『WebSphere MQ Workflow 응용프로그램 수정』을 참조하십시오. 추적 사용 및 설정 방법을 포함한 추적에 대한 자세한 정보는 *Connector Development Guide*를 참조하십시오.

커넥터 추적 메시지에선 다음과 같은 레벨 옵션을 사용할 수 있습니다.

- 레벨 0 커넥터는 자체 버전만을 나열합니다.
- 레벨 1 커넥터는 폴링 시 메시지가 검색될 때마다 또는 요청 처리 시 메시지가 게시될 때마다 기록합니다.
- 레벨 2 커넥터는 gotAppEvent() 또는 executeCollaboration()에서 InterChange Server로 Business Object가 게시될 때마다 기록합니다.
- 레벨 3 커넥터는 이 프로세스에서 실행된 논리 및 구문 분석되는 메시지에 대해 보다 자세한 사항을 제공합니다.

- 레벨 4 커넥터가 스스로 입력 또는 종료하는 각 메소드를 추적합니다.
- 레벨 5 커넥터는 자체 초기화를 기록하고 처리되는 모든 Business Objects를
 덤프하며 읽기 및 쓰기가 가능한 모든 XML 문서를 나열합니다.

제 5 장 FDLBORGEN 유틸리티를 사용한 Business Object 정의 작성

- 『FDLBORGEN 유틸리티 정보』
- 78 페이지의 『FDLBORGEN 유틸리티를 사용하기 전에』
- 78 페이지의 『FDLBORGEN 구문』
- 79 페이지의 『FDLBORGEN 예제』
- 79 페이지의 『변환 참고사항』

이 장에서는 WebSphere MQ Workflow 내보내기 파일에서의 Business Object 정의 추출을 자동화하는 독립형 명령행 유틸리티인 FDLBORGEN에 대해 설명합니다. 결과 Business Object 정의 파일은 XML Data Handler와 호환될 수 있습니다.

WebSphere MQ Workflow 정의 언어의 머릿글자인 FDL은 FDLBORGEN이 입력으로 사용하는 WebSphere MQ Workflow 내보내기 파일의 접미부입니다.

FDLBORGEN 유틸리티 정보

FDLBORGEN 유틸리티 기능은 다음과 같습니다.

- FDL 구조 정의에 따라 Business Object 정의를 작성하여 Business Object 정의 파일에 기록합니다. 이 파일은 repos_copy 유틸리티를 사용하여 InterChange Server 저장소에 로드할 수 있습니다.
- XML Data Handler의 요구사항을 따르는 Business Object 정의를 작성합니다. 이 Business Object 정의는 추가로 편집하지 않아도 됩니다.
- 모든 Business Object 정의에 필수 ObjectEventId 속성을 추가합니다. 또한 이 속성을 지정하는 경우 Business Object 파일 상단에 저장소 버전 번호를 추가합니다.

FDLBORGEN은 connectors\MQWorkflow\utilities\fdlborgen 디렉토리의 일부로 설치되는 다음 두 파일로 구성됩니다. 전체 설치 제품 디렉토리는 21 페이지의 『설치된 파일 구조』를 참조하십시오.

- FDLBORGEN 유틸리티를 호출하는 다음의 실행 파일
 - Windows 환경의 경우, 이 실행 가능 파일은 fdlborgen.bat입니다.
 - UNIX 환경의 경우, 이 실행 가능 파일은 fdlborgen입니다.
- FDLBORGEN 조작에 필요한 모든 Java 클래스가 들어 있는 Java 아카이브 파일인 fdlborgen.jar

FDLBORGEN 유틸리티를 사용하기 전에

FDLBORGEN 유틸리티를 사용하려면 하나 이상의 입력 파일을 작성해야 합니다. 이 작업은 WebSphere MQ Workflow에서 .fd1 유형의 파일을 생성 및 내보냄으로써 수행할 수 있습니다.

1. WebSphere MQ Workflow Buildtime을 여십시오.
2. Buildtime 메뉴에서 내보내기를 선택하십시오.
3. 대화 상자에서 파일 이름(예: MyExport.fdl)을 선택한 후 확인을 누르십시오.

FDLBORGEN 구문

FDLBORGEN 명령행 유틸리티를 실행하려면 다음을 수행하십시오.

1. DOS 명령 프롬프트 창(Windows) 또는 셸 창(UNIX)을 여십시오.
2. FDLBORGEN 명령을 입력하십시오.

FDLBORGEN 명령 형식은 다음과 같이 정의됩니다.

```
fdlborgen -i[input-file] -o[output-file] -p[prefix]
          {-n[object-name]} {-r[repos_version]} {-v[verblist]} {-V}
```

위의 각 구성요소에 대한 설명은 다음과 같습니다.

-i[input-file]	fd1 파일이 현재 디렉토리에 위치하지 않는 경우 fd1 파일 이름 및 해당 경로를 지정합니다.
-o[output-file]	생성된 Business Object 정의가 저장될 위치 및 이름을 지정합니다.
-n[object-name]	Business Object 정의로 변환될 fd1 파일에 최상위 레벨 데이터 구조의 이름을 지정합니다.
-p[prefix]	생성된 Business Object 정의 이름 앞에 삽입되는 접두부를 지정합니다. fd1 오브젝트 이름과 Business Object 이름을 구별할 때 유용합니다. 접두부 옵션이 필요합니다.
-r[repos_version]	생성된 Business Object 정의 파일 시작 부분에 ReposCopy 머리글을 추가합니다. 예를 들어, 다음과 같이 파일 시작 부분 다음에 -r1.0.2를 추가합니다. [ReposCopy] Version = 1.0.2 [End] 기존 Business Object 정의 파일 버전에 겹쳐쓰고 있는 경우, 다음 매개변수를 사용하여 버전 정보를 보존하십시오.

- v[*verblist*] 각 Business Object에 포함될 verb 목록을 지정합니다. Create, Retrieve, Update 및 Delete verb가 지원됩니다. 각 verb를 쉼표로 분리하고 공백을 추가하지 마십시오.
이 매개변수를 지정하지 않는 경우, 표준 Create, Retrieve, Update 및 Retrieve verb가 추가됩니다.
- V 프로그램을 상세 모드로 전환하고 모든 해당 항목, 속성, 요소 및 주석을 인쇄합니다.
- [*mapping-file*] 맵핑 파일의 이름(및 파일이 현재 디렉토리에 위치하지 않는 경우의 해당 경로)을 지정합니다. 맵핑 파일에 관해서는 다음 "변환 참고사항"을 참조하십시오.

주: FDLBORGEN에서 발생한 모든 오류는 stderr로 전송됩니다.

FDLBORGEN 예제

WebSphere MQ Workflow 데이터 구조 MyCustomer를 Wf_MyCustomer Business Object로 변환하려면 먼저 WebSphere MQ Workflow Buildtime으로 입력 파일을 생성해야 합니다. 현재 디렉토리의 MyExport.fdl을 사용하여 다음과 같이 FDLBORGEN을 실행할 수 있습니다.

```
fdlborgen -iMyExport.FDL -oMyBO.txt -nMyCustomer -pWf_ -vCreate,Update -r2.0.0
```

이를 통해 MyCustomer 데이터 구조에 따라 MyExport.FDL FDL 파일에 Wf_MyCustomer Business Object가 생성됩니다. 모든 필수 하위 데이터 구조 또한 이 정의 파일에 포함됩니다.

변환 참고사항

STRING, LONG 또는 FLOAT 유형의 데이터 구조 구성원은 동일한 이름의 STRING 유형 속성으로 맵핑됩니다.

데이터 구조의 오브젝트 구성원은 이름은 동일하나 <boprefix><data structure name>의 유형이 아닌 하위 오브젝트로 변환됩니다.

Business Object 이름 및 속성 이름은 80개까지의 영숫자와 밑줄을 가질 수 있습니다. 최상위 레벨 데이터 구조와 fdl의 구성원 이름을 지정할 때 이 규정을 지키십시오. 그렇지 않으면, 맵핑 테이블을 준비하거나 -m 명령 옵션을 사용하여 맵핑 테이블 파일을 지정하십시오.

맵핑 테이블 파일은 다음 CSV 형식을 따라야만 합니다.

- <name1 in FDL>,<mapped name1>

- <name2 in FDL>,<mapped name2>

-m 옵션과 맵핑 파일 이름 및 경로(필요에 따라)를 지정할 경우, <name1 in FDL>은 Business Object 정의 파일에서 생성된 <mapped name1>으로 대체됩니다.

제 6 장 문제점 해결

이 장에서는 Connector for WebSphere MQ Workflow 시작 및 실행 시 발생할 수 있는 문제점에 대해 설명합니다.

시작 시 문제점

문제점

커넥터가 시작 시 갑자기 종료되고 다음 메시지가 보고됩니다.
java.lang.NoSuchMethodError at com.ibm.workflow.api.
Agent\$OsaLocator.locateController(Agent.java:219) at
com.ibm.workflow.api.Agent\$OrbLocator.locate(Agent.
java:173) at com.ibm.workflow.api.Agent.setName
(Agent.java:401)

메시지 처리 중 커넥터는 종료되며 다음과 유사한 오류를 보고합니다.
Exception thrown: com.crossworlds.connectors.
mqworkflow.exceptions.FatalProcessingException: [Type:
Fatal Error] [MsgID: 40007] [Msg: Failed to read message
content. An IO error occurred: java.io.
UnsupportedEncodingException Cp437.]]

초기화 중 커넥터는 갑자기 종료되며 다음 예외가 보고됩니다.
java.lang.UnsatisfiedLinkError: no mqjbnd01 in shared
library path

초기화 중 커넥터는 갑자기 종료되며 다음 예외가 보고됩니다.
Exception in thread "main" java.lang.NoClassDefFoundError:
com/ibm/workflow/api/FmcException

가능한 솔루션 / 설명

이 오류는 기본 IBM Java ORB 라이브러리가 IBM Java ORB 에이전트를 통해 원격 Workflow 서버로 통신하는 경우 WebSphere MQ Workflow API에 필요한 라이브러리와 호환되지 않음을 표시할 수 있습니다. IBM 요구사항에 따라 이러한 라이브러리는 기본 IBM Java ORB 라이브러리에 우선하도록 부트스트랩되어야 합니다. 이를 수행하려면 start_connector.bat 또는 start_connector.sh를 열고 3단계를 시작하는 섹션으로 화면이동한 후 IBM에서 요구하는 대로 올바른 IBM Java ORB 라이브러리를 지정했는지 확인하십시오.

주: WebSphere MQ Workflow API에서는 제공되는 라이브러리와 다른 IBM Java ORB 라이브러리가 필요할 수 있습니다. 자세한 정보는 *IBM WebSphere MQ Workflow: Programming Guide*를 참조하십시오.

사용 중인 JVM 버전이 메시지 문자 세트 지원에 필요한 라이브러리를 가지고 있지 않습니다. 이 문제점에 대한 가장 간편한 솔루션은 Sun Microsystems의 최신 JDK 버전을 다운로드한 후 이 새 JVM에서 커넥터를 실행하는 것입니다.

start_connector.bat(또는 start_connector.sh) 파일을 열어 %CROSSWORLDS%\bin\java의 모든 인스턴스를 새 JVM 경로로 대체하십시오.

커넥터가 IBM WebSphere MQ Java 클라이언트 라이브러리에 서 필수 런타임 라이브러리(mqjbnd01.dll [Windows] 또는 libmqjbnd01.so [UNIX])를 찾을 수 없습니다. 경로에 라이브러리 폴더가 포함되었는지 확인하십시오.

start_WebSphereMQWorkflow.bat(Windows) 또는 start_WebSphereMQWorkflow.sh(UNIX)에서 MQSeries Workflow 클라이언트 라이브러리에 대한 정확한 경로를 지정했는지 확인하십시오. 추가 지시사항은 52 페이지의 『시작 파일 구성』을 참조하십시오.

문제점

초기화 중 커넥터는 갑자기 종료되며 다음 예외가 보고됩니다.
java.lang.UnsatisfiedLinkError: no fmcojprf (libfmcojprf.a or .so) in java.library.path

가능한 솔루션 / 설명

커넥터는 필수 런타임 라이브러리를 찾을 수 없습니다. 시스템에서 지정된 라이브러리(예: libfmcojprf.a)를 찾아 이 파일의 상위 디렉토리가 사용자 경로에 포함되어 있는지 확인하십시오. 라이브러리가 찾을 수 없는 경우, 모든 필수 전제조건 소프트웨어 (WebSphere MQ Workflow 응용프로그램 및 WebSphere MQ 클라이언트 라이브러리)를 설치했는지 확인하십시오. 라이브러리를 사용자 경로에 위치시키고 추가한 경우, start_connector 스크립트를 수정한 후 명령행에 전달한 java.library.path 옵션에 이 경로를 추가하여 어댑터를 시작할 수 있습니다. 이를 수행하려면, 시작 스크립트의 아래쪽으로 화면이동하여 어댑터를 시작하는 명령을 찾으십시오(이는 ProductDir\bin\java로 시작할 수 있습니다). 명령행은 옵션을 지정합니다. _Djava.library.path=.를 시작하는 옵션을 찾으십시오. _Djava.library.path 옵션용으로 지정한 기타 디렉토리의 목록에 문제의 라이브러리에 해당하는 상위 디렉토리를 추가하십시오. 예를 들어, 수정 후에 c:\program\files\webspheremqworkflow\bin이 libfmcojprf.a를 포함한 경우, 명령행은 _Djava.library.path=ProductDir\bin;%CONNDIR%;c:\program files\webspheremqworkflow\bin을 포함할 수 있습니다.

이벤트 처리**문제점**

커넥터가 커넥터 구성에 지정된 내용에 관계없이 사용자 ID가 "fmc"인 모든 메시지를 전송합니다.

가능한 솔루션 / 설명

커넥터에 지정된 WebSphere MQ 서버 채널의 등록 정보를 점검하십시오. MCA 사용자 ID 등록 정보에 값을 지정하지 않았는지 검증하십시오.

부록 A. 커넥터의 표준 구성 등록 정보

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

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

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

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

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

신규 및 삭제된 등록 정보

다음 표준 등록 정보가 이 릴리스에 추가되었습니다.

새 등록 정보

- XMLNameSpaceFormat

삭제된 등록 정보

- RestartCount
- RHF2MessageDomain

표준 커넥터 등록 정보 구성

어댑터 커넥터에는 두 가지 유형의 구성 등록 정보가 있습니다.

- 표준 구성 등록 정보
- 커넥터 특정 구성 등록 정보

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

Connector Configurator 사용

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

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

등록 정보 값 설정 및 갱신

등록 정보 필드의 기본 길이는 255자입니다.

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

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

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

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

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

표준 등록 정보 요약

표 13은 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 13. 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 13. 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 13. 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 13. 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 커져서 오류 또는 심각한 오류가 발생할 때 InterchangeSystem.cfg 파일에 지정된 MESSAGE_RECIPIENT에 대한 전자 우편 공고를 생성합니다.

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

MaxEventCapacity

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

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

MessageFileName

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

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

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

MonitorQueue

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

기본값은 CONNECTORNAME/MONITORQUEUE입니다.

OADAutoRestartAgent

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

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

기본값은 false입니다.

OADMaxNumRetry

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 1000입니다.

OADRetryTimeInterval

RepositoryDirectory가 <REMOTE>인 경우에만 올바릅니다.

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

기본값은 10입니다.

PollEndTime

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

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

PollFrequency

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

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

기본값은 10000입니다.

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

PollQuantity

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

PollStartTime

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

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

RequestQueue

Business Object를 커넥터에 전송하기 위해 통합 브로커에서 사용하는 대기열입니다.

기본값은 CONNECTOR/REQUESTQUEUE입니다.

RepositoryDirectory

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

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

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

ResponseQueue

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

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

RestartRetryCount

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

기본값은 3입니다.

RestartRetryInterval

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

기본값은 1입니다.

SourceQueue

DeliveryTransport가 JMS이고 ContainerManagedEvents가 지정된 경우에만 적용 가능합니다.

JMS 이벤트 저장소를 사용하는 JMS 사용 커넥터에 대해 보증된 이벤트 전달을 지원하는 커넥터 프레임워크의 JMS 소스 대기열을 지정합니다. 자세한 정보는 90 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.

- 101 페이지의 『Connector Configurator 개요』
- 102 페이지의 『Connector Configurator 시작』
- 103 페이지의 『커넥터 특정 등록 정보 템플릿 작성』
- 106 페이지의 『새 구성 파일 작성』
- 109 페이지의 『구성 파일 등록 정보 설정』
- 117 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(103 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, 104 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(109 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 104 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 - 커넥터 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 템플

리트에 나타납니다.

등록 정보가 종속되도록 지정하고 종속 조건을 설정하려면, 다음을 수행하십시오.

1. 사용 가능한 등록 정보 화면에서 종속될 등록 정보를 선택하십시오.
2. 등록 정보 선택 필드에서 드롭 다운 메뉴를 사용하여 조건 값을 보유할 등록 정보를 선택하십시오.
3. 조건 연산자 필드에서 다음 중 하나를 선택하십시오.

== (같음)

!= (같지 않음)

> (초과)

< (미만)

>= (이상)

<= (이하)

4. 조건 값 필드에 종속 등록 정보를 템플릿에 포함하기 위해 필요한 값을 입력하십시오.
5. 사용 가능한 등록 정보 화면에서 종속 등록 정보가 강조표시된 상태에서 화살표를 눌러서 종속 등록 정보 화면으로 이동하십시오.
6. 완료 버튼을 누르십시오. Connector Configurator가 Connector Configurator를 설치한 \bin 디렉토리의 \data\app에 XML 문서로 입력한 정보를 저장합니다.

새 구성 파일 작성

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

브로커를 선택하려면 다음을 수행하십시오.

- Connector Configurator 홈 메뉴에서 파일 > 새로 작성 > 커넥터 구성을 누르십시오. 새 커넥터 대화 상자가 나타납니다.
- 통합 브로커 필드에서 ICS, WebSphere Message Brokers 또는 WAS 연결을 선택하십시오.
- 이 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 템플릿이 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 .out입니다.
- 커넥터의 이전 구성 파일.
이러한 파일의 확장자는 일반적으로 *.cfg입니다.

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

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

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

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

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

*.txt 파일이 커넥터용 어댑터 패키지로 전달되거나 확장자가 다른 정의 파일을 사용할 수 있는 경우 이 옵션을 선택하십시오.
3. 디렉토리 화면에서 적당한 커넥터 정의 파일로 이동하고 이를 선택한 후 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 > 프로젝트에서를 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 112 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 메시징을 사용하는 커넥터의 경우 적용 가능)

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

ICS에서 실행되는 커넥터의 경우, 다음 등록 정보의 값도 필요합니다.

- 연관된 맵
- 자원
- 메시징(적용 가능할 경우)

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

표준 등록 정보는 다음과 같이 커넥터 특정 등록 정보와 다릅니다.

- 커넥터의 표준 등록 정보는 커넥터의 응용프로그램 특정 구성요소와 해당 브로커 구성 요소에서 모두 공유됩니다. 모든 커넥터에는 동일한 표준 등록 정보 세트가 있습니다. 이러한 등록 정보는 각 어댑터 안내서의 부록 A에 설명되어 있습니다. 일부는 변경할 수 있지만 모든 값을 변경할 수는 없습니다.
- 응용프로그램 특정 등록 정보는 커넥터의 응용프로그램 특정 구성요소, 즉 응용프로그램과 직접 상호작용하는 구성요소에만 적용됩니다. 각 커넥터에는 커넥터의 응용프로그램에 고유한 응용프로그램 특정 등록 정보가 있습니다. 이러한 등록 정보 중 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. 값은 변경할 수 있으나 이름을 변경하거나 등록 정보를 제거할 수는 없습니다.
- 흰색 배경의 필드는 응용프로그램 특정 등록 정보를 나타냅니다. 이들 등록 정보는 응용프로그램 또는 커넥터의 특정 필요에 따라 다릅니다. 값을 변경하고 이 등록 정보를 삭제할 수 있습니다.
- 값 필드는 구성 가능합니다.

- 갱신 메소드 필드는 정보용이므로 구성할 수 없습니다. 이 필드는 값이 변경된 등록 정보를 활성화하기 위해 필요한 조치를 지정합니다.

표준 커넥터 등록 정보 설정

표준 등록 정보의 값을 변경하려면 다음을 수행하십시오.

1. 값을 설정하려는 필드를 누르십시오.
2. 값을 입력하거나, 값이 나타나는 경우 드롭 다운 메뉴에서 선택하십시오.
3. 표준 등록 정보의 모든 값을 입력한 후, 다음 중 하나를 수행할 수 있습니다.
 - 변경사항을 취소하고 원래 값을 보존한 상태에서 Connector Configurator를 종료하려면, 파일 > 종료를 누르고(또는 창을 닫고) 변경사항을 저장할 것인지 묻는 메시지가 표시될 때 **아니오**를 누르십시오.
 - Connector Configurator에서 다른 카테고리의 값을 입력하려면 카테고리 탭을 선택하십시오. 표준 등록 정보(또는 다른 카테고리)에 입력하는 값은 사용자가 다음 카테고리로 이동할 때 보존됩니다. 사용자가 창을 닫을 때 모든 카테고리에 입력한 값을 전체적으로 저장할 것인지 버릴 것인지 묻는 메시지가 표시됩니다.
 - 변경된 값을 저장하려면, 파일 > 종료를 누르고(또는 창을 닫고) 변경사항을 저장할지 묻는 메시지가 표시되면 **예**를 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 **추가**를 누르십시오. 하위 등록 정보를 추가하려면 상위 행 번호를 마우스 오른쪽 단추로 누른 다음 **하위 추가**를 누르십시오.
2. 등록 정보 또는 하위 등록 정보의 값을 입력하십시오.
3. 등록 정보를 암호화하려면 **암호화** 상자를 선택하십시오.
4. 『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

84 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(ICCS에만 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 제어가 등록 협업으로 전송하는 해당 일반 오브젝트가 있습니다. 맵의 연관은 응용프로그램 특정 Business Object를 일반 Business Object로 변환하거나 일반 Business Object를 응용프로그램 특정 Business Object로 변환하는 데 사용할 맵을 결정합니다.

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

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

연관된 맵 탭은 다음 필드를 표시합니다.

- **Business Object** 이름

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

- 연관된 맵

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

- 명시

어떤 경우에는 연관된 맵을 명시적으로 바인드해야 할 수도 있습니다.

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

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

맵을 명시적으로 바인드하려면 다음을 수행하십시오.

1. 명시 열에서, 바인드하려는 맵에 대한 선택란에 체크 표시를 하십시오.
2. Business Object와 연관시키려는 맵을 선택하십시오.
3. Connector Configurator 창의 파일 메뉴에서 프로젝트에 저장을 누르십시오.
4. 프로젝트를 ICS에 전개하십시오.
5. 변경사항이 적용되도록 서버를 다시 시동하십시오.

자원(ICS)

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

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

메시징(ICS)

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

추적/로그 파일 값 설정

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

로깅 및 추적 값을 변경하려면 다음을 수행하십시오.

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

- 콘솔에(STDOUT):

로깅 또는 추적 메시지를 STDOUT 화면에 기록합니다.

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

- 파일에:

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

주: 로깅 및 추적 파일은 단순한 텍스트 파일입니다. 파일 이름을 설정할 때 원하는 파일 확장자를 사용할 수 있습니다. 그러나 추적 파일의 경우, 사용자 시스템에 상주할 수 있는 다른 파일과의 혼동을 피하기 위해 .trc보다는 .trace 확장자를 사용하는 것이 좋습니다. 파일 로깅의 경우에는 .log 및 .txt가 일반적인 파일 확장자입니다.

Data Handler

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

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

구성 파일 저장

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

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

- System Manager에서, 통합 구성요소 라이브러리에 *.con 확장자를 갖는 파일로 저장.
- 지정하는 디렉토리에 저장.

- 독립형 모드에서, 디렉토리 폴더에 *.cfg 확장자를 갖는 파일로 저장.

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

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

구성 파일 변경

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

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

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

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

구성 완료

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

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

국제화된 환경에서 Connector Configurator 사용

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

Connector Configurator는 다음 위치에서 영어 이외의 문자를 지원합니다.

- 모든 값 필드

- 로그 파일 및 추적 파일 경로(추적/로그 파일 탭에 지정됨)

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

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

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

부록 C. 자습서

이 부록에서는 어댑터와 함께 제공되는 샘플의 설치, 구성 및 사용 방법에 대해 설명합니다. 이 자습서는 ICS(InterChange Server) 통합 브로커에서 사용할 수 있도록 디자인되었습니다. 주석 규칙에 대한 안내는 이 문서의 머리말을 참조하십시오.

전제조건

이 자습서를 사용하기 전에 IBM WebSphere Business Integration 제품을 설치하고 이 제품에 대해 잘 알고 있어야 합니다. 샘플을 설치하기 전에 다음 작업을 완료하십시오.

1. Aadapter for WebSphere MQ Workflow를 설치하십시오. 이 자습서의 샘플은 WebSphere MQ Workflow 버전 3.2.2, 3.3.2 및 3.4.x에 맞게 디자인되었으며 기타 버전에 대한 지원은 다를 수 있습니다.
2. 포트 커넥터를 설치하십시오. 포트 커넥터와 실제로 연관된 에이전트는 없지만 커넥터 정의가 저장소에 있어야 합니다.
3. 저장소에 WebSphere MQ Workflow Adapter 정의 및 포트 커넥터 정의가 없는 경우 Connector Configurator를 사용하여 로드하십시오.
 - a. 파일에서 열기 메뉴 항목을 선택하십시오.
 - b. \connectors\WebSphereMQWorkflow\samples 폴더에 있는 저장소 파일 Sample_MQWorkflow_Order_Connectors.in을 로드하십시오.
 - c. MQWorkflowConnector 및 PortConnector 정의가 로드되었는지 확인하십시오.
4. Java용 IBM WebSphere MQ 클라이언트 라이브러리를 설치하십시오.

설치 전 점검 목록

샘플을 설치하기 전에 다음 정보를 수집하십시오.

- IBM WebSphere InterChange Server의 이름(기본값: LocalHost)
ICS 이름 = _____
- WebSphere MQ Workflow 대기열 관리자(기본값: FMCQM)
대기열 관리자 이름 = _____
- 대기열 관리자의 CCSID(기본값: 819)
대기열 관리자 CCSID = _____
- 대기열 관리자 리스너에 설정된 포트(기본값: 5010)
대기열 관리자 포트 = _____

- 대기열 관리자의 호스트(기본값: LocalHost)
대기열 관리자 호스트 = _____
- 대기열 관리자에서 사용하는 서버 연결 채널(기본값: FMCQM.CL.TCP)
대기열 관리자 채널 = _____
- WebSphere MQ Workflow 시스템 이름(기본값: FMCSYS)
WebSphere MQ Workflow 시스템 이름 = _____
- WebSphere MQ Workflow 시스템 그룹(기본값: FMCGRP)
WebSphere MQ Workflow 시스템 그룹 = _____
- WebSphere MQ Workflow 계정 사용자 이름(기본값: ADMIN)
WebSphere MQ Workflow 계정 사용자 이름 = _____

주: 이 계정은 시스템 계정의 MQM 그룹의 일부로서 WebSphere MQ Workflow에 있어야 합니다. 그렇지 않은 경우, 어댑터에서 WebSphere MQ를 통해 메시지를 발행하지 못하거나 WebSphere MQ Workflow에서 WebSphere MQ Workflow 프로세스를 실행할 수 없습니다.

- WebSphere MQ Workflow 구성 이름(기본값: FMC)
WebSphere MQ Workflow 구성 이름 = _____

환경 설정

이 섹션에서는 샘플을 사용할 수 있도록 환경을 준비하는 방법에 대해 설명합니다. 이 후부터 *sample_folder*는 샘플이 있는 폴더를 나타내고 *WBI_folder*는 현재 IBM WebSphere Business Integration이 설치되어 있는 폴더를 나타냅니다.

1. 대기열 작성 이 자습서를 사용하려면 WebSphere MQ Workflow 대기열 관리자에서 여섯 개의 로컬 대기열을 정의해야 합니다. WebSphere MQ Explorer 응용 프로그램을 사용하거나 명령행에 RUNMQSC FMCQM을 입력하고 다음 명령을 실행하여 작성할 수 있습니다.
 - DEFINE QL('MQWFCONN.ERROR')
 - DEFINE QL('MQWFCONN.ARCHIVE')
 - DEFINE QL('MQWFCONN.IN_PROGRESS')
 - DEFINE QL('MQWFCONN.REPLYTO')
 - DEFINE QL('MQWFCONN.UNSUBSCRIBED')
 - DEFINE QL('CWLINPUTQ')
2. **Business Object** 정의 작성 -- 선택적 WebSphere MQ Workflow 데이터 구조에 대한 Business Object 정의 파일은 이미 존재하지만, 이 단계를 수행하면 데이터 구조를 Business Object로 변환하는 FDLBORGEN 유틸리티를 보여줍니다. 이 유틸리티를 사용하려면 다음을 수행하십시오.

- a. 명령행에서 `WBI_folder/connectors/WebSphereMQWorkflow/utilities` 디렉토리로 변경하십시오.
- b. 다음을 입력하십시오.

```
FdlBorgen -isample_folder/WebSphereMQWorkflow_Samples.fdl -
osample_folder/SampleItem.in -nSampleItem -pMQWF_Structure_ -r3.1.0
```

```
FdlBorgen -isample_folder/WebSphereMQWorkflow_Samples.fdl -
osample_folder/SampleItemOrder.in -nSampleItemOrder -pMQWF_Structure_
-r3.1.0
```

3. 저장소로 샘플 **Business Object** 로드 Business Object Designer를 사용하여 IBM WebSphere ICS를 시작한 후 파일”파일에서 열기를 선택하십시오. 저장소 파일 `Sample_MQWF_Order_Objects.in`을 로드하십시오. 이 파일은 `WBI_folder/connectors/WebSphereMQWorkflow/Samples` 폴더에 있습니다. 샘플 Business Object가 로드되었는지 확인하십시오.
4. 저장소로 샘플 협업 템플릿 및 협업 오브젝트 로드 WebSphere Business Integration System Manager를 사용하여 파일에서 열기를 선택하십시오. 저장소 파일 `sample_MQWF_Order_Collaborations.in`(`WBI_folder/connectors/WebSphereMQWorkflow/Samples` 폴더에 있음)을 로드하십시오.
5. 협업 템플릿 컴파일 WebSphere Business Integration System Manager를 사용하여 Collaboration Templates 폴더를 마우스 오른쪽 단추로 누르고 드롭 다운 목록에서 모두 컴파일을 선택하십시오.
6. **IBM WebSphere ICS** 다시 시작 InterChange Server를 다시 시동하여 모든 변경사항을 적용하십시오. System Monitor를 사용하여 모든 협업 오브젝트 및 커넥터 제어기를 녹색 상태로 유지하십시오.

샘플, 템플릿, 어댑터 및 맵 구성

이 섹션에서는 샘플 및 그 구성 방법과 어댑터에 대해 설명합니다.

샘플 내용 정보

샘플은 다음과 같습니다.

- `MQWF_DataStructure_SampleItemOrder` WebSphere MQ Workflow의 데이터 구조 `SampleItemOrder`를 나타내는 Business Object입니다.
- `MQWF_DataStructure_SampleItemOrder_Item` WebSphere MQ Workflow의 데이터 구조 `SampleItemOrder`에 포함되어 있는 하위 데이터 구조 `Item`을 나타내는 Business Object입니다.
- `MQWF_DataStructure_SampleItem` WebSphere MQ Workflow의 데이터 구조 `SampleItem`을 나타내는 Business Object입니다.

- MQWF_SampleItemOrder MQWF_DataStructure_SampleItemOrder의 컨테이너 오브젝트입니다. 어댑터에서 사용하는 여러 Meta Object와 함께 입/출력 데이터 구조를 보유합니다. 이 오브젝트는 데이터 구조 SampleItemOrder 처리 시 어댑터와 협업 간에 전달되는 오브젝트입니다.
- MQWF_SampleItem MQWF_DataStructure_SampleItem의 컨테이너 오브젝트입니다. 어댑터에서 사용하는 여러 Meta Object와 함께 입/출력 데이터 구조를 보유합니다. 이 오브젝트는 데이터 구조 SampleItem 처리 시 어댑터와 협업 간에 전달되는 오브젝트입니다.
- MO_MQWorkflow_ProcessInstance 이 오브젝트는 WebSphere MQ Workflow 프로세스를 추적 및 제어하는 데 사용됩니다.
- MQWF_SampleItemRequest MQWF_DataStructure_SampleItemRequest의 컨테이너 오브젝트로서 WebSphere MQ Workflow가 InterChange Server로 요청을 전송할 때 사용됩니다. 어댑터에서 사용하는 여러 Meta Object와 함께 입력 데이터 구조를 보유합니다. 데이터 구조 SampleItemRequest 처리 시 어댑터에서 협업으로 전달되는 오브젝트입니다.
- MQWF_SampleItemResponse MQWF_DataStructure_SampleItemRequest의 커넥터 오브젝트로서 InterChange Server가 WebSphere MQ Workflow로 응답을 리턴할 때 사용됩니다. 어댑터에서 사용하는 여러 Meta Object와 함께 출력 데이터 구조를 보유합니다. 데이터 구조 SampleItemRequest 처리 시 협업에서 어댑터로 전달되는 오브젝트입니다.
- MQWF_GBO_SampleItem MQWF_DataStructure_SampleItem의 컨테이너 오브젝트입니다. 입/출력 데이터 구조를 보유하며 협업의 가상 일반 Business Object를 나타냅니다.
- SampleItemOrderSync_MQWF_to_Port 및 SampleItemOrderSync_Port_to_MQWF 어댑터와 포트 커넥터 간에 Business Object MQWF_SampleItemOrder를 교환하는 데 사용되는 협업 오브젝트입니다.
- SampleItemSync_MQWF_to_Port 및 SampleItemSync_Port_to_MQWF 어댑터와 포트 커넥터 간에 Business Object MQWF_SampleItem을 교환하는 데 사용되는 협업 오브젝트입니다.
- SampleWorkflowProcessControl_Port_to_MQWF Business Object MO_MQWorkflow_ProcessInstance의 검색 및 제어에 사용되는 협업 오브젝트입니다.
- SampleItemActivity_MQWF_to_MQWF 어댑터에서 MQWF_GBO_SampleItem을 받아 포트 커넥터 에이전트로 전달한 후 다시 어댑터로 전송하는 데 사용되는 협업 오브젝트입니다.
- MQWF_Sample_RequesttoGBO 요청 Business Object MQWF_SampleItemRequest에서 가상 일반 Business Object MQWF_GBO_SampleItem으로 데이터를 전송하는 맵입니다.

- MQWF_Sample_GBOtoResponse 가상 일반 Business Object MQWF_GBO_SampleItem 에서 응답 Business Object로 데이터를 전송하는 맵입니다.
MQWF_SampleItemResponse.

템플릿 클래스 파일 복사

협업 템플릿을 사용하려면 관련 클래스 파일이 있어야 합니다.

- *sample_folder/classes*의 파일을 *WBI_folder/collaborations/classes/UserCollaborations/classes*로 복사하십시오.

주: 샘플 협업은 CollaborationFoundation 템플릿(개별적으로 사용 가능)를 기반으로 하므로 IBM은 협업 수정에 필요한 .CLM 또는 .java 구성요소를 제공하지 않습니다.

커넥터 구성

커넥터 등록 정보를 다음과 같이 구성하십시오.

- AgentTraceLevel=3
- ApplicationPassword=(*username*의 암호)
- ApplicationUserName=*username*
- ArchiveQueue=MQWFCONN.ARCHIVE
- BOPrefix=MQWF_
- DeliveryTransport=IDL(선택적)
- ErrorQueue=MQWFCONN.ERROR
- InputQueue=CWLDINPUTQ
- MQSeriesCCSID=*CCSID*
- MQSeriesChannel=*CHANNEL*
- MQSeriesHostname=*HOST*
- MQSeriesPort=*PORT*
- MQSeriesQueueManager=대기열 관리자
- OutputQueue=FMC.FMGRP.EXE.XML
- ReplyToQueue=MQWFCONN.REPLYTO
- UnsubscribedQueue=MQWFCONN.UNSUBSCRIBED

샘플 Business Object 지원

샘플 Business Object를 사용하려면 어댑터가 이 오브젝트를 지원하는지 확인해야 합니다.

1. System Manager에서 WBI Adapter for the WebSphere MQ Workflow 및 포트 커넥터 정의를 여십시오.

2. 지원되는 Business Object 탭을 누르고 다음 Business Object를 추가하십시오.
 - MO_MQWorkflow_ProcessInstance
 - MO_MQWorkflow_ProcessTemplateConfig
 - MO_MQWorkflow_ContainerInfo
 - MO_MQWorkflow_ProcessInfo
 - MO_MQWorkflow_ActivityInfo
 - MO_MQWorkflow_ActivityRequest
 - MO_MQWorkflow_ActivityResponse
 - MO_DataHandler_Default
 - MO_DataHandler_DefaultXMLConfig
 - MQWF_SampleItem
 - MQWF_Structure_SampleItem
 - MQWF_SampleItemOrder
 - MQWF_Structure_SampleItemOrder
 - MQWF_Structure_SampleItemOrder_Item
 - MQWF_SampleItemRequest
 - MQWF_SampleItemResponse
 - MQWF_Structure_SampleItemRequest

맵 바인딩

맵에서 데이터를 전송하려면 어댑터와 연관시켜야 합니다.

1. System Manager에서 어댑터 정의를 여십시오.
2. 연관된 맵 탭을 누르십시오.
3. 다음 맵에 대해 명시적 바인딩을 선택하십시오.
 - MQWF_Sample_RequesttoGB0
 - MQWF_Sample_GB0toResponse

최종 구성 단계

1. 모든 변경사항을 적용하려면 InterChange Server를 다시 시동하십시오.
2. *sample_folder/WebSphereMQWorkflow_Samples.fdl*을 IBM WebSphere MQ Workflow Runtime Server로 가져오십시오.

주: Runtime으로 FDL 파일을 가져오는 데 대한 정보는 *IBM WebSphere MQ Workflow: Getting Started with Buildtime*을 참조하십시오. 기존 런타임 데

이터를 겹쳐써도 문제가 없는 경우, 명령행 창을 열고 다음을 입력하여(경고: 이 작업은 모든 런타임 데이터를 겹쳐쓰는) 빨리 샘플 Workflow를 로드할 수 있습니다.

```
fmcibie /i=WebSphereMQWorkflow_Samples.fdl /u=ADMIN /y= /t  
/o (password: password)
```

시나리오 실행

시나리오를 실행하기 전에 다음을 수행하십시오.

1. 아직 실행하지 않은 경우 InterChange Server를 시작하십시오.
2. 아직 실행하지 않은 경우 -fkey 옵션(자동 폴링 방지)을 사용하여 WBI Adapter for WebSphere MQ Workflow를 시작하십시오.
3. 아직 실행하지 않은 경우 Visual Test Connector를 시작하십시오.

Visual Test Connector를 시작하고 PortConnector의 프로파일을 정의한 후 에이전트를 바인딩하여 포트 커넥터를 시뮬레이션하십시오.

동기 요청

이 시나리오에서는 정의된 Workflow 프로세스로 비즈니스 데이터를 전달하여 종료 결과를 검색합니다. 이는 WebSphere MQ Workflow에 대한 요청을 발행한 후 초기화된 Workflow 프로세스가 완료될 때까지 어댑터가 차단하므로 동기 호출입니다.

이 시나리오에서는 주문 키를 Workflow 프로세스 Lookup_Order_Status로 전달하여 가상 주문의 상태를 검색합니다. 이 Workflow에는 IBM WebSphere Business Integration Server에 retrieve를 실행하여 주문 정보를 가져오는 한 가지 조치만 포함됩니다. 이 조치는 어댑터가 WebSphere MQ Workflow에 동기 요청을 발행하는 방법 및 WebSphere MQ Workflow가 어댑터에 동기 요청을 발행하는 방법을 보여줍니다.

1. 샘플 주문 작성 Visual Test Connector를 사용하여 Create Verb로 Business Object MQWF_SampleItemOrder의 새 인스턴스를 작성한 후 다음과 같이 채우십시오(정의되지 않은 값은 CxIgnore여야 함).

MQWF_SampleItemOrder

- Input_ItemOrder
 - TrackingNumber = ABC123
- MO_Config
 - ProcessTemplateName = Lookup_Order_Status
 - KeepName = false
 - UserId = UserName
 - ExecutionMode = Synchronous

- ResponseTimeout = 600000
 - TimeoutFatal = false
2. 어댑터로 이 오브젝트 전송 어댑터는 이 오브젝트를 요청 메시지로 변환하여 WebSphere MQ Workflow 서버에 발행합니다. 어댑터는 즉시 리턴하지 않고 대신 WebSphere MQ Workflow에서 응답을 대기해야 합니다. WebSphere MQ Workflow는 어댑터에서 요청을 받아 MQWF_SampleItemOrder.Input_ItemOrder 오브젝트의 데이터를 사용하여 프로세스 템플릿 Lookup_Order_Status를 동시에 작성 및 호출합니다. 이 Workflow 프로세스의 처음이자 유일한 단계는 추적 번호를 키로 사용하여 IBM WebSphere Business Integration Server에서 데이터 구조 SampleItemOrder를 검색하는 것입니다. MQ Workflow는 이를 수행하기 위해 어댑터의 입력 대기열에 요청 메시지를 발행하고 응답을 대기합니다. 이는 WebSphere MQ Workflow Client 응용프로그램을 검사하여 확인할 수 있습니다.
 3. 어댑터 에이전트 창에서 p를 눌러 이벤트를 폴하십시오. 어댑터는 WebSphere MQ Workflow(원래 요청에 의해 트리거됨)에서 발행한 요청을 찾습니다. 어댑터는 Retrieve Verb로 요청 메시지를 MQWF_SampleItemOrder 오브젝트로 변환하여 SampleItemOrderSync_MQWF_to_Port 협업에 게시합니다.
 4. Visual Test Connector를 통해 요청을 승인하십시오. MQWF_SampleItemOrder.Input_ItemOrder 오브젝트의 TrackingNumber 속성이 ABC123(초기 요청의 속성과 동일함)인지 확인하십시오. 다음과 같이 MQWF_SampleItemOrder.Output_ItemOrder 오브젝트를 채운 후 응답 성공을 선택하여 요청을 완료하십시오.
 - Output_ItemOrder
 - TrackingNumber = ABC123
 - Approved = YES

어댑터는 WebSphere MQ Workflow로 응답을 리턴한 후 MQWF_SampleItemOrder.Output_ItemOrder에 포함되어 있는 비즈니스 데이터로 다시 전달합니다. WebSphere MQ Workflow는 어댑터에서 응답을 받아 원래 요청에 대한 응답 메시지로 데이터를 통합합니다. 원래 요청은 어댑터의 ReplyTo 대기열에 다시 발행됩니다. 어댑터는 응답 메시지를 검색하여 변경사항 또는 오류를 협업에 리턴합니다. 이로써 동기 Workflow 요청이 완료됩니다. 협업으로 리턴된 오브젝트에는 다음과 같이 값을 입력해야 합니다.

 - MQWF_SampleItemOrder
 - Input_ItemOrder
 - TrackingNumber = ABC123
 - MO_Config
 - ProcessTemplateName = Lookup_Order_Status

- KeepName = false
- UserId = UserName
- ExecutionMode = Synchronous
- ResponseTimeout = 600000
- TimeoutFatal = false
- Output_ItemOrder
 - TrackingNumber = ABC123
 - Approved = YES
- ProcessInstance 이 값을 채워야 하지만 값을 예측할 수 없습니다. 프로세스 상태는 TERMINATED이어야 합니다.

주: 10분(Meta Object에는 600000밀리초로 구성됨) 이내에 이 프로세스를 완료하지 못하면 어댑터가 WebSphere MQ Workflow에서 응답을 받지 못했다고 보고합니다.

비동기 요청

이 시나리오에서 어댑터는 정의된 Workflow 프로세스로 비즈니스 데이터를 전달하고 프로세스가 완료될 때까지 대기하지 않습니다. 이는 WebSphere MQ Workflow에 요청을 발행한 후 어댑터가 병렬 실행 시 프로세스를 추적하는 데 사용할 프로세스 ID를 수신한다는 점에서 비동기 호출입니다. 이 시나리오에서 어댑터는 Workflow 프로세스 Approve_Order로 주문을 발행하여 주문 승인 작업을 시작(또는 승인 완료 여부를 나중에 확인)합니다. Workflow 프로세스에서는 주문 항목에 대한 정보를 검색하여 해당 항목의 재고 수량이 충분한지 여부에 따라 주문 승인을 갱신합니다. 이 시나리오에서는 어댑터가 비동기식으로 Workflow 프로세스의 시작을 트리거하는 방법 및 어댑터가 병렬로 실행되는 Workflow 프로세스의 상태를 모니터링하는 방법을 보여줍니다.

1. Visual Test Connector를 사용하여 Create Verb로 Business Object MQWF_SampleItemOrder의 새 인스턴스를 작성한 후 다음과 같이 채우십시오(정의되지 않은 값은 CxIgnore여야 함).
 - MQWF_SampleItemOrder
 - Input_ItemOrder
 - TrackingNumber = ABC123
 - Customer = Billy Bob
 - Item
 - Name = Hammer
 - Quantity = 1
 - MO_Config
 - ProcessTemplateName = Approve_Order

- KeepName = false
 - UserId = UserName
 - ExecutionMode = Asynchronous
 - ResponseTimeout = 500
 - TimeoutFatal = false
2. MQWF_SampleItemOrder를 어댑터로 전송하십시오. 어댑터는 이 오브젝트를 요청 메시지로 변환하여 WebSphere MQ Workflow 서버에 발행합니다. 그런 다음 어댑터는 프로세스 인스턴스 ID가 포함된 응답을 대기합니다. WebSphere MQ Workflow는 어댑터에서 요청을 받아 MQWF_SampleItemOrder.Input_ItemOrder 오브젝트의 데이터를 사용하여 프로세스 템플릿 Approve_Order를 비동기식으로 작성 및 호출합니다. 프로세스가 시작되면, WebSphere MQ Workflow는 즉시 시작된 Workflow 프로세스의 ID를 포함하는 어댑터로 다시 응답을 발행합니다. Workflow 프로세스는 또한 IBM WebSphere Business Integration Server에서 name = Hammer로 데이터 구조 SampleItem을 검색하는 첫 번째 단계를 시작합니다. 이는 두 개의 별도 조치입니다. 어댑터의 ReplyTo 대기열로 응답이 발행되는 동시에 어댑터의 입력 대기열로 요청이 발행됩니다. 어댑터는 WebSphere MQ Workflow에서 응답을 받아 호출 협업으로 Business Object를 리턴합니다. 이 오브젝트는 다음과 유사합니다.

- MQWF_SampleItemOrder
 - Input_ItemOrder
 - TrackingNumber = ABC123
 - Customer = Billy Bob
 - Item
 - Name = Hammer
 - Quantity = 1
 - MO_Config
 - ProcessTemplateName = Approve_Order
 - KeepName = false
 - UserId = UserName
 - ExecutionMode = Asynchronous
 - ResponseTimeout = 5000
 - TimeoutFatal = false
 - ProcessInstance 값을 채워야 하지만 값을 예측할 수 없습니다.

주: 이 자습서에서 나중에 사용할 수 있도록 ProcessInstanceID 속성 값을 기억해두십시오.

이제 Workflow 프로세스는 협업 처리에 병렬로 실행됩니다. Workflow 프로세스를 추적 또는 제어할 수 있는 유일한 방법은 MQWF_SampleItemOrder. ProcessInstance 오브젝트에서 리턴된 ProcessInstanceID를 사용하는 것입니다.

3. Visual Test Connector를 사용하여 Terminate Verb로 Business Object MO_MQWorkflow_ProcessInstance의 새 인스턴스를 작성하십시오. WebSphere MQ Workflow 3.4에서 XML API를 사용해야 하며 WebSphere MQ Workflow 3.3.2에서도 권장됩니다. XML API를 사용하려면 어댑터 구성 등록 정보 JavaCorbaApi = False를 설정하고, Workflow 프로세스(MO_MQWorkflow_ProcessInstance)를 모니터링하려면 ProcessInstanceName = ProcInstName(이전 단계에서 리턴된 ProcInstName)을 설정하십시오.

주: XML API의 경우, Restart 및 Delete Verb만 지원됩니다.

4. 어댑터로 MO_MQWorkflow_ProcessInstance를 전송하십시오. Workflow 프로세스의 상태를 리턴해야 합니다. ProcInstState 속성은 RUNNING입니다.
5. 이 시나리오에서 시작된 Workflow 프로세스를 재개하려면 어댑터 에이전트 창에서 p를 눌러 이벤트를 풀하십시오. 어댑터는 WebSphere MQ Workflow(원래 요청에 의해 트리거됨)에서 발행한 요청을 찾습니다. 그런 다음 어댑터는 Retrieve Verb로 이 요청 메시지를 MQWF_SampleItem 오브젝트로 변환하여 SampleItemSync_MQWF_to_Port 협업에 게시합니다.
6. Visual Test Connector를 통해 요청을 승인하십시오. MQWF_SampleItem.Input_Item 오브젝트의 Name 속성이 Hammer인지 확인하십시오. 다음과 같이 MQWF_SampleItem.Output_Item 오브젝트를 채운 후 응답 성공을 선택하여 요청을 완료하십시오.

- Output_Item
 - Name = Hammer
 - Price = 14.99
 - Stock = 20

어댑터는 WebSphere MQ Workflow로 응답을 리턴한 후 MQWF_SampleItem.Output_Item에 포함되어 있는 비즈니스 데이터로 다시 전달합니다. WebSphere MQ Workflow는 어댑터에서 응답을 받아 Stock 값이 원래 주문의 Quantity 값보다 큰지 여부를 확인합니다. 값이 큰 경우 해머의 재고가 충분하여 주문을 완료할 수 있으므로 승인됩니다. Workflow 프로세스에서는 InterChange Server에서 주문을 갱신하여 마지막 단계를 수행하고, 원래 주문과 키가 동일하지만 현재 Approve 속성이 Y인 어댑터의 입력 대기열에 데이터 구조 SampleItemOrder를 발행합니다.

7. WebSphere MQ Workflow의 이 마지막 요청을 처리하려면 어댑터 에이전트 창에서 p를 눌러 이벤트를 풀하십시오. 어댑터는 WebSphere MQ Workflow에서 발행한 요청을 찾아 Update Verb로 MQWF_SampleItemOrder 오브젝트를 작성한 후 SampleItemOrderSync_MQWF_to_Port 협업에 오브젝트를 게시합니다. 이 시나리

오에는 갱신할 레코드가 없으므로 이 요청을 승인할 수 있습니다. 어댑터가 응답을 발행하면 IBM WebSphere Business Integration Server 및 Workflow 프로세스 모두의 요청이 완료됩니다.

Workflow 프로세스 제어

이 시나리오에서는 진행 중인 프로세스를 종료하여 Workflow 프로세스를 제어하는 방법에 대해 설명합니다.

1. 127 페이지의 『비동기 요청』에서 설명하는 대로 1단계에서 시작하되, Retrieve 용어로 Business Object MO_MQWorkflow_ProcessInstance를 발행하지 않고 용어를 Suspend로 변경하십시오. 이 오브젝트를 어댑터로 전송하고, WebSphere MQ Workflow Client 응용프로그램을 통해 프로세스가 일시중단되는지 확인하십시오.

주: 프로세스는 InterChange Server에 대한 첫 번째 요청을 완료할 때까지 SUSPENDING 상태를 유지합니다. 이는 어댑터가 아닌 WebSphere MQ Workflow의 기능을 반영합니다.

2. Verb를 Resume으로 변경하고 오브젝트를 다시 전송하십시오. Workflow 프로세스의 상태는 다시 RUNNING으로 변경됩니다.
3. Verb를 Terminate로 변경하고 오브젝트를 다시 전송하십시오. Workflow 프로세스의 상태는 다시 TERMINATED로 변경됩니다. Retrieve Verb로 오브젝트를 발행하여 이를 확인할 수 있습니다. 이런 방법으로 ICS를 통해 Workflow 프로세스의 상태를 제어 및 모니터링했습니다.

주: 이 요청을 생성한 프로세스가 종료되어도 어댑터의 입력 대기열에 요청 메시지가 남아 있을 수 있습니다. 이는 정상적인 현상입니다. 어댑터가 이 요청을 처리해도 WebSphere MQ Workflow는 생성된 응답을 무시합니다.

WebSphere MQ Workflow의 동기 요청

이 시나리오에서는 WebSphere MQ Workflow에서 ICS로의 동기 요청 및 응답을 시뮬레이션합니다. 시나리오 1과의 차이점은 어댑터가 비동기식으로 협업을 호출한다는 것입니다(시나리오 1에서는 모든 호출을 동기식으로 수행). 이 시나리오는 Workflow 프로세스가 완료되도록 다른 Workflow 프로세스가 대기하지 않아도 된다는 점에서 시나리오 1보다 유용합니다.

1. 갱신 요청을 작성하십시오. WebSphere MQ Workflow 클라이언트에서 Wowkflow 프로세스 인스턴스를 작성 및 시작한 후 다음과 같이 입력 데이터 구조 SampleItemRequest의 값을 입력하십시오.

- SampleItemRequest
 - Name = Hammer
 - Price = 14.99
 - Stock = 20

WebSphere MQ Workflow는 이 요청을 어댑터의 입력 대기열에 동기식으로 발행하고 응답을 대기합니다.

2. 어댑터 에이전트 창에서 p를 입력하여 이벤트를 폴하십시오. 어댑터는 WebSphere MQ Workflow에서 발행한 요청을 찾습니다. 어댑터는 Update 용어로 이 요청 메시지를 MQWF_SampleItemRequest 오브젝트로 변환합니다. 어댑터는 MQWF_Sample_RequesttoGBO 맵을 사용하여 Business Object를 일반 Business Object MQWF_GBO_SampleItem으로 변환한 후 비동기식으로 공개합니다. 여기서 어댑터는 응답을 대기하지 않고 다른 요청을 받을 수 있습니다.
3. Visual Test Connector를 통해 요청을 승인하십시오. 일반 Business Object를 등록하는 SampleItemRequest_MQWF_to_MQWF 협업은 오브젝트를 받고 Visual Test Connector에 의해 승인됩니다. MQWF_GBO_SampleItem.InputItem 및 MQWF_GBO_SampleItem.OutputItem 오브젝트의 Name 속성이 Hammer(초기 요청의 해당 속성과 동일)인지 확인하십시오. 다음과 같이 MQWF_GBO_SampleItem.OutputItem 오브젝트의 비어 있는 속성을 입력한 후 ReplySuccess를 선택하여 일반 Business Object를 어댑터로 전송하십시오.

- MQWF_GBO_SampleItemOrder
 - ContainerInfo
 - ReturnCode를 제외한 모든 속성을 입력합니다.
 - (MQWF에서 ID로 사용되는 ActImplCorrelID를 변경하지 마십시오.)
 - InputItem
 - Name = Hammer
 - Price = 14.99
 - Stock = 20
 - OutputItem
 - Name = Hammer
 - Price = 11.25
 - Stock = 8

어댑터는 일반 Business Object를 받아 MQWF_Sample_GBOtoResponse 맵을 사용하여 MQWF_SampleItemResponse 오브젝트로 변환합니다. 어댑터는 MQWF_SampleItemResponse.Output_Item에 포함되어 있는 비즈니스 데이터를 사용하여 WebSphere MQ Workflow로 응답을 리턴합니다. WebSphere MQ Workflow는 어댑터에서 응답을 받아 ActImplCorrelID 값을 확인합니다. ActImplCorrelID 값과 일치하는 Workflow 프로세스가 있는 경우 프로세스가 완료됩니다. WebSphere MQ Workflow 클라이언트의 해당 프로세스 인스턴스는 없어야 합니다(창을 새로 고쳐야 합니다).

주의사항

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

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

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.

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

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

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

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

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

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

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

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

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

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

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

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

프로그래밍 인터페이스 정보

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

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

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

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

상표 및 서비스표

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

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

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

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

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

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



IBM WebSphere Business Integration Adapter Framework V2.4.0

IBM